A PRAGMATIC COMMUNICATION MODEL FOR WAY-FINDING INSTRUCTIONS

PAUL WEISER

Department of Geodesy and Geoinformation
Research Group Geoinformation
Vienna University of Technology

DISSERTATION
2014

ZUSAMMENFASSUNG

Diese Arbeit präsentiert ein pragmatisches Agenten-basiertes Kommunikationsmodell für den Austausch von Wegbeschreibungen. Pragmatik ist in diesem Fall als linguistische Theorie zu verstehen, die Kommunikation als ein kontextabhängiges System ansieht. Das Ziel dieser Arbeit ist die Schaffung von konzeptionellen und formalen Grundlagen für das Design von neuartigen Informationssystemen, die mit Hilfe von dynamischen Interaktionsmethoden individuell anpassbare Routeninformationen bereitstellen.

Routenbeschreibungen die von heutigen Informationssystemen zur Verfügung gestellt werden, unterscheiden sich von Instruktionen, die von Menschen generiert werden, in zwei wesentlichen Punkten. Zum einen, ist die Art der Information anders. Menschen produzieren hauptsächlich qualitative Instruktionen, z.B. solche bei denen Orientierungspunkte eine zentrale Rolle spielen. Computer-generierte Instruktionen hingegen basieren auf quantitativen Informationen wie Längen- oder Zeitangaben. Diese Art von Information ist für Menschen nicht intuitiv kognitiv erfassbar. Zum anderen ist die Art wie mit einem Informationssystem interagiert werden kann nicht mit der Interaktionsweise von Menschen mit anderen Menschen vergleichbar.

Menschen stellen sich auf ihre Gesprächspartner ein, in dem sie ein mentales Modell (auch „Theory of mind“ genannt) von ihnen vorhalten, d.h., Informationen werden dynamisch und individuell generiert und adaptiert falls dies als nötig empfunden wird. Zudem kann, wenn eine übermittelte Information von einem Gesprächspartner als nicht ideal empfunden wird, dieser eine gewünschte Anpassung der vormals übermittelten Informationen signalisieren.

Diese zwei Prinzipien der pragmatischen Kommunikation („Theory of mind“ und Signale) wurden in dieser Arbeit konzeptionell und formal erfasst. Das Resultat ist ein ausführbares und testbares Modell, das in der funktionalen Programmiersprache Haskell formalisiert wurde. Der Beitrag dieser Arbeit besteht darin, dass sie formale Grundlagen schafft, damit eine Form der Interaktion wie sie Menschen intuitiv durchführen, in Zukunft auch zwischen Menschen und Computern möglich wird.

Das erste Kapitel spezifiziert die Problemstellung und die Hypothese. Ein Vergleich zwischen Routenbeschreibungen, die von Informationssystemen und Menschen generiert wurden, veranschaulicht das Problem.

Das zweite und dritte Kapitel arbeitet die bestehende Literatur zu den Bereichen Kommunikation und Routenbeschreibungen auf. Aus dieser Recherche wird eine Theorie entwickelt, die aufzeigt, dass pragmatische Kommunikation sich der “theory of mind“ und dem Austausch von Signalen zur Bedeutungsfestlegung von Konzepten bedient.

Kapitel 4 definiert zuerst eine einheitliche Taxonomie von Routenbeschreibungen. Anschließend werden aus Sprachdaten Signale extrahiert die Aufschluss über den genauen Verlauf des Gesprächs geben. Diese Signale werden systematisch erfasst und klassifiziert. Zudem findet eine detaillierte Analyse der einzelnen Gesprächsphasen statt, die aufzeigt wann und warum die extrahierten Signale auftreten.

Kapitel 5 präsentiert das formale Modell, dass die Konzepte der in Kapitel 2 bis 4 entwickelten Theorie mathematisch beschreibt. Dazu wurde die funktionale Programmiersprache Haskell gewählt, die aufgrund ihrer statischen Typsicherheit und der kompakten Schreibweise eine konsistente und präzise Formulierung des zuvor konzeptuell Aufgearbeiteten ermöglicht. Das Modell ist ausführbar und produziert Resultate, die die aus den Sprachdaten gewonnenen und beobachteten Ergebnisse bestätigten.

Kapitel 6 schließt diese Arbeit mit einer Zusammenfassung, einer Diskussion sowie einem Ausblick auf zukünftige Forschungsmöglichkeiten ab. Ein besonderer Augenmerk wird dabei auf die nicht-verbale Kommunikation gelegt, auf der ein Großteil unserer „Gespräche“ basieren.

Schlüsselwörter: Routenbeschreibungen, Pragmatik, Theory of Mind, Signalisierung, Kommunikation, Interaktion

ABSTRACT

This thesis presents an agent-based pragmatic communication model for way-finding instructions. Pragmatics should be understood as the linguistic theory that views communication as a context-dependent system. The goal of this work is to develop the conceptual and formal foundations for the design of next-generation spatial information systems. Such systems would allow more dynamic forms of interaction and be capable of delivering routing information that is tailored to a user's specific and individual needs.

Way-finding instructions provided by today's information systems are fundamentally different from instructions generated by humans.

First, the type of information is different. Humans produce instructions that include mostly qualitative information, e.g., in the form of landmarks. Computer-generated instructions, on the other hand, rely on quantitative (metric) information, such as references to time and distance between decision points. However, for many people this type of information is difficult to process, thus introducing additional cognitive strain in an already demanding way-finding task.

Second, the way today's information systems allow users to interact with the presented content seems extremely limited if compared to human forms of interactions. For example, humans can adjust to their dialog partner's expectations by keeping track of a mental model of them ("theory of mind"). This implies that information is generated individually and dynamically, allowing to make changes during information presentation, in case this seems necessary. In addition, humans have the ability to signal whether some piece of presented information has been understood or not.

In this work, these two mechanisms, i.e., humans having a "theory of mind" and the ability to use signals to indicate the progress of a conversation, were systematically analyzed and formally specified. The result is an executable and testable computational model, implemented in the functional programming language Haskell. The contribution of this thesis is that it provides the formal and conceptual foundations for the design of future information systems that can mimic human forms of interaction.

Keywords: Way-finding Instructions, Pragmatics, Theory of Mind, Signals, Communication, Interaction

ACKNOWLEDGMENTS

This thesis would not have been possible without the support and advice from a number of people!

First, I would like to thank my advisers Prof. Andrew U. Frank and Prof. Sabine Timpf. The discussions with you had a profound influence on my thinking and style of work. Special thanks go to Prof. Frank for giving me access to his extensive library. Unfortunately, four years are not nearly enough to read all the books that looked interesting.

Thanks to Prof. Martin Raubal for giving me the opportunity to finish my dissertation at ETH Zürich.

Thanks to my colleagues old and new, at TU Vienna and ETH Zürich. In particular, I would like to thank Amin Abdalla, Eva-Maria Holy, Jürgen Hahn, Paolo Fogliaroni, Markus Mayr, Thomas Fuhrmann, Gerhard Navratil, Heidi Hobel, Simon Scheider, Ioannis Giannopoulos, David Rudi, and Christian Sailer. Without you, I would have been lost!

Last but not least, I thank my family. In particular, my parents and my sister for their support.

Thank you Irmi for your love and understanding. Thank you Annika, you are the sunshine of my life.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

Part I

INTRODUCTION

1 INTRODUCTION

1.1 MOTIVATION

Navigation is the combined endeavor of way-finding (i.e., the planning and decision making necessary to reach a destination) and locomotion [Montello, 2005]. Whether we need to carry out some errands or plan a trip to a foreign country, chances are we need to complete navigational tasks in order to be successful. In this context, we often rely on some kind of stored expert knowledge, e.g., in the form of maps or route instructions. Such cognitive artifacts, i.e., devices "designed to maintain, display, or operate upon information in order to serve a representational function" [Norman, 1986], allow to transfer spatial knowledge (e.g., a route) between humans in a coded form (pictorial or verbal). In addition, cognitive artifacts reduce the need to carry out computations in the head [Hutchins, 1995].

Formally, a route consists of a series of decision points (e.g. intersections or landmarks) at which the way-finder needs to perform an action in order to proceed (e.g., "turn right at the bank"). Thus, route instructions communicate a sequence of such actions with the intent to guide a way-finder from a source to a goal destination (See Section 3 for a detailed treatment). Note, in this work the terms way-finding and route instructions are used interchangeably.

Route (Way-finding) instructions are navigation aids

1.1.1 *Way-finding Instructions offered by current Web Routing Services*

Today's Web routing services are popular information systems for getting instructions between two points of interest [Schmidt and Weiser, 2012]. Typically, a user needs to specify start and goal as well as a preferred mode of transport. Based on a graph representation of the street network, a routing algorithm (See [Delling et al., 2009] for a comparison) takes the user input to calculate a route between the two specified points. The output, i.e., the presented route, is usually a function of the selected transport mode and an optimization criterion. For example, walking instructions impose other restrictions on the computed route than directions for driving (cf. highways and one-way streets). The result of a Web routing query typically looks like the instructions shown in the following example[1]:

1 Taken from maps.google.com

1. Head northeast on Gußhausstr. toward Karlsg. (20m)
2. Slight left onto Karlsg. (240m, 3min)
3. Slight left toward Treitlstr. (260m, 3min)
4. Continue straight onto Treitlstr. Your destination will be on the right (62m, 54 secs)

An analysis of the type of information used in such computer-generated route instructions[2] reveals several distinct features:

Type of Information

Routing services provide way-finding instructions based on street names and quantitative information.

- Instructions mention **only street names**. However, research has shown that proper names are both badly remembered and recalled from memory [Cohen and Burke, 1993]. Furthermore, during the way-finding task street names might be difficult to identify in the environment. For example, signs could be temporarily occluded (e.g., due to construction work), not be at the place a person expects them to be, or miss entirely. Note that street names are not even spelled out, e.g., "Karlsgasse" is refereed to as "Karlsg". This might further impair the cognitive processing of the instructions, e.g., for foreigners.
- Instructions contain **quantitative (metric) information** (e.g., 240 meters or 3 minutes) to indicate progression along the route. However, in order to make use of quantitative information in instructions, a user needs to be able to measure distances (either by estimation or using a device). This may introduce additional cognitive strain in an already demanding way-finding task. In addition, it is well known that humans are notoriously inaccurate when asked to estimate distances (cf. [Crompton, 2006] for some recent findings).

Note, one could also mention the use of absolute directions (e.g., "northeast") and vague concepts (e.g., "slight left"), but their treatment would be beyond the scope of this thesis. However, judging from aforementioned empirical evidence, one could argue that the type of information (street names and quantitative) present in computer-generated route instructions is not optimal in terms of most people's cognitive abilities.

In addition to an analysis of the type of information, one can also look at the way how human-computer interaction takes place with the Web service:

Interaction

Current Information Systems are user-blind, context-blind, and static.

- The system is **user-blind**. The presented content is the same, regardless of the type of user who requested the information. Possible different needs of different users are not taken into account. However, the "spatial suitability" [Jonietz and Timpf, 2013] of a route is determined by the capabilities of an agent. For example, elderly may have different demands a route. Therefore, a suggested route that is perfectly fine for one person is not walkable for someone else (cf. stairs). Furthermore, prior knowledge of the person requesting the instructions is not considered. However, a person's geographic knowledge is individually different. This fact might cause redundancy if information is presented in an inflexible manner.

- The system is **context-blind**. The presented content is independent of a person's intended activity. For instance, a user in case of an emergency might need the fastest route, taking current traffic into account. A tourist, on the other hand, might rather be interested in the most scenic route, featuring landmarks of historic and cultural importance (cf. [Hirtle et al., 2011]).

- The system is **static** because it offers **little or no possibility for interaction** with the presented content. The only form of interaction the system provides is to specify certain input parameters (e.g., transport mode, start, goal), but once the information is displayed there is no way to modify it dynamically. One can of course run another query with different input parameters to get a (potentially) different result, but the system does not offer a user the possibility to adjust specific parts of the route. For example, one could think of scenarios in which a user would like to have more in-depth information on a particular route segment that seems more difficult to navigate (cf. [Hoelscher et al., 2011].

1.1.2 *Way-finding Instructions as Communicated by Humans.*

Both the type of information presented and the ways of interaction offered by today's Web services stand in stark contrast to how humans communicate (spatial) information. The following excerpt [Weiser and Frank, 2013] demonstrates an example of an oral exchange of route information between two human individuals. Note, the terms *T* and *S* refer to the target (receiver) and source (presenter) of the instructions, respectively:

2 While this particular example is based on instructions offered by Google, a quick evaluation of other commercial routing services showed that none differs substantially in terms of how instructions are presented and how a user can interact with the system.

1. T: Can you tell me how I get to your place?
2. S: So, you start here at our institute, then you continue toward Naschmarkt.
3. T: Ok...
4. S: [...] then take the street before you come to Naschmarkt...
5. T: Is that Margarethenstrasse?
6. S: Yes!
7. S: [...], do you know the University of Applied Arts?

Careful study of such a real-world example reveals a feature that is missing from the type of information offered by Web routing services:

Type of Information

- Human-produced instructions use mostly **qualitative information** (e.g., landmarks). They do not rely on the measurement of time or distances. The example conversation illustrates the use of landmark information to identify crucial decision points (cf. Naschmarkt and University). An analysis of two corpora (Klein [1979]; Weiser and Frank[2013]) of human-produced route instructions in dialog showed that people use mostly qualitative terms throughout the conversation. In human produced instructions, statements such as "turn right at the bank" occur more likely than "turn right after 250 m". In addition, research has confirmed that qualitative statements play an important role for both the communication ([Frank, 1992], [Frank, 1996], [Klippel, 2003]) and conceptualization [Lakoff and Nunez, 2001] of (spatial) information. In terms of the usability of navigation systems, qualitative information is essential for providing efficient, effective, and satisfactory instructions [Burnett, 2000].

Humans are capable of providing qualitative (landmark based) way-finding instructions that are context-dependent.

The apparent difference between information provided by a Web service compared to humans is also reflected by the opposing types of data available to each source of information. Often, a Web service has only access to 2-D network data (i.e., coordinates and street names) while humans have access to wide range of 3-D visual input from which landmark information can be constructed. Although a theoretical method to extract landmarks from existing datasets has been proposed [Raubal and Winter, 2002], no Web service to date has implemented such a model. Approaches that address the need to bridge the discrepancy between computer-generated (quantitative) and human-generated (qualitative) information have recently been proposed by de Felice et al.[2011] and Fogliaroni[2013].

Even more striking is an analysis of the actual interactive communication process that occurs between two humans.

Interaction

- Humans use **different information for different users**, in case they perceive this is appropriate [Clark, 1996]. For example, the same route described to a tourist or a local may differ in terms of how landmarks are named [Hahn and Weiser, 2014]. A reference to "Stephansdom" (a large Viennese church) is likely to be more appropriate for a Viennese local, whereas "major church" might be a better description for the same real-world object, in case the addressee is a tourist. One could speculate that for a person from out of town, the reference's level of detail is too high to infer that "Stephansdom" maps onto an object that represents a church. In the example conversation (See 1.1.2 above) the use of the proper name "Naschmarkt" by S (a popular Viennese market) indicates that S assumes that T can understand this reference.

The exchange of instructions between humans takes place interactively and involves both speaker and hearer.

- A person providing an instruction can **take the intended activity** of its addressee **into account**. If the (assumed) intended activity is recognized, speakers may adjust the information presentation and content as they see fit. Hirtle et al. [2010], Hirtle et al. [2011], as well as Tenbrink and Winter[2009] provided some empirical evidence highlighting this fact.

- Humans can **ask for clarification** if they receive information that does not fit their expectations. Information is not just presented by a sender and then in turn passively received by a recipient. The recipient may actively intervene if they are not satisfied with the information presented ([Clark, 1996], [Weiser and Frank, 2013]). Explicit requests for changes in the level of detail of the content (e.g., "Stephansplatz" vs. "major church"), summaries of already received information, and the repetition of unclear parts are common methods to foster understanding of the discussed subject matter. This can be observed in lines 4-6 of the example conversation (See 1.1.2 above). S gives the instruction "take the street" at which T should make a turn. T wants to make sure whether this is the same street they had in mind and asks S to confirm the name of "the street", namely "Margarethenstrasse". This assumption is then approved by S as being correct.

- Humans may **probe information** in case they are not sure if the knowledge of some piece of information can be presupposed. During the exchange of instructions in the example conversation (See 1.1.2 above), S is not sure if T has the knowledge required to identify the landmark (as part of the next description) by its proper name "University of Applied Arts" (See 1.1.2, line 7 above). With the intent to get his assumption verified (or rejected) S asks T for confirmation.

Note, this thesis does not claim that human-produced way-finding instructions are generally superior to computer-generated ones. Humans can fail to deliver useful directions for various reasons, e.g., they may lack the necessary knowledge, make errors during the information presentation, or do not agree to cooperate. However, the way we can interactively exchange (routing) information is still unique to human-human communication settings.

1.2 STATEMENT OF THE PROBLEM

Section 1.1 highlighted the differences that exist between human-produced and computer-generated way-finding instructions. First, humans produce mostly qualitative instructions featuring landmarks, while instructions produced by a Web service rely on quantitative information and street names. Second, the way we can interact and exchange information with other humans is fundamentally different from the type of interaction offered by a Web routing service. Clearly, Human-Human interaction is more dynamic, flexible, and adaptive than Human-Computer interaction offered by today's information systems. This argument is further empirically supported by an in-depth analysis of relevant theories on communication and way-finding instructions (See Chapters 2 and 3, respectively).

While questions related to the effective communication of various types of information (qualitative vs. quantitative) are interesting on its own, this work focuses on the question of how the interaction between a user and an information system can be improved.

One central claim is that current information systems do not offer a way of interaction that lives up to the meaning of the word communication. The literal Latin meaning of the word communication is "to make common"[3]. Therefore, one can argue that the goal of communication is to agree on the interpretation of a concept (e.g., a word or sentence), i.e., to make the concept common between the participants of a conversation. Likewise, if participants do not agree on the meaning of a concept it is not common among them. Communication helps to form a shared understanding of concepts between two or more individuals.

If one accepts the semantics of the word communication (as suggested above), the analysis presented in Section 1.1 reveals that today's information systems fail to communicate information effectively because they offer no (or only limited) ways of making pieces of information common between the system and the user. This is mostly due to the fact that commercial systems do not allow for feedback whether the information demand is met by the user.

In other words, the adaption of the used vocabulary (to represent concepts) is all on the user side who has to learn the terms from the perspective of the machine. This means that the programmer of the system defined the meaning of the terms beforehand, leaving them unchangeable. This has also been noted by Timpf [2002] who

3 cf. www.etymonline.com

compared way-finding ontologies from the perspective of the traveler and the system.

This thesis is in the tradition of Pragmatics ([Levinson, 1983], [Yule, 1996]), in the sense that the successful communication of information depends on the user's ability to interact with other conversation partners and takes the user's intentions into account. In other words, Pragmatics is a branch of linguistics that views communication as a context-dependent system. Therefore, human communication is always pragmatic communication.

Frank and Mark [1991], Winter and Wu [2009], as well as Weiser and Frank [2013] suggested that in order to improve the usability of information systems, new ways of interacting with spatial information that come closer to human-style (pragmatic) communication need to be defined.

1.3 COMMITMENTS, HYPOTHESIS AND RESEARCH QUESTIONS.

This work is based on several assumptions. Note, a detailed treatment of the theory sketched here is provided in Section 2:

- Instructions are communicated between two humans, a knowledgeable source and a target person lacking the knowledge required to navigate between a specified start and goal. The communicated instructions are assumed to be walking directions. No implications on other modes than walking are drawn. This work does not explicitly model human-computer or computer-computer communication processes, although possible implications on them can be drawn (See Chapter 6).

- Instructions are presented in the form of installments. An installment is a chunk of information, that can either (in its minimal form) correspond to a decision point and an associated action, or encompass several decision points and associated actions (cf. spatial chunking [Klippel, 2005]). Human communication makes frequent use of installments with the intention to ease the cognitive workload of its receiver. For example, Clark [1996] noted that "speakers divide presentations into brief repeatable installments because they tacitly recognize that people have limited immediate memory spans". Generally, the amount of information that can be held in memory is limited to about "seven plus or minus two" pieces at a time [Miller, 1956].

- Humans engage in a form of interactive dialog during which each person indicates whether information the other person presented met their demand. It is implicitly assumed that an addressee's demand is met unless indicated otherwise, i.e., "doing nothing" is taken as a confirmation. In case an expectation is met, both conversation partners have agreed on a particular interpretation of the discussed statement. In case the expectation is not met initially, both speaker and hearer continue to collaborate on finding a shared interpretation.

- The process of finding a shared interpretation of a statement between two human individuals is referred to as **"Negotiation Of Meaning"**. This work applies the notion of meaning negotiation to the exchange of route instructions between two human agents.
- The negotiation takes place for each installment of the discussed route until both participants have agreed on its interpretation or decide that no common interpretation is possible.
- Indications that contribute to advancing the communication by either person are called **"Signals**". For example, an explicit confirmation of a suggested instruction is a signal (See Sections 2.4.1 and 4 for a detailed treatment).
- The instruction giving agent presupposes some form of spatial knowledge or preferences to be present in the instruction receiving agent. The concept of presupposing information is referred to as **"Theory of mind"** (See Section 2.4.2).

Hypothesis

Based on the assumptions and commitments made above, the hypothesis of this work is as follows:

> THE NEGOTIATION OF MEANING DURING THE EXCHANGE OF ROUTE INSTRUCTION IS A PROCESS BETWEEN A HUMAN INFORMATION SOURCE AND A HUMAN TARGET. THE PROCESS EMERGES THROUGH THE USE OF SIGNALS AND A THEORY OF MIND.

Research Questions

The hypothesis gives rise to several research questions that are addressed in this work:

- What are the conceptual requirements for a pragmatic communication model that captures the human exchange of wayfinding instructions?
- What are possible signals agents use to indicate either successful or unsuccessful communication of a particular installment of an instruction?
- What are the concepts and processes that need to be formalized to yield a computational model of the negotiation process?

1.4 APPROACH AND METHODOLOGY

The main contribution of this work consists of two consecutive parts. First, based on linguistic theories and empirical evidence from language data a conceptual model of the suggested negotiation process is constructed. Second, the conceptual model is formalized using the functional programming language Haskell [Peyton Jones, 2003] to allow for consistency checks and testing. The resulting computational model serves as a proof of concept.

In particular, Section 2 presents empirical evidence from Cognitive Linguistics and Psychology indicating that the meaning of terms during a conversation is negotiated with the help of signals and by taking other people's (assumed) knowledge into account, i.e., applying the concept of a "theory of mind". In addition, Section 3 discusses the state-of-the-art research on way-finding instructions. This helps to identify potential impediments in the construction of more cognitively adequate information systems.

These theoretical foundations are then applied to analyze real-world observations of a number of route instructions exchanges between two persons. The analysis is carried out in two steps and attempts to identify potential signals used by both persons during the spatial knowledge transfer. First, a data collection based on a number of dialog-based exchange of route instructions is conducted [Weiser and Frank, 2013]. The result is a preliminary notion of the used signals. Second, the initial findings are then verified by a coding process [Montello and Sutton, 2006] of additional language data. Specifically, an existing corpus of (dialog-based) route instructions in the form of transcripts is used (cf. [Klein, 1979]). Because transcripts can only contain some of the signals explicitly, their content needs to be analyzed and interpreted. The goal of the coding-process is to identify the signals people use to indicate how the conversation should progress.

The results from the signal extraction process are subsequently used to construct a conceptual model of the negotiation process. The main goal of this step is to define and analyze the structure of the negotiation process (Section 4).

In the final step, the conceptual model is formalized using the functional programming language Haskell [Peyton Jones, 2003] to yield an agent-based model. Using an executable programming language for this task has two main advantages. First, it allows to check the model's consistency using Haskell's strong and static type system. Second, the model can be tested by feeding it data and thus allows its validation. The model is valid if it can reproduce the same behavior (i.e., the negotiation process) as one would observe when humans engage in the exchange of route instructions.

1.5 RELEVANCE AND CONTRIBUTION

While most research on route instructions has been concerned with analysis of structural properties, models describing the communication of routes have received less attention (See Chapter 3). More specifically, research has mainly addressed questions how instructions can be made cognitively adequate for humans processing. What is still missing, however, are studies and formal models that focus on the "form of the communication" [Winter and Wu, 2009].

This thesis attempts to contribute to improving the interactive capabilities of current spatial information systems. An in-depth understanding of human communication is one of the necessary prerequisites because, humans are the ones that interact and communicate with a information system. Existing information systems do not take the pragmatic way of human-based communication into account. The first step to achieve the goal of more pragmatic forms of interaction is a formal model of the human communication process. One of the advantages of a formal treatment of the communication process is that computers enforce constraints on the consistency of a model. Models specified in a natural language are often too vague in their definitions and hard to check for consistency.

The contribution of this research is as follows:

1. This work unifies theories from Cognitive Linguistics and Psychology that emphasize the interactive nature of human communication and applies them to the field of Geographic Information Science. Specifically, a holistic treatment of signals and a "theory of mind" contributes to the understanding of communicative processes. This in turn can as one of the foundations to improve the interaction capabilities of today's information systems.

2. This work suggests a classification of the signals that can occur during the exchange of way-finding instructions. In addition, an in-depth analysis of the spatial knowledge transfer that occurs between humans negotiating route instructions is provided. To the best of my knowledge, such a signal classification and analysis do not exist.

3. This work develops a formal communication model that includes the hearer of the route instruction giving process and put them on par with the speaker. As a consequence, both people are contributors to the conversation. In addition, agents are assumed to have a "theory of mind". The meaning of statements (e.g., instructions) is negotiated among all participants of the conversation. Previous formal models of communication only concentrated on the speaker part but left out the hearer. This implies that only one interpretation is possible, the one of the speaker. This, however, impairs the development of truly interactive information systems. If only the system is in charge

of the interpretation of terms, a user may experience difficulties in using it. To the best of my knowledge, no holistic formal model that unifies signals and a theory of mind and applies it to a scenario of spatial knowledge transfer exists.

In the future, an information system that implements the suggested model could allow for a dynamic form of interaction with the presented content and thus mimic human-style (pragmatic) communication. For example, in case of an routing service, a user could then modify certain parts of a route for which she wishes to retrieve either less, more, or entirely different types of information. In addition, the system would be able to anticipate information demands by the user.

1.6 STRUCTURE

The remainder of this this work is structured as follows. Chapter 2 introduces communication theories relevant to this thesis, in particular the notion of signalling and the concept of a "theory of mind". In Chapter 3, a literature review of the existing research concerned with way-finding instructions is provided. Chapter 4 presents the proposed conceptual model. Chapter 5 discusses the computational model of the negotiation process. Note, the complete and testable Haskell code of the model can be found in Appendix A. Chapter 6 concludes with a summary and discussion of the presented work and suggests future research directions.

Part II

THEORETICAL FOUNDATIONS

2

REPRESENTATION, MEANING, AND COMMUNICATION

This chapter reviews research findings related to the study of human communication. It develops the theoretical background of this work and empirically supports the hypothesis stated in Section 1.3. The results are used to design and implement the conceptual (Chapter 4) and computational model (Chapter 5).

This chapter is structured as follows. First, research findings concerned with spatial mental representations are discussed. Mental representations are the foundations of knowledge on which meaningful communication is based. The treatment includes theories of how humans form and store mental representations of their environment, what elements constitute such representations and how representations are externalized, i.e., communicated. Second, theories on the nature of (spatial) information communication are presented. Unidirectional models are contrasted to theories that emphasize a collaborative approach to communication. Third, the grounding problem is addressed. Grounding refers to challenge how an abstract symbol system can be given meaning. Human grounding strategies, physical and social, are reviewed. Finally, empirical evidence suggests a theory that argues that the interpretation of terms can be negotiated through bi-directional communication (taking both speaker and hearer into account), signals and a "theory of mind".

2.1 REPRESENTATION

Humans have conscious and unconscious systems that handle responses to environmental stimuli, both of which form the basis for human thought and decision making [Kahneman, 2011].

Decision making is based on two cognitive Systems.

The first system is unconscious thought, resulting in fast and direct responses, e.g., in the form of emotions. Immediate responses to stimuli, e.g., from predators, without having to consciously think about what is going on is a hard-wired survival mechanism likely to have developed through an evolutionary selection process. Critical situations call for computational inexpensive and fast decision making if they are to be handled successfully.

Effective spatial behavior requires an internal representations of the world.

The second system is conscious thought, resulting in planned responses towards a goal. Crick and Koch [1998] argued that our perceptual visual consciousness allows us to interpret scenes in comparison with past experiences and store them for the future. Thus, for humans to carry out high-level cognitive processes, such as planning and goal pursing, some form of internal representation of the world needs to exist.

In terms of Marr [1982], a representation is simply "a formal system for making explicit certain entities or types of information, together with a specification of how the system does this". An example for representations can be found in the domain of abstract algebras, a division of mathematics. Algebras define entities (e.g., numbers), procedures (e.g. addition), and rules (e.g., $a + 0 = 0 + a = a$) that specify how the entities can be manipulated by the procedures.

One of the central hypotheses of cognitive science is that the process of thinking makes use of representations of the world and (computational) procedures that operate on them [Thagard, 2005]. This has become known as the **computational theory of mind**. As such, mental representations are a type of formal system. However, for example Lakoff and Nunez [2001] proposed that humans can use abstract thought only indirectly through the use of conceptual metaphors shaped by our daily experiences and interactions with reality. Other prominent scholars such as Searle [1980], have also criticized the computational approach for various other reasons (See also Section 2.3).

This thesis accepts the "computational theory of mind", i.e., it assumes that the human mind can be modeled by the use of computational procedures. This allows to specify a formal model of cognitive processes using an executable programming language. Note this stance does not imply that the mind works exactly like a computer, it is rather a way to model and explain human cognitive processes using computational approaches. For similar arguments, see the current discussions in the emerging field of Quantum Cognition [Busemeyer and Bruza, 2012].

Thought can be understood as a computational procedure operating on a representation of the world.

The term "cognitive mapping" describes "a series of psychological transformations by which an individual acquires, codes, stores, recalls and decodes information about the relative locations and attributes of phenomena in his everyday spatial environment." [Downs and Stea, 1973]. This definition stresses the importance of such a cognitive representation for the implementation of "any strategy of spatial behaviors", e.g., the process of way-finding.

Note, the metaphor "map" should only be understood in a functional sense, i.e., a cognitive map serves the same function as a cartographic map (e.g., self-localization), but it does not necessarily have the same degree of truthfulness or a similar way of knowledge representation [Downs and Stea, 1973]. For example, Stevens and Coupe [1978] showed that the judged relative direction between two cities are often distorted if qualitative relationships are inferred from superordinate structures (e.g., states) and imposed onto subordinate structures (e.g., cities) that do not share the same relationship. To illustrate, Portland (Oregon, USA) is often (wrongly) assumed to be located south of Toronto (Canada). This is assumed to be inferred from the correct fact that Canada is to the north of the US. In a similar fashion, Kuipers [1982] argued that knowledge encoded in cognitive maps is not symmetric, noting that the knowledge of how to get from A to B does not imply one knows how to get from B to A. Furthermore, metric information as opposed to topological relations

A cognitive map serves the same function as a cartographic map. However, this does not imply that both have the same degree of truthfulness.

is rarely (or only in a limited sense) preserved in cognitive representations[Lynch, 1960].

Barbara Tversky [1993] criticized the term "cognitive map" because it does not capture the intrinsic richness of the knowledge one can have about the environment. She suggested the terms "cognitive collage" and "spatial mental model". A cognitive collage refers to diverse snippets of information (personal experiences, narratives, etc.) that cannot necessarily be integrated into a single representation. A spatial mental model refers to representations that allow for "perspective taking and inferences about spatial locations", as it is the case for familiar areas.

According to Johnson-Laird [1989] a mental representation or model is a body of knowledge that meets three conditions:

1. Its structure corresponds to the structure of the situation that is represented by the model. Couclelis [1996] applies this to the process of exchanging route directions and suggests that the situation represented includes the spatial context and knowledge required for the task.

2. A mental model can consist of both perceptible and abstract elements. Couclelis argues that mental models in this context include a cognitive map (perceptible elements) as well as abstract cognitive structures, such as image schemeta [Johnson, 1987] and basic categories [Rosch, 1973].

3. The mental model does not contain variables but instances of elements, e.g. specific images that are stored for certain elements.

Kuipers [2000] proposed a computational model that contains four interrelated and hierarchically structured representational levels of environmental space.

Kuipers' computational model of an internal representation of the world.

1. A control level that consists of qualitatively uniform segments of space. An agent can bind them to continuous control laws ("walk towards the tower").

2. A discrete causal level that includes a representation of states and actions as well as the knowledge of how to manipulate states by applying actions to them ("turn right at the intersection").

3. A topological level that consists of a representation of places, paths, regions, and how they are linked qualitatively through connectivity, order, boundary, and containment relations ("street X is connected to street Y by connection C").

4. A metrical level that consists of a quantitative (metric) map that indicates distance, direction, and shape between, to and of objects ("the distance to the goal is 2 km").

2.1.1 *Elements of Spatial Mental Representations*

Lynch[1960] argued that people naturally form an individual image of the city in which they live, i.e., they have some specific form of spatial mental representation of their environment. He goes on and claims that "there seems to be a public image of any given city which is the overlap of many individual images"[Lynch, 1960, p. 46] . His study attempted to identify the invariants that can be found in such a public image of a city, i.e., the physical and perceptible objects that each individual image exhibits. According to his analysis, based on the collected observations of inhabitants in three U.S. cities, the image of a city consists of five elements: (1) Paths, (2) Edges, (3) Districts, (4) Nodes, and (5) Landmarks.

Paths, Edges, Districts, Nodes, and Landmarks are the basic elements of spatial mental representations.

1. **Paths** are linear elements along a person can move, e.g., streets, roads, or railroads. Since paths are one of the most prominent features in people's images they are often used as a reference to which other elements are oriented and related. The prominence of a particular path may be affected by an associated special use or activity (e.g., a shopping street), its spatial characteristics (e.g., a particular wide street), salient buildings along it (e.g., a spectacular facade), or its proximity to other features (e.g., a road along a river).

2. **Edges** are linear elements that are not conceptualized as paths. They can act as boundaries between two areas, e.g., walls, or rivers. Edges can either be perceived as barriers (e.g., a waterfront) or as seams, connecting and relating two regions (e.g. a street joining two areas together).

3. **Districts** are rather large two-dimensional sections of a city an observer can enter and evoke an "inside of" feeling, e.g., a particular neighborhood. Districts are determined by "thematic continuities", such as form, function, or inhabitants. Due to its distinct character people often instantly recognize in what district they are in.

4. **Nodes** are points at which decisions can be made, e.g., junctions, "places of a break in transportation", or "shifts from one structure to another". Nodes can also be perceived as concentrations of a certain characteristic, e.g., squares. The physical characteristics of a node may not be point-like at all (e.g., a large square can be a node). Thus, geographic features may be conceptualized as nodes although their geometry is not point-like.

5. **Landmarks** are physical point features that have salient characteristics making them distinct from other elements, e.g., a large tower, a store, or a sign. From the perspective of the way-finder, landmarks can either be distant (global) or local. Global landmarks can serve as a reference over a long distance symbolizing a "constant direction", e.g., a church tower. Local landmarks on

the other hand are often only visible from a particular point of view, e.g., a store.

The proposed distinction, however, is not clear cut. Lynch [1960] notes that the same physical element can serve different conceptual functions, i.e., they have different affordances [Gibson, 1979] for different users. A highway, for example, is a path for a car driver but an edge for a pedestrian. In addition, mental representations can have differing scales, ranging from small to large.

In this thesis the focus is on mental representations that encompass large scale or environmental space, i.e., space that needs to be apprehended through locomotion because it exceeds direct perception.

2.1.2 *The Acquisition of Mental Representations*

Mental representations of environmental space are formed in an iterative and (often time-consuming) fashion [Montello, 1993]. There are two ways one can acquire an internal representation of the world.

Mental representations can be acquired through first-hand experiences and cognitive artifacts.

1. In an **indirect fashion**, i.e., through the use of cognitive artifacts and other people's experiences communicated orally. Examples include (spatial) narratives, maps, and route directions.

2. In a **direct fashion**, i.e., through perceptional processes during which agents observe the environment and form beliefs about it.

Note that the resulting representation can be incomplete, imprecise, or even plain wrong (cf. [Frank, 2000], [Tversky, 1993]). Lynch [1960] noted that our perception is "partial, fragmentary, mixed with other concerns" and that the resulting "image is the composite of them all".

Marr [1982] suggested a computational theory consisting of three consecutive steps that may explain how one can acquire a representation through visual processing:

Marr's computational theory of representation acquisition is based on visual processing.

1. Important information from the two-dimensional retina image is gathered. This includes geometrical organization, distribution, and intensity changes of the perceived environment.

2. A 2.5-D sketch is generated that makes orientation and depth of the visible surface from an observer's perspective explicit.

3. The viewer-centered surface is transformed into an object-centered surface. This 3-D object-centered view makes the relationships between the object themselves explicit. Johnson-Laird [2004] notes that this final step is crucial if one wants to move safely through an (unfamiliar) environment, i.e., without running into objects.

Siegel and White [1975] proposed three types of learning systems that attempt to explain how humans acquire a spatial representation of geographical areas of which they had no prior knowledge:

1. The acquisition of landmark knowledge ("the figurative core") takes place by "taking snapshots" of salient landmarks and the context (spatial and temporal) in which they were perceived. The result of such a representation is the mental organization of landmarks in form of decision points.

2. This is followed by the learning of routes between the previously acquired landmark information. At first, such a route only consists of decisions when to change directions between two landmarks. Through experience, however, routes are "scaled" in order to add metric information, e.g., the number of blocks passed.

3. Finally, several routes are integrated and organized into a network-structure forming a "survey knowledge" of the environment.

Discrete vs. continuous theories of spatial knowledge acquisition.

Siegel and White's framework was criticized by Montello [1998] as well as Ishikawa and Montello [2006] who argued that spatial knowledge acquisition is a continuous accumulation and refinement of metric information rather than discrete shifts from non-metric to metric information.

2.1.3 *The Externalization of Mental Representations*

Once mental representations have been acquired they can be externalized either through artifacts (e.g., sketch maps, written descriptions) or speech (e.g., oral instructions). Johnson-Laird noted that "discourse [...] enables individuals to experience the world by proxy" [2004, p.189].

However, the oral communication of spatial information faces a particular problem. The environment as well as any mental representations acquired through it are three-dimensional in nature. Language, on the other hand enforces linearity on linguistic expressions, i.e., the communication of facts that describe the world is limited to one at a time and one after the other. This "linearity problem" [Levelt, 1981] requires speakers to map descriptions about the three-dimensional world onto a linearly ordered structure. Linde and Labov [1975] showed that people do this naturally, for example, when asked to describe the static spatial layout of their homes to someone who is not familiar with the apartment. Linde and Labov found that most people prefer to "transform spatial layouts into temporally organized narratives". For example, it is natural to say "...then you come to the living room with the kitchen to the right".

Language enforces linearity onto the expression of facts.

Note, the linearization problem does not occur if the spatial structure to be described itself happens to be linear, for example in the case of a road network. This fact makes route instructions an interesting research subject from the perspective of Linguistics (See also Chapter 3).

2.2 COMMUNICATION

Humans are social animals spending a considerable amount of their time interacting with one another [Levinson, 2006]. From an evolutionary standpoint, it has been argued that this "interactional intelligence" made the development of language possible in the first place. This would imply that interaction is prior to language ([Grice, 1989], [Levinson, 2006]). This section discusses conventional models of communication and their relevance to this thesis.

2.2.1 *Shannon's Model*

A first attempt to formalize the notion of communication and transmission of messages led to the mathematical theory of communication proposed by Shannon [1948]. The model is a theory of communication in the context of telecommunication, such as the transmission of telephone or television signals. It introduces and formalizes the notion of entropy as a measure of disorganization and its negative, i.e., information as a measure of organization [Wiener, 1989]. The model (See Figure 1) consists of a transmitter (sender) who encodes a signal so it can be sent over a (potentially noisy) channel to a receiver who in turn decodes the message. The model requires that both the sender and the receiver have the same encoding and decoding rules, i.e., a common language [Brennan et al., 2010]. Note, the term signal as used by Shannon is not the same as the notion of a signal used in this thesis (See Section 4.2).

Shannon's communication model allows only for a quantitative measure of information and is biased towards the producer of the information.

Shannon's approach has several drawbacks. For example, it allows only for a quantitative measure of information (the less probable a message is the more information it contains). The theory does not account for the effect a message has, i.e., its pragmatic content [Frank, 2003]. Even though Shannon acknowledged the fact that messages have a meaning he saw semantics as being "irrelevant to the engineering problem" [Shannon, 1948].

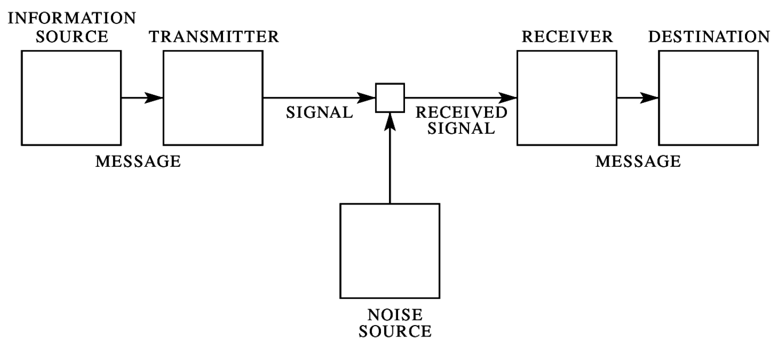

Figure 1: The communication model proposed by Shannon [1948]

Another consequence of the Shannon's model is that language speaking (encoding) and listening (decoding) are separate from each other, i.e., they are independent activities. This is contrary to how human communication takes place because it involves interdependent activities by both speakers and hearers. Critiques of Shannon's approach, most notably Grice [1989] and Clark [1996] have long called for an alternative to his view, in particular because it has had a long standing impairing influence on linguistic models of communication.

Clark argues that language is a collaborative joint action between two or more conversation partners and that meaning is generated (grounded) as a result of this interaction.

In the terminology of this thesis meaning is negotiated through the process of communication. It is noteworthy to mention that this stance is also conveyed by the literal Latin meaning of communication, i.e., "to make common". To make something common implies a joint effort of people.

2.2.2 *Game Theory*

Game Theory is "the mathematical study of interaction among independent, self-interested agents" [Shoham et al, 2008, p.48]. In this context, "self-interested" refers to the idea that each agent has an interest (e.g., in the form of individual preferences) in achieving a specific state of the world. An agent may use a set of actions available to them to transform the current state of the world into the desired state.

The standard approach to model the interest of an agent is utility theory. Formally, utility theory consists of a set of decision alternatives X and a binary preference relation on the elements contained in X. From these two constructs one can then deduce theorems [Fishburn, 1968].

In game theory, one can distinguish between non-cooperative and cooperative games. Note, "non-cooperative" does not necessarily mean that agents have conflicting interests. Likewise, the term "cooperative games" does not imply that all agent's interests align at all times. The difference is that in non-cooperative games the agent is modeled individually (i.e., their beliefs, desires, etc.) while cooperative games assume a group of agents as the basic modeling unit [Shoham et al, 2008].

Game Theory models agents as having preferences acting towards a desired goal. Agents take other agents perspective into account and interact with each other.

Game theory distinguishes between several types of games. For example, in a so-called "common payoff game" [Shoham et al, 2008] agents coordinate their actions, such that the resulting state of the system is maximally beneficial to everyone.

Agents may use different strategies to achieve their desired goals. For example, if each player makes the best possible move by taking into account what the other agents would do, and neither player gains by changing the strategy, the game is said to be in a "Nash equilibrium"[Osborne et al., 1994]. The outcome to such a strategy is defined as the equilibirum point [Davis, 1997].

In conclusion, game theoretic models assume that (1) agents have individual preferences and act towards achieving a desired goal, (2) agents can take another agent's perspective into account, and (3) agents can interact with each other to achieve their goals.

2.2.3 *Open Systems Interconnetion (OSI) Model*

The Open Systems Interconnection model (OSI) is a conceptual model aimed at standardizing the functionality of a communication system [Zimmermann, 1980]. The OSI architecture consists of seven layers grouping together functions that are different with respect to the performed process or the technology involved (cf. [ISO - OSI, 1994]):

- Layer 7 - **Application**: The top layer is concerned with application-specific exchange of information (e.g., HTTP).
- Layer 6 - **Presentation**: This layer is concerned with the transformation between application and network-specific formats of data[1] (e.g., transformation from and to XML).
- Layer 5 - **Session**: This layer is concerned with managing connections between entities. This includes services to synchronize data transfer.
- Layer 4 - **Transport**: This layer is concerned with the segmenting of data and responsible for end-to-end error recovery.
- Layer 3 - **Network**: This layer is concerned with the routing of data.
- Layer 2 - **Data Link**: This layer is concerned with encoding and decoding bit streams into frames and adding check-sums. It detects possible errors that may occur in the physical layer.
- Layer 1 - **Physical**: The bottom layer consists of a bit stream (e.g., light signals) and provides the means to manage physical connection between linked entities.

The functionality of the model can be explained as follows. Each layer services the next higher layer and is served from the next lowest. Instances on the same level can communicate horizontally using protocols that define rules on how to process data. The actual exchange of information between two communicating entities, however, happens vertically. Information exchange starts at the top layer (application) on the one side, passed through the lower layers to end up at the bottom layer (physical), then handed to the other entity, passed through the intermediate layers to finally end at top layer on the other side (See Figure 2).

2.2.4 *Summary*

This section presented conventional approaches to communication. In particular, Shannon's communication model, game theoretic no-

1 http://en.wikipedia.org/wiki/OSI_model

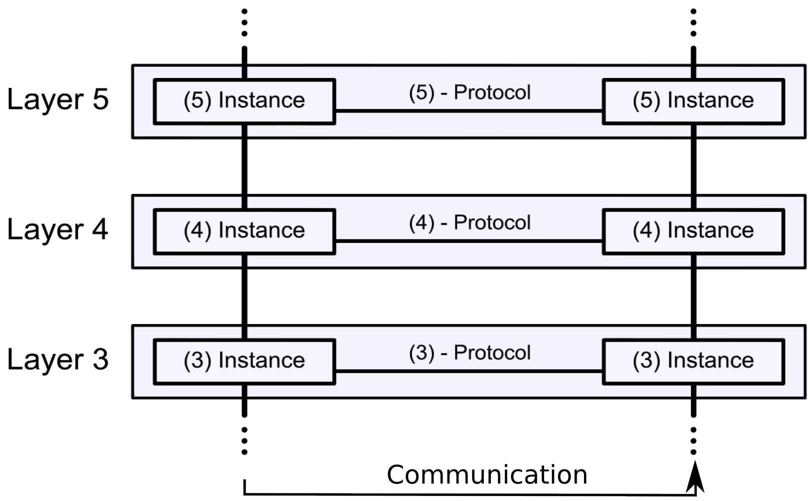

Figure 2: The Open Systems Interconnection (OSI) Model. Conceptual View adapted from: http://upload.wikimedia.org/wikipedia/commons/4/41/OSI-model-Communication.svg

tions of communication, as well as the Open Systems Interconnection (OSI) model.

The OSI model advocates the separation of different features of communication into layers. Each layer's function is limited to a specific set of tasks and each layer can only directly communicate with neighboring layers to exchange data. The model allows a clear specification of the semantics of each task and benefits from its "divide-and-conquer" approach to the complex problem of information exchange.

Game Theoretic approaches model all agents in a communication setting explicitly, thus make all contributors to a conversation equivalent. This includes a model of an agent that has specific and individual preferences about the state of the world and acts towards achieving them. In addition, agents can interact with each other by considering not only their own but also the other agent's goals.

While Shannon's model was the first formal approach to model communication it is biased towards the producer of information. In a sense it is a uni-directional model because it does not put potential hearers on par. In addition it does not account the meaning of information. How is meaning created and how is it inferred from a message?

2.3 MEANING

The previous section discussed conventional approaches to communication. However, such models do not explain how communicated content is given meaning by the speaker, and how receivers of information infer its meaning. This section describes theories concerned with identifying the processes by which meaning is created.

2.3.1 *What is meaning?*

Meaning can be understood as the relationship between mental constructs (See Figure 2.1), reality, and the words we use to describe both mental constructs and reality. In semiotics ("The science of signs", cf. [Eco, 1979]) this triad of relations is called a sign and can be formally described by the triangle shown in Figure 3. This model proposed by Peirce and many others [as cited in Chandler, 2007] contains three elements:

Sign: The triadic relationship between a) mental constructs, b) reality, and c) words humans use to describe reality and mental constructs.

1. The representamen describes how a sign is represented symbolically, e.g., the word "flower".
2. The interpretant describes how the sign is interpreted, e.g., the concept one has about a flower in their mind.
3. The object describes what a sign represents in reality, e.g., the actual real-world object flower.

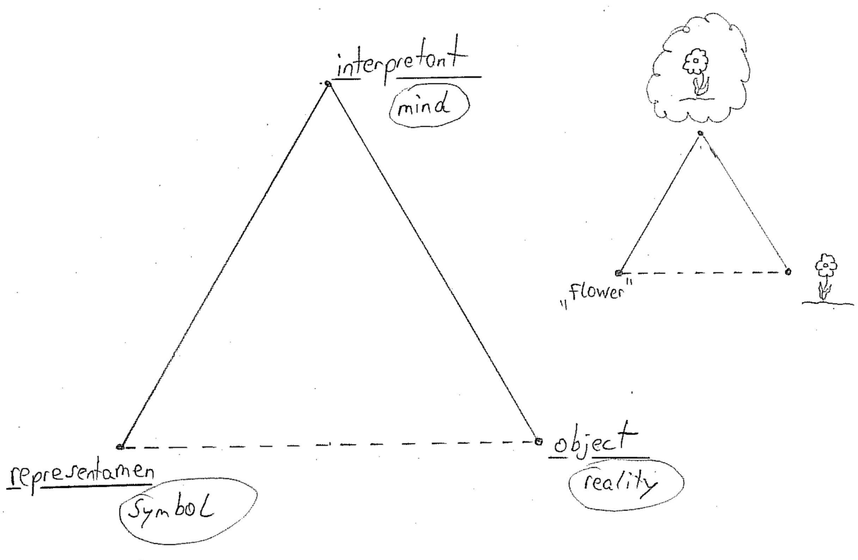

Figure 3: The triadic relationship between symbol - representamen (flower as a word), mind - interpretant (flower as a concept), and reality - object (flower as an object)

The following example (adapted from [Chandler, 2007]) illustrates how the process of interpretation of a sign works: Imagine a wooden box with a label attached to it. Reading the label (representamen) evokes a picture in your mind (interpretant) that resembles the object you suspect to be contained by the box. The box, however, cannot be opened and the object (reality) contained by it cannot inspected directly, because if you could, there would be no need to use a sign to represent it's content.

Our daily experiences with the world shape the way we conceptualize and think of objects. Note that objects can be both in the concrete or abstract domain, e.g., mathematical concepts (cf. [Lakoff and Nunez, 2001]). In the case of a concrete object its conceptualization forms a mental representation shaped by real-world experiences (e.g., landmarks encountered during way-finding). This is represented by the right side of the semiotic triangle in Figure 3. Similarly, the process of decoding a sign (e.g., reading an instruction) evokes a concept in the mind of the decoder (reader). This is represented by the left side of the semiotic triangle in Figure 3.

Humans use language to express their intentions [Grice, 1989] and to convey information. They communicate messages that can be used by others, e.g., for decision taking. Without access to adequate information, our lives would be a very ineffective endeavor [Wiener, 1989]. Persons engaging in a communication are bound to infer meaning from utterances produced by their conversation partner.

The presented model exhibits at least two problems. First, the connection between the symbol (representamen in Peirce's terminology) and reality is not straight forward (See bottom side of the semiotic triangle in Figure 3). One general problem with words is that they do not have any meaning inherently attached to it [Brennan et al., 2010]. There is nothing that intrinsically connects the word "flower" with the real-world object flower. This implies that the relationship representamen (symbol) and interpretant (mind) "is ontologically arbitrary" [Chandler, 2007].

Second, how can one make sure that the same symbol evokes the same concept across several people? The decoding of a symbol has to use the "detour" via the interpretant (mind) of its decoder. If one person attempts to decode a sign, e.g., a word, it evokes a concept that is experientially connected to either a concrete or an abstract object. However, from the decoder's perspective there can be no verification of the intended conceptualization by the producer of the sign. In other words, there is no way of telling whether the decoder of the sign conceptualizes the object as the producer of the sign.

2.3.2 *The Grounding Problem*

The grounding problem is a direct consequence of the arbitrary connection between the symbol and the object (See Section 2.3.1): How can meaning be created in the first place? Traditional approaches have attempted to ground meaning in the machine by using formal specifications. Humans, however, achieve symbol grounding without the use of formal specifications. This section contrasts theories addressing both views, i.e., grounding from the perspective of the machine vs. man.

Searle's Chinese Room Argument.

Searle's [1980] famous "Chinese Room argument" demonstrates that formal symbol manipulating systems (e.g., a computer) cannot know the meaning of its internally stored representation:

> Imagine you do not speak nor understand a single word of Chinese. You sit in a room with a book that contains nothing than a set of Chinese characters. You also find a list of instructions in your native language. Through a slit in the door you receive a sheet of paper full of Chinese characters. The instructions allow you to correlate characters you find on the sheet of paper with characters in the book. Note that the correlation process is based on matching shapes alone, since you don't know the meaning of the Chinese characters!
> Using the instructions, the book and the sheet of paper you can also produce Chinese characters yourself. You are instructed to write them down on yet another sheet of paper. Once you are done you pass it through the slit in the door.
> Unknown to you, the people outside the room call the characters on the sheet of paper they gave to you "questions" while they call the characters on the sheet of paper they received from you "answers to the questions".
> Lets assume the Chinese book you have at your disposal is called a story by the people outside the room. For them it seems that you have answered all their questions they had about the story written in the book. The problem, however, is that you don't understand the meaning of the story. You simply transformed one set of meaningless characters into another set of meaningless characters by applying rules you found in the set of instructions.

Searle [1980] uses his argument as a metaphor for computers that can process instructions (i.e., a program) to produce answers to questions somebody (e..g, another human or computer) asks about a given knowledge base (e.g., a database). The computer, however, does not know the meaning behind the process.

In this context, Searle introduces the notion of weak and strong Artificial Intelligence (AI). Weak AI can simulate mental processes (e.g., thinking) by using computational procedures which can then be tested and verified for correctness and consistency. Strong AI, however, can simulate mental processes such that the entity who does the simulation can actually "understand" what they do in a fashion comparable to humans. The distinction strong AI vs. weak AI, however, is not clear cut. For example, Brooks ([1991],[1995]) argued for forms of artificial intelligence not based on explicit representations (See also [Braitenberg, 1986] and [Both et al., 2013]).

While strong AI might not be practically feasible (at least now), the problem of how to attach semantics to meaningless symbols remains, even with the notion of weak AI. This challenge has become known as the symbol grounding problem [Harnad, 1990]. There are several approaches one could attempt to give symbols meaning in a computer:

Symbol Grounding in the Machine

- One Symbol can be grounded in another symbol. Nothing is gained, however, as the Chinese room argument itself illustrates. In a sense, the Chinese characters are grounded in the instructions on how to transform them. Still, the instructions need to be grounded in something else, leading to the problem of infinite regression (A is grounded in B is grounded in C is grounded in ...). The entity (man or machine) sitting in the room knows nothing about the meaning of the performed processes.

- One can specify a taxonomy of symbols using a recursive subclass relation. For example, we could say that the symbol combination "Strasse" is a "road". A road might then be specified using a more general concept, such as "a way". This concept in turn might then be further abstracted until we have reached some general upper term, e.g., "thing" (cf. [Fellbaum, 1998]). This approach has at least two problems (See Kuhn [2005]). First, it assumes that upper level concepts are universal across all domains (which they are clearly not). Second, it is not expressive enough to capture multiple meanings the same entity can have, nor can it describe processes in which entities take part. One could also argue that this approach is just a more sophisticated way of grounding symbols through other symbols, leaving the problem of infinite regression unsolved.

- Wierzbicka [1996] suggested a list of around 100 words universal across all languages. It is proposed that with this list of words (assuming one can agree on their semantics), one could then construct all other meanings as combinations of the basic words.

- Kuhn [2005] proposed to ground symbols in observations, e.g., by using sensor networks. The method relies on the fact that any observation involves a measurement and any measurement in turn is grounded in a physical process related to the Earth. For example, a sensor that associates the symbol "green" with a specific hue, saturation, and lightness value of an object can be said to have grounded the meaning of the color green.

2.3.3 *Human Strategies to Ground Meaning*

Humans generate meaning although they have no formal specification for the semantics they use. Steels [2008] in the tradition of Brooks ([1991],[1995]) notes that humans are capable of generating their own semantics while computers need a programmer to define the meaning of a symbol. Cangelosi [2006] makes a distinction between physical and social symbol grounding. For an overview see also Glenberg and Robertson [2000], Steels [2008] and Coradeschi et al. [2013].

2.3.3.1 *Physical Symbol Grounding*

Physical symbol grounding is "the ability of each individual to create an intrinsic link between world entities and internal categorical representations" [Cangelosi, 2006]. There exist several theories how this is done:

Physical Grounding.

- Meaning is grounded through perception. For example, this would assume that language acquisition works through a mapping process from real-world objects to words. However, it is not clear how this is achieved. For instance, how are rich sensory inputs "stripped down" to a symbolic representation. How does it work the other way around?

- Meaning is grounded through multi-dimensional stream encoding. This would assume that the collage of input streams (words, images, sounds) of which reality exists grounds meaning. However, empirical observations contradict this claim. For instance, a child watching a foreign TV program cannot learn the language, although it has been exposed to multiple streams of information (cf. Pinker [1994]).

- Meaning is grounded through a concept called closed loop semantics which grounds terms in observations and actions [Frank, 2003]. Using the principle of closed loop semantics an agent grounds terms by linking reality to beliefs via observations and beliefs to reality via actions.

- Meaning is grounded through embodied cognition, particularly through affordances [Gibson, 1979], scripts [Schank and Abelson, 1977] or frames [Minsky, 1974], and image schemata [Johnson, 1987]. According to this view meaning is given to a symbol through our experiences with reality. This implies that meaning is generated through the evaluation of a particular situation in which we perceive particular actions (affordances), physically constrained by our bodies (image schemata), and socially constrained (optional) by scripts/frames. This "experimentalist" [Lakoff, 1987] approach has been verified through empirical observations and is the most universal model of the physical grounding process to date.

2.3.3.2 *Social Symbol Grounding*

Social symbol grounding refers to "the process of developing a shared lexicon of perceptually-grounded symbols in a population of cognitive agents" [Cangelosi, 2006]. For example, Chandler [2007] noted that "meaning of words is determined intralinguistically not extralinguistically". This means that the meaning of words can be established through repeated interaction with other people.

Social Grounding.

Steels [1999] demonstrated, using embodied robotic agents that it is possible for each agent to first each individually perceive and conceptualize features of reality, and then invent and share a common language. This common language allowed them to communicate about the individually perceived real-world entities. In addition, Galantucci [2005] showed that an alternative communication system emerges if humans need to coordinate joint actions, but cannot access already established communication systems (e.g., speech and writing).

From the perspective of linguistics, there are two opposing views on language (cf. [Clark, 1996]). On the one hand, there is the *language-as-product* view. It studies language "abstracted away from speakers, times, places, and circumstances in which it might have been produced". This pretty much resembles Shannon's approach to communication (See Section 2.2.1). As such the meaning of words is conventional and additional meaning can be produced by the combination of words according to some rules. On the other hand, the *language-as-action* view opposes the notion that language is only a static structure (product) but argues that language needs to be studied by taking into account the conversational partner's attitudes, the circumstances, and interactions of the conversation (cf. Pragmatics [Levinson, 1983]). The notion of language-as-action view implies that the meaning of terms (words, sentences) is not contained by the words themselves but needs to be determined by an interactive process, i.e., the communication between at least to individuals.

Language-as-product vs. Language-as-action

Using the example of navigating a ship, Hutchins [1995, pp. 232-39] provides empirical evidence of how people negotiate the meaning of terms. Ship navigation is an elaborate task that requires the seamless interaction of several people. The use of a partially controlled natural language [Tobias Kuhn, 2014] and standardized procedures (e.g., readbacks and acknowledgements) are among the requirements that make this task feasible. Despite the artificially constrained context, there is still some room for interpretation of terms. As a consequence they need to be negotiated among the interacting participants of a given conversation.

A shared perspective is essential for the successful completion of a given task. However, finding a shared perspective needs only be as good as it needs to be for a given task [Simon, 1998]. This implies that if all participants of a conversation agree that some given information is suitable, it is, by definition, suitable in this context. Note that this does not imply that meaning once it has been negotiated is not subject to revision.

An agreement among communication partners on the meaning of a given term can be achieved through one, or several of the following:

- The interaction of agents that need to find a common perspective (Do you mean X or Y when you say A?).
- The reference of an utterance to objects in the real world [Clark et al., 1986].
- The relation between cognitive artifacts (e.g., a protractor and a map).
- Previously established facts during a communication [Clark, 1996].
- The context or structure of a task at hand (e.g., navigating a ship).

In addition, Levinson [2006] has shown that the negotiation of the meaning of terms is universal and independent of the communication mode. For example, the negotiation of meaning can also be witnessed during the process of signing between two people who share little or no common culture or (sign) language. Another example would be chat rooms where "ambiguous" statements, such as irony often require elaborations on the intended meaning of a given utterance.

2.3.3.3 *Relation between Physical and Social Grounding*

Note, one could argue that social grounding does not imply that terms are also physically grounded. This is true, in the sense that at least one agent taking place in a social grounding process has to also physically ground the negotiated term, e.g., experimentally (See Lakoff [1987]). If this was not the case the entire system would not be grounded in physical reality. However, this does not imply that every agent needs to physically ground the meaning of words in order to use them in a meaningful way. In this context, Putnam [1975] suggested the term "division of linguistic labor" and argued that "for everyone to whom gold is important for any reason has to acquire the word gold; but he does not have to acquire the method of recognizing if something is or is not gold".

2.3.4 *Summary*

This section discussed the grounding problem that arises as from the need to give abstract symbols a meaning. This section also contrasted conventional approaches to symbol grounding in man and the machine. Specifically, humans can ground meaning either physically, i.e., through the creation of a link between real-world objects and corresponding objects, or socially, i.e., through the creation of a shared lexicon. Such a shared lexicon is created in a collaborative fashion, i.e., meaning is negotiated during communication. The following next section develops a theory of meaning negotiation, in particular its components - signals and a "theory of mind".

2.4 A THEORY OF MEANING NEGOTIATION

This section introduces the theory of meaning negotiation. It consists of signals and a "theory of mind" both of which are used to socially ground the meaning of terms among the participants of a conversation. The here presented approach is a pragmatic one [Levinson, 1983], as the successful communication of information depends on the user's ability to interact with other conversation partners and takes the user's intentions into account.

Clark et al. [1986] proposed that the attempts to agree on the meaning of a term during a communication are essentially "joint hypothesis testing". The goal for participants engaging in a conversation is to establish an agreement on the interpretation of a term. To achieve this, the meaning for the terms in question has to be negotiated. For example, if person A presents a statement (hypothesis: I believe the utterance i is of help to you) she cannot per se assume that B understood i unless B indicates so (as such B confirms A's hypothesis). On the other hand, if B indicates that she did not understand i (as such B rejects A's hypothesis), A has to revise her hypothesis, construct some new utterance i' and present it again. Clark [1996] suggested that people engaged in a communication indicate how the conversation should progress using a device called signals.

Furthermore, Fussel and Krauss [1992] have shown that if a person presents a message to somebody else, the message (and its intended meaning) is designed such that it fits the assumed needs of its receiver. This anticipation of another person's state of knowledge and beliefs is called a "theory of mind".

The following subsections discuss the notions of signals and theory of mind in detail and present empirical evidence for their existence.

2.4.1 *Signals*

For Clark [1996] signals are coordination devices used by people engaged in a conversation to coordinate their next steps. As such, they are "acts by which one person means something for another". The successful exchange of signals helps to advance the communication.

Signals can be compared to the concept of a feedback mechanisms. In system theory, the principle of feedback is a "method to control a system by reinserting results from its past performance" [Wiener, 1989]. If we accept the idea that communication is a system with people as their parts (such as the pragmatic view on language, cf Levinson [1983]), signals are used to control or change the progress of the communication by evaluating their success or failure and subsequent adjustments. This can be described as a mutually depended observe-act cycle in which both conversational partners evaluate signals (observe) and then adapt their utterances (act) according to the perceived signal. A changed utterance then can potentially generate new signals which in turn have to be interpreted again.

Brennan et al. [2010] noted that signals should follow three criteria if they are to be perceived as an communicative act:

- Signals must be informative, i.e., have some information attached to it.
- The addressee must attend to the information attached to the signal and be able to recover it (See also Austin [1955], Clark [1996], and Levinson [1983] for similar arguments).
- A presented signal must be modifiable or adjusted if necessary, i.e., it is a function of the interactive communication process and not a static concept.

If any of these conditions is violated during the communication, signals become meaningless and the conversation is not likely to be successful. To understand what signals are it is worthwhile to first look at the theory of speech acts.

2.4.1.1 *Speech Act Theory*

Speech acts could be interpreted as signals from the speaker's perspective. Speech acts do not consider the fact that for a successful communication a hearer and their signals are also required. Signals are an extension to speech act theory in an attempt to include both the hearer and the speaker in a communication.

Austin's [1955] theory of "how to do things with words" is the precursor to Searle's [1969] Speech Act Theory. Prior to Austin's theory, a wide-spread hypothesis in linguistics was that statements have only a meaning if they can be tested for truth or falsity [Levinson, 1983]. According to this view, utterances such as the famous "I now pronounce you man and wife" in U.S. wedding ceremonies cannot be verified to be either true or false, thus they are meaningless. Austin noticed that there are two types of utterances, one of which "just says something" and others that that "perform an action" (performatives). Although a "performative" is not subject to verification in the classical logical sense, it still can be "null and void" [Levinson, 1983] if they do not adhere to certain felicity conditions [Austin, 1955]:

- There must be "an accepted conventional procedure having a certain conventional effect" in which the performative is uttered. This condition stresses the hearer of a performative who has to accept the effects to make it valid. In particular, Austin notes that it needs to be both heard and understood.

- The persons and circumstances involved in the performative act must be appropriate.

- The procedure that includes performative acts needs to be executed by all participants correctly and completely. As such, the performative act needs to trigger an "uptake" by the addressee, otherwise it is not complete.

- The persons involved in the procedure must be sincere, i.e., the intentions of the speaker must match the actual procedure ("do not say what you know to be false") and if the act implies a future conduct they must follow the conduct accordingly ("don't promise something you will not do").

Later, Austin realized problems with his initial theory and attempted to identify the elements of a more general theory that can explain all the "senses there are in which to say something is to do something". The revised theory includes three acts that can be performed with an utterance:

- **The Locutionary Act:** This is the physical utterance made by the speaker. At this level a speaker says words with certain literal sense and reference. Austin notes that this is "roughly

Locutionary Act: Literal Meaning. Illocutionary Act: Context-dependent Meaning. Perlocutionary Act: Effect on the state of the world.

equivalent to meaning in the traditional sense". This assumes that utterances have (only) a literal meaning.

- **The Illocutionary Act:** This captures the way how the locutionary act is used. In addition to their literal meaning (locution) messages may have additional meaning that depends on the context of the utterance. For example, while it might be clear what "I am cold" means literally, it is not so easy to determine whether it is meant as a request to shut the window or just a statement of discomfort.

- **The Perlocutionary Act:** This is the effect an act (locutionary and illocutionary) has on an audience. The perlocutionary act may change the beliefs or actions of its addresses. It determines the consequences of an utterance. If somebody tells me to shut the window, I may or may not comply. In any case the effect of the utterance is the perlocutionary act.

Searle's [1969] notion of speech acts extends and refines Austin's work. Searle proposes five classes of actions a speaker can perform to mean something:

- **Representatives**: They are meant to inform the receiver of a message about something. For example, a speaker can assert some belief ("It is cold today"), provide a conclusion, or offer a description (cf. [Yule, 1996]).

- **Directives**: They are meant to get the receiver of the message to do something. For example, a speaker can request something ("Close the door, please") or ask questions ("How do I get to A?").

- **Commissives**: They indicate that the sender of the message is committed to something. An example is the statement "I promise I will have finished my work by tomorrow".

- **Expressives**: They express a psychological state [Levinson, 1983]. For example, a speaker can thank, apologize, or congratulate someone for something.

- **Declarations**: They have a direct effect on the state of the world, i.e., they change the state of the world. For example, declarations of war.

2.4.1.2 *Signals: An Extension to Speech Act Theory*

Both Levinson [1983] and Clark [1996] argue that Searle's work on speech acts neglects the interactive nature of communication. This fact can be illustrated by the following dialog:

- T: Can you tell me how I get to the old opera?

- S: Excuse me?

- T: To the old opera

- S: To the old opera; straight on [...]

In terms of Searle's speech act theory (See discussion above), T has performed a directive with the utterance "Can you tell me how I get to the old opera?". However, this notion does not consider the fact that an uptake by the addressee S is required. By uttering "what?" S indicated that she has not understood T's question and needs clarification. In this example, T's intention to ask S about instructions on how to get to the old opera is only recognized by S after requesting her to restate the question. Austin[1955] noted that an illocutionary act cannot successfully be performed by a speaker, if its audience cannot hear the utterance or simply does not take it in the sense as intended by the speaker. However, Searle's notion of speech acts does not model the uptake explicitly.

To remedy this shortcomings Clark[1996] introduces the notion of signals. Signals are more than speech acts in at least two ways:

- According to Searle, speech acts are bound to utterances. In contrast, signals are independent from a particular modality such as speech. For example, a signal in the form of a gesture can also convey meaning but does not produce a physical utterance.
- Speech acts are centered around the speaker and do not explicitly model the hearer. Clark [1996] argues that messages need to be secured by the speaker and accepted by the hearer. Speech acts do not comply to the assumption that meaning of terms is negotiated between individuals. If participants of a communication do not "jointly establish" meaning, it does only take place in a uni-lateral fashion. Speech acts account for the speaker's meaning, such as "by presenting i to B, A means for B that p". The meaning of the presented statement ("i means that p"), however, is not accounted for from the perspective of the receiver of the statement. Clark noted that there is a difference between the illocutionary act, i.e., the intended meaning by the speaker, and the illocutionary effect, i.e., the received meaning by the hearer. Speech act theory assumes an alignment between the speaker's and receiver's meaning. As such they do not consider potential divergences in meaning.

Signal Recognition

Clark [1996] concludes that speech acts are not expressive enough to suffice for the notion of communication as a process where meaning is negotiated between participants. Clark introduces the notion of joint actions between speaker (S) and hearer (H). Note that the hearer and speaker should not be taken literally. The speaker cannot speak but gesture instead and the hearer can not hear but recognize a gesture. Joint actions are used to achieve a "mutually desired goal", e.g., to solve a particular problem. Signaling can occur at all four levels of a joint action (See also Figure 4):

- Level 1 - Execution (S) and Attention (H): At this level S executes some behavior b for H to perceive; H in turn attends to the execution of behavior b. For example, if S utters a sound or exhibits a movement, H needs to hear the sound or see the movement for a successful joint action on level 1.
- Level 2 - Presentation (S) and Identification (H): At this level S presents a signal s for H to identify. H is required to identify s. For example, the behavior b intended to be a signal s should be identified by H not just as a behavior but also as a signal. Note that on this level the actual meaning of the signal is not yet identified.
- Level 3 - Meaning (S) and Understanding (H): At this level S means that p by presenting signal s. H is required to understand that p is the case from signal s.
- Level 4 - Proposal (S) and Consideration (H): This level resembles the illocutionary act discussed above but additionally includes the notion of uptake by H. For example, by presenting a signal, S proposes, suggests, or asserts a specific action to be taken into consideration by H.

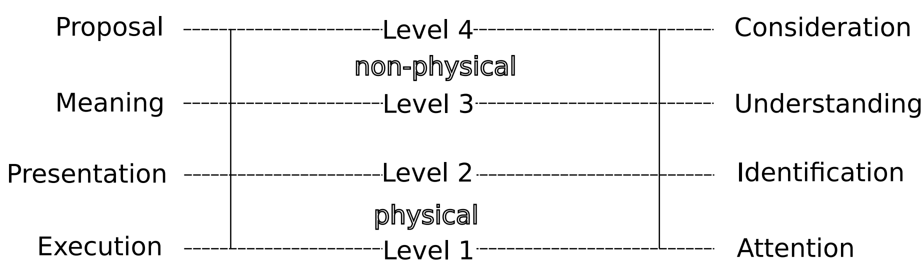

Figure 4: Layers of Abstraction according to Clark [1996]. Levels 1 and 2 are physically observable. Levels 3 and 4 are not physically observable.

If any of the the four steps fails the entire communication, is not successful. The four levels exhibit upward causality, upward completion, and downward evidence [Clark, 1996]. Thus, S must get H to attend her voice (level 1) so H can identify the word(s) S is presenting (level 2). Identification of the words is also required if S wants H to recognize the meaning of the utterance or gesture while this in turn is the requirement for the action that follows. Similarly, evidence that H recognized the intended meaning by S is an indication that H has also successfully identified the signal presented by S which in turn is evidence that H has attended the behavior executed by S.

Note that Clark's model can be compared to the Open Systems Interconnection model (OSI) introduced in Section 2.2.3. Similar to Clark's approach, the OSI model exhibits upward causality, upward completion, and downward evidence.

Signal Tracks

Signalling in communication occurs on two collateral (in the sense of parallel) tracks [Clark, 1996]. In track 1, participants of a communication try to "carry out official business", i.e., they try to construct communicative acts. For example, in a scenario where person T asks S for the way to some destination, the official business is the route description as a response to T's inquiry. Signals that fall in track 1 are usually meant to assert or ratify a belief. For example, "you turn right at the intersection" is the assertion of the belief that this helps the wayfinder to navigate. In track 2, participants are concerned with "attempts to create a successful communication", i.e., they try to construct meta-communicative acts. It is in track 2 where people ask for confirmation of an utterance or provide evidence that understanding was achieved.

Clark [1996] notes that signals in track 2 can address any of the levels of joint actions mentioned above. For example, if B says "you said turn right?", the utterance can refer to each one of the four levels at which a joint action occurs, i.e., level of consideration (level 4), understanding (level 3), identification (level 2), and attention (level 1). Also, the collateral nature of signals implies that any statement that is meant to talk official business, implicitly asks for its confirmation on all four levels of the joint actions. This is due to the fact that persons attempt to ground (at least implicitly, e.g., by the absence of negative evidence) the meaning of every statement before the conversation can advance. Grounding can only occur if the hearer of a statement does or does not signal understanding on his side.

2.4.1.3 *Common Ground*

How do people realize that the negotiation has ended for a given installment, i.e., have agreed on its meaning? This leads to the notion of common ground. Stalnaker [2002] defines common ground as "the mutually recognized shared information in a situation in which an act of trying to communicate takes place". To a certain degree, common ground can exist prior to a conversation. For example, Steels [1999] has shown that members of a community language share some common form of mutually understood concepts that are semantically defined.

In addition, common ground can develop during conversation. First, common ground is inferred from what both communication partners can assume to be known based on available evidence. This evidence can include the knowledge that both partners are from the same city, speak the same language, or are located at the same place where the communication takes place (cf. [Clark, 1996]). Second, the interactive process of meaning negotiation that consists of signaling and a theory of mind (See Section 2.4.2) also contribute to the development of a common ground. In other words, the progress of the conversation (signals used and possible responses) indicates whether some negotiated statement get added to common ground (See Chapter 4).

2.4.2 *Theory of Mind*

Most people have the ability to attribute mental states to other people. For example, Melzoff [1995] has shown that children as early as 18 months can differentiate between the surface behavior and intentions of people. Examples of the states we attribute to others are intention, beliefs, and knowledge. This ability of humans is also called **theory of mind**. Note, the concept of a theory of mind is also implicitly modeled by game theoretic approaches to communication (See Section 2.2.2). Thus, the actions of an agent depend on what they believe other agents would do in a certain situation.

Although mental states are not directly observable they are used to predict [Premack and Woodruff, 1978] and make sense of other people's behavior [Baird and Baldwin, 2001]. For example, if person A knows some fact x and person B knows that A knows x (See Figure 5) it could influence B's future behavior. During a game of chess, for instance, I may benefit from the fact that my opponent knows that I know that some move m leads to an unfavorable position.

Note that unlike projection (the assumption that another person's mental state equals my own mental state, cf. [Krueger, 2007]), theory of mind requires to separate one's knowledge, feelings, and desires from somebody else. Malle [2013] mentions at least three sources of information humans can use to successfully apply theory of mind:

- **General Knowledge:** For example, agent A would not be happy if I gave him the wrong directions.
- **Agent-Specific Knowledge:** For example, if I know that agent A is a professional geographer I expect him to have a reasonable degree of spatial knowledge.
- **Perceived facts of a given situation:** For example, agent A signaled me that p or not p. Thus, the agent might not be satisfied with a presented assertion at a given level of detail.

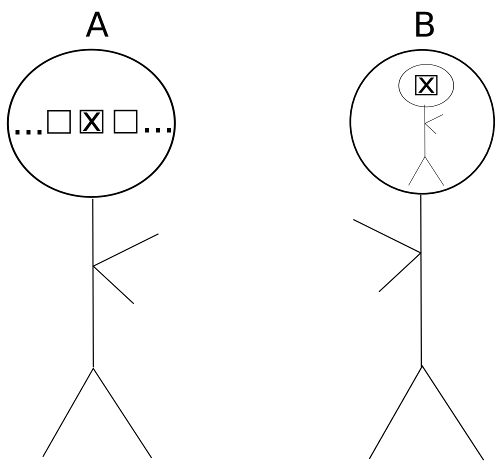

Figure 5: Illustration of theory of mind (ToM): "A knows that x" (left) and "B knows that A knows that x" (right)

The application of theory of mind could (potentially) lead to a scenario where infinite regress occurs, e.g., A knows that B knows that A knows that ... and so on. In reality, however, this is not likely to happen.

Consider the following: There exists a fixpoint $x_n \iff f(x_n) = x_n$. Consequently, $f(f(...f(x_n))) = f^n(x_n) = x_n$. Applied to the theory of mind this means that at a certain stage of mutual state attribution there is nothing to gain by further imagining the other person's mental state, so the process stops. This is similar to the terminating condition specified for recursive functions. Note that the commitment to theory of mind does not necessarily mean that people keep and update complete models of another person's assumed knowledge. For example, Pickering and Garrod [2004] argued that modeling the other person's mind completely is cognitively too costly (cf. principle of economy [Simon, 1998]) to be applicable for all communication scenarios.

2.4.2.1 *Empirical Evidence for a "Theory of Mind"*

The previous section argued that mental models of another person's mind ("I know that he knows that x") is used for cognitive processes, e.g., decision making. This allows to apply a "theory of mind" to a conversation setting. For example, statements made by one speaker and intended for a hearer are influenced by what the speaker thinks the hearer expects to hear. This implies that the meaning of statements is not only a function of the speaker and the discussed content but also takes the hearer into account.

Audience Design

Fussel and Krauss [1992] as well as Isaacs and Clark [1987] showed that people design the content and meaning of messages according to their expectation of the person who is supposed to receive it ("audience design hypothesis"). For example, a speaker A who suspects hearer B to not be from the same town and as a consequence adjusts the presentation of information to a level that B is likely to be capable of handling. Clark [1996] proposed that evidence about a person's membership to a cultural community can determine what we assume the other person to know. For example, if person A believes person B is Viennese, a route description could include the actual names of landmarks ("walk towards Stephansdom") as opposed to generic descriptions offered to foreigners ("walk towards the major church"). Hahn and Weiser [2014] proposed a formalization based on a quantum theoretic model for the problem that involves perspective changes and changes in the level of detail of such referring expressions (Stephansdom vs. major church) during the exchange of route instructions.

Once a theory of mind has been formed it can be subject to change. Clark [1996] argued that direct personal experience during conversation can provide additional evidence about another person's state of knowledge. For example, if person A has used the term "Bim" to refer to a tram and person B signaled understanding, person A can assume "Bim" to be a good description ("Bim" is the Viennese term for tram). Naturally, each type of evidence can be used to change beliefs about the other person or their state of knowledge. For example, if person B fails to understand a term T chosen by A she may reject her hypothesis that B belongs to cultural community C. Also, the belief about B belonging to community C forms her beliefs about the terms she can use in order to be understood.

Mirror Neurons

There exists neurological evidence that the concept of a "theory of mind" is based on a certain type of neurons in the brain. These, so called mirror neurons, are activated both when one perceives and when one carries out an action [Rizzolatti and Arbib 1998]. Thus, from an neurological point of view, playing tennis actively and watching tennis passively are nearly the same. In order to understand and interpret somebody's actions I have to know what they mean. For example, I can only mean what being hostile means, if I have an idea of how to produce this behavior myself. The main advantage of such a mirror neuron system is that it does not require me to learn all possible signals there are. If I know how to do X, I also know how it looks like if someone else is intending to do X, i.e., they are dual.

In fact, proponents of the embodied mind theory ([Johnson, 1987], [Lakoff and Johnson, 1980], [Lakoff, 1987], [Brooks, 1991]) claim that almost any aspect of cognition is also deeply connected to the physical aspects of our bodies. For example, it has been shown that getting primed with words related to old will make you walk more slowly. Conversely, getting primed by walking at a slow pace for 30 minutes will make you think of words related to being old [Kahneman 2011].

2.4.2.2 *Strategies for Audience Design*

In addition to designing a suitable message for the hearer from the perspective of the speaker, successful communication also requires a listener to recognize the intentions a speaker has with a given utterance [Grice, 1989][Sperber and Wilson, 1986]. Grice [1989] termed the process of any implication made by the speaker and subsequent recognition of what is implied by the listener **conversational implicature**. In general, there is a difference between what is said ("saying") and what is implied ("implicating") by a given utterance [Clark, 1996]. For example, if a tourist stops me in the streets and asks "How do I get to Karlsplatz" I may reply "You can take U2". With this statement, I have only said that I believe that taking the subway (U2) will get the tourist to his preferred destination. What is implicated by this sentence, however, is that the subway still operates and that it indeed serves Karlsplatz. Otherwise the information would be of little help to the tourist.

Grice [1989] suggested four maxims that a speaker should obey to make the inference of what was implied by a contribution ("theory of mind") communication more effective:

1. **Maxim of Quantity:** A contribution should be as informative as it is required for the task at hand, but not more!

2. **Maxim of Quality:** A contribution should be truthful (Things that are known to be false must not be said).

3. **Maxim of Relation:** A contribution should be relevant.

4. **Maxim of Manner:** A contribution should avoid obscurity and ambiguity of expression, be brief, and orderly.

Grice mentioned that there is a general way how conversation partners can work out the conversational implicature, unless aforementioned maxims are not violated:

> "He said that *p; [...] he could not be doing this unless he thought that q; he knows (and knows that I know that he knows) that I can see that the supposition that he thinks that q is required; he has done nothing to stop me thinking that q; he intends me to think, or is at least willing to allow me to think, that q; and so he has implicated that q."*

In fact, Grice' statement implies a person to have a theory of mind. With "...he could not be doing this unless he thought..." he refers to the anticipation of behavior that follows from applying theory of mind, i.e., "he knows (and knows that I know that he knows).

Sperber and Wilson [1986] later claimed that all of Grice's principles could be unified under two principles of relevance:

1. Cognitive Principle: "Human cognition is geared to the maximization of relevance"

2. Communicative Principle: "Utterances create expectations of relevance"

Both approaches, Grice's maxims and Sperber and Wilson's notion of relevancy, however, were criticized by Clark [1996] who argued that the maxims are "paradoxically specified for the speaker" but do not consider the listener. As such, Grice assumes that the listener is always able to recognize what is implied if the speaker adheres to the proposed maxims. However, this would imply that messages have the same meaning across all domains and all speakers, i.e., all representaments (symbols linked to real world objects, cf Figure 3) are clearly specified and no ambiguities are present. In general, however, this is not always true. Often, words have multiple semantics such as "bank" which can be interpreted as "river bank" or "money institute", depending on the context.

2.5 CONCLUSIONS

This chapter reviewed theories related to communication of information relevant to this thesis. First, spatial mental representation that are the foundation on which meaningful communication can be based were treated. Section 2.1 discussed their development, nature, and ways how representations can be externalized. Second, in Section 2.2 conventional models of communication were discussed. This included Shannon's mathematical model of communication, game theoretic approaches and the Open Systems Interconnection (OSI) model. Third, the issue of meaning creation (See Section 2.3) was discussed. In particular, it addressed the problem of grounding meaning in formal systems (i.e., computers). A review of human grounding strategies, either physical or social complemented this discussion.

Based on the reviewed theories (Representation, Communication, and Grounding) a theory of meaning negotiation was developed. In particular, it was argued that this theory includes signals and a theory of mind. The main assumption of this thesis is that each installment of a set of route instructions is negotiated between the source and the target, i.e., both participants in a communication. The developed theory suggested that:

1. A model needs to put all contributors to a conversation on par. Conventional models (such as Shannon's approach, See Section 2.2.1) have failed to address this issue and often only consider the producer of an information.

2. A model needs to consider that communication partners use signals during the conversation (See Sections 2.4.1.2 and 4.2). Signals indicate the meaning of utterances and they suggest whether something has or has not been understood. In particular, they indicate how the conversation should progress. Signals can be of type 1 (concerned with official business), or of type 2 (i.e., concerned with meta-communication). Successfully negotiated terms are added to the common ground of all participants.

3. A model needs to address the ability of people to form beliefs about other people's beliefs (a "theory of mind"). This implies that people engaged in a conversation form an initial hypothesis about the knowledge or capabilities of their counterpart. For example, they form an image of the action and the appropriate level of detail they expect to be good or fitting somebody's expectations of the same action. In addition, a "theory of mind" is subject to revision in case an initial assumption is not met.

4. A model needs to consider the interplay between signals and a "theory of mind". Consider that some information source (S) formulates an appropriate action based on some information target's (T) assumed preferences. Then, S receives feedback (in the form of signals) whether the initial expectation was met or

not. As a result a "theory of mind" might need to be adjusted to further successfully contribute to the conversation.

The theoretical foundations developed here are later used to construct a conceptual model (See Chapter 4) of the communicatino process that is then formalized (See Chapter5). The next chapter, however, first gives an analysis of the theory of route instructions to argue that information systems that provide such information can benefit from the theory developed here.

3 WAY-FINDING INSTRUCTIONS

The preceding chapter developed a theory of communication that:

1. includes both agents, i.e., the hearer and the speaker, and proposed that the meaning of terms is negotiated between all participants of a conversation.

2. advocates the notion of agents that use signals to indicate how the conversation should progress.

3. assumes that agents model other agent's mental states (i.e., have a "theory of mind") and evaluate them to decide on further actions.

4. proposed that successfully negotiated terms are added to the common ground of both agents.

This chapter reports research findings concerned with the theory on way-finding instructions. The goal is to identify and analyze potential impediments for a cognitively adequate information system, as sketched in Section 1.1. In particular, it is argued that most of conventional research on way-finding instructions focused on studying their structural (i.e., static) parts. However, the interpretation of an instruction (thus its usefulness in a given situation) is not fixed, but a function of the user and their requirements. Therefore, an information system needs to offer ways allowing users to dynamically change information presentation to their needs. Although, there exist some models that mention the interactive nature of an exchange of way-finding instructions most of them fail to make the interaction processes explicit. In addition, this chapter identifies and discusses crucial elements necessary to provide cognitively adequate instructions. The findings from this chapter are one of the inputs for developing the conceptual model presented in Section 4.

This chapter is structured as follows. First, using a top-down approach, the question "What are way-finding instructions?" is answered. This includes a review of existing semi-formal classification schemes in a top-down manner. Second, formal models of route instructions are analyzed, including ways to generate and classify instructions automatically. In addition, approaches that account for the fact that instructions can be represented over multiple levels of detail are reviewed. Third, quality aspects that can arise during the presentation of instructions are discussed. This includes general and individual (context-dependent) principles. Fourth, a discussion of existing communication models in the context of route instructions is provided. The concluding section summarizes the discussed findings with respect to the requirements of a next-generation route information system as sketched in Section 1.1.

3.1 SEMANTICS

Existing literature uses the terms route instructions, route directions, or way-finding instructions interchangeably. Generally, way-finding instructions can be considered as a type of cognitive artifact. Norman [1986] defined a cognitive artifact as "an artificial device designed to maintain, display, or operate upon information in order to serve a representational function". Thus, cognitive artifacts can help to increase the (limited and individual) cognitive abilities of humans.

Way-finding instructions are cognitive artifacts.

More specifically, way-finding instructions are a cognitive artifact intended to assist humans with the navigation of a route between two or more points in geographic space. Formally, a route consists of a series of decision points at which the way-finder needs to perform an action in order to proceed. Thus, way-finding instructions communicate a sequence of such actions with respect to crucial decision points along a route. The following paragraphs attempt to describe the structural nature of instructions in a top-down manner.

Route Instructions can include prescriptive and descriptive statements.

Route instructions can be viewed as a special type of discourse that includes both prescriptive and descriptive statements [Tom and Denis, 2004]. Prescriptive statements mention actions a way-finder can carry out to move between several decision points while descriptive statements describe the environment (e.g., decision points) along the way.

This point of view is complemented by an early classification provided by Riesbeck [1980], who makes a distinction between three types of instructions.

Riesbeck's classification: Motions vs. Descriptions vs Comments

1. **Motions:** They "tell you what to do", e.g., "Turn right".
2. **Descriptions:** They "tell you what a place looks like", e.g., "There is bank to the left".
3. **Comments:** They refer to the process of direction giving/receiving process itself, e.g., "You cannot miss it".

A more detailed classification (from a linguistic point of view) analyzing the types of terms used in way-finding instructions is provided by Wunderlich and Reinelt [1982]. They defined four "functionally relevant" categories:

Wunderlich and Reinelt's classification: Nominals vs. Directives vs. Position Markers vs. Verbs of Movement

1. **Nominals:** Terms related to real-world entities (i.e., objects and their attributes) used for orientation. Examples for nominals are "large street", "intersection", or "traffic light".
2. **Directives:** Terms used to split instructions into smaller parts and indicate the beginning or end of temporal or spatial units. For example, "then" can be used to refer to an upcoming spatial object, while "until" indicates a temporal duration.
3. **Position Markers:** Terms indicating spatial relations, often correlating with directives. Examples for position markers are spatial prepositions such as "on the left", or "in front of".

4. **Verbs of Movement:** Terms such as “go”, “follow”, or “continue”.

Allen ([1997];[2000]) extended Wunderlich and Reinelt’s classification by adding “state-of-being” verbs and grouping directives and position markers into the class “delimiters”:

Allen’s extension includes state-of-being verbs vs. delimiters

1. **State-of-Being verbs:** Descriptions about the environment, such as the expression “There is a large owl in front of the building”.

2. **Delimiters:** They subsume Wunderlich and Reinelt’s directives and position markers. This class makes a distinction between distance and direction designations. Distance designations may come in the form of quantitative statements that are either spatial (e.g., “two miles”) or temporal (e.g., “five minutes”). Direction designations are essentially spatial relations and may feature absolute (e.g., “north of”), intrinsic (e.g., “to the left of you”), or relative (e.g., “to the right of the green building”) reference frames.

Another attempt to classify the structure of route directions was offered by Denis [1997], who intended to find the invariants in route descriptions that hold true over a variety of different instances of instructions. He analyzed a corpus of human-produced route instructions, and found two consistently used components in the descriptions, i.e., landmarks and (prescribing) actions. Based on these two main components, Denis identified five classes into which instructions can be divided:

Denis’ classification: prescribing actions vs. introducing landmarks vs. describing landmarks vs. comments

1. **Prescribing Actions without a landmark:** Refers to motion statements mentioning no landmark, e.g., "Go straight".

2. **Prescribing actions with a landmark:** Refers to motion statements that include a reference to a landmark, e.g., "Pass the university building".

3. **Introducing Landmarks:** Refers to statements that introduce landmarks without reference to a specific action, e.g., "There is a bank".

4. **Describing Landmarks:** Refers to statements that describe landmarks based on their non-spatial attribute properties, e.g., "The bank is called Erste".

5. **Comments:** Refers to meta-statements, e.g., "The route is 15 min. long".

Denis [1997] also provides a brief and informal analysis of the content of each class. For example, he mentions that members of the first class (i.e., prescribing actions without a landmark) either consist of actions that instruct the way-finder to proceed (e.g., "go straight") or to execute a reorientation (e.g., "turn right"). The third class consists of instances that make explicit references to a spatial location (e.g., "There is a house to your right").

In a more recent study, Schwering et al. [2013]offer a distinction between self-orientations, turning-movements, and non-turning movements:

Schwering et al distinguish between self-orientations, turning-movements, and non-turning movements.

1. **Self-orientations:** Statements that help the way-finder to self-orient. They can either mention a local landmark ("behind the church") or a global landmark ("you are looking in the direction of a tower").

2. **Turning movements:** Statements that include turning movements, e.g., "turn right at the church".

3. **Non-turning movements:** Statements that include non-turning movements, e.g., "pass the church".

The plethora of existing classifications suggests that there is no "shared form of understanding" [Gruber, 1993] of the semantics of route instructions. It could be speculated that this is one aspect why commercial Web services fail to offer instructions that are "cognitively sound". Information system designers would certainly benefit from a clear specification of the elements that make up human-produced instructions. As a consequence, a unifying taxonomy of the above mentioned approaches is attempted in Chapter 4 and later formalized in Chapter 5.

3.2 FORMAL MODELS

This section treats formal approaches to way-finding instructions. First, various attempts to automatically generate route instructions are reviewed. Second, a formal model aimed at the classification of instructions is discussed. Finally, models that focus on the representation of instructions over multiple levels of detail (LoD) are presented.

3.2.1 *Generation*

One of the first attempts to formalize spatial knowledge in the context of route instructions was provided by Kuipers ([1978], [1983]). The basis of his proposed TOUR model [Kuipers, 1978] is the notion of an agent that moves through the environment and acquires knowledge of the geographic space in the form of a cognitive map through observations (cf. Section 2.1). The modeled agent keeps track of its current position and a set of available inference rules that allow for problem solving, such as navigating between two places. A representation of the agent's position stores information about the current place and path an agent is located, i.e., its direction, heading (2D), and orientation.

Kuipers TOUR model

The agent can receive two types of instructions ("turn", or "go-to") based on its observation of the environment. Inference rules combine information from the agent's current position and the observed environment to fill in parts of the instruction that are unspecified. For example, the partial instruction "turn right" can be enriched by using the local geometry and current position to add information how the path segments and the facing direction of the agents changes if the action is performed. This information is then used to update the position of the agent.

In subsequent work, Kuipers [1983] formalized sensorimotor procedures, i.e., the knowledge of how to travel between two places. He argues that this type of knowledge can be represented by views and actions:

- **A view** is defined as a sensory image of an agent at a decision point, e.g., an intersection.
- **An action** is defined as an operation that changes the current view by locomotion.

Kuipers Model for Sensorimotor Routines and Partial Knowledge

Thus, routes observed by an agent can be formalized as a sequence of views and actions (V_0, A_1 , V_1, ..., A_n, V_n). In order to recall a route the agent requires two types of "associative links". The first link V -> A encodes the knowledge that at the current view V action A should be carried out, if the route is to be followed correctly. The second link (V, A) -> V′ encodes the knowledge that the agent will end up at view V′ if the action A is carried out at view V. Thus, an agent's complete knowledge of a route can be represented as a sequence of type 1 and 2 links.

Kuipers [1983] also discussed various types of partial knowledge and their implications if a route is retrieved from memory. For example, in case an agent has only a sequence of type 1 links (V -> A) stored in memory, the route needs to be reconstructed in-situ. This means that the route has to be traveled physically by the agent because there is no stored connection of the views available. If the agent does not have access to the view V′ with respect to V, the environment "returns" the result if the agent carries out action A at view V. This means that the agent ends up at V′ after carrying out the action, thus returning the missing link.

Habel [1988] proposed a semantically enriched knowledge model for the generation of route instruction. He argued against the direct and literal generation of instructions, based on a street network represented by an underlying graph structure. Instructions produced in this manner would have many redundant features. To illustrate, if a simple route is represented by a graph as a sequence of nodes (place) and edges (street), i.e., "$n_1 - e_1 - n_2 - e_2 - n_3 - ... - e_k - n_k$", one could directly generate a route instruction such as "Start at $place_1$, follow $street_1$ until $place_2$, follow $street_2$ until $place_3$, etc...". However, such a sequence can contain unnecessary parts, e.g., if no reorientation takes place between $place_1$ and $place_3$. Thus, the instruction "Start at $place_1$, then go straight until you reach $place_3$" would be less redundant.

Habel argued that information systems should be capable of emphasizing relevant and omitting irrelevant information in route instructions. The semantics of a purely sequential representations based on nodes (place) and edges (path) alone are not rich enough to allow questions related to relevance and similarity. To remedy this fact, Habel developed a model that encodes connectivity of street segments and landmark information. Nodes (places) are given semantics in terms of what landmarks are present. Furthermore, sequences of edges (street-segments) that allow a direct connection between nodes are grouped together and form a higher level entity (street). For example, assume the sequence "$n_1 - e_1 - n_2 - e_2 - n_3$" can be grouped as "$s_1$". From this one can generate an instruction such as "To go from $place_1$ to $place_3$ take s_1" instead of "Start at $place_1$, then follow $street_1$ until $place_2$, then follow $street_2$ until $place_3$.

Habel's semantically enriched knowledge model

More recent research was concerned with formalizations of the structural parts of route instructions. For example, Klippel ([2003, a], [2003, b]) developed a formalism grounded in human mental conceptualizations of actions at decision points. It defines routes as a sequence of tuples that consist of a route segment (RS) and a decision point (DP). Decision points are points along the route that require an agent to decide on an action, e.g., which direction to take. Klippel's main contribution was to show that humans have at least seven different conceptualizations of turning actions at decision points. These action primitives (or wayfinding choremes (wc), in Klippel's terminology) allow to express turn-by-turn route descriptions with the help of a formal grammar. The following example shows an excerpt from his formalization (slightly adapted as used in Weiser and Frank [2013]):

Klippel's wayfinding choremes

```
<Route> ::= <DecisionPoint>&lt;Segment&gt;[<RoutePart>]<DecisionPoint>
<RoutePart> ::= <DecisionPoint>&lt;Segment&gt;
<DecisionPoint> ::= wcl | wcr | wcs | wcRMl | wcRMr | wcRMs
```

In this formal grammar turning actions, such as "turn left", "turn right", and "go straight" are denoted by "wcl ", "wcr ", and "wcs ", respectively. Landmarks are represented as separate elements. For example, turning actions at landmarks are indicated by wcLMl, wcRMr , and wcRMs , meaning "turn left at landmark", "turn right at landmark", and "pass landmark", respectively. Subsequently, Klippel et al. [2009] defined a data structure that captures the semantics of wayfinding choremes.

Klippel's formalism can also produce higher-order elements from a sequence of elementary wayfinding choremes (cf. Habel [1988]). The method proposes rewriting rules to "spatially chunk" functional elements together [Klippel et al., 2005]. Note, the term refers to the conceptual or verbal chunking of instructions, not the combination of the geometrical features (path segments or intersections). For example, the instruction "go straight, then straight, then turn right" can be expressed in terms of way-finding choremes as "wcs, wcs, wcr ". The functionally equivalent statement "turn right at the third intersection" can be rewritten as "dwc3r". This research is in line with Frank's [2003] proposed theory of pragmatic information content, taking into account that messages of different length can convey the same information (See Section 3.2.3).

Spatial Chunking

A different approach proposed by Srinivas and Hirtle [2006] and based on Klippel's formal grammar addressed the issue of how one can formally represent partially familiar routes. The research was motivated by the apparent redundancy of fully detailed instructions for familiar areas. Testing their approach through experiments, Srinivas and Hirtle [2006] found that people prefer schematic instructions over fully detailed ones for routes they were partially familiar with. Their research is in line with this thesis, as commercial systems fail to make a distinction between users who are totally unfamiliar with a route and those that have partial knowledge (See Section 1.1).

Several other formal approaches meant to provide more cognitively adequate route instructions exist. For example, the research done by Raubal and Winter [2002], and Nothegger et al. [2004] was concerned with defining a "formal measure" for the saliency of landmarks. Landmarks are an ubiquitous feature of human-produced route directions. In addition it has been shown that people prefer instructions that include landmarks over instructions without them [Lovelace et al., 1999]. For a more detailed treatment of the special role of landmarks see Section 3.3.

Raubal and Winter's as well as Nothegger et al's landmark model based on saliency

Raubal and Winter's [2002], as well as Nothegger et al.'s [2004] research was motivated by the lack of formal approaches to determine the "salient characteristics" that make landmarks "distinct from other elements" [Lynch 1960]. The measure included parameters for the visual (e.g. shape, color), semantic (e.g., cultural and historical importance), and structural (e.g., placed at intersections) attraction of landmarks. The results showed that (with the help of the specified formal measure) it is theoretically possible to automatically extract landmarks from already existing datasets.

Hansen et al. [2006] proposed a data structure capable to represent the multi-faceted semantics of landmarks. The data structure is based on an taxonomy capturing several aspects of landmarks as integral part of route instructions.

Yet, as to date no commercial implementation of such formal landmark-based models exist.

3.2.2 *Classification*

Brosset et al. [2007] proposed a conceptual model based on boolean functions that can encode linguistic constructs found in typical route instructions. It is meant to support the analysis of a route instruction's structural properties through a classification process. The model, similar to Kuipers' approach [1983], is based on locations at which actions can be performed. However, their model also includes landmarks and other spatial entities (e.g., forests) explicitly.

The formalism introduces a boolean function f that evaluates to true if a location contains a landmark or spatial entity ($f :: Loc- > Bool$). Another function g evaluates to true if an instruction contains an action mentioning a landmark or spatial entity (g :: $action- > Bool$). A route segment can be represented by the triple of the type A (p_i,a_i,p_{i+1}). Here, the location p_i is connected to a location p_{i+1} via action a_i. Using the functions f and g for the location and the action, respectively, additional information can be encoded by a tripe of type B $(f(p_i), g(a_i), f(p_{i+1}))$.

Brosset et al.'s boolean model for structural analysis of route instructions

For example, an action that starts at an location and ends at a location but neither action nor the locations mention landmarks or spatial entities is encoded by the triple of type B $(0, 0, 0)$. An action that itself mentions a landmark between two locations not mentioning a landmark is encoded by the triple of type B (0,1,0). Thus, an action mentioning a landmark that starts and ends at a location with a landmark is encoded by the triple of type B (1,1,1). The model can be extended to include directional (e.g., "go right") and elevation (e.g., "go up") information by introducing additional boolean functions.

3.2.3 *Level of Detail*

In order to make sense of an infinitely complex world, humans have developed the ability to abstract unnecessary detail. Abstraction can be defined in terms of a subset operation reducing a large finite number of objects or actions to are smaller more perceivable quantity [Timpf, 1999]. Recursive application of this operation generates further subsets and results in a hierarchy, i.e., a system that is "analyzable into successive sets of subsystems" [Simon, 1998]. This implies that each successive level of a hierarchy is a less detailed representation of its superordinate structure. Other terms describing essentially the same concept are "granularity" or "level of detail". In this thesis the term level of detail (LoD) is used.

LoD = Level of Detail

As a special type of message, route instructions are usually communicated to enrich an agent's spatial knowledge, such that they can carry out way-finding in an area of which they have no or little spatial knowledge. As such, instructions can have varying levels of detail depending on the individual user needs (See Section 3.3.2). Human-produced route instructions use no fixed level of detail, and if communicated orally, switches between several levels are rather common. A comparison of human-generated and automatically produced descriptions showed that although computer-generated instructions are comprehensive they are often redundant and do not take a way-finder's prior knowledge into account. Humans, on the other hand, are capable of "providing information of changing detail in a consistent and coherent way, adapting to the addressee's assumed asymmetric information needs" [Tenbrink and Winter, 2009].

However, today's information systems are basically implementations of Shannon's [1948] model, i.e., they do not allow the user to manipulate presented instructions (See 2.2.1). In addition, current models do not account for the fact that two messages with the same content can have different semantics (lead to different actions), and two messages with different content can have the same semantics

[Frank, 2003]. Frank's theory of "pragmatic information content" formalizes this notion and provides a measure to determine whether two messages are pragmatically equivalent. This is the case if, in a determined context, two messages trigger the same action by an agent. It is a theory that includes the state of the receiver of an instruction. This notion of level of detail is based on what agents do with the instructions, i.e., their actions, rather than the approaches that are concerned with describing the level of detail of the environment. For example, for some agent the instructions "Take the second right" and "Continue at the first intersection, then turn right" may be pragmatically equivalent (See also Klippel et al. [2003]).

Pragmatic Information Content of Route Instructions

Timpf et al. [1992] identified and modeled three conceptual levels of which a way-finding process consists. This was later extended by Timpf and Kuhn [2003], and although it only considers the physical properties of a street network and does not treat instructions explicitly, it argues strongly for the conceptual differences of a way-finding task in varying contexts. The three conceptual levels and possible corresponding instructions are defined as follows:

Conceptual Level of Detail in Route Instructions

1. **Planning level:** At this level an agent considers the start and goal of the route, its duration and possible places in between. A possible instructions at this level could state "To get from A to B you need to go via C and D. The trip takes 2 hours".

2. **Instruction Level:** At this level, the entry and exit points for each leg of the trip are included. On this level, instructions such as "take entrance En1, then follow highway H2, finally take exit Ex3" are required.

3. **Driver Level:** At this level, the agent receives very detailed information of the type "When should I drive on which lane of the road?". This implies instructions including very specific decisions, e.g., "take ramp R1, change onto left lane, slow down".

Tenbrink and Winter [2009] proposed three types of granularity (LoD) found in natural language descriptions of a route:

- **Linear Granularity:** This reflects the underlying linear geometric properties of the mentioned entities. For example, a change in linear granularity (from abstract to specific) can be observed in instructions such as "go straight on until you see a large green building, there you turn right, then the next left". The change in granularity is reflected by the fact that the statement "go straight..." spans several decision points and line segments, while the subsequent reorientations ("...turn...") are specific to a decision point.

- **Areal granularity**: This reflects the underlying 2D geometric properties of the street network. An example for a change in areal granularity (from specific to abstract) can be witnessed in an instruction such as "walk towards the lake, in the middle of the park". The action "walk towards..." refers to a concrete areal object while the "middle of the park" is a region that is no clearly defined.

Linguistic approach to Level of Detail in Route Instructions

- **Elaboration**: This is based on the notion of a hierarchical conceptualization of concrete objects (cf. [Rosch et al., 1976], [Lakoff, 1987]). For example, the same geometric feature of a road can be referred to by its basic category term "road" (general) or its proper name "Gusshausstrasse" (specific). Alternatively, one can view elaboration as referring to an object's attributive information. For example, "turn right at the green building" is more specific than "turn right at the building". Elaboration corresponds to the notion of LoD that can be modeled by using partial function application as proposed by Weiser et al. [2012] and more recently, a quantum approach to category switches [Hahn and Weiser, 2014].

Timpf [2002] proposed another way-finding model based on the activity theory of Leontiev [1974]. Leontiev's theory uses the hierarchically structured concepts of activities, actions, and operations. Activities are the most abstract concept and driven by an agent's motive while actions are more specific and driven by a conscious purpose. It is worthwhile to state that motive is "something (as a need or desire) that causes a person to act and purpose" while purpose is "the reason why something is done" [1].

Way-finding and Activity Theory

Thus, the task of way-finding can be seen as an activity (driven by desire to find the way to a destination) that can consist of several actions (driven by the purpose to achieve the activity), e.g., "walk from A to B, take train from B to C". The most specific concept "operations" is related to conditions. This implies that under certain circumstances the conditions can change (thus the operations) while the goals remain constant (thus the actions). For example, the action "walk from A to B" may imply the operations "start at A, leave building through exit E1, [...], you see B". The operation "leave building" depends on various conditions, for example, that the particular exist E1 is not blocked. If the condition is violated the operation has to be adapted, i.e., an alternative exit E2 has to be found. This, however, does not affect the superordinate action "walk from A to B".

Destination Descriptions

A model called "destination descriptions" exploiting the hierarchical and dynamical nature of human-produced instructions was developed by Tomko and colleagues ([Tomko and Winter 2006]; [Tomko, 2007]; [Tomko and Winter, 2009]). Commercial services provide turn-by-turn instructions that emphasize "how" to get to a destination by providing a constant LoD across the entire route. In contrast, human-produced route descriptions do not keep track of a constant LoD for every part of the route but tend to, (1) coarsely mention the destination, and (2) add more detail as the description progresses. Destination descriptions emphasize the "where", i.e., they are place descriptions, which in turn requires that the receiver has at least some partial knowledge of the environment.

1 www.merriam-webster.com

Tomko and Winter [2009] mention the example of a taxi driver who receives a destination description, for instance, "To Leitgebgasse please. It is in the 5th district, close to Reinprechtsdorferstrasse". This information usually suffices to get a taxi driver started. Once the destination is close, the taxi driver could then ask "Where should I go now?". This triggers a more detailed description (i.e., turn-by-turn) from the passenger. Destination description are usually shorter than turn-by-turn descriptions, reducing the cognitive load of a way-finder who is partially familiar with the geographic area of which they want receive navigational advice.

An algorithmic approach to combine classical turn-by-turn and destination descriptions was presented by [Richter et al., 2008]. In their interactive dialog-driven model the way-finder is first presented with a (coarse) destination description. In case the user requests more detail the system can switch to a turn-by-turn description from the current position to the next intermediate location along the route.

3.3 QUALITY ASPECTS

This section reviews quality issues related to the presentation of way-finding instructions. This includes an assessment of general (i.e., how instructions should be communicated) and contextual (e.g., user-dependent) properties.

3.3.1 *General principles*

Allen [2000] conducted human experiments to evaluate principles he considered useful for the communication of route instructions. The results showed the validity of three "principle-based practices" meant to foster both the ability to comprehend and follow route instructions:

1. **"The principle of natural order"**: Instructions should be presented in the correct spatio-temporal order. This ensures an increased accuracy at which route instructions can be carried out. In general, the order at which things can be described has an inherent "left-to-right" structure, bound by the "design characteristics of spoken language" [Levelt, 1982]. Language is only capable of expressing a linear structure. As such, any higher-dimensional structure needs to be mapped onto the one-dimensional nature of language. This implies that listeners expect (easy to follow) information to be arranged in an ordered fashion. Information that is presented out-of-order is usually perceived as either not useful or difficult to disentangle. Because any route is a linear sequence of places, the instructions describing a route are bound to be linear.

Allen's communication principles: Principle of natural order, Referential Determinacy, Principle of mutual knowledge.

2. **"Referential Determinacy"**: A route instruction should attempt to eliminate, or at least reduce potential uncertainties at decision points. This implies a special treatment of the environmental features (in particular, landmarks) an agent encounters along

the route, as well as unambiguous instructions concerning the action one should perform at each decision point.

3. **"Principle of mutual knowledge":** Instructions should include information that can be comprehended by anyone, thus they should not require special skills. For example, the use of cardinal directions in instructions might not be equally appropriate for every person. The principle of "mutual knowledge" can be generalized if one tries to minimize the amount of quantitative information and rely on qualitative information instead. Studies have shown that people mostly rely on qualitative information for the conceptualization [Lakoff and Nunez, 2001] and communication [Frank, 1992] of (geographic) features. For example, in the context of route instructions it has been shown that humans often group several instructions into one single abstract concept, i.e., the spatial information is conceptualized in qualitative terms [Klippel, 2003]. A direct consequence of this principle is that, because knowledge can not be assumed to be universal across all domains, the establishment of mutual knowlege between several persons may require signals and negotiation (as suggested in this thesis).

Route Instructions as Spatial Narratives

Mark and Gould [1992] proposed that route directions share many features of narratives, i.e., stories. For example, both stories and route instructions include deictic information. According to Levinson [1983] deixis refers to the way language encodes contextual features in an utterance. Assuming that instructions take an agent on an imaginary journey, the agent has to keep track of dynamic information related to "where", "when", "who", "whose". "Where" refers to the location, "when" to the time, "who" to the person in focus, and "whose" to the point of view (speaker or hearer) that is assumed in the instruction. Other features that make human-produced instructions similar to narratives is their use of repetition to indicate distance (e.g., the statement "go straight straight straight" to indicate a long way down the road), and summaries [Mark and Gould, 1992]. The notion of route instructions as spatial narratives has also implications for the design of information systems. For example, good instructions need to be internally consistent in the way deictic terms are used in order to allow the agent to follow the instructions effectively (cf. Allen's [2000] communication principles mentioned above).

Another approach concerned with the quality of route instructions was pursued by Denis[1997] and Denis et al. [1999]who attempted to determine the essential elements (invariants) found in any human-produced route instruction. Denis' research is in line with the empirical findings presented by Allen [2000], Lovelace et al. [1999], as well as Michon and Denis [2001], all of which emphasize the important role of landmarks at decision points and their associated actions for route instructions that are perceived as good by humans. In addition, Ross et al. [2004] compared conventional (indicating distance to turn and street names) to landmark based instructions. The authors found that landmarks included in route directions increased user confidence and navigation performance. Tom and Denis [2003] showed that street information is less effective than landmark based information for navigational task and that street names are also less likely to be mentioned by people if they are asked to provide instructions.

It is interesting to note that the special role of actions and landmarks at decision points is independent of external environmental factors. In this respect it does not matter whether navigation takes place in natural (cf. [Brosset et al., 2008], [Sarjakoski et al., 2012]) or underground [Fontaine and Denis, 1999] environments.

Useful instructions include landmarks at decision points and provide adequate associated actions. They are concise and avoid redundancy. These principles are context-independent.

Additionally, Denis [1997] findings showed that descriptions rated as good by humans adhere to Grice's [1989] maxims of effective communication, e.g., they are concise and avoid redundancy. In one study, Daniel and Denis [2004] asked participants to provide instructions as concise as possible while at the same time ensuring that recipients of the instructions would still be able to follow them. The results showed that concise instructions feature "action prescriptions and landmarks at points on the route where key actions have to be taken" while landmark descriptions without an associated action were significantly reduced.

3.3.2 *Individual (Context-Related) Factors*

Although the effective communication of route instructions should adhere to the principles mentioned in Section 3.3.1, people might have different expectations of what a route instruction should include or not. There are at least three factors that determine the usefulness of a route instruction on an individual level:

- **The activity at hand:** For example, Hirtle et al. [2011] looked at the activity the instructions should support and its effect on the relevancy and level of detail needed from the perspective of a human agent. For example, someone in case of an emergency who needs to find the way to the nearest hospital is only interested in getting the fastest route. This requires a concise description, preferably with the most salient landmarks. On the other hand, someone on a sightseeing trip does not mind to take a longer but more scenic route. Such a description might include landmarks that are of cultural and historical importance, but pay less attention to saliency-related aspects of landmarks. Furthermore, Hirtle et al. [2011] showed that human-generated instructions reflect the underlying activity through the use of specific keywords. This indicates that humans adjust their instructions to include the perceived activity needs, implictly communicated by the person who requires the instructions.

The activity at hand, the environment, and the recipient influence the LoD in route instructions

- **The Environment:** Humans have the ability to emphasize certain features of specific parts of a route, in case they perceive them as being difficult to navigate. In a case study that analyzed Web-based way-finding instructions containing the phrase "tricky part", Hirtle et al. [2010] attempted to identify factors that make some instructions more difficult than others. They found that the "tricky parts" were mostly based on ambiguous geometrical situations (e.g., sharp turns), bad signage (either missing entirely or misleading), naming conventions (e.g., an agent turns at an intersection, expects the street name to change but finds it to be the same), or traffic regulations (e.g., no left turn).

- **The recipient:** In a study conducted by Hölscher et al. [2011] it was shown that route instructions varied depending on whether the route was described to somebody else or to oneself. Both the content of the instruction and the actual route choice were different. For example, factors such as the number of turns to take, the street types included, and the efficiency of the route were adapted to what was assumed to be the addressee's information needs.

Drawing on above mentioned research it can be concluded that the information considered to be useful for an agent depends on the context, specifically (1) the activity at hand, (2) the environment to be encountered, and (3) the perceived or actual capabilities of the instruction receiving agent. Jonietz and Timpf [2013] suggested the term "spatial suitability" to account for the three aforementioned factors.

3.4 COMMUNICATION MODELS

This section reviews approaches that address aspects of way-finding instructions in a communication setting from a systemic point of view. This includes a discussion of its parts, i.e., the participants involved including their roles, and the phases that occur during a communication setting. A series of authors have attempted to identify the phases that occur during the human exchange of way-finding instructions. Most of the early research on route directions was done in the realm of Cognitive Linguistics and Psychology and focused on the analysis of both structure and generation of spatial descriptions ([Linde and Labov, 1975], [Klein, 1979], [Klein, 1982], [Wunderlich and Reinelt, 1982], [Allen, 1997], [Denis, 1997], [Couclelis, 1996]).

The communication of routes consists both of a cognitive and a linguistic task

Note, in the following discussion of existing models, the receiver and sender of the instructions are denoted by T (Target) and S (Source), respectively. In general, instruction giving and receiving consists of a cognitive (mental) and linguistic (verbal) task [Couclelis, 1996], both of which are needed for the successful communication of route instructions. During the cognitive task, for example, S has to activate a mental representation (i.e., cognitive map) and locate the start and goal destination. Then S needs to plan and construct a path between a start and a destination position including potential choice points. T in turn, has to memorize the instructions and integrate any new information in their already existing mental representation. The linguistic task is concerned with the process of how the route instructions are verbalized and communicated (externalized) between the target and the source.

One of the first models that discussed the cognitive and linguistic tasks was provided by Klein [1979] who attempted to construct a generic model for the process of "asking for directions". Based on the study of language data, he proposed three separate and consecutive phases:

Klein's generic communication model: Introduction, Route Description, and Closure.

1. **Introduction Phase:** This phase consists of the first contact between T and S, the statement of the problem by T, and finally the acceptance or rejection of the direction giving task by S.

2. **Route Description Phase**: In this phase, S plans the route (i.e., locates start and goal as well as constructs a path in between based on his cognitive map), articulates the plan, and makes sure T understands the directions.

3. **Closure Phase:** In this phase S ends the description. T in turn can accept or reject the description. If T rejects the description, phase 2 (route description) will be initiated again. Otherwise the conversation ends.

In another study done by Wunderlich and Reinelt [Wunderlich and Reinelt, 1982], Klein's model was extended to include an optional securing phase that can occur between the route description and the closure phase. This extended model is functionally equivalent to Allen's model [1997]:

Wunderlich and Reinelt's extension adds a securing phase to Klein's model.

- **Securing Phase (optional):** In this phase, S may summarize, repeat, paraphrase, or complete parts of the description. This phase can be triggered either by S (who wants to make sure T has understood), or by T if the closure phase fails.

Later, Couclelis [1996] proposed a more detailed model of the underlying (cognitive) processes during the communication of instructions. The model consists of five phases:

Couclelis' five stages model: Initiation, Representation, Transformation, Symbolization, and Termination.

1. **Initiation:** T requests directions, S registers, and confirms the request. The request is understood by S as referring to destination.

2. **Representation:** S forms an intention to help, S evokes a mental image, S plans the route hierarchically (from coarse to fine in a two stage process).

3. **Transformation:** The route is linearized and segmented and parts are selected.

4. **Symbolization:** The route is expressed and (if needed) reinforced.

5. **Termination:** The interaction ends.

An analysis of aforementioned models reveals that they do not sufficiently stress the interactive nature of human communication. Although the mentioned models discuss potential phases in the communication process they only mention the features of which each phase consists. The actual processes necessary to build a realistic model (i.e., in the sense of pragmatic communication discussed in Chapter 2) are not sufficiently spelled out. For example, Klein [1979] notes that the person who has to give instructions has to modify or expand their opinion about the receiver. This relates to agents having a "theory of mind" (See Section 2.4.2). Klein also acknowledges that the sender needs to make sure that the receiver understands everything about the description correctly. Similarly, Couclelis [1996] briefly mentions the need to reinforce statements during the symbolization stage. This relates to agents using signals to indicate how the conversation should progress (See Section 4.2). However, none of these crucial points is explicitly modeled or discussed in more detail.

3.5 CONCLUSIONS

This chapter presented the current state-of-the-art research on way-finding instructions. First, it answered the question "What are way-finding instructions?". This included a top-down approach to identify the various types of information that make up way-finding instructions. Second, formal models were discussed, including the generation, classification, and abstraction of instructions. Third, quality-related properties (general and individual) of route instructions were reviewed. Finally, conventional models addressing the exchange of instructions within a communication setting were critically analyzed.

The research reviewed in this chapter has demonstrated that:

1. Landmarks play an important role for a successful and effective communication of way-finding instructions, independent of context.

2. Whether an instruction is considered to be useful depends on the user, the activity at hand, and the environmental features along the route.

3. Human-produced way-finding instructions have no fixed level of detail. Humans use varying levels of granularity, and they can adjust information presentation to another person's assumed needs.

4. Most of the existing formal models were concerned with the representation of an instruction's structural properties, but did not or only superficially consider communicative aspects.

5. Existing dialog models failed to address the complete interactive nature of communication. Human-style communication makes use of signals and agents have a "theory of mind". In

addition, existing models are only spelled out in vague natural language terms. A formal treatment would make models testable and allow for consistency checks.

To conclude, existing approaches have failed to sufficiently address the pragmatic aspects of communication. There is no fixed set of information that fits everyone's needs at all times. The effective and successful communication of information is based on the interaction of the user with information, as well as a user's intentions prior and during communication. These empirical findings are valuable input for the design and formalization of the conceptual (See Chapter 4) and computational (See Chapter 5) communication model presented in the following chapters.

Part III

PRAGMATIC COMMUNICATION MODEL

4 CONCEPTUAL MODEL

This chapter presents the conceptual model developed to capture the aspects of meaning negotiation between communication partners in the context of route descriptions.

This chapter is structured as follows. First, a taxonomy is proposed that is grounded in principles that allow for cognitively adequate instructions. The taxonomy follows from the empirical research findings presented in Chapter 3. Second, the proposed signals used in human communication settings that exchange way-finding instructions are presented. The signals were extracted from an exploratory data collection [Weiser and Frank, 2013] and verified using an additional language corpus collected by Klein [1979]. Third, the language data is also analyzed with respect to the actual negotiation process and the signals that occur. This analysis further supports the hypothesis stated in this thesis and provides empirical evidence for the use of signals to negotiate the meaning of utterances in the context of interactive exchanges of way-finding instructions.

4.1 TOWARDS A SHARED UNDERSTANDING OF WAY-FINDING INSTRUCTIONS

This section proposes a unifying and light-weight taxonomy that specifies the semantics of way-finding instructions using principles that allow a cognitively adequate information presentation. One could argue that the reason why current information systems fall short in presenting (cognitively) easy-to-process instructions is that they have no shared (formal) understanding of the semantics of instructions. Although some classifications have been proposed (See Section 3.1), they are often too coarse, overlapping, and provide no formal specifications. Additionally, the computational model presented in the next section needs to specify the language agents can use to make assertions about space.

The taxonomy presented here was developed in a top-down manner. Denis ([1997], [1999]), as well as Tom and Denis [2004] have shown that human-produced instructions use predominately two general types of instruction. While instructions are typically seen as procedural discourse ("do x"), they also include descriptive elements ("there is y"). Thus, the two main classes of the taxonomy are actions and descriptions.

Instructions consist of actions and descriptions.

4.1.1 *Actions*

Here, an action is defined as a procedure that changes an agent's location on a street network, i.e., from one decision point to another. A

Action	**Progression**	**Reorientation**
LM	"Go straight at X"	"Turn right at X"
NoLM	"Go straight"	"Turn right"

Table 1: Example Table for Action Instances

Progression vs. Reorientation

primary distinction can be made between actions that either tell the way-finder to proceed or to reorient themselves (cf. Denis[1997]). A progression keeps an agent's orientation constant, while a reorientation changes the direction an agent faces. Furthermore, both types of actions may or may not contain references to either local or global landmarks, i.e., landmarks that are either along a road or visible but not contiguous to a road. The information can be combined to end up with 4 different action type combinations:

Reference to a LM vs. No reference to a LM

1. **ProgressionLM:** Progression with a reference to a landmark. Possible instances of this class are "Pass the university", or "Walk towards the clock tower".

2. **ProgressionNoLM:** Progression without a reference to a landmark. Possible instances or this class are "Go straight", or "Continue".

3. **ReorientationLM:** Reorientation with reference to a landmark. Possible instances of this class are "Turn left at the large intersection", or "Turn right after the green building".

4. **ReorientationNoLM:** Reorientation without a reference to a landmark. Possible instances of this class are "Turn right", or "Make a left".

The aforementioned classes subsume the previous classifications mentioned in Section 3.1. For example, Schwering et al.'s [2013] class "turning movements using local landmarks" and statements mentioning Lovelace et al.'s [1999] "landmark at choice points" are subsumed in the class **ReorientationATLM.** Also, Schwering et al.'s class "non-turning movement using local landmark", as well as local and global landmarks can be subsumed in the class **ProgressionAtLM**. In addition, Klippel's [2003] turning concepts could also be represented by using these classes. See Table 1 for example instances of each action class presented above.

4.1.2 *Descriptions*

Here, a description is defined as a procedure that updates an agent's knowledge about the world. This can include information about their current position, attributes concerning visible landmarks, or meta-information (e.g., the total duration) about a route. Descriptions can either be of type locating or non-locating.

Locating vs. NonLocating Descriptions

Locating descriptions help the way-finder to self-locate, by explicitly referring to a landmark. This implies a matching process between

Desc	**NonLocating**	**Locating**	
		SpatialRef	**NoSpat.Ref**
LM	“X is red”	“X is to the left”	“There is X”
NoLM	“You can’t miss it”	-	

Table 2: Example Table for Description Instances

an agent’s current location and the mentioned landmark. Locating descriptions can or cannot include explicit relative references to the spatial location of a landmark, i.e., they may mention the relations between the agent’s heading and the landmark. The former is represented by a statement such as "There is a bank to your left". The latter is represented by a statement such as "There is a bank". The locating character of this type of instruction, however, is implicit. If the agent can identify the feature mentioned in the description she will be able to update her knowledge about the current position.

Spatial References vs. No Spatial References

In contrast, non-locating descriptions can either provide additional attributive information on specific landmarks (e.g., "The building has a greenish facade"), or include meta-information on the entire route or route sections (e.g., "It takes 15 min. to travel the route").

Furthermore, it is possible to make a distinction between local and global landmarks for both locating and non-locating descriptions. The following combinations are possible:

1. **LocatingLMSpatialRef:** Locates the way-finder by making an explicit spatial reference to a (local/global) landmark. Possible instances of this class are “There is a green building to your left”, or “In the distance you can see a large church”.

2. **LocatingLMNoSpatialRef:** Locates the way-finder by mentioning a visible (local/global) landmark but without explicitly referring to its spatial location . Possible instances of this class are “Then you see a bank”, or “There is a large building”.

3. **NonLocatingLM:** Mentions attributive information of a (local / global) landmark. Possible instances of this class are "The bank has a greenish facade", or “The building features an owl sculpture”.

4. **NonLocatingNoLM:** (Meta)-Comments about the entire route or sections of the route. Possible instance of this class are “The route is 15 minutes long”, or “It is not far”.

With this classification, Schwering et al.’s [2013] self-orientations are subsumed in class LocatingAtLMSpatial. Denis’ [1997] class "landmark attribute information" can be subsumed in class NonLocatingAtLM while Denis’ comments are subsumed in Meta. See Table 2 for example instances of each description class presented above.

4.1.3 *Discussion*

This section proposed a light-weight taxonomy that unified existing classification aimed at describing invariants in route instructions. The taxonomy makes a basic distinction between actions (indicate movement of the agent) and descriptions (indicate updates of the agent's knowledge-base). Actions can be further divided into reorientations (i.e., an agent's heading changes) and progressions (i.e., an agent's heading remains constant). Descriptions are either of type locating (i.e., provide an agent with orientation information) or non-locating (provide an agent with attributive information) descriptions. See Figure 6 for a lattice representation of the proposed taxonomy.

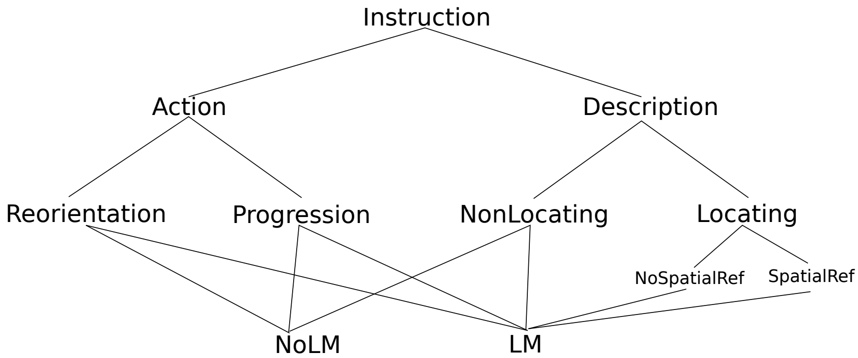

Figure 6: Proposed Taxonomy of Route Instructions (on a type level)

Note, for each type of instruction it is possible to include (optional) landmark. In addition, locating descriptions can be enriched by adding a spatial reference. These two features allow a basic notion of level of detail. For example, the instruction "turn right at landmark_X" (ReorientationLM) is more specific than "turn right" (Reorientation). Another example is the instruction "There is a landmark_Y" (Locating) is more general than "There is a landmark_Y to the right" (LocatingSpatialRef). Furthermore, through the combination of "atomic" instruction types, an additional notion of level of detail can be introduced. See Figure 7 for a possible combination of instructions, all belonging to the super type Description.

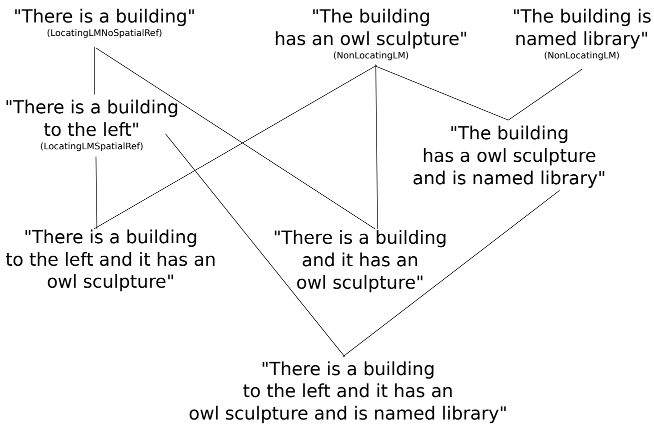

Figure 7: Lattice of a possible combination of basic instructions (instances are all of type Description)

The here developed taxonomy represents one of the basic components for the computational model introduced in the next chapter. This includes a formal specification of the discussed taxonomy.

4.2 EMPIRICAL OBSERVATION OF SIGNALS

The signals presented here and used to negotiate the meaning of route instructions were extracted from two language corpora of an exchanges of route instructions between two individuals. This feature distinguishes the corpora from other language data collected in a monologue setting. Dialog based language data is the input for the conceptual and computational models developed in this thesis. Note, this section only discusses the signals themselves. The interaction, i.e., the process of meaning negotiation using signals is discussed in the next section.

The first dataset was collected from an exploratory data analysis in the city of Vienna, Austria [Weiser and Frank, 2013] and consists of 10 dialog sequences. The second dataset is based on language data collected by Klein [1979] in the city of Frankfurt/Main in Germany. The dataset contained 20 instructions to two different destination. From the original 40 dialog sequences of the Frankfurt dataset only 35 were used for the analysis (18 to destination 1 and 17 to destination 2) because either there was more than one person answering the request, or the dialog data was not complete due to technical difficulties.

4.2.1 *Method*

In the Vienna case, the author asked 10 participants (5 female, 5 male) with a mean age of 31 to take part in the exploratory data collection. The participants were asked to described a route they travel frequently and know well (e.g., from work to home) to the experimenter. This particular set-up ensured that participants were likely to have a spatial mental model which allows for making inferences about spatial locations, and ”total” knowledge of the situation [Tversky, 1993]. Note, “total knowledge” should only be understood as being able to describe a route sufficiently detailed such that the recipient of the instructions can navigate the route. Total does not imply that the person has a complete knowledge of the route. The experimenter had no or little knowledge of the geographic space part of the described routes. The conversations were recorded and later transcribed to allow for the analysis presented here.

For the Frankfurt/Main case, the full transcripts of the conversations were part of the paper published by Klein [1979] and also used for signal extraction discussed here. The author of this study approached participants on the street and asked them to give directions to two destinations near the city center. The conversations were recorded without knowledge of the participants.

4.2.2 *Coding Process and Analysis*

Transcripts collected from human subjects are open-ended records, i.e., the data is unstructured, and thus difficult to analyze. To allow for an analysis, the collected transcripts were interpreted using a coding approach proposed by Montello and Sutton [2006]. Coding consists of a segmentation and classification process. During segmentation the data records are broken into "appropriate" and analyzable units. Montello and Sutton [2006] note that this process is largely a matter of theoretical and conceptual consideration.

The goal of the analysis was to extract the signals that are used by contributors to a conversation in order to coordinate how the conversation should progress. Thus, it was assumed that appropriate segments are transitions of speech from one contributor to the other, i.e., moments a speaker becomes a hearer and vice versa (turn-taking). During classification, segments are grouped into "meaningful" categories. The classification produced the entities (signals) used to negotiate the meaning during conversation.

The coding process was carried out in two runs. The first run was done using the Vienna dataset (cf. [Weiser and Frank, 2013]) with the goal to perform an exploratory signal collection. The notion of signals was then refined by carrying out the same analysis with the transcripts available from the Frankfurt/Main dataset [Klein, 1979]. The analysis involved both a bottom-up and a top-down approach. While a bottom-up approach extracts categories based on the data itself, a top-down approach defines categories before the analysis is carried out. Categories formed in a top-down manner are based on theoretical considerations. For example, it was assumed that two types of signals [Clark, 1996] do exist (cf. 2.4.1): Type 1 signals are concerned with the official business of the conversation. As such they are communicative acts. Type 2 signals are concerned with making sure the communication is successful.

Furthermore, the coding process was aided by suggestions provided by Clark [1996], as well as Clark and Wilkes-Gibbs [1986] on how to spot signals. The signal types can roughly be classified by temporal placement, language rhythm, and gestures (not considered). For example, acknowledgment and truncation of speech can be interpreted as "I understand what you said" and "I'm already understanding what you are saying". Uncertainty makers, such as expressions of discomfort, can give hints on a possible not understanding. Collaborative Completion of sentences on the other hand can be interpreted as "I think you mean this but I am not sure". Also, the presentation of information in a bit-wise fashion can be interpreted as a request for confirmation.

4.2.2.1 *Note on data collection and analysis*

The study of language data, i.e., in the form of transcripts, is notoriously difficult. This is especially true if one wants to minimize the introduction of biases during data collection. For example, in order to collect dialog data in the context of giving and receiving route direction, several approaches are possible.

First, the experimenter may ask different people for instructions to different destinations the participants are familiar with but the experimenter is not (e.g., directions to the participant's homes). The advantage of this approach is that the experimenter is not likely to be biased towards the geographic space in question. It is relatively easy to control that he is unfamiliar with it. Also, signals are likely to be used from both the participants and the experimenter. The disadvantage, however, is that destinations are not directly comparable because they are different. Note, this approach has been carried out for the exploratory data analysis (i.e., the first step) in this work.

Second, language data may be collected by asking different people to give directions to the same location by the same experimenter. The advantage of this approach is that instructions are easier to compare, because they come from different people but have a common destination. The disadvantage is that the experimenter becomes familiar with the instructions over time and builds a mental model over the course of the experiment. This might have an effect on the frequency of signals used by both the experimenter and the participants. The deteriorating effect of getting used to the presented instructions in terms of signals, however, cannot be tested. Also, participants in this particular set-up might not have the same excellent knowledge of the geographic space as compared to the first set-up (People gave instructions to their homes). Note, data collection in the Frankfurt/Main case was conducted in this manner.

Third, language data can be collected by studying direction giving among participants, leaving the experimenter in only a passive role. This approach has the advantage that signal use is likely to be unaffected over time. However, the disadvantage of this approach is, similar to the first set-up, no comparison of the destinations in terms of the signals can be carried out.

As a compromise, this thesis extracted signals based on data that was collected using the first and second approach. This can be justified because the main research question of this work (related to signals) was to determine what signals do exist in an interactive exchange of route directions. The study of their frequency as a function of the geographic space and mental models of the person who requested the instructions would be another research topic and is not addressed here.

4.2.3 *Signal Presentation*

Note, in the following discussion, the receiver and sender of the instructions are denoted by T (Target) and S (Source), respectively. Since the goal of signaling is to agree on the meaning of an assertion of a particular instruction, any signal can belong to either of the following two categories: Present and Accept-Reject. For example, one may assume that in case an assertion presented by S is directly ac-

Signal Categories: Present vs. Accept-Reject cepted by T, the meaning of the assertion has been negotiated. This is because of this act both individuals have agreed on its meaning. In case an assertion from S is rejected by T, a repair sequence needs to be initiated. This repair sequence then consists of further present and accept-reject exchanges between the two participants. This view is in line with Clark [1996], as well as Clark and Schaefer ([1987], [1989]), who argue that any contribution to discourse can be viewed as either belonging to a presentation or an acceptance phase.

This thesis, however, introduces a finer distinction of the signals, necessary to model conversations in the context of an exchange of instructions. The notion that an assertion can be rejected and then needs to be refined distinguishes the models (conceptual and computational) presented in this work from previously proposed models that implicitly assume an alignment of speaker and hearer (See Section 3.4).

Note, all examples are tagged and refer to the Klein data set included in the Appendix B. Examples can belong to either of the two destinations, i.e., Oper (O), or Goetheplatz (G). Any example is tagged according to the person who made the utterance (S=Source vs. T= Target) and a number that indicates when the utterances occurred. For instance, the tag "O2-S5" means that the utterance is part of the second transcript dataset (destination Oper) and was made by the source as the fifth utterance during the conversation.

4.2.3.1 *Present (Perspective from S = Source)*

Some signals are exclusive to the source (provider) of the information during a conversation:

- One type of a present signal used by S is an **assertion**. An assertion can be of any of the assertion classes mentioned in Section 4.1, i.e., on the most general level either an action or a description. An assertion is a **type 1** signal (official business). See for example *O2-S5*: "[go] across the large square" ([geh] ueber den grossen Platz).
- Another type of a present signal used by S is a **probe**. A probe is used to inquiry about knowledge the source suspects to exist in the target. Note, a probe implies the use of a "theory of mind". In order to suspect some information to exist in T, the information providing agent S has to model the mental state of T before. The constructed theory is then used to determine an instruction, S thinks would be appropriate for T. A probe is a **type 2** signal (meta-communicatoin). See for example *O9-S1*: "Are you somewhat familiar with the area here?" (Kennt ihr euch hier einigermassen aus?).

4.2.3.2 *Present (Perspective from T = Target)*

Similar to the signals that belong to the source, some signals are uttered by the target alone:

- One present signal by T is a **ratification**. With T's attempt to ratify an utterance made by S (e.g., an assertion) they signal that the assertion was accepted. Ratify is a **type 1** (i.e., official business) signal. In terms of a "theory of mind", a ratification by T means that S can assume that the demand of the presented type of assertion was met.

- **R-StartConversation**: During the initiation of a conversation, a request by T is used to establish whether S knows the destination for which T want the description. More generally, T can initiate the spatial knowledge transfer by framing the question that makes clear the intention of what kind of information is needed. See for example *O8-T1*: "Can you tell me how to get to the old opera house?" (Koennen Sie mir sagen, wie man zum alten Opernhaus kommt?). This signal is of **type 1**, i.e., concerned with official business.

4.2.3.3 *Present (Perspective from S=Source or T = Target)*

- **Acknowledge:** A signal both participants of a conversation can use. Acknowledge is a **type 2** signal, i.e., concerned with meta-communication. Its goal is to signal that for example a R-Confirm signal (type 2) was understood correctly. This can happen through explicit assertions of understanding (e.g., "yes", or "ok") or through implicit assertions of understanding, e.g., continued attention. Note that there is a difference between ratify and acknowledge. Ratify is a type 1 signal and is meant to signal that an assertion can be ratified. Acknowledge is only used to signal understanding as a meta-communicative act. For example, this implies that the utterance "yes" as response to an assertion does not have the meaning "yes, I ratify" but rather "yes, I understood what you said". Ratification happens implicitly (cf. Clark [1996]) while acknowledgment can occur either implicitly or explicitly.

Furthermore, S and T can choose among a class of different possible **requests**:

- **R-LoDChange:** With this signal S and T request a change in the level of detail of an utterance. This signal is of **type 2**, i.e., concerned with meta-communication. From the perspective of S, this is a typical signal used during the initiation of the conversation in which S tries to negotiate the meaning of T's request. See for example G1-T2: "To the Goethehaus" (Zum Goethehaus?) and the response by G1-S2: "Goethehaus? Where?" (Goethehaus? Wo?). S is not able to produce instructions by T's reference to the destination alone and therefore requests more detail that might help to identify the destination. On the other hand, the signals can also be used by T who wants to get S to change the level of detail (LoD) because she cannot ratify a statement at the current presented granularity. See for example O13-T2:

"and how do I get there?" (Und wie komme ich dahin?) as a response to S making the assertion O13-S1: "Yes, the best way is through here" (Ja, da gehen Sie am besten hier durch). Note that a request for a change in the LoD of an assertion can go both ways. For example, in case the source presents too much information the target thinks it can not recall during the way-finding process they can ask for a more concise (decreased LoD) description.

- **R-EndConversation:** In case T thinks instructions are sufficient for the way-finding task to be performed, she can initiate the end of the conversation by indicating that she has received enough information. This is usually done by thanking. In case S cannot continue the instructions to the T's preferred destination S may initiate the end of the conversation themselves and refer to somebody else as the possibly better source of the information. One could argue that S can initiate a request for the end of conversation by uttering the final instruction that leads the target to the destination. However, in this thesis it is assumed that S waits for T to signal that the instructions are sufficient for her purposes (i.e., S is committed to provide T the information she needs). For example, it could be the case that T still wants to receive a summary of the route or any other information that might be relevant for understanding. This signal is of **type 1**, i.e., concerned with official business.

- **R-Confirm**: The goal of this signal is to make sure that the other person has understood a presented assertion ("Did you understand that I asserted X") or ratification. Requests for confirmation are signals of **type 2** (concerned with meta-communication). They can either be explicit or implicit. Note, implicit requests for confirmation are inherent to every utterance. This is because according to the grounding through communication model the source wants to make sure that their utterance has been understood (following the cooperation principles suggested by Grice [1989]). An explicit request for confirmation is sometimes added after an assertion was made. See for example *G16-S5*: "You did understand, right?" (Sie haben verstanden, oder?).

- **R-Repeat:** This signal is often used to make sure some information has been understood correctly. Secure is a **type 2** signal (meta-communication). See for example the utterance G1-S1: Goethehaus? as a response to the utterance G1-T1: "Can you tell me how to get to the Goethehaus?" (Koennen Sie mir sagen wie man zum Goethehaus kommt?). The intention behind a repeat request might be caused by the goal to reinforce some information i or to signal that it was not understood. Note, a repeat signal by T can trigger an adjustment of a "theory of mind" in S. For example, this is the case if S thinks the asserted instruction did not meet the information demand of T.

Signals Role/Type	**Type 1 - Official Business**	**Type 2 - Meta-Communication**
S	Assertion	Probe
T	Ratify, R-Start	-
S or T	R-End	Acknowledge, R-Confirm, R-LoDChange, Repeat

Table 3: Type 1 and 2 Signals (Present) as used by S and T

Table 3 provides a overview of the various signals made by S and T as well as a classification whether they belong to type 1 (official busines) or type 2 (meta-communication):

4.2.4 *Signal Acceptance and Rejection*

Morrow et al. ([1993], [1994]) and Clark [1996] provide a classification of evidence that indicates whether a signal was understood and accepted. In general, there are explicit and implicit signs of understanding and acceptance:

Explicit and Implicit Signs of Understanding

- **Explicit** confirmation of understanding: Responses such as hm, yes, nodding, etc.
- **Explicit** exemplification of understanding: Verbatim Repetition, Paraphrases, etc.
- **Implicit** presupposition of understanding: Uptake of a proposed joint action, or through the initiation of the relevant next turn
- **Implicit** display of understanding: The way the addressee responds to the speaker's utterance, or through continued attendance

Furthermore, Clark and Brennan [1991] argued that the absence of negative evidence (e.g., T has not rejected S's assertion, therefore it is accepted) is usually not enough in real conversations between humans. They proposed that a sign of acceptance has to be given providing positive evidence. Note that this is often done by gesture and thus not explicitly considered in the material used for this study. However, positive evidence can also provided through explicit acknowledgments or by an initiation of the relevant next turn (T has requested i, therefore S asserts i).

Clark and Brennan [1991] also note that the problem of infinite regress (T's acceptance of S's acceptance of T's acceptance, etc...) does not occur in real conversations because positive evidence of acceptance is always provided, at least at the level of continued attention. In other words, as long as the recipient attends my utterances I can assume that she accepts it. Additionally, infinite regress does not occur because some types of evidence for acceptance (e.g., initiation of relevant next turn, continued attention) do not have a separate presentation phase.

4.2.4.1 *Acceptance of Signals*

Judging from the data analyzed in the study, the source (S) usually accepts T's signals by initiating the relevant next turn. In our example, S continues with the route description, unless T signals otherwise. In case a signal is rejected, the response is a present signal indicating the reason why the signal was rejected. This in turn is the start of a repair sequence.

Consider the following excerpt from the example conversation O4 (See Appendix B):

- T1: "Can you tell me how to get to the Opera?" (Können Sie uns sagen wie wir zum alten Opernhaus kommen?).
- S1: "Yes. Now, you go on, until the end [of the street]" (Ja. Jetzt gehen Sie vor, bis ganz vorn hin).
- T2: "Yes" (Ja).
- S2: "Until you get to Kaufhof" (Bis Sie an den Kaufhof stossen).

The first contribution starts with T asking for directions to the old opera house. This means that T presents a **R-StartConversation** signal This is accepted by S by initiating the next turn, i.e., S accepts T's request by giving the first relevant instruction. This also marks the start of the next contribution to which S asserts the first instruction (utterance S1). The source's assertion is acknowledged, i.e., accepted by T through utterance T2. By presenting this utterance, T also implicitly ratifies the assertion previously made by S. The source in turn accepts the target's ratification by initiating the next relevant turn, i.e., the next instruction S2.

As can be seen from the example, the acceptance of a signal occurs either through initiating of the relevant next turn, or by explicit acknowledgment. The concrete type of acceptance depends on the context in which the preceding signal was uttered and the role of the communication partners. This sample also illustrates the "ideal" case of an exchange of route instructions between two individuals (See O5 in Appendix B for a complete example of such a sequence). Thus, the ideal case is a sequence of exchanges in the following form:

- T1: Request_Instruction
- S1: Accept Request through Assert_Instruction_1
- T2: Accept Assertion_Instruction1 through Acknowledgment and Ratify

- S2: Accept Ratify and Acknowledgment through Assert_Instruction_2
- T3: Accept Assertion_Instruction2 through Acknowledgment and Ratify
- [...]

4.2.4.2 *Rejection of Signals*

In case a signal is rejected by either the source (S) or the target (T), the conversation cannot simply go on. The utterance that was not understood needs to be repaired. This is because, in case a signal is rejected the meaning of an utterance has not been successfully negotiated, i.e., one does not agree. The following example illustrates various repair sequences between both conversation partners. Consider the following example (See Appendix, G3):

- T1: "Can you tell us how to get to the Goethehaus?" (Können Sie uns sagen wie man zum Goethehaus kommt?)
- S1: "What?" (Wie?)
- T2: "To the Goethehaus?" ("Zum Goethehaus?")

In this example, T1's request for receiving way-finding instructions to the destination Goethehaus was rejected by S because he could not understand the type of request. Therefore, T's presentation of a request was rejected on grounds of action level 3, i.e., its meaning was not understood.

Another way how a signal can be rejected is by ignoring it. Consider the example from conversation O4-T13/S13:

- T13: "Ok, thank you!" (Gut, dankeschön)
- S13: Then you pass by Hauptwache, yes? (Dann kommen Sie an der Hauptwache vorbei,ja?)

T wants to end the conversation by initiating a R-EndConversation (thanking) because he thinks the received instructions are sufficient for carrying out the way-finding task. Note, utterances by S at this stage of the conversation are part of a summary of the route, but apparently not needed by T. This request, however, is rejected (by ignoring it) because S simply continues with the instructions.

Yet another way how present signals can be rejected is illustrated in the following example from conversation O13 (Also see Appendix B):

- S2: [...] or through Fressgasse, if you know it?
- T3: mh?

S sends a probe signal by asking whether T knows a particular street (Fressgasse). However, the probe is rejected by T.

4.2.5 *Discussion*

This section presented various types of signals that were extracted from the analyzed language data. The nature of the signals and their potential effects on an agent's "theory of mind" were discussed. The results suggested here empirically supports the hypothesis of this work (See Section 1.1) by making the signals used to negotiate the meaning of instructions explicit.

It should be mentioned that in most conversations the roles of source (S) and target (T) of some information exchange are not fixed, but vary. Thus, during the course of a communication a source becomes a target, and vice versa. In this respect the exchange of route instructions is special because the roles of source and target usually remain fixed and do not switch. However, this commitment does not affect the generality of the signals that were presented here.

An analysis of the dataset revealed that explicit requests from T for more information **(R-LoDChange)** did not occur very often. However, this might be due to the fact the transcripts can only show effects that were evident in verbal utterance. Signals made on a gestural level could have led to an adjustment of level of detail. Furthermore, instances where the source asked T to confirm an assertion were rare (**R-Confirm**). This is probably due to the fact that an implicit request for confirmation is always inherent in any assertion. The source simply assumes that if T cannot acknowledge and ratify the assertion then she would say so. Therefore, there is no need to add this information every time, following Grice's [1989] maxims ("Include only as much information as needed").

4.3 WAY-FINDING INSTRUCTION DIALOGS - AN ANALYSIS

To illustrate the use of signal combinations this section introduces the notion of contribution trees, as suggested by Clark and Schaefer ([1987], [1989]). In communication analysis, a contribution tree shows two different things. First, the sequence of present and accept/reject signals over the course of the conversation. Second, the hierarchical structure of the conversation. In case a signal is rejected and a repair sequence is initiated, the "depth" needed to negotiate the utterance is evident from the tree. Note, during the conversation participants themselves do not know at which hierarchical level they are. Therefore, the contribution is completed when the participants are done negotiating the meaning of each contribution (See also Clark and Schaefer [1989]). Note a successfully negotiated contribution is added to the common ground of both participants as described in Section 2.4.1.2

Consider the following example of a simple contribution tree (See Figure 8) and the corresponding utterances:

- T1: “Can you tell me how to get to the Opera?” (Können Sie uns sagen wie wir zum alten Opernhaus kommen?).
- S1: “Yes. Now, you go on, until the end [of the street]” (Ja. Jetzt gehen Sie vor, bis ganz vorn hin).

Here, both conversation partners negotiate the meaning of an instruction (denoted by C for contribution) through a sequence of present and accept/reject signals. The first contribution (See Figure 8) is concerned with establishing the destination and initiated by utterance T1 (“Can you tell me how...”). This is accepted through the initiation of the relevant next turn by S (the diagonal line from right to left connecting the first C with S1). The next contribution is initiated by S making an utterance (“Yes...”) and giving the first instruction. In other words, the acceptance of T’s initial request happens through S offering the first instruction which in turn is the start of the next contribution (the offered instruction by S) that requires T to signal acceptance (or rejection).

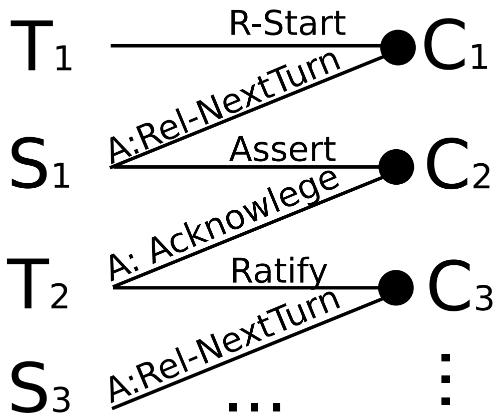

Figure 8: Contribution tree visualization of a sequence of successfully negotiated instructions.

If a presented signal is not directly accepted but rejected instead, it initiates a new present phase that specifies the reason why the signal was rejected. This is the repair sequence during which dialog partners attempt to agree on an interpretation. For example, during a repair sequence the level of detail of an utterance can either be increased or decreased, until both participants have agreed on its interpretation. Consider the following dialog (See O9 in the Appendix B) and Figure 9 for an example of a repair sequence:

- T1: “We’re looking for the old opera house” (Wir suchen’s alte Opernhaus).
- S1: ... Are you somewhat familiar with the area around here? (... kennt ihr euch hier einigermassen aus?).
- T2: “No” (Nein).
- S2: “So, you go...” (Also, ...).

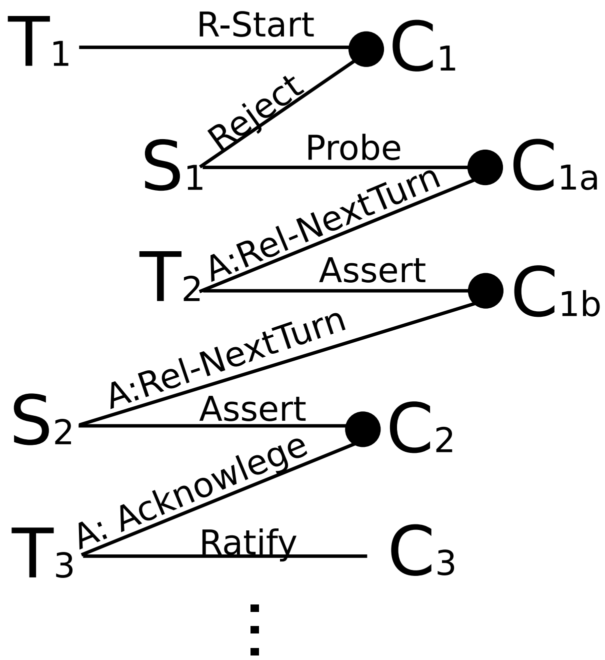

Figure 9: Contribution tree visualization of a repair sequence.

- T3: "Ok"

T starts with making a request (T2) for instructions to the old opera (This starts the first contribution C). However, this request is not accepted by S (as in the example above). Instead, S sends a probe (utterance S1) to figure out how much T knows about the geographic area. Note, the probe initiates a new contribution phase that is one level deeper in the hierarchy of the tree. The probe in turn is accepted by T through utterance T2 ("no") by giving S the information they needed. S can now initiate the relevant next turn, i.e., giving the first instruction to the requested destination. This also marks the start of the third contribution, now at the top level of the tree.

The first thing the analysis of the data revealed was that not only the exchange of instructions, but also initiation and the closure of the conversation was negotiated. The distinction of three phases, initiation, route instructions, closure is at par with previous suggestions of how a conversation in this context can be structured (See Section 3.4).

4.3.1 *Negotiation during the Initiation Phase*

4.3.1.1 *The Initiation Phase Fails Completely*

In case the initiation phase fails no route instructions can be given. However, this does not mean that the source is not willing to help the target completing the way-finding task. Consider example conversation G1 (Also see Appendix B) and the corresponding contribution tree (See Figure 10):

- T1: "Can you tell me how to get to the Goethehaus?" (Können Sie mir sagen wie man zum Goethehaus kommt?).

- S1: Goethehaus?
- T2: “To the Goethehaus” (zum Goethehaus).
- S2: “Goethehaus? Where?” (Goethehaus? Wo?).
- T3: “Well we don’t where exactly” (Wissen wir eben nicht genau).
- S3: “Goethehaus? No address?” (Goethehaus? Keine Adresse).
- T4: “No, Großer Hirschgraben I think it was” (Nee, Großer Hirschgraben war das, glaub ich).
- S4: Sorry? (Bitte?).
- T5: “Großer Hirschgraben, the street...” (Großer Hirschgraben, die Straße).
- S5: [5 sec pause]
- T6: “You don’t know, we’ll ask again” (Wissen Se nicht, fragen wir nochmal).

An analysis of the contribution tree reveals the following hierarchically structured attempts to (unsuccessfully) negotiate the meaning of the destination. With utterance T1, the target makes a request to S for getting instructions to the Goethehaus. Instead of initiating the relevant next turn (thus accepting the request) and providing instructions the source rejects T1:requestStart by initiating a S1:RequestSecure. This is accepted by T2 initiating the relevant next turn and acknowledging the secure signal. However, S rejects this again and provides a S2:requestLoDChange signal. T cannot provide more information (T3 rejects S2:requestLoDChange). Both T and S further attempt to negotiate the meaning of the initial request but eventually fail to achieve agreement. The target finally ends the conversation with T6:requestEnd.

4.3.1.2 *The Initiation Phase Succeeds (Simple Case)*

In the simplest possible case the initiation phase is ended by S who initiates the relevant next turn as response to T’s signal requestStart. For example, consider dataset O7:

- T1: “Can you tell me how one gets two the old opera house?” (Können Sie mir sagen, wie man zum alten Opernhaus kommt?).
- S2: “Yes, you go until ...” (Ja, Sie gehen bis ...).

4.3.1.3 *The Initiation Phase Succeeds (Extended Case)*

Often, the initiation phase requires longer for both participants to establish a mutual understanding of the destination for which the instructions should be given. Consider the initiation phase of dataset G3 (Also see Appendix B):

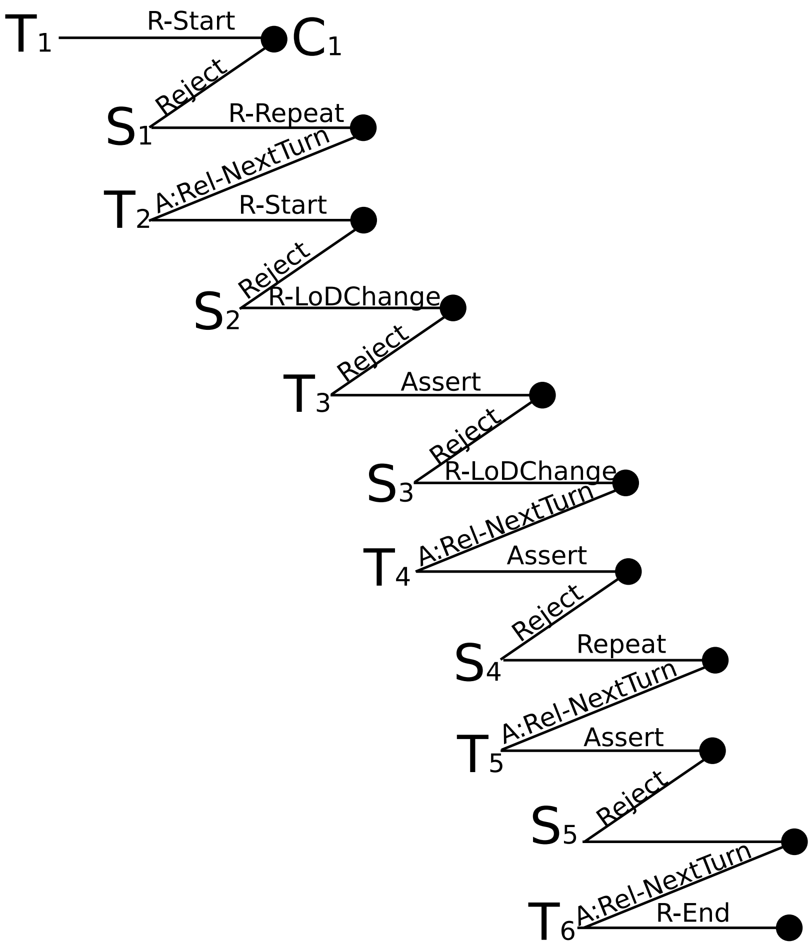

Figure 10: An example for the failure of the initiation phase

- T1: "Can you tell us how to get to the Goethehaus?" (Können Sie uns sagen, wie man zum Goethehaus kommt?).
- S1: "What?" (Wie?).
- T2: "To the Goethehaus?" ("Zum Goethehaus?").
- S2: "Gueterhaus?".
- T3: "Goethe".
- S3: "Hm, I think, it's somewhere, Goethe, Goethesquare, Goethesquare and Goethehaus, heh, I think this is it, isn't it?" (Hm, ich glaube, das ist hier etwa, Goethe, Goethe, Goetheplatz, Goetheplatz und Goethehaus, he ich glaube da ist das oder?).
- T4: "Very close of there" (Ganz in der Naehe davon).
- S4: [starts with instructions].

An analysis of the contribution tree (See Figure 11) reveals the following: First, T's T:requestStart signal for route instructions is not identified by S. S rejects the signal by presenting a requestRepeat signal indicating that she did not understood what T meant. T accepts the request by initiating the relevant next turn, which is in this context the repetition of the requestStart (T2).

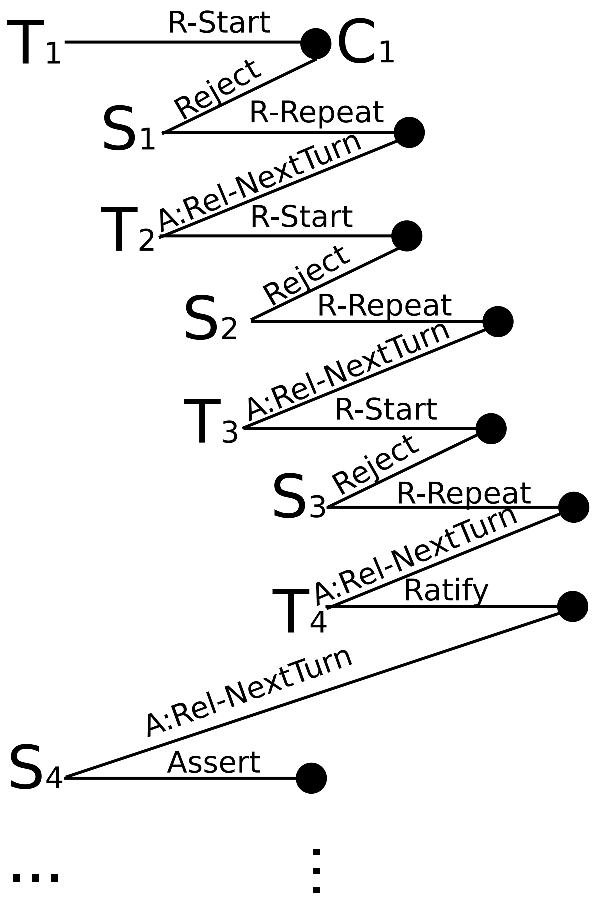

Figure 11: An example for a successfully negotiated initiation phase

The source responds to this by a S2:requestConfirm signal but the corresponding utterance is rejected by T making a third request indicating that S has not correctly understood the first word in the destination (Gueter vs. Goethe). With utterance S3 the source indicates that she has finally understood the meaning of Goethehaus as the desired destination but is not entirely sure where it is located (The source does not (yet) offer an uptake of the request). Instead, S frames a question that is an attempt to secure the destination (requestSecure). The contribution concerned with the identification of the destination (such that the source can provide directions), is finally ended after T acknowledges the securing attempts by S.

Note that T's three consecutive requests exhibit themselves a LoD change. First, the entire question is framed (T1). Then a shorter version is framed (T2). Finally the request only states Goethe (T3). This is because the target suspects that S has understood that he wants a route instruction to a destination (correctly identified after the first request) and the destination consists of the word "haus" (Correctly identified after the second request).

4.3.1.4 *Discussion - Initiation Phase*

For the Opernhaus dataset (See Appendix B: O1-O18) cases where the initiation phase consisted of more than just a present and an accept by initiating the relevant next turn (i.e., starting the instruction) occurred

5 out of 18 times. In the other 13 cases the initiation phase did not occur on a deeper hierarchical level. For the Goethehaus case (See Appendix: B G1-G17), 1 out of 17 initiation phases failed completely. Also, 7 out of 17 phases required some negotiation of meaning that exceeded a simple present and accept signal exchange. 9 out of 17 phases did not require a more in-depth negotiation of the destination.

4.3.2 *Negotiation during Route Instruction Phase*

The following examples illustrate some cases during which it was required to negotiated the meaning of instructions.

4.3.2.1 *Explicit requests for a LoD change*

Consider the following excerpt from example conversation O13 and the corresponding contribution tree (See Figure 12).

- S1: "Yes, the best way to go is through there" (Ja; da gehen Sie am besten hier durch).
- T2: "And how do I get there? (Und wie komme ich dahin?).

T explicitly asks for more information concerning the previously suggested instruction (requestLoDChange). The now following exchange of utterances are concerned with establishing the name of the street, S refers to as "through there". Once this fact has been established (sequence S2-T3-S3-T4-S4), S starts with the presentation of the route instruction again. T, however, now wants to get more information on how to get across (The assertion of the source is not directly accepted). Apparently it is not clear to T how to get across the street (it might not be possible to cross the street. S gives this information by accepting the requestLoDChange and offering the requested information. The conversation can then continue after the meaning of this information was negotiated.

- S5: "Well, there [you go] straight" (Also hier geradeaus)
- T5: "How do I get across?" (Wie komm ich denn da rueber?)
- S6: "Underneath, yes underneath [...]" (Unten durch ja unten durch [...])

Another example where the target requests a level of detail change can be witnessed in the transcript for conversation G7 (See Apendix):

- S3: "Across, over there across to the left, it's almost there, you need to go left, I believe the next one left again, There it should be. In any case it's not far." (Rüber, da vorne links rüber, ist's schon fast da, müssen Se links durch, ich glaub die nächste wieder links, da müsste es dann sein; also weit von hier ist es auf jeden Fall net).
- T4: "So, here" (Also hier).

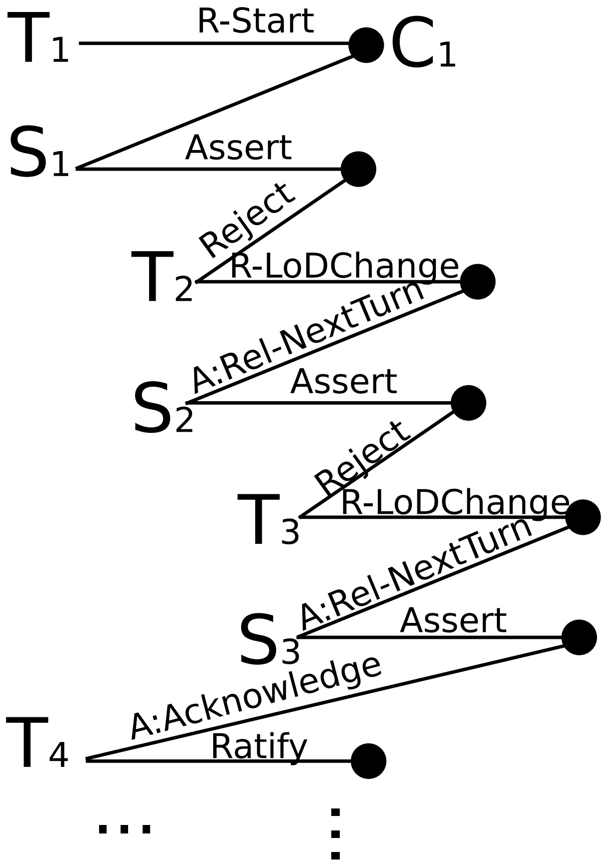

Figure 12: An example for two explicit requests for a LoD Change made by T

- S4: "So, over there enter the street on the left, and then you need to turn left again, then you're almost there" (Also hier vorne die Straße links rein, und dann müsse Se wieder links gehen, und dann Sind Se schon fast dort).
- T5: "Alright" (Ist gut).

It seems T does not judge the assertion S3 as good enough to accept it. Instead, T requests a level of detail change (requestLoDChange) which is in turn answered by S with a condensed version of what he said before. T judges the repetition to be much clear now and accepts it to be sufficient for the current purpose.

4.3.2.2 *Discussion - Route Instruction Phase*

An analysis of the contribution trees revealed that the negotiation of contributions during the exchange of route instruction phase was not as common as initially suspected. In the Goethehaus case, only 16 out of 17 transcripts were analyzed in terms of their exchange of route instructions (In one case the source could not identify the destination). Out of the 16 remaining transcripts an extended negotiation of utterances during the actual exchange of instructions occurred 6 times. The other 10 times only the standard sequence of present-accept signal pairs occurred. In the Opernhaus case, 6 out of the 18 contained at least one instance where the meaning of an actual instruction had to be negotiated. The remaining 12 transcripts did not show any sign of divergence between the interpretations of terms from the perspective of S and T.

4.3.3 *Negotiation during Closure Phase*

The following examples illustrate some cases during which it was necessary to negotiated the meaning of conversation closure.

4.3.3.1 *The Closure Phase Succeeds (Simple Case)*

In case the requestEnd signal by T (meant to end the conversation) is accepted by the source the conversation simply ends. Note that in many cases S does not give explicit evidence of understanding. This could be seen as an instance where the absence of negative evidence is used as a way to accept a presented signal.

4.3.3.2 *The Closure Phase Succeeds (Extended Case)*

There are cases during which the T's signal to end the conversation is simply ignored by S. As a result S continues the conversation. Consider the (extreme) example of dataset O4 (See also Appendix B) during which S does not accept T's requests for the conversation to end. In fact, T's attempts and signals to end the conversation are ignored two times (T11 and T13) until with utterances T16-S16 the conversation finally ends.

- T11: "Mhm, thank you" (Mhm, dankeschoen)
- S11: "Well, you can't go wrong ..." (Also sie koennen gar net fehlgehn ...)
- [Instructions continue]
- T13: "Good, thank you" (Gut, dankeschoen)
- S13: "Then you pass Hauptwache, yes?" (Dann kommen Sie an der Hauptwache vorbei, ja?)
- [Instructions continue]
- T16: "Thank you" (Dankeschoen)
- S16: "You're welcome" (Bitteschoen)

4.3.3.3 *Discussion - Closure Phase*

The case where T's endRequest signal was ignored by the source occurred only once for each of the two datasets. In addition, most endRequests by T were accepted tacitly by S without making an additional comment.

4.4 conclusions

This chapter developed a conceptual model illustrating the components and processes that are part of a negotiation of meaning in the context of way-finding instructions. Several signals were extracted from real language data and a contribution tree analysis of the corpus was performed to analyze the hierarchical nature of conversations. More specifically, the contribution of this chapter is the following:

1. A shared understanding of way-finding instructions was proposed in the form of a taxonomy. The taxonomy attempted to unify previous classifications that were specified either too coarsely or did not consider crucial types of information (in the sense of cognitively adequate), such as landmarks.

2. In a two-step coding process a number of language transcripts was analyzed to extract a set of signals agents can use during the exchange of instruction to negotiate an instruction's meaning. Possible relations between signals and an agent's "theory of mind" were also drawn. Previous models that attempted to explain the communication of route instructions did neither consider signaling between between human individuals nor treated their relations to agents having a "theory of mind".

3. An analysis of three phases that occur during the exchange of instructions was performed with respect to the occurring signals. In particular, during each phase of an exchange of instructions, i.e., initiation, route instructions, and closure, the negotiation of the meaning of utterances is needed if the conversation is to be successful. Signals are used to indicate the success or failure of conversations between a knowledgeable source and another target agent.

The in-depth analysis of the conversations on a signal level as well as the notion of signals as ubiquitous negotiation devices is one of the main contribution of this thesis. In addition, the first two steps serve as one of the inputs necessary to specify the computational model developed in chapter 5. The third step helped to empirically support the hypothesis stated in the introduction, i.e., the meaning negotiation process emerges through the use of signals and a theory of mind.

5 COMPUTATIONAL MODEL

This chapter discusses the developed computational model and is the main contribution of this thesis. It formalizes the notion of meaning negotiation (including a "theory of mind", and signals), a theory developed from empirical arguments presented in Chapters 2 and 3 and conceptually analyzed in Chapter 4. The computational model was implemented using the functional programming language Haskell [Peython Jones, 2003], and allows for a simulation of a negotiation process during which agents exchange route instructions. The presented model is thus a proof of concept of the previously developed theory. Note, the presented code listings are only excerpts to support the argumentation in the text. The complete executable Haskell code can be found in the Appendix A.

This chapter is structured as follows. First, a general overview on the basic architecture of each module is given. This includes a discussion of the used formal representation of the negotiated route, the capabilities of the modeled agents (e.g., having preferences and evaluating other agent's preferences), and a formal treatment of the route instruction taxonomy introduced in Section 4.1. Second, the processes that occur during the the exchange of instructions are discussed. This includes a description of how instructions are generated, based on the evaluation of a "theory of mind" and the signals used during the conversation. Third, the functionality of the computational model is demonstrated using several test cases. The chapter concludes with a discussion of the presented results.

5.1 OVERVIEW OF THE BASIC ARCHITECTURE

This section gives an overview of the architecture for each component of the computational model, i.e., the used data structures. The design choices are a direct consequence of the empirical evidence presented in previous chapters. The first two subsections (See 5.1.1 and 5.1.2) discuss the modules "route" and "landmarks", respectively. These two modules formalize the representation of the geographic space on which the exchange of instructions between agents is based. The following section (See 5.1.3) presents the basic data structure for each agent with a particular focus on an agent's "theory of mind". This is followed by the formal representation of signals (See 5.1.4), originally introduced in Section 4.2. In the last subsection (See 5.1.5), a formal model of the taxonomy of instructions, introduced in Section 4.1 is presented.

5.1.1 *Module Route*

As argued in Section 2.1, an agent needs some form of mental representation of the geographic space in which way-finding decisions are to be made. This module describes the representation of the route each agent stores during the exchange of instructions.

Note, the model assumes that the source agent has already preselected the necessary parts to navigate between a given start and goal when the conversation starts. Therefore, the route is not generated dynamically from a richer geographic representation, but assumed to be static. This commitment, however, does not affect the generality of the described approach. This thesis emphasizes the communication of instructions between agents, not the generation of potential routes.

It is assumed that an agent's representation of a route consists of a list of segments (See Listing 1). A segment connects two decision points with a directional information on how to get from one decision point to the next (This somewhat resembles Klippel's formalism described in Section 3.2). Directions are coded relative to the agent who is assumed to be located at a starting node (n) and facing the next node (n+1). Decision points are assumed to link to an underlying graph structure (i.e., a node id equals the corresponding decision point id) and can include one (optional) landmark (See also Section 5.1.2).

Note, a decision point could, in principle, host several landmarks. The model, however, assumes the agent has either selected one particular landmark or none at all, but not several landmarks. The necessary selection process could be based on a saliency measure for landmarks, as for example proposed by Raubal and Winter [2002].

Listing 1: Module Route: Data Structure

```
type Segment = (DecisionPoint, Direction, DecisionPoint)
type Route = [Segment]
data DecisionPoint = DP {getNode :: Node, getLM :: LM} deriving (
    Show, Eq)
-- relative directions: An agent can go straight, left, or right.
data Direction = Str | L | R | EmptyDir deriving (Show, Eq)
```

In addition to the model's limitation that no more than one landmark is present at a decision point, three other commitments are made.

First, the route is assumed to be correct, meaning it does not mention non-existing landmarks, wrong turns, or wrong connections between segments. This commitment, however, does not affect the exchange of instructions, because the incorrectness of information can only be discovered by the target agent during the way-finding task (e.g., if an instruction to turn is not possible because there is no intersection) but not during the conversation. During the exchange the target has no means of determining the correctness of an instruction because it can not directly be compared to the environment. Thus, lifting this commitment would not add to the usefulness of the presented model.

Second, the route is assumed to be complete, meaning that at least the turning information between decision points are available to the source (S) agent. This would result in a minimal set of instructions, e.g., "turn right, then left, then go straight...". We can assume that an agent who is asked by somebody else to provide instructions does only agree to the task if she has at least this kind of basic information. Lifting this commitment then allowed an agent to produce at least a partial route to the requested destination, but the actual process of exchanging instructions would be unaffected.

Third, apart from the preselected route, there are no alternative routes available to S. This implies that if both agents cannot agree on a particular segment of the route, no alternative route is calculated on the fly and the conversation ends (For details see Section 5.2). Lifting this commitment would require an agent to recalculate alternative routes (here, assumed to be static) based on a richer representation of the environment. This would likely increase chances that a route is successfully negotiated, but has no impact on the generality of the approach presented here.

Three example routes (one complete, one minimal, and one partial) are illustrated in the following code snippet:

```
exampleRoute, minRoute, partialRoute :: Route
exampleRoute = [(DP (N 1) newEI, Str, DP (N 2) restaurant),
                (DP (N 2) restaurant, L, DP (N 3) bikeShop),
                (DP (N 3) bikeShop, Str, DP (N 11) cafe)]
minRoute = [DP (N 1) NoLM, Str, DP (N 2) NoLM]
partialRoute = [(DP (N 1) newEI, EmptyDir, DP (NoNode) NoLM),
                (DP (NoNode) NoLM, EmptyDir, DP (NoNode) cafe)]
```

In case of the target agent, it is assumed that at lesat some minimal landmark information is available at the start and end of the route for which the agent wants to receive instructions.

In addition, the module includes some methods the agents can use to modify and retrieve (i.e., constructors and observers) decision points and segments. The methods are used to integrate new information (i.e., way-finding instructions) with an already existing representation of the route. Note, in the following code (See Listing 2) only the type classes for operations on decision points (DPOps) and segments (SegOps) and their corresponding function signatures are given. See Appendix A.1 for the actual implementations of the specified methods.

Listing 2: Module Route: Constructor and Observer Functions

```
class (Eq dp) => DPOps dp where
        isNodeEqual :: dp -> dp -> Bool
        isLMEqual :: dp -> dp -> Bool
        isNoNode :: dp -> Bool
        isNoLM :: dp -> Bool
        updateDP :: dp -> dp -> dp

class SegOps seg dp dir | seg -> dir, dir -> dp where
        getStartDP, getEndDP :: seg -> dp
        getDir :: seg -> dir
        setStartDP, setEndDP :: seg -> dp -> seg
        setDir :: seg -> dir -> seg
        isSegConnected, isSeqEqual :: seg -> seg -> Bool
        updateSeg :: seg -> seg -> seg
        updateStartDP, updateEndDP :: seg -> dp -> seg
        updateDir :: seg -> dir -> seg
        isStartSeg, isEndSeg  :: seg -> seg -> Bool
```

5.1.2 *Module Landmark*

Landmarks are physical point features that have salient characteristics, making them distinct from other elements in the environment [Lynch, 1960]. Numerous studies (See Chapter 3) have shown that humans prefer instructions that include landmarks over instructions based on other kinds of information.

This module specifies the data structure and the corresponding constructors and observables for a formalization of landmark information. This information can be included in the generated instructions and is an integral part of the representation of a route to be negotiated.

The data type LM (See Listing 3) has three different data constructors, i.e., SimpleLM, LM, and NoLM. The constructor SimpleLM mentions only the category (Cat) of a landmark and the associated identifier (ID). The data constructor LM includes additional attribute information, e.g., a proper name or a general descriptive term (e.g., green, tall, etc.) for the represented landmark. The distinction is made because one can refer to the same landmark using its basic category (cf. [Rosch, 1978]), such as "building", or by using (a combination of) its proper name and other attributive information, e.g., "Technical University". Finally, the constructor NoLM is used as the formal internal representation in the case where no landmark is presented at a decision point.

Listing 3: Module Landmark: Data Structure

```
data LM = SimpleLM {getID :: LMID, getLMCat :: LMCat} | LM {getID
     :: LMID, getLMCat :: LMCat, getAttributes :: Attributes} |
    NoLM
newtype LMCat = LMCat Cat
data Cat = House | Restaurant | Shop | Bank | Fountain | Church |
     NoCat
type Attributes = Map.Map String String
```

The operations to construct and modify landmark information are specified by the type class Landmarks (See Listing 4). See the Appendix A.2 for an implementation of the specified function signatures.

Listing 4: Module Landmark: Constructors

```
class Landmarks i cat atts lm | lm -> i, lm -> cat, lm -> atts
    where
        makeSimpleLM :: i -> cat -> lm
        makeLM :: i -> cat -> atts -> lm
        lmToSimpleLm :: lm -> lm
        updateLM :: lm -> lm -> lm
```

The following code snippet shows two examples of a formal landmark representation, one including attribute information (library), the other not (house):

```
library = LM {getID = LMID 10, getLMCat = LMCat House,
    getAttributes = fromList [("fascade","owl sculpture"),("name"
    ,"Library")]}
house = SimpleLM {getID = LMID 1, getLMCat = LMCat House}
```

5.1.3 *Module Agent*

This module describes the behavior and representation of an agent that takes part in a negotiating process of way-finding instructions. For the complete code of this module see Appendix A.3.

Note, agents are tagged according to their roles (Source vs. Target). Type tags (See Listing 5) allow to make a distinction between the role of each agent statically on the type level. The data type that represents the agent consists of three data constructors and their corresponding functions (See Listing 5). The data constructor Route uses the function getRoute for a basic getter and setter functionality, i.e., to retrieve and modify the stored route. The route can be conceptualized as a form of mental representation that stores the geographic knowledge necessary to generate the instructions (See Sections 2.1 and 5.1.1).

Listing 5: Module Agent: Data Structure

```
data Agent = Agent {getRoute :: Route, getInstrList :: InstrList,
     getPrefMap:: PrefMap}

-- type tags
data Source
data Target
data Role r where
        Source :: Agent -> Role Source
        Target :: Agent -> Role Target

-- example agents
testAgentSource = Source $ Agent exampleRoute emptyInstrList
    exPrefMap
testAgentTarget = Target $ Agent partialRoute emptyInstrList $ PM
     []
```

The data constructor InstrList is used to keep track of possible, tried, and succeeded instructions that are exchanged during the negotiation process. The third data constructor PrefMap stores the assumed preferences of the other agent taking part in the conversation. Note, this is the model's implementation of a data structure that represents an agent's "theory of mind" (See Section 2.4.2). For example, for any given source agent, PrefMap stores the assumed preferences of the target agent. Note, stored preferences are updated dynamically during the conversation depending on the reactions (signals) of the other agent. The implementation of agents having a "theory of mind" is further discussed in the next subsection.

5.1.3.1 *The Dialog Partner's Assumed Preferences (Perspective of the Source)*

Each agent keeps track of both their own preferences and the preferences of their dialog partner. Assumed preferences of each agent's dialog partner are updated dynamically, depending on the progress (indicated by signals) of the conversation (See Section 5.2 for a detailed discussion of the negotiation process). Having a set of assumed preferences of dialog partners and the capability to update them dynamically, models the concept of a "theory of mind" (As introduced in Section 2.4.2).

The model assumes that there is a semantic correspondence between both agent's concepts. For example, if agent A talks about a progression at a landmark the other agent knows what is meant, given they have the capability to understand the concept of a progression at a landmark. In other words, there is no inter-agent uncertainty about a concept if it is shared by both agents. However, if one agent uses a concept the other cannot understand a repair sequence has to be initiated.

The source agent needs to evaluate the assumed preferences of their dialog partner (if they exist) in order to generate possible instructions. This behavior is implemented by the type class EvalPrefs (See Listing

6). The code describes the specification of the type class and its instantiation for the instruction type "Progression". In case an agent has no assumed preferences of their dialog partner, the most general instruction of an instruction type is selected instead. This models the behavior of an agent who wants to minimize the cognitive strain introduced by longer and more complicated instructions. In other words, the agent is assumed to operate by the principle of economy (cf. [Simon, 1998]) if no better information on the preferences of the target agent exists. Alternatively, an agent who wants to maximize the information content could be modeled by selecting the most specific type of instruction.

Listing 6: Module Agent: Evaluation of Assumed Preferences (ToM)

```
class EvalPrefs t where
        evaluateP :: String -> PrefMap -> Maybe [(DecisionPoint
            -> Direction -> Maybe t)]

instance EvalPrefs Progression where
        evaluateP p prefs = if null (getAssumedPrefs p prefs)
            then Just [makeProgress] else Just $ map prefToInstr
            (getAssumedPrefs p prefs)
```

The function evaluateP takes a String, representing the type of an instruction (e.g., "progression"), and a PrefMap ("theory of mind") and returns a list of partial functions. The partial functions are then used to generate possible instructions at a given decision point (See Section 5.2.3).

In addition, agents have the capability to update already stored preferences should the conversation require it (e.g., if dialog partners send corresponding signals). The concept of a "theory of mind" assumes that a mental model of another agent is a function of the conversation during which it is constantly updated. An agent's methods to update preferences are implemented by the type class UpdatePrefs (See Listing 7):

Listing 7: Module Agent: Update of Assumed Preferences (ToM)

```
class UpdatePrefs prefMap where
        getPrefs :: String -> prefMap -> [(String,Bool)]
        isIn :: String -> prefMap -> Bool
        switchPrefs :: String -> prefMap -> prefMap
        addPrefs :: (String, Bool) -> prefMap -> prefMap
```

The functions getPrefs and isIn are observers to check the state of the assumed preferences (stored in PrefMap). The function switchPrefs is called in case the agent assumed some preference to hold true which turned out to be false, or vice versa. The function addPrefs is called if the source agent had initially not stored some preference but during the course of the conversation the preference turned out to exist. Note, for the complete code of the discussed type classes, see Appendix A.3.

5.1.3.2 *An Agent's Actual Preferences (Perspective of the Target)*

Naturally, the preferences an agent assumes to be present in a dialog partner ("theory of mind") can be different to that partner's actual preferences. An agent's actual preferences can be modeled generically via the type class BasicPref (See Listing 8).

The type class makes use of a default implementation of the specified preference functions, all of which evaluate to True. In other words, it is assumed by default that an agent would accept any type of instruction. It is fair to assume that the producer of some instruction understands its meaning, otherwise it would not be used in the conversation. Thus, a source agent can be modeled by making the agent an instance of the type class BasicPrefs.

A target agent who can have limited capabilities, e.g., one that requires all instructions of type action to include additional landmark information, can be modeled by overwriting the default functions (See Listing 8). Note, for the complete implementation of the dicussed type classes, see Appendix A.3.

Listing 8: Module Agent: Individual Preferences

```
class BasicPrefs prog reor loc nonLoc agent | agent -> prog,
    agent -> reor, agent -> loc, agent -> nonLoc where
        prefersProg :: agent -> prog -> Bool
        prefersProg _ _ = True
        prefersReor :: agent -> reor -> Bool
        prefersReor _ _ = True
        prefersLoc :: agent -> loc -> Bool
        prefersLoc _ _ = True
        prefersNonLoc :: agent -> nonLoc -> Bool
        prefersNonLoc _ _ = True

-- an agent who is capable of processing any type of instruction
instance BasicPrefs Progression Reorientation Locating
    NonLocating (Role Source)

-- an agent who has only limited capabilities, e.g., needs a
    landmark for every instruction that is also an action
instance BasicPrefs Progression Reorientation Locating
    NonLocating (Role Target)  where
        prefersProg _ (ProgressionAtLM _ _) = True
        prefersProg _ _ = False
        prefersReor _ (ReorientationAtLM _ _ _) = True
        prefersReor _ _ = False
```

5.1.4 *Module Signals*

This module represents the formalization of signals discussed in Section 2.4.1 and subsequently extracted from the language data (See Section 4.2). The data type Signals consists of two data constructors, i.e., one for signals in Track 1 and one for signals in track 2. Signals in Track are concerned with the official business during a conversation, e.g., an assertion. Signals in track two are concerned with

Instruction Class	mentions Landmark	is an Action	is a Description
ProgressionAtLM	x	x	
ProgressionNoLM		x	
ReorientationAtLM	x	x	
ReorientationNoLM		x	
Locating	x		x
NonLocating	x		x
Comment			x

Table 4: Classes of the proposed Route Instruction Taxonomy (Overview)

meta-talk, e.g., acknowledgment. For a detailed theoretical treatment of the mentioned signals see Section 4.2.3.

Listing 9: Module Signals: Overview of the used Signals.

```
data Signals = Signals Signal1 Signal2 | EmptySignals

data Signal1 = Assert | Ratify | EndConv | StartConv| LoDChange |
     NoSignal1
data Signal2 = Acknowledge | RConfirm | Reject | NoSignal2

standardAssert = (Signals Assert  RConfirm)
standardRatify = (Signals Ratify Acknowledge)
```

The standard exchange during a negotiation process consists of an assert/request-for-confirmation signal by S and a ratify/acknowledge reply by S. For an explanation of the effect of signals during the exchange of instructions see Section 5.2. See also Appendix A.4 for the complete code of this module

5.1.5 *Module Instruction*

This module provides the formalization of the proposed taxonomy of route instructions introduced in Section 4.1.

To recapitulate, the taxonomy made a basic distinction between actions (i.e., if followed they move the agent from one decision point to the next) and descriptions (i.e., if followed they update an agent's knowledge of the world). Actions can further be divided into progressions and reorientations. The former type of instruction keeps (if executed) an agent's heading constant while the latter requires an agent to adjust their heading. Another basic distinction is between instructions that include landmark information and those that do not. For an overview of these distinctions See Table 4.

In addition, each type of instruction can be interpreted as having several levels of detail. For example, we can assume that an instruction of type ProgressionNoLM (e.g., "go straight") is more general than an instruction of type ProgressionAtLM (e.g., "go straight at

Listing 10: Module Instructions: Type Hierarchy (Excerpt).

```
class Instructions instr where ...
class (Instructions actions) => Actions actions ...
class (Actions progression) => Progressions dp dir progression
    ...
```

the building"). Furthermore, additional information can be added by combining two or more instruction types. For example an instruction of type ProgressionAtLM could also include a non-locating (attributive information) to generate instructions such as "Go straight at the red building".

5.1.5.1 *Type Hierarchy*

The taxonomy can be modeled in Haskell by creating type classes that capture the desired behavior of each instruction. Type variables (for the instruction type) are then instantiated for each particular type class. Listing 10 shows an excerpt of the type hierarchy that is part of the the formalized taxonomy (See Appendix A.5 for the complete code of this module). The behavior can be optionally constrained by introducing instance contexts from one or several super class(es). See for example Kuhn [2002], who applied this approach to modeling conceptual integration, such as combining the characteristics of the concept "house" and "boat" into the combined concept "houseboat".

Applied to the modeled instructions, the specified methods in the type class **Progressions** need to make sure they can satisfy the constraints introduced by the super class **Actions**. This class in turn, needs to make sure that it can satisfy constraints specified by the class **Instructions**. In such a way, the semantics of each instruction type can be defined in a consistent manner.

For example, if we want to provide an axiom whether a certain type of instruction requires a change in heading, we can specify this by making default implementations of a function that checks whether an instruction includes a change in an agent's heading. Thus, the type class Actions uses a default method (headingChange = False) which is overwritten by the instance for type Reorientation. Similarly, we can define the semantics on the action level assuming that the successful completion of an instruction of type action involves a location change of the agent (See Listing 11).

Listing 11: Module Instructions: Constraints imposed by Super Classes.

```
class (Instructions actions) => Actions actions where
        hasHeadingChange :: actions -> Bool
        hasHeadingChange _ = False
instance Actions Progression
instance Actions Reorientation where
        hasHeadingChange _ = True

class Instructions instr where
```

```
        updateLocation :: instr -> Bool
        updateLocation _ = False

instance Instructions Reorientation where
        updateLocation _ = True
instance Instructions Progression where
        updateLocation _ = True
```

Testing the axiom on two different types of instructions we get the desired behavior:

```
> hasHeadingChange ReorientationNoLM (N 1) R :: Reorientation
True
> hasHeadingChange ProgressionNoLM (N 2) Str :: Progression
False
```

5.1.5.2 *Data Types and Class Implementations*

The data structure for the modeled instructions consists of a data type corresponding to each class of the proposed taxonomy (See Table 4). Listing 12 illustrates the code for the action types progression and reorientation.

Listing 12: Module Instructions: Data Types for Instructions

```
data Progression = ProgressionAtLM LM Node | ProgressionNoLM Node
     | NoProg deriving (Eq, Show)
data Reorientation = ReorientationAtLM LM Node Direction |
    ReorientationNoLM Node Direction | NoReor deriving (Eq, Show)
```

Using the data constructors ProgressionAtLM, as well as ProgressionNoLM we can make a distinction between progressions that mention a landmark and one that do not. To capture the behavior of the instructions, e.g., to add constructors and specify axioms, type classes were defined. Listing 13 illustrates the abstract behavior for the class that includes instructions of type ProgressionAtLM.

The constructors makeProgressAtLM and makeProgress generate instructions of the corresponding types if they are fed a decision point (dp) and some directional information (dir). This information is based on the underlying route representation stored by the source agent (See 5.1.1). See Listing 13 for an implementation of the function makeProgressAtLM.

Listing 13: Module Instructions: Type Classes for Instructions

```
class (Actions progression) => Progressions dp dir progression |
    dp -> progression, dir -> progression where
        makeProgressAtLM :: dp -> dir -> Maybe progression
        makeProgress :: dp -> dir -> Maybe progression

instance Progressions DecisionPoint Direction Progression where
        makeProgressAtLM (DP _ NoLM) _ = Nothing
        makeProgressAtLM dp dir | dir == Str = Just $
            ProgressionAtLM (lmToSimpleLm $ getLM dp) (getNode dp
            )
```

```
                  | otherwise = Nothing
```

For the here presented computational model, the instructions were defined such that they can represent the Progression, Reorientation, Locating, and nonLocating level of above mentioned taxonomy. A higher-level classification of progression/reorientation locating/non-locating into action and description, respectively, would be possible. A distinction between actions and descriptions would be necessary, if the agents carried out a mental walk of the route, e.g., in order to distinguish between locomotion and updates on encountered landmarks. For the negotiation process, this distinction is less relevant and therefore not considered explicitly. Existing constraints, however, are still expressed in the code through type class - super class relationships (See Listing 10).

```
newtype Action = Act ActionType
data ActionType = Progression | Reorientation
newtype Description = Desc DescriptionType
data DescriptionType = Locating | NonLocating

testActions = map Act [Progression, Reorientation]
testDescription = map Desc [Locating, NonLocating]
```

For the remaining discussion, the type of an instruction is assumed to consist either of type Progression, Reorientation, Locating, or Non-Locating. This is represented by the data type Instr wrapping the other types and shown in Listing 14.

Listing 14: Module Instructions: Data Type for Instructions

```
data Instr = Pr Progression | Ro Reorientation | Lo Locating |
    NoLo NonLocating
```

5.2 MODEL OF THE NEGOTIATION PROCESS

The previous section discussed the basic architecture of each of the modules used in the computational model. This section discusses the formal processes that capture the behavior of a negotiation of meaning between two agents. The implementation that handles the conversation and thus the exchange of instructions is specified in the module Conversation (See Appendix A.6 for the full code).

This section first starts with a short pseudo-code illustrating the basic principles of the negotiation process. This is followed by a brief explanation of the architecture for the module conversation. Then, the process of how S generates instructions based on T's assumed mental model ("theory of mind") and the communicated signals is discussed. Finally, the evaluation process of both signals and messages communicated by S and T is discussed.

5.2.1 *Basic Principles*

To briefly repeat the principles introduced in previous sections, in particular, the concept of agents having a "theory of mind" (See Section 2.4.2) and using signals (See Section 2.4.1) to indicate the progress of the conversation, the following pseudo-code captures the basic nature of a negotiation process. The process requires two prerequisites. First, it is assumed that T wants to request some operations from S (e.g. instructions) that are needed to achieve a particular task (e.g., successfully navigate a route). Second, S has the necessary set of operations and agrees to communicate them.

```
S = Source, T = Target, sig = Signal, (T, f(a1,a2,...,an) = set of
    operations, S assumes T can carry out. Result of "theory of mind
    ".
```

```
1  T: request help from S // what operations do you have?
2  S: apply ToM (S,T) -> (T,f(a1,a2,a3)) // S thinks T should do a1,a2,a
       3
3  S: present i = f(a1,a2,a3) to T with sig = assert(i) // S presents
       operations
4  T: evaluate i // T checks if operations meet their demand
5  T: If (possible i) then (send sig = accept(i))
6   else (send sig = reject(i) + have not f(a'))
7          where f(a') = filter (isNotPossible) i
8  S: if (sig(T) = accept) then (continue) else
9          update ToM (S,T) -> (T,i'=f(a1,a2,a3)\f(a')) // S now knows T
               can only do (a1,a2,a3) without f(a')
10    + present i' to T with sig = assert(i') // S presents updated
          operations
11
12  ... process continues at (4): T evaluating operations presented by S
```

The procedure relies on both signals and agents having a "theory of mind". First, information (a set of possible operations) is constructed based on T's information needs which are assumed (but not known) by S. The assumption includes a model of T's assumed preferences, e.g., "I only use very specific (highly detailed) references to landmarks". In case an assumption is not met (T signals rejection), S has to update the mental model and try sending an adjusted set of operations.

In general, adjustments are made based on signals that are communicated between both agents. In this example, two basic signals are used by T to either accept or reject information presented by S. The communication ends, if T has received the requested information that meets the demands. Should this be the case, both S' and T's interpretation of the communicated information match.

5.2.2 *Architecture of the Negotiation Process*

The implemented model assumes that agents exchange messages that consist of an instruction and any signals attached. The data type that represents a message consists of a data constructor "Maybe Instr" (Note, "Maybe" is used to account for the fact that an instruction can be empty, i.e., "Nothing") and a constructor for the communicated signals.

Note, the model represents the exchanged signals from an abstract point of view, independent from a particular modality (e.g., speech). In other words, the model does not explicitly specify the modality of an exchanged signal, i.e., it just assumes that there is such a thing as a signal attached to the message. In addition, the model assumes that T only responds to the messages presented by S through the use of signals. An alternative implementation, i.e., to let the target send instructions (e.g., with an attached signals such as repeat - "Did you say m?", where m is some previously communicated message) would be possible but is not considered here. In other words, the target only sends messages that include signals but no instructions, while the source sends messages that can consist of both instructions and signals.

```
data Message = Message {instr:: Maybe Instr, signals ::Signals} |
     EmptyMessage
```

The conversation is implemented using concurrency, in particular, by using a buffered channel variable CVar (See Listing 15) that works as a container for messages produced by the source (S) and target (T) agent.

Listing 15: Module Conversation: Buffered Channel Variable.

```
type CVar a b = (MVar a, -- source message container
              MVar b) -- target signal container
-- create a new buffered channel variable
newCVar :: IO (CVar Message Signals)
newCVar = do
        m <- newEmptyMVar
    s <- newEmptyMVar
        return (m,s)
```

The exchange of messages works as follows. Both source and target put messages on the communication channel (See Listing 16). First, the source evaluates any signals that were received from T. Then, S generates an instruction (See Section 5.2.3) based on both the received signals and dynamically updated preferences (represented by S having a mental model of T, i.e., a "theory of mind"). Instructions are then communicated with an attached signal (depending on the situation) in the form of a message.

Listing 16: Module Conversation: Putting Messages on the Channel.

```
putMsg :: CVar Message Signals -> Role r -> IO (Maybe Message,
    Role r)
putMsg (sMsg, tSig) r = do
        sig <- takeMVar tSig -- evaluate signal from T
        let (msg, role) = cMessage sig r -- create message
            according to T's signal a theory of mind
        putMVar sMsg (Maybe.fromMaybe EmptyMessage msg) --
            respond with message (instruction and signal)
        return (msg, role)
```

The target then gets any messages from the communication channel (See Listing 17). First, the target consumes the attached signals and evaluates any instructions according to their individual preferences (See Section 5.2.4). Based on the outcome of the evaluation process, the target then responds with a signal (e.g., accept or reject) which is send back to S. In case the instruction is accepted the route information gets integrated with the target's stored representation of the route.

Listing 17: Module Conversation: Getting Messages from the Channel.

```
getMsg :: CVar Message Signals -> Role r ->  IO (Maybe Signals,
    Role r)
getMsg (sMsg, tSig) r = do
        msg <- takeMVar sMsg -- retrieve message from S
        let (sig, role) = eMessage msg r -- evaluate message from
             S
        putMVar tSig (Maybe.fromMaybe EmptySignals sig) --
            respond with a message (signal)
        return (sig, role)
```

5.2.3 *Instruction Generation*

The generation of instructions by S works as follows. In general, the source agent creates a list of possible instructions for the currently negotiated segment. Generated instructions are based on the assumed preferences of T ("theory of mind") and any signals that were received during the conversation.

Listing 18 illustrates the process during which instructions are generated. Depending on whether the process of following a particular route requires the agent to carry out a progression (e.g., "go straight"), or a reorientation (e.g., "turn left") to get to the next decision point, the corresponding instruction is generated. Note, in this particular example (Listing 18) only instructions of type progression are generated.

Listing 18: Module Agent: Generation of Instructions

```
class CreateInstructions agent instr where
        createInstr :: agent -> instr

instance CreateInstructions (Role r) [Maybe Instr] where
        ...
        createInstr (Source (Agent  (seg@(_,dir,_):_) _ p)) =
            ... [acts] ...
           where acts = if dir == Str then progs else reors
                                 pProgs = preferedProgs seg p
                                 progs = if null pProgs then
                                     allProgs seg else pProgs
                                 ...

class PreferredInstructionsPerSegment seg prefs instr where
        preferedProgs :: seg -> prefs -> instr
```

```
instance PreferredInstructionsPerSegment Segment PrefMap [Instr]
    where
        preferedProgs seg prefs = map Pr $ instrList seg (
            evaluateP "Progression" prefs )


instrList (dp1, dir, _) xs = applySegInfo dp1 dir $ fromMaybe []
    xs

applySegInfo :: dp-> dir-> [dp -> dir -> Maybe t] -> [t]
applySegInfo dp dir xs = map fromJust $ filter (isJust) $ map (
    $dir) $ map ($dp) xs
```

The way how instructions are generated depends on whether S has a mental model of T. In case S has no mental model (i.e., no assumed preferences of T are available), S generates all instructions that are possible for a given segment. In case S has an idea what T prefers (either assumed, or confirmed/rejected by the signaling process) only a subset of the possible instructions is generated. For example, assume at a given decision point it would be possible to generate a locating instruction (e.g., "There is a house to your left") and a progressionNoLM/progressionAtLM (e.g. "Continue" vs. "Continue at the House"). In case S does not know anything about T's preferences, S would first communicated the locating instruction and then one of the instructions of type progression (which one depends on whether S operates under the principle of economy, cf. Section 5.1.3.1). Let us assume, however, S thinks that T does not want any locating information but instead always prefers instructions of type progression that include landmarks (ProgressionAtLM). In this case the only communicated instruction would be "Continue at the house".

Note, during the first step of the generation process, instructions exist only in the form of a list of partial functions (type :: [dp -> dir -> Maybe t]). This list is generated by the function "evaluateP" that checks T's assumed preferences and returns a list of possible instructions (See Section 5.1.3.1). The actual instructions (ones that are communicated) are generated by applying the function "applySegInfo" to the list of partial functions. The function "applySegInfo" retrieves its arguments from the underlying representation of the route and thus evaluates the list of partial functions according to the present real-world features.

The process can be explained as follows. Let us assume S evaluates T's mental model and concludes that T only prefers progressions that include landmark information but does not wish to retrieve locating information. The function evaluateP therefore returns the list [makeProgressAtLM] which is of type [dp-dir->Maybe t]. Now, S applies any available route information for the currently negotiated route segment, i.e., the corresponding decision point and directional information. The result is a list that includes the actual instruction, e.g., [Just (ProgressionAtLM (SimpleLM {getID = LMID 1, getLMCat

= LMCat House}) (N 1))]. From this list the instruction is then selected and communicated to T.

Of course, the process is not only carried out for progressions (or reorientations) but also for instructions of type locating and nonLocating. In general, the result is a list of instructions S assumes to fit T's demand or (if S does not take T's mental model into account) a list of all instructions S can possibly produce. In case the list contains more than just one possible instruction, first any locating information (to introduce the landmark) is communicated, followed by possible actions. For the full code listing that refers to the generation of instructions see Appendix A.3.

Once S has come up with a list of possible instructions a message is communicated to T that includes one selected instruction with an attached signal, e.g., one that asserts the communicated instruction. In case no instruction is possible (e.g., because T had signaled preferences for a particular instruction S cannot produce) the communicated message does not contain an instruction but only a signal that tells T that their demand cannot be met.

5.2.4 *Message Evaluation*

Once a target agent receives a message produced by S, the agent uses two functions to evaluate them. Note, the function eMessage (evaluates a message) could be used by S as well (its type "Role r" is generic) in case T produces a message. This could, in principle, happen as a response to a "repeat" signal. The second function, "evalSMsg" is specific to the target and called by the generic function eMessage. It uses pattern matching to identify, and respond to the signals attached to messages communicated by S.

```
eMessage :: Message -> Role r -> (Maybe Signals, Role r)
evalSMsg :: Message -> Agent -> (Maybe Signals, Agent)
```

Listing 20 illustrates how the function "evalSMsg" works for the particular case during which S communicated a message that includes an instruction and the signal "assert". First, T evaluates whether she is capable of understanding the instruction. This depends of course on T's actual preferences.

Listing 19: Module Agent: Check of Instructions

```
class CheckInstructions agent instr | agent-> instr where
        checkInstr :: agent -> instr -> Bool

instance CheckInstructions (Role Target) (Maybe Instr) where
        checkInstr agent (Just (Pr instr)) = prefersProg agent
            instr
        ...
        checkInstr _ Nothing = False
```

In case, the instruction can be interpreted (i.e., assumed the correct type of instruction), T sends an acceptSignal to S. The evaluation whether an instruction can be interpreted by a target agent is carried out once they have received it. Using pattern matching the correct function, based on an agent's preferences, is called (See Listing 19). If the instruction cannot be interpreted, however, T decides whether the instruction is rejected entirely or a request for a change in level of detail is sent. The latter would be the case if T interprets the instruction as useful in principle, but would need more or less detail to fully accept it. In this case, a request for LoDChange is returned. The former would be the case, if T rejects the instruction because its type does not meet T's demand at all.

Listing 20: Module Conversation: Evaluation of Messages

```
evalSMsg (Message inst (Signals Assert RConfirm)) agent = if
    isCapable then (Just acceptSignal, nAgent) else rejectOrLOD
                                            where rejectOrLOD = if
     isIndeterminate (Target agent) then (Just rejectLoDSignal,
    agent) else (Just rejectSignal, agent)
                isCapable = checkInstr (Target agent) inst
                seg = instrToRouteSeg inst
                nAgent = agent{getRoute = route'}
                route' = getRoute $ unRole (integrateRS (Maybe.
                    fromJust seg) (Target agent))
```

Furthermore, in case an instruction is interpreted as meeting T's demand, T adds the information that can be extracted from the instruction to their own representation of the route. Listing 21 shows the type class that specifies the behavior needed to integrate information retrieved from an accepted instruction with an existing representation of a route. See Appendix A.3 for the implementation of the type class RouteOps.

Listing 21: Module Agent: Operations to Adjust Routes

```
class RouteOps role seg | role -> seg where
        isSegIn :: seg -> role -> Bool
        addRouteSeg :: Maybe seg -> role -> role
        replaceRouteSeg :: Maybe seg -> role -> role
        integrateRS :: seg -> role -> role
        updatePreviousAndAdd :: seg -> role -> role
```

5.2.5 *Signal Evaluation*

T responds to messages communicated by S with signals that indicate whether the information demand was met or. The source agent, in turn, uses two functions to evaluate any signals produced by T and create corresponding messages. The function cMessage creates messages according to T's assumed (or in case the conversation is already in progress, confirmed or rejected) preferences as well as T's response signals. T's signals are evaluated using function evalTSig.

```
cMessage :: Signals -> Role r -> (Maybe Message, Role r)
evalTSig :: Signals -> Agent -> (Maybe Instr, Agent)
```

Listing 22 illustrates the conversation in the case where T signaled ratification and acknowledgment of a presented instruction. Thus, in this particular case, T's behavior implies that the assumed capabilities for this given type of instruction can be confirmed.

Pattern matching is used to distinguish three cases of possible behavior.

First, in case the remaining route is empty, the function returns to its caller (cMessage) initiating the end of the conversation. Second, in case the route is not empty, and the set of possible instructions is empty (i.e., the function is called for the first time for the given segment), the function returns the first possible instruction based on the set of instructions generated from the head of the remaining route. Third, in case the list of possible instructions is not empty, the next possible instruction is returned. Note, once S has agreed on providing route information, the list of possible instructions can be not empty for two reasons. First, there are more instructions possible at a given route segment (at least one description and one action). T, in case T signaled that the previously presented instruction was inadequate, S attempts to present another type of instruction instead, thus filling the list of possible instructions.

Listing 22: Module Conversation: Evaluation of Signals

```
evalTSig (Signals Ratify Acknowledge) agent@(Agent [] _ _) = (
    Nothing, agent)
evalTSig (Signals Ratify Acknowledge)  agent@(Agent (_:rs) (Su _,
     Tr (ts), Po []) _) = (p, nAgent{getInstrList = (Su [],Tr ts
    ',Po $ ps)})                        where nRoute = rs
                nAgent = agent{getRoute = nRoute}
                nInstr = if null nRoute then [Nothing] else
                    createInstr (Source nAgent)
                p = head nInstr
                ts' = if p == Nothing then ts else ts ++ [(Maybe.
                    fromJust p)]
                ps = map Maybe.fromJust $ tail nInstr
evalTSig (Signals Ratify Acknowledge)  agent@(Agent _ (Su ss, Tr
    (_:ts), Po ((p:ps))) _) = (Just p, agent {getInstrList = (Su
    $ ss', Tr $ ts', Po ps)})         where  ss' = ss ++ ts
                ts' = ts ++ [p]
```

5.3 TESTING THE MODEL

This section illustrates the negotiation process between a source and target agent with varying parameters. Note, the code for the test cases can be found in Appendix A.8.

The example route used here is limited to four decision points (start, end, and two intermediate ones) to keep the negotiation process at a minimum without giving up generality (See Figure 13).

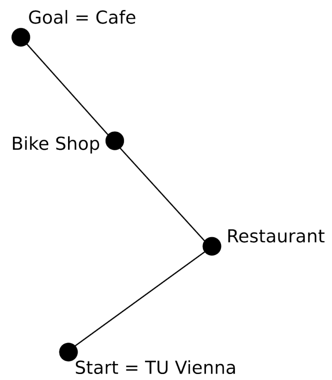

Figure 13: A sketch of the negotiated route used for the test cases (Mental model of S).

Let us first assume that S has landmark information available for each decision point along the route, and T has only minimal knowledge. In particular, T knows that she is located at the start of the route, that there is a landmark called "Tu Vienna (newEI)", and she wants to get instructions to another landmark (i.e., the goal) called "Cafe Kunsthalle (cafe)".

Note, a request for instructions starting at the same place where one is currently located is somewhat special (e.g., one could request instructions from A to B without actually being at A). This however, does not have an effect on the actual negotiation process. For the following examples, the formal representation of both routes (from the perspective of S and T) looks as follows:

```
-- the route S has selected for communication
routeS = [(DP (N 1) newEI, Str, DP (N 2) restaurant),
                                (DP (N 2) restaurant, L, DP (N 3)
                                     bikeShop),
                                (DP (N 3) bikeShop, Str, DP (N
                                    11) cafe)]
-- information T has available
partialRouteT = [(DP (N 1) newEI, EmptyDir,DP (NoNode) NoLM),
                                (DP (NoNode) NoLM, EmptyDir, DP (
                                    NoNode) cafe)]
```

5.3.1 *Case 1: T has no specific preferences or they match all assumptions made by S*

Let us assume that T would accept any type of instruction that is presented by S. In this case, the negotiation process is only an exchange of Assert/R-Confirm and Ratify/Acknowledge signals (See for instance one of the example conversations, presented in Section 4.3). Note, the same behavior can be observed, if S has a "theory of mind" (assumed preferences) that matches T's actual preferences.

To create a target agent that accepts any type of instruction, it is sufficient to make the agent an instance of the type class "BasicPrefs". As described in Section 5.1.3.2 and illustrated in Listing 8, the type class BasicPrefs has default implementations for each type of instructions that evaluate to true.

```
--create an agent that accepts any type of instruction
instance BasicPrefs Progression Reorientation Locating
    NonLocating (Role Target)
```

The negotiation process can be started by calling the function "converse" in Module Conversation (A.6). The function is of type "Role Source -> Role Target -> Signals -> IO()" and requires one source and one target agent, as well as a start signal (This is assumed to be T's request for instructions).

The actual negotiated content also depends on the nature of S having a "theory of mind". For example, if S starts the conversation without a "theory of mind" and T accepts any type of instruction, S will only present instructions that do not include landmark information. This is because the agent adheres to the principle of economy [Simon, 1998] and does not present more than is needed to navigate the route.

Listing 23: Test Case 1a: T accepts any type of instruction, S starts with an empty theory of mind

```
"T: Can you give me instructions to the Cafe Kunsthalle?"
"S: I assert Continue. Do you confirm?"
"T: I ratify and I acknowledge"
"S: I assert Turn left. Do you confirm?"
"T: I ratify and I acknowledge"
"S: I assert Continue. Do you confirm?"
"T: I ratify and I acknowledge"
"S: This is all I have! There is Cafe Kunsthalle."
"T: Ok thx!"
```

After the conversation, T has an updated representation of the route that now includes an additional segment in between, as well as directional information on how to get from one segment to the next. However, no landmark information is available apart from the start and goal:

```
-- T's updated route (no lm information)
getRoute target = [(DP {getNode = N 1, getLM = "New EI"},straight
    ,DP {getNode = N 2, getLM = NoLM}),
(DP {getNode = N 2, getLM = NoLM},left,DP {getNode = N 3, getLM =
     NoLM}),
(DP {getNode = N 3, getLM = NoLM},straight, DP {getNode = N 11,
    getLM = "Cafe Kunsthalle"})]
```

Alternatively, we can model the case where S has a "theory of mind" to start the conversation with, and it matches T's expectations. For example, let us assume S thinks that T wants landmark information to be included in instructions of type progression and reorientation. As expected, we will get the following output on the console:

Listing 24: Test Case 1b: T accepts any type of instruction, S starts with a theory of mind that matches T's expectations

```
"T: Can you give me instructions to the Cafe Museum?"
"S: I assert Continue at the New EI. Do you confirm?"
"T: I ratify and I acknowledge"
"S: I assert Turn left at the Viennese Restaurant". Do you confirm?"
"T: I ratify and I acknowledge"
"S: I assert Continue at the Bike Shop. Do you confirm?"
"T: I ratify and I acknowledge"
"S: This is all I have! There is Cafe Museum."
"T: Ok thx!"
```

In this particular case, T's resulting representation of the route equals the communicated representation that was selected by S.

```
-- T's updated route (lm information)
getRoute target = [(DP {getNode = N 1, getLM = "New EI"},straight
    ,DP {getNode = N 2, getLM = "Viennese Restaurant"}),
(DP {getNode = N 2, getLM = "Viennese Restaurant"},left,DP {
    getNode = N 3, getLM = "Bike Shop"}),
(DP {getNode = N 3, getLM = "Bike Shop"},straight,DP {getNode = N
     11, getLM = "Cafe Museum"})]
```

5.3.2 *Case 2: T's preferences do not match the assumptions made by S*

Let us now model the case where T prefers instructions that include landmark information, i.e., does not accept instructions of type "ProgressionNoLM" and "ReorientationNoLM". Let us further assume that S has applied a "theory of mind" and (wrongly) expects that T does not want landmark information to be included, regardless of the type of instruction. Thus, preferences assumed to exist in T are contrary to T's actual preferences with respect to instruction types progression and reorientation.

T's actual preferences can be adjusted by overwriting the corresponding default functions and making the target agent an instance of the type class BasicPrefs. Preferences S assumes to exist in T are represented by the function "theoryOfMind" (See Listing 25).

Listing 25: Test Case 2: Adjusting an agent's preferences

```
-- The target's acutal preferences
instance BasicPrefs Progression Reorientation Locating
    NonLocating (Role Target)  where
        prefersProg _ (ProgressionAtLM _ _) = True
        prefersProg _ _ = False
        prefersReor _ (ReorientationAtLM _ _ _) = True
        prefersReor _ _ = False

-- S assumption of T's preferences
theoryOfMind = PM [("ProgressionAtLM", False), ("ProgressionNoLM"
    , True), ("ReorientationNoLM", True), ("ReorientationAtLM",
    False)]
```

The console output of the conversation presented in Listing 26 reflects the above described assumptions. The first time, S needs to generate an instruction of type progression, S includes no landmark information (based on the initial assumption of a "theory of mind"). Once S presents this type of instruction, T rejects it and requests a LoD change instead. Then, S updates his mental model of T (S suspects that T does want landmark information to be included in instructions of type progressions). Note, this assumption is later proven correct and can be witnessed when S creates the last progression, but this time mentioning a landmark. In a similar fashion, a initially produced instruction of type "ReorientationNoLM" is rejected and triggers an updated instruction of type "ReorientationAtLM".

Once S received a request for a change in the level of detail of the previously presented instruction, S then checks if such an adjusted instruction is available. Finally, S responds with an adjusted instruction (In general only, if it exists).

Listing 26: Test Case 2: A negotiation process with diverging interpretations - Success

```
"T: Can you give me instructions to the Cafe Museum?"
"S: I assert Continue. Do you confirm?"
"T: I request a level of detail change and I cannot accept this"
"S: I assert Continue at the New EI. Do you confirm?"
"T: I ratify and I acknowledge"
"S: I assert Turn left. Do you confirm?"
"T: I request a level of detail change and I cannot accept this"
"S: I assert Turn left at the Viennese Restaurant. Do you confirm?"
"T: I ratify and I acknowledge"
"S: I assert Continue at the Bike Shop. Do you confirm?"
"T: I ratify and I acknowledge"
"S: This is all I have! There is Cafe Museum."
"T: Ok thx!"
```

The result of the negotiation process is also reflected in the representation of S having a "theory of mind":

```
-- S now has an updated theory of mind that reflects T's actual preferences
theoryofMind' = PM [("ProgressionAtLM",True),("ProgressionNoLM",False),("
    ReorientationNoLM",False),("ReorientationAtLM",True)]
```

We could also model a case that reflects T being insisting on the particular type of instruction she wishes to receive.

```
instance BasicBehavior (Role Target) where
        isInsisting _ = True
```

In case T is insisting on a particular type of instruction and only sends a reject signal (without specifying a LoD change), S first attempts to produce any other instructions available at the currently negotiated route segment. This is because if T only signals "reject", S cannot know what type of information is preferred, so it is checked what other types of information would be possible at the particular decision point. For example, S may assume that T wants a locating information ("There is ...") to precede any action (progression or reorientation) so it would be tried to provide this information instead.

Alternatively, S could end up presenting the actually preferred instruction (i.e., in the example above ProgressionAtLM), thus ending up with the same result as if T had signaled a request for a change in the level of detail.

If, however, S cycled through all possible instructions at a given decision point and any one was rejected, S requests the end of the conversation. Should this be the case, S provides a final locating instruction for the next decision point. We could imagine such an agent to be a person who concludes that she cannot provide the preferred instruction, but ends the conversation with telling the other person that there is an upcoming decision point without explicitly specifying how to get there.

5.3.3 *Case 3: S has only partial information for the requested route*

The first two test cases assumed S had landmark information for every decision point along the route. Instead, let us now assume S has no landmark information for the second decision point (See Figure 14). Furthermore, lets assume S thinks T prefers instructions of type reorientation with landmarks attached.

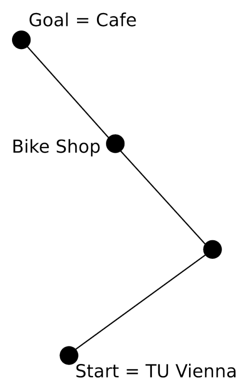

Figure 14: A sketch of the negotiated route in case S has only partial information of the route (Mental model of S).

The modified representation of the route can be represented by the following code snippet:

```
routeSPartial = [(DP (N 1) newEI, Str, DP (N 2) NoLM),
                 (DP (N 2) NoLM, L, DP (N 3) bikeShop),
                               (DP (N 3) bikeShop, Str, DP (N 11)
                                   cafe)]
theoryOfMind = PM [("ReorientationNoLM", False),
        ("ReorientationAtLM", True)]
```

If S has only partial information of the route, the success of the negotiation process depends on the preferences and behavior of T. First, in case T prefers landmark information for the given segment and is insisting on the correct type of information the conversation is ended by S because such an information cannot be provided. Second, in case T has no preferences (i.e., accepts any type of instruction) and

S does not store any assumed preferences, S produces an instruction without a landmark. This is because S has no landmark information at this particular decision point.

Thus, T will not notice that S has no landmark information available at the second decision point. Finally, if T prefers another type of instruction but is not insisting on a particular type, T will request a change in level of detail. Listing 27 illustrates this case.

Listing 27: Test Case 3: Console output of S having only partial route information

```
"T: Can you give me instructions to the Cafe Kunsthalle?"
"S: I assert Continue. Do you confirm?"
"T: I request a level of detail change and I cannot accept this"
"S: I assert Continue at the New EI. Do you confirm?"
"T: I ratify and I acknowledge"
"S: I assert Turn left. Do you confirm?"
"T: I request a level of detail change and I cannot accept this"
"S: This is all I have! There is Bike Shop"
"T: Ok thx!"
```

The first instruction is rejected because T prefers progressions that do include a landmark. Therefore, T requests a LoD change and S produces the adjusted instruction instead. Then, at the next decision point S produces an instruction of type reorientation not mentioning a landmark. This is because, S has no information for this type of instruction. As a response T requests a level of detail change but this is rejected by S in turn because there is no other information available. If T is insisting on this particular type of information (as in this case), S requests to end the conversation and mentions the next landmark ("bikeshop") T will encounter.

5.3.4 *Case 4: S has no information for the requested route*

In the trivial case where S has no route information to offer, the conversation is ended immediately.

```
routeS = EmptyRoute
-- examplary output if route is empty
"T: Can you give me instructions to ..."
"S: I have no information to provide"
"T: Ok thx!"
```

5.4 CONCLUSIONS

This chapter reported on the implementation of a computational model that formalizes the notion of meaning negotiating, including agents having a "theory of mind" and using signals to indicate the progress of the conversation. First, the basic architecture of the model was explained. This was followed by a description of the formalized processes that make up the negotiation of meaning. Finally, a number of test cases was presented to demonstrate the functionality of the model.

The test cases demonstrated that the model can replicate the behavior that was observed while analyzing the language data (See Chapter 4). In addition, different types of behavior (including the complexity and duration of the negotiation process) can be triggered by adjusting both agents' preferences.

The exchange of way-finding instructions is complete with respect to T's actual preferences, and if the conversation is successful (S has adjusted to T's preferences or assumed them in the first place), results in at least a minimal representation of the route which would allow the target agent to travel the route from start to end. Thus, the model demonstrated the capability of an instruction giving agent to adjust information presentation such that it meets the (varying) demands of an instructing receiving requesting agent.

Note, while the functionality of the model (in terms of the types of instructions that can be exchanged) was kept at a minimum, it can still demonstrate its applicability to real world situations. Various extension to the model could be made, e.g., the source could attempt to calculate an alternative route in case no information is possible at a given requested level of detail. Also, additional instruction types and their aggregations (e.g., spatial chunking [Klippel, 2003]) could be implemented. However, this would only increase complexity of the model without gaining further insights on the process of meaning negotiation.

Part IV

CONCLUSIONS

6 CONCLUSIONS AND FUTURE WORK

This chapter presents the concluding remarks and is structured as follows. First, a summary of the presented work is given, including the motivation and an explanation of the used methodology. This is followed by a discussion of the results and major findings. Finally, some directions for future research are given. This includes a discussion of the commitments and assumptions that were made for this thesis, and what can be expected if they are lifted.

6.1 SUMMARY

This thesis presented an agent-based pragmatic communication model for way-finding instructions. Pragmatics is a branch of linguistics that views communication as a context-dependent system. The goal of this work was to develop the conceptual and formal foundations for the design of next-generation information systems that can mimic human forms of interaction and present information that is tailored to a user's specific and individual needs.

This work started with a comparison between way-finding instructions provided by today's Web routing services and ones that are generated by humans. It was found that the two forms of instructions are fundamentally different from each other. First, humans produce instructions that include mostly qualitative information, e.g., in the form of landmarks. Computer-generated instructions, on the other hand, rely on quantitative (metric) information, such as references to time and distance between decision points. Numerous studies, however, have shown that for many people this type of information is difficult to process, thus introducing additional cognitive strain in an already demanding way-finding task. Second, the way today's information systems allow users to interact with the presented content seems extremely limited, if compared to human-human forms of interactions. For example, humans can adjust to their dialog partner's expectations by keeping track of a mental model of them ("theory of mind"). This implies that information is generated individually and dynamically, allowing to make changes during information presentation, in case this seems necessary. In addition, humans have the ability to signal whether some piece of presented information has been understood or not. As such, dialog partners negotiate the meaning of some information piece until both have agreed on a particular interpretation that meets both people's needs.

Based on this initial analysis it was then argued that current information systems could provide better forms of interaction with the user, if they specified and implemented pragmatic communication principles, as found in human-human communication settings. The following paragraphs give an overview on how this thesis contributed to achieving this goal.

Chapter 2 reviewed conventional theories of communication, most notably Shannon's model [1948] and Searle's speech acts [1969], with the goal to identify the shortcomings that underlie such approaches. In particular, it was found that the reviewed communication models do only consider the producer of some information explicitly, but leave out both the hearer and potential side effects that can occur during the conversation. As such they view language as a static structure (i.e., a product) not as a dynamic system. In an attempt to remedy this fact, two alternative theories, one based on the notion of signals [Clark, 1996], the other based on agent's having a "theory of mind", were presented, and suggested instead as two important principles that make up an pragmatic approach to communication. This view stresses the processes between dialog-partners that underlie communication.

Chapter 3 presented the current state-of-the-art research in the field of way-finding instructions. The literature review attempted to answer the question "What are way-finding instructions?", by discussing their semantics in a top-down manner. Formal approaches to the generation and classification, as well as models that focus on the presentation of instructions over multiple levels of detail were also discussed. This was followed by an analysis of quality related aspects both from a general and an individual (i.e., user-dependent) point of view. Finally, conventional approaches that modeled the communication of way-finding instructions were critically reviewed. It was found that they, similar to the generic communication models reviewed in Chapter 2, do not put speaker and hearer on par and, but rather focus on their static structure not treating communication processes sufficiently.

Chapter 4 presented a conceptual model that captures the components and dynamics of an exchange of way-finding instructions between two human agents. First, a taxonomy of way-finding instructions was suggested. The design choices were based on the insights gathered from the literature study on way-finding instructions conducted in Chapter 3. Then, two different language corpora that included the exchange of way-finding instructions between two humans were analyzed. The goal of this process was to extract the signals humans use to indicate the progress of the conversation. The signals were then systematically classified. Chapter 4 concluded with an analysis of the dynamics of signal use during the exchange of instructions.

Chapter 5 presented the computational model. It formalized both the notion of signals (as introduced in Chapter 2 and systematically classified in Chapter 4) and agents having a "theory of mind". The formal model was implemented in the functional programming language Haskell. In the first section, the basic architecture of each module was presented. Then, the processes that underlie the formalized communication setting were explained. Finally, several test cases were presented that illustrated the applicability of the model to real-world situations. The adjustment of an agent's preferences and general properties allows for a simulation of the negotiation process. Note, each section of this chapter provides excerpts of code listings used to support the argumentation. The complete executable code can be found in Appendix A.

It was finally concluded that the developed computational model can simulate the processes humans use to negotiate the meaning of way-finding instructions, until the interpretation of an instruction meets both dialog-partners needs. In particular, this process is achieved through the use of signals and keeping track of the other person's mental state (i.e., having a "theory of mind").

6.2 results and major findings

This section presents the results and major findings of this thesis.

First, from a theoretical perspective it was argued that the process of meaning negotiation between two human agents exchanging way-finding instructions, needs to include both signals and the notion of an agent who has a "theory of mind". Signals are devices used during communication to indicate how the conversation should progress. A "theory of mind" refers to the concept of an agent being able to attribute mental states, such as knowledge or preferences, to another person.

Previously, the two concepts were studied independently from each other. This is probably due to the fact that signals are a theory developed in Cognitive Linguistics while the notion of a "theory of mind" is a construct from Psychology. This work treated them as interrelated processes in a pragmatic communication setting.

Second, from a conceptual point of view the components and processes of meaning negotiation were identified through the analysis of dialog-based language data. This led to a systematic classification of the signals agents use to indicate the progress of the conversation. In addition, it was shown that signals and applying a "theory of mind" are interrelated processes, each having an effect onto the other. The analysis of language data further showed that signals are used during each phase of the conversation. In particular, during the initiation and the closure of the conversation, as well as the actual exchange of instructions, the negotiation of the meaning of utterances through signals and a "theory of mind" is carried out in order to secure the conversation's success.

In addition a shared understanding of way-finding instructions was proposed in the form of a light-weight taxonomy. The taxonomy unified previous classifications that were specified either too coarsely or did not consider crucial types of information (in the sense of cognitively adequate), such as landmarks.

Third, the major scientific result is the presented formal model that can simulate the process of meaning negotiation between two agents. The two previously identified mechanisms, i.e., humans having a "theory of mind" and the ability to use signals to indicate the progress of a conversation, were formally specified.

The model was implemented in the functional programming language Haskell. This allowed for automatic consistency checks of the model and executing and testing various use cases. The use cases differed in terms of the preferences and background knowledge of each agent. A modification of these parameters led to changes during the negotiation process.

Thus, the model is able to reproduce and simulate the different types of behavior that were observed during the study of the language data.

The main contribution of this thesis is thus the suggestion of formal and conceptual foundations for the design of future information systems that can mimic human forms of interaction. Ideally, the user of an information system should be able to adjust the information content, i.e., the level of detail of the presented information, to their individual and specific needs. In case the information system presents some information that does not fit the needs of the user, the user should be able to communicate the divergence between what she expects and what the system expects to fit the user. Since the system cannot be expected to have a perfect model of the user and therefore always presents the best information for any context, the solution is to allow the user to adjust the information to signal the system whether the demand was met. Although, this concept is, in principle straight-forward, it has not been found application in any navigation system, to the extend a person is used to it during the communication of other people.

To conclude, this thesis contributed to the vision of an information system that designs the information content and its level of detail according to the user's expected needs. In addition, if these expected needs are not met, the system would offer ways of providing feedback to indicate in what way the different interpretations of the presented content exist. Over time, and through the interaction between user and system, the correct amount and type of information is found.

6.3 FUTURE WORK

This thesis made some simplifying assumptions for the design and implementation of the presented pragmatic communication model. In order to answer the research questions and prove the hypothesis stated in Chapter 1.1, only the most relevant concepts and processes were formalized. This section discusses the effects that can be expected if the commitments are lifted and how this can lead to future research.

6.3.1 *Multimodal Signaling*

The formalized pragmatic communication model presented in this thesis is based on the assumption that humans use signals to indicate the (successful) progress of communication. The signals presented in this work were extracted from two dialog-based language corpora that included only the audio parts of communication. Therefore, the extracted signals were based on speech alone. In general, however conversations between humans are very rich in terms of non-verbal forms of communication, such as body language, eye gazes, and gestures (cf. [Navarro and Karlins, 2014]). This implies that signaling takes also place in these alternative communication modes.

Thus, it would be worthwhile to collect new data from conversations that exchange way-finding instructions and put, for example, special emphasis on the gestures and eye gazes used. Recent technological advance with mobile and light-weight eye-tracking devices have opened up ample research opportunities to record eye gazes outside from stationary laboratory settings. See for example, Kiefer et al. [2013] who applied eye-tracking to the process of self-localization, studying the effects of visual matching processes between symbols on a map and landmarks in the environment.

Eye-tracking could be utilized to gain evidence whether a statement is accepted or rejected or if a dialog-partner still attends the conversation. In this work, we assumed that attention on the level of eye-gazes is the bottom acceptance signal exhibited by the communication partners, to avoid infinite regression of accept and present cycles (See for example Section 4.2.4). While this seems a reasonable assumption, it could benefit from an empirical verification through experimental data.

Note, while the signals used for this work were extracted from transcripts alone (thus, only reflect speech) it seems very unlikely that lifting this limitation would make the here gained insights invalid. Evidence for this claim can be gained from the formal model that allows to simulate the negotiation process and shows that the implemented signals suffice for agents to agree on a common interpretation.

6.3.2 *Layered Signal Model*

In this thesis, the signaling process between both agents was formalized as a one-tier process. Similar to the Open Systems Interconnection model (OSI [1994]) presented in Section 2.2.3, however, it would be possible to handle the signaling between agents with a layered architecture. Each layer would then group together the different methods and functions needed to perform the negotiation process. As such each layer would be limited to a specific set of tasks and allow a clear specification of the semantics of each task. For example, considering two given layers, the first layer could secure whether a concept exists at all in both agents. If this is the case, the second layer could then negotiate the preferred form of the instruction on top of the result of the first layer.

6.3.3 *Way-finding Complexity and Equivalence of Actions*

Recent research [Giannopoulos, et al., 2014] has argued that the complexity of way-finding situations is not just a function of the environment (e.g., number of branches at a decision point), but also includes the user and the instructions offered during the task. Related to this the issue is the level of detail at which instructions need to be offered, if they are of value to a user in a given way-finding situation.

In this thesis level of detail was seen as an integral part of way-finding instructions and agents were modeled to have individual preferences that determined whether they would accept a given instruction. However, it is known that two instructions with the same level of detail can have different effects on different users, while at the same time two instructions with varying levels of detail can have the same effect on a user (cf. [Frank, 2003]). While this was no explicitly taken into account for this work it is certainly of interest for extensions to the presented model. For example, what are the implications on an action performed by the way-finder as a result of a presented instruction. What could be a (pragmatic) measure to determine when two actions can be said to be equivalent?

6.3.4 *Accounting for Uncertainty in Conversations*

During the analysis of the language data it was observed that the source (S) of information sometimes made statements that indicated uncertainty about assertions (e.g., by adding "I'm not sure"). This could have led the target (T) to believe that S is not reliable and therefore caused T to not ask for more information, as opposed to the case, where S had never used any uncertainty markers. The issue whether an assertion made by S has an uncertainty marker and whether this influences T was not considered for the model but could be interesting future research.

One of the commitments to the study of spatial knowledge transfer (in particular the computational model) was that the source had "complete" knowledge of the situation. Complete in this respect means that S can offer a (minimal) description to T such that it enables them to find the way to the destination. This claim, however, is in reality often too strong. Even participants who know the geographic area in question well, might signal some form of uncertainty. This implicit communication of uncertainty might influence the progress of the communication. For example, as a result T could (if uncertainty is suspected) request clarifications for understanding less often.

However, the commitment to not consider uncertainty has no direct effect on the presented model because the model simulates the process during which participants agree or do not agree on a particular interpretation of an instruction. This process terminates independently from the consideration of uncertainty.

Part V

APPENDIX

A

HASKELL CODE

The language pragmas neccessary to run the code can be included by adding the following line of code at the beginning of each module:

```
{-# LANGUAGE MultiParamTypeClasses, FlexibleInstances, FunctionalDependencies,
     DeriveDataTypeable, EmptyDataDecls, GADTs, UndecidableInstances, FlexibleContexts  #-}
```

A.1 MODULE ROUTE

```haskell
{-- Copyright: Paul Weiser, Oct 2014 --}
module Route where

import Graphs
import Landmarks

{--------------architecture of route representation---------------------}
-- a decision point has an associated node (from the underlying graph structure) and a landmark (optional).
data DecisionPoint = DP {getNode :: Node, getLM :: LM} deriving (Show, Eq)

-- the direction an agent has to move to get to the next decision point. Initial position and heading: n = Start, heading towards n+1 Node
data Direction = Str | L | R | EmptyDir deriving (Show, Eq)

{-- instance Show Direction where
        show Str = "Straight"
        show L = "Left"
        show R = "Right"
        show EmptyDir = "" --}
-- a segment connects to decision points with directional information

type Segment = (DecisionPoint, Direction, DecisionPoint)
-- a route consists of a list of segments
type Route = [Segment]

{-----------------------------------------------------------------------}

updateRoute :: Route -> Segment -> Route
updateRoute [] _ = []
updateRoute (r:rs) seg = updateSeg r seg : updateRoute rs seg

getDestination :: Route -> LM
getDestination [] = NoLM
getDestination route =  getLM $ getEndDP $ last route

{---------------- basic operations on decision points -------------------}
class (Eq dp) => DPOps dp where
        isNodeEqual :: dp -> dp -> Bool
        isLMEqual :: dp -> dp -> Bool
        isNoNode :: dp -> Bool
        isNoLM :: dp -> Bool
        updateDP :: dp -> dp -> dp

instance DPOps DecisionPoint where
        isNodeEqual dp1 dp2 = getNode dp1 == getNode dp2
        isLMEqual dp1 dp2 = getLM dp1 == getLM dp2
        isNoNode dp = getNode dp == NoNode
        isNoLM dp = getLM dp == NoLM
        updateDP (DP (NoNode) (lm1)) (DP (N n') lm2) = DP (N n') (updateLM lm1 lm2)
        updateDP (DP (N n) (NoLM)) (DP _ lm') = DP (N n) (lm')
        updateDP (DP NoNode NoLM) (DP n lm) = (DP n lm)
        updateDP dp _ = dp
{------------------------------------------------------------------------}
{------------ operations on segments ------------------------------------}
class SegOps seg dp dir | seg -> dir, dir -> dp where
        getStartDP, getEndDP :: seg -> dp
        getDir :: seg -> dir
        setStartDP, setEndDP :: seg -> dp -> seg
        setDir :: seg -> dir -> seg
        isSegConnected :: seg -> seg -> Bool
        isSegEqual :: seg -> seg -> Bool
        updateSeg :: seg -> seg -> seg
        updateStartDP, updateEndDP :: seg -> dp -> seg
        updateDir :: seg -> dir -> seg
        isStartSeg, isEndSeg  :: seg -> seg -> Bool
        isPartialSeg, isFullSeg :: seg -> Bool

instance SegOps Segment DecisionPoint Direction where
        getStartDP (dp1,_,_) = dp1
```

```
        getEndDP (_, _, dp2) = dp2
        getDir (_,dir,_) = dir
        setStartDP (_,dir,dp2) dp1'  = (dp1', dir,dp2)
        setEndDP (dp1, dir, _) dp2'  = (dp1,dir,dp2')
        setDir (dp1, _, dp2) dir' = (dp1,dir',dp2)
        updateStartDP (dp1, dir,dp2) dp1'  = (updateDP dp1 dp1', dir, dp2)
        updateEndDP (dp1, dir, dp2) dp2' = (dp1, dir, updateDP dp2 dp2')
        updateDir seg@(_,dir,_) dir'  = if dir == EmptyDir then setDir seg dir' else seg
        updateSeg seg1@(dp1, dir, dp2) (dp1', dir', dp2') = if (getNode dp1) == (getNode dp1
             ') then (updateDP dp1 dp1', updDir, updateDP dp2 dp2') else seg1
             where updDir = if dir == EmptyDir then dir' else dir
        isSegConnected (_, _, dp2) (dp1', _, _) = dp2 == dp1'
        isSegEqual seg1 seg2 = (getNode . getStartDP) seg1 == (getNode . getStartDP) seg2
        isStartSeg seg  (DP (startNode) startLM, EmptyDir, DP (NoNode) NoLM) = (getNode $
             getStartDP seg) == startNode || (getLM $ getStartDP seg) == startLM
        isStartSeg _ _ = False
        isEndSeg segToTest segInRoute  = ((getLM . getEndDP) segInRoute) == ((getLM .
             getStartDP) segToTest)
        isPartialSeg = not . isFullSeg
        isFullSeg (dp1, dir, dp2) = all  (\x-> x== False) $ (map isNoNode dps) ++ (map isNoLM
              dps) ++ [dir == EmptyDir]
               where dps = [dp1, dp2]

{--------End of Module Route-------------------------}
```

A.2 MODULE: LANDMARKS

```
{-- Copyright: Paul Weiser, Oct 2014 --}

module Landmarks where
import Data.Map as Map


{-------------------data structure-------------}
-- a landmark can either be simple (only id and category) or include additional attribute
      information (proper name, color, etc...)
data LM = SimpleLM {getID :: LMID, getLMCat :: LMCat} | LM {getID :: LMID, getLMCat :: LMCat,
       getAttributes :: Attributes} | NoLM

-- a landmark has a unique identifier
newtype LMID = LMID Int deriving (Show, Eq)
-- a landmark belongs to a category
newtype LMCat = LMCat Cat deriving (Show, Eq)
-- example categories data Cat = House | Restaurant | Shop | Bank | Fountain | Church | NoCat
       deriving (Show, Eq)
-- a landmark has additional attributes type Attributes = Map.Map String String

-- landmarks are equal if they share the same identifier
instance Eq LM where
        (SimpleLM id1 _) == (SimpleLM id2 _) = id1 == id2
        (LM id1 _ _) == (LM id2 _ _ ) = id1 == id2
        NoLM == NoLM = True
        (LM id1 _ _) == (SimpleLM id2 _) = id1 == id2
        (SimpleLM id1 _) == (LM id2 _ _) = id1 == id2
        _ == _ = False

instance Show LM where
        show (SimpleLM _ cat) = show cat
        show NoLM = "NoLM"
        show (LM _ cat atts ) = show $ fromMaybe ("") $ Map.lookup "name" atts
{----------------------------------------------}
{---------- basic operations on landmarks-----------}
class Landmarks i cat atts lm | lm -> i, lm -> cat, lm -> atts where
        makeSimpleLM :: i -> cat -> lm
        makeLM :: i -> cat -> atts -> lm
        lmToSimpleLm :: lm -> lm
        updateLM :: lm -> lm -> lm

instance Landmarks LMID LMCat Attributes LM where
        makeSimpleLM i cat = SimpleLM {getID = i, getLMCat = cat}
        makeLM i cat atts = LM {getID =i, getLMCat = cat, getAttributes = atts}
        lmToSimpleLm (LM i cat _) = SimpleLM i cat
        lmToSimpleLm _ = NoLM
        updateLM lm1@(LM id1 cat1 atts1) (LM id2 cat2 atts2) = if id1 == id2 && cat1 == cat2
             then lm1' else lm1
               where lm1' = if atts1 == Map.empty then (LM id1 cat1 atts2) else (LM id1 cat1
                     atts2')
                               atts2' = Map.union atts1 atts2
        updateLM lm1@(SimpleLM id1 cat1) lm2@(LM id2 cat2 _) = if id1 == id2 && cat1 == cat2
             then lm2 else lm1
        updateLM  lm2@(LM _ _ _) lm1@(SimpleLM _ _) = updateLM lm1 lm2
        updateLM sLm1@(SimpleLM _ _) (SimpleLM _ _) =sLm1
        updateLM NoLM lm = lm
        updateLM lm NoLM = lm

{--------------------------------------------------}
-- test lms
newEI, restaurant, bikeShop, house, fountain, church, coffeeshop, bank, supermarket, library,
       cafe :: LM
newEI = LM (LMID 1) (LMCat House) $ Map.fromList $ [("name", "New EI"), ("color", "green")]
```

```
restaurant = LM (LMID 2) (LMCat Restaurant) $ Map.fromList $ [("name", "Viennese Restaurant")
    ]
bikeShop = LM (LMID 3) (LMCat Shop) $ Map.fromList $ [("name", "Bike Shop")]
house = LM (LMID 4) (LMCat House) $ Map.fromList $ [("age", "old"), ("color", "white")]
fountain = LM (LMID 5) (LMCat Fountain) $ Map.fromList $ [("size", "big")]
church = LM (LMID 6) (LMCat Church) $ Map.fromList $ [("color", "yellow")]
coffeeshop = LM (LMID 7) (LMCat Shop) $ Map.fromList $ [("name", "Coffeeshop")]
bank = LM (LMID 8) (LMCat Bank) $ Map.fromList $ [("name", "Erste Bank")]
supermarket = LM (LMID 9) (LMCat Shop) $ Map.fromList $ [("name", "Supermarket")]
library = LM (LMID 10) (LMCat House) $ Map.fromList $ [("name", "Library"), ("fascade", "owl
    sculpture")]
cafe = LM (LMID 11) (LMCat Shop) $ Map.fromList $ [("name", "Cafe Museum")]
{-----------End of Module Landmarks-------------}
```

A.3 MODULE AGENT

```
{-- Copyright: Paul Weiser, Oct 2014 --}

module Agent where

import Data.List
import Data.Maybe
import Graphs

import Route
import Instruction
import Landmarks

{--------------- data structure agent ---------------}

{--
an agent has a representation for:
1. the stored route (cf. Module Route)
2. list of instructions of which he keeps track during the conversation
3. a list of preferences an agent assumes to exist in the other agent (ToM)
--}

data Agent = Agent {getRoute:: Route, getInstrList :: InstrList, getPrefMap:: PrefMap}
    deriving (Show)

-- type tags define a role for each agent (Source vs. Target)

data Source
data Target
data Target2
data Target3

data Role r where
        Source :: Agent -> Role Source
        Target :: Agent -> Role Target
        Target2 :: Agent -> Role Target2
        Target3 :: Agent -> Role Target3

instance Show (Role r) where
        show (Source agent) = show agent
        show (Target agent) = show agent
        show (Target2 agent) = show agent
        show (Target3 agent) = show agent

unRole :: Role r -> Agent
unRole (Source s) = s
unRole (Target t) = t

-- 1. imported from Module Route

-- 2. InstrList

type InstrList = (Succeed, Tried, Possible)
newtype Succeed = Su [Instr] deriving (Show)
newtype Tried = Tr [Instr] deriving (Show)
newtype Possible = Po [Instr] deriving (Show)

--generic method to apply a modifier function to an instruction list
modInstrList :: (InstrList -> InstrList) -> Agent -> Agent
modInstrList f agent = agent {getInstrList = f old}
        where old = getInstrList agent

emptyInstrList :: InstrList
emptyInstrList = (Su [], Tr [], Po [])

-- 3. PrefMap

newtype PrefMap = PM [(String, Bool)] deriving Show unPM (PM x) = x

-- generic method to apply modifier functions to a prefMap
modPrefMap :: (PrefMap -> PrefMap) -> Agent -> Agent
modPrefMap f agent = agent {getPrefMap = f old}
        where  old = getPrefMap agent

-- generic method to apply a list of modifier functions to a prefMap
modPrefMap2 :: [PrefMap -> PrefMap] -> Agent -> Agent
```

```
modPrefMap2 [] agent = agent
modPrefMap2 (f:fs) agent = modPrefMap2 fs $ agent {getPrefMap = f old}
        where old = getPrefMap agent

{-----------------------------------------------------}

{------ AGENT PREFS (Theory of Mind)-----------------}

{-- The map that contains the assumed preferences and helper functions to adjust them --}

{-- depending on the assumed preferences, instructions are produced,
as default a lazy agent is assumed (no prefs -> take simplest instr)
--}

class EvalPrefs t where
        evaluateP :: String -> PrefMap -> Maybe [(DecisionPoint -> Direction -> Maybe t)]

instance EvalPrefs Progression where
        evaluateP p prefs = if null (getAssumedPrefs p prefs) then Just [makeProgress] else
             Just $ map prefToInstr (getAssumedPrefs p prefs)

instance EvalPrefs Reorientation where
        evaluateP p prefs = if null (getAssumedPrefs p prefs) then Just [makeReorient] else
             Just $ map prefToInstr (getAssumedPrefs p prefs)

instance EvalPrefs NonLocating where
        evaluateP p prefs = if null (getAssumedPrefs p prefs) then Nothing else Just $ map
             prefToInstr (getAssumedPrefs p prefs)

instance EvalPrefs Locating where
        evaluateP p prefs = if null (getAssumedPrefs p prefs) then Nothing else Just $ map
             prefToInstr (getAssumedPrefs p prefs)

--helpers
getAssumedPrefs :: String -> PrefMap -> [(String, Bool)]
getAssumedPrefs p' prefs = filter (\(p,_) -> isInfixOf p' p) . unPM $ filterFalsePrefs prefs

filterFalsePrefs :: PrefMap -> PrefMap filterFalsePrefs (PM prefs) = PM $ filter (\(_,b) -> b
      == True) prefs

-- an agent can update the assumed preferences depending on the feedback during the signaling
      process

class UpdatePrefs prefMap where
        getPrefs :: String -> prefMap -> [(String,Bool)]
        isIn :: String -> prefMap -> Bool
        switchPrefs :: String -> prefMap -> prefMap
        addPrefs :: (String, Bool) -> prefMap -> prefMap

instance UpdatePrefs PrefMap where
        getPrefs p' prefs = filter (\(p,_) -> p'==p) . unPM $ filterFalsePrefs prefs
        switchPrefs p' (PM prefs) = PM $ map (\(p, b) -> if p==p' then (p,not b) else (p, b))
              prefs
        addPrefs pb@(p,_) pm@(PM prefs)= PM $  if isIn p pm then prefs else pb:prefs
        isIn p' (PM prefs) = elem p' (map fst prefs)

{--------------------------------------------------------------}


{----------- operations to adjust route ----------------------}

class RouteOps role seg | role -> seg where
        removeFstRouteSeg :: role -> role
        addRouteSeg :: Maybe seg -> role -> role
        replaceRouteSeg :: Maybe seg -> role -> role
        isSegIn :: seg -> role -> Bool
        integrateRS :: seg -> role -> role
        updatePreviousAndAdd :: seg -> role -> role
        integrateEnd :: seg -> role -> role

instance RouteOps (Role Source) Segment where
        addRouteSeg _ agent = agent
        removeFstRouteSeg (Source (agent)) = Source (agent {getRoute = tail $ getRoute agent
             })
        replaceRouteSeg (Just seg) agent@(Source (Agent route instr p)) = if seg `elem` route
              then (Source (Agent newRoute instr p)) else agent
               where i = fromJust $ elemIndex seg route
                      newRoute = x ++ seg : ys
instance RouteOps (Role Target) Segment where
        addRouteSeg (Just seg) (Target agent) = Target agent'
               where old = getRoute agent
                      agent' = if seg `elem` old then agent else agent {getRoute =
                          old ++ [seg]}
        isSegIn _ (Target (Agent [] _ _)) = False
        isSegIn seg@(dp1, _,  _) (Target agent@(Agent (r:rs) _  _)) = (getNode . getStartDP)
             r == getNode dp1 || isSegIn seg (Target nAgent)
                      where nAgent = agent{getRoute=rs}
        integrateRS seg (Target agent@(Agent [] _ _)) = Target (agent{getRoute = seg : []})
        integrateRS seg role@(Target agent@(Agent route _ _))
              | isSegIn seg role  = Target (agent{getRoute=updateRoute route seg})
              | otherwise = if isEndSeg seg (last route) then integrateEnd seg role else
                    updatePreviousAndAdd seg role
        updatePreviousAndAdd _ role@(Target (Agent [] _ _)) = role
        updatePreviousAndAdd seg@(dp1',_,_) (Target agent@(Agent route _ _)) =  Target nAgent
             '
```

```
                    where nAgent' = agent{getRoute =((init . init) route) ++ [butLast']++ [seg] ++ [endSeg]}
                          (dp1, dir, _) = (last . init) route
                          butLast' = (dp1,dir, dp1')
                          endSeg = last route
        integrateEnd _ role@(Target (Agent [] _ _)) = role
        integrateEnd seg (Target agent@(Agent xs _ _)) = Target agent{getRoute=(init (init xs)) ++ [endSeg']}
                where endSeg = last xs -- final partial segment of initial route
                          (dp1,dir,_) = (last . init) xs -- final partial segment of communicated route
                          lmInfo2 = (getLM . getEndDP) endSeg -- already existing lm info
                          lmInfo1 = (getLM . getStartDP) seg -- newly communicated lm info
                          lm' = updateLM lmInfo1 lmInfo2   -- updated lm
                          endSeg' = (dp1,dir,dp2')
                          dp2' = (DP (getNode $ getStartDP seg) lm') -- the new endDP
{--------------------------------------------}

{-- create instructions based on preferences of target --}

class CreateInstructions agent instr  where
        createInstr :: agent -> instr

instance CreateInstructions (Role r) [Maybe Instr] where

createInstr (Source (Agent [] _ _ )) = [Nothing]
createInstr (Source (Agent  (seg@(_,dir,_):_) _ p)) =  map Just $ locs ++ [head acts] ++ nonLocs
        where acts = if dir == Str then progs else reors
                    pProgs = preferedProgs seg p
                    progs = if null pProgs then allProgs seg else pProgs
                    pReors = preferedReors seg p
                    reors = if null pReors then allReors seg else pReors
                    locs = preferedLocs seg p
                    nonLocs = preferedNonLocs seg p

class PreferredInstructionsPerSegment seg prefs instr where
        preferedProgs :: seg -> prefs -> instr
        preferedReors :: seg -> prefs -> instr
        preferedLocs :: seg -> prefs -> instr
        preferedNonLocs :: seg -> prefs -> instr

instance PreferredInstructionsPerSegment Segment PrefMap [Instr] where
        preferedProgs seg prefs = map Pr $ instrList seg (evaluateP "Progression" prefs )
        preferedReors seg prefs = map Ro $ instrList seg (evaluateP "Reorientation" prefs)
        preferedLocs seg prefs = map Lo $ instrList seg (evaluateP "Locating" prefs)
        preferedNonLocs seg prefs = map NoLo $ instrList seg (evaluateP "NonLocating" prefs)

applySegInfo :: dp-> dir-> [dp -> dir -> Maybe t] -> [t]
applySegInfo dp dir xs = map fromJust $ filter (isJust) $ map ($dir) $ map ($dp) xs

class AllInstructionsPerSegment seg instr where
        allProgs :: seg -> instr
        allReors :: seg -> instr
        allLocs :: seg -> instr
        allNonLocs :: seg -> instr

instance AllInstructionsPerSegment Segment [Instr] where
        allProgs seg = map Pr $ instrList seg $ Just progList
        allReors seg = map Ro $ instrList seg $ Just reorList
        allLocs seg = map Lo $ instrList seg $ Just  locList
        allNonLocs seg = map NoLo $ instrList seg $ Just nonLocList

instrList (dp1, dir, _) xs = applySegInfo dp1 dir $ fromMaybe [] xs
allInstrs "Progression" = allProgs
allInstrs "Reorientation" = allReors
allInstrs "Locating" = allLocs
allInstrs "NonLocating" = allNonLocs

{-- check instructions based on agent's capabilities --}

class CheckInstructions agent instr | agent-> instr where
        checkInstr :: agent -> instr -> Bool

instance CheckInstructions (Role Target) (Maybe Instr) where
        checkInstr agent (Just (Pr instr)) = prefersProg agent instr
        checkInstr agent (Just (Ro instr)) = prefersReor agent instr
        checkInstr agent (Just (Lo instr)) = prefersLoc agent instr
        checkInstr agent (Just (NoLo instr)) = prefersNonLoc agent instr
        checkInstr _ Nothing = False

class InstrToRS instr seg | instr -> seg where
        instrToRouteSeg :: Maybe instr -> Maybe seg

instance InstrToRS Instr Segment where
        instrToRouteSeg (Just (Pr (ProgressionNoLM (N n)))) = Just (DP (N n) NoLM, Str,  DP (NoNode) NoLM)
        instrToRouteSeg (Just (Pr (ProgressionAtLM lm (N n)))) = Just (DP (N n) lm, Str,  DP (NoNode) NoLM)
        instrToRouteSeg (Just (Ro (ReorientationNoLM (N n) dir))) = Just (DP (N n) NoLM, dir, DP (NoNode) NoLM)
        instrToRouteSeg (Just (Ro (ReorientationAtLM lm (N n) dir))) = Just (DP (N n) lm, dir, DP (NoNode) NoLM)
```

```haskell
        instrToRouteSeg (Just (Lo (LocatingLM lm (N n)))) = Just (DP (N n) lm, EmptyDir, DP (NoNode) NoLM)
        instrToRouteSeg (Just (Co (ProgAtLMNonLoc lm (N n)))) = Just (DP (N n) lm, Str, DP (NoNode) NoLM)
        instrToRouteSeg (Just (Co (ReorAtLMNonLoc lm (N n) dir ))) = Just (DP (N n) lm, dir, DP (NoNode) NoLM)
        instrToRouteSeg (Just (Co (LocNonLoc lm (N n)))) = Just (DP (N n) lm, EmptyDir, DP (NoNode) NoLM)
        instrToRouteSeg _ = Nothing

class InstrToString instr where
        instrToStr :: Maybe instr -> String
        instrToStr2 :: Maybe instr -> String

instance InstrToString Instr where
        instrToStr (Just (Ro _)) = "Reorientation"
        instrToStr (Just (Pr _)) = "Progression"
        instrToStr (Just (Lo _)) = "Locating"
        instrToStr _ = ""
        instrToStr2 (Just (Ro (ReorientationAtLM _ _ _))) = "ReorientationAtLM"
        instrToStr2 (Just (Ro (ReorientationNoLM _ _ ))) = "ReorientationNoLM"
        instrToStr2 (Just (Pr (ProgressionAtLM _ _))) = "ProgressionAtLM"
        instrToStr2 (Just (Pr (ProgressionNoLM _ ))) = "ProgressionNoLM"
        instrToStr2 (Just (Lo (LocatingLM _ _))) = "Locating"
        instrToStr2 (Just (NoLo (NonLocatingLM _ _))) = "NonLocating"
        instrToStr2 _ = ""

class PrefToInstr t where
        prefToInstr :: (String, Bool) -> (DecisionPoint -> Direction -> Maybe t)

instance PrefToInstr Progression where
        prefToInstr (s@"ProgressionNoLM", _) = (makeProgress)
        prefToInstr (s@"ProgressionAtLM", _ ) = (makeProgressAtLM)

instance PrefToInstr Reorientation where
        prefToInstr (s@"ReorientationNoLM", _) = (makeReorient)
        prefToInstr (s@"ReorientationAtLM", _) = (makeReorientAtLM)

instance PrefToInstr NonLocating where
        prefToInstr (s@"NonLocating", _) = (makeNonLocating)

instance PrefToInstr Locating where
        prefToInstr (s@"Locating", _) = (makeLocating)

{----------------------------------------------------------------}

{---------- AGENT CAPABILITIES -------------------------------}
-- default implementation of basic capabilities

class BasicPrefs prog reor loc nonLoc agent | agent -> prog, agent -> reor, agent -> loc, agent -> nonLoc where
        prefersProg :: agent -> prog -> Bool
        prefersProg _ _ = True
        prefersReor :: agent -> reor -> Bool
        prefersReor _ _ = True
        prefersLoc :: agent -> loc -> Bool
        prefersLoc _ _ = True
        prefersNonLoc :: agent -> nonLoc -> Bool
        prefersNonLoc _ _ = True

-- instance for the source agent (can process any type of instruction)
instance BasicPrefs Progression Reorientation Locating NonLocating (Role Source)

-- a target agent with adjusted capabilities --> always wants to receive a instruction with a landmark
{--instance BasicPrefs Progression Reorientation Locating NonLocating (Role Target) --}

instance BasicPrefs Progression Reorientation Locating NonLocating (Role Target)  where
        prefersProg _ (ProgressionAtLM _ _) = False
        prefersProg _ _ = True
        prefersReor _ (ReorientationAtLM _ _ _) = True
        prefersReor _ _ = False

{-- another target agent with adjusted capabilities
-> does not care about the progression type
but prefers a reorientation that includes a landmark
--}

instance BasicPrefs Progression Reorientation Locating NonLocating (Role Target2) where
        prefersReor _ (ReorientationNoLM _ _) = False
        prefersReor _ _ = True

-- an agent that has the same capabilities as the source
instance BasicPrefs Progression Reorientation Locating NonLocating (Role Target3)

{-----------------------------------------------------------------}
class ExtendedPrefs combtype agent | agent -> combtype where
        prefersComb :: agent -> combtype -> Bool
        prefersComb _ _ = True

instance ExtendedPrefs CombType (Role Source)
instance ExtendedPrefs CombType (Role Target) where
        prefersComb _ (ProgAtLMNonLoc _ _)  = True
        prefersComb _ (ReorAtLMNonLoc _ _ _) = False
        prefersComb _ (LocNonLoc _ _) = True
```

```
class BasicBehavior agent where
        isLazy, isHelpful :: agent -> Bool -- agent produces lest detailed instr
        isHelpful = not . isLazy
        isInsisting, isIndeterminate :: agent -> Bool -- agent wants most detailed instr
        isIndeterminate = not . isInsisting

instance BasicBehavior (Role Source) where
        isLazy _ = True
instance BasicBehavior (Role Target) where
        isInsisting _ = False
```

A.4 MODULE SIGNALS

```
{-- Copyright: Paul Weiser, Oct 2014 --}

module Signal where

data Signals = Signals Signal1 Signal2 | EmptySignals deriving (Show, Eq)

-- Signal1 is in track1 and concerns official business data
Signal1 = Assert | Ratify | EndConv | StartConv | NoSignal1 deriving (Show, Eq)
-- Signal2 is in track2 and concerns meta-talk data
Signal2 = Acknowledge | RConfirm | Reject | LoDChange | NoSignal2 deriving (Show, Eq)

instance Show Signal1 where
        show Assert = "I assert"
        show Ratify = "I ratify"
        show StartConversation = "Can you give me instructions to the "
        show EndConversation = "This is all I have!"
        show NoSignal1 = ""

instance Show Signal2 where
        show Acknowledge = "I acknowledge"
        show RConfirm = "Do you confirm?"
        show Reject = "I cannot accept this"
        show LoDChange = "I request a level of detail change"
        show NoSignal2 = ""

-- test signals
emptySignal, standardSignal, acceptSignal, rejectSignal, rejectLoDSignal, endSignal,
      startSignal :: Signals
emptySignal = Signals NoSignal1 NoSignal2
standardSignal = (Signals Assert  RConfirm)  -- asserts instr and asks for confirmation
acceptSignal = (Signals Ratify Acknowledge) -- confirms signal 2 and ratifies instr
rejectSignal = (Signals NoSignal1 Reject)   -- reject instr
rejectLoDSignal = (Signals  NoSignal1 LoDChange) -- reject current LoD of instr but not the
      instr itself
endSignal = (Signals EndConv NoSignal2) -- asks for termination of conversation
startSignal = (Signals StartConv NoSignal2) -- asks for the route
```

A.5 MODULE INSTRUCTION

```
{-- Copyright: Paul Weiser, Oct 2014 --}

module Instruction where

import Data.Char
import Data.Maybe

import Graphs
import Route
import Landmarks

{-- data types --}
data Instr = Pr Progression | Ro Reorientation | Lo Locating | NoLo NonLocating |  Co
      CombType | EmptyInstr deriving (Show, Eq)

{-- actions --}
data Progression = ProgressionAtLM LM Node | ProgressionNoLM Node | NoProg deriving (Show, Eq
      )
data Reorientation = ReorientationAtLM LM Node Direction | ReorientationNoLM Node Direction |
       NoReor deriving (Eq, Show)
{-- descriptions --}
data Locating = LocatingLM LM Node | NoLoc deriving (Eq, Show)
data NonLocating = NonLocatingLM Attributes Node |NoNonLoc deriving (Eq, Show)
type Comment = String

-- composite instructions (Progression/Reorientation at LM and NonLocating Attributes,
      Locating a LM and NonLocating Attributes)
data CombType = ProgAtLMNonLoc LM Node | ReorAtLMNonLoc LM Node Direction | LocNonLoc LM Node
       deriving (Eq, Show)

{-- type hierarchy --}

```

```
class Instructions instr where
        updateLocation :: instr -> Bool
        updateLocation _ = False
        updateLMKnowledge :: instr -> Bool
        updateLMKnowledge _ = False
        isLoc, isNoLoc, isProg, isReor :: instr -> Bool
        isLoc _ = False
        isNoLoc _ = False
        isProg _ = False
        isReor _ = False
        getLoc, getNoLoc, getProg, getReor :: [instr] -> [instr]
        removeLoc, removeNoLoc, removeProg, removeReor :: [instr] -> [instr]

instance Instructions Instr where
        getLoc = filter ( isLoc)
        getNoLoc = filter (isNoLoc)
        getProg = filter (isProg)
        getReor = filter (isReor)
        removeLoc = filter (not . isLoc)
        removeNoLoc = filter (not .isNoLoc)
        removeProg = filter (not . isProg)
        removeReor = filter (not . isReor)

instance Instructions Progression where
        isProg _ = True
        updateLocation _ = True

instance Instructions Reorientation where
        isReor _ = True
        updateLocation _ = True
instance Instructions Locating where
        isLoc _ = True
        updateLMKnowledge _ = True

instance Instructions NonLocating where
        isNoLoc _ = True
        updateLMKnowledge _ = True

class (Instructions actions) => Actions actions where
        hasHeadingChange :: actions -> Bool
        hasHeadingChange _ = False

class (Instructions descs) => Descriptions descs

instance Actions Progression
instance Actions Reorientation where
        hasHeadingChange _ = True

instance Descriptions Locating
instance Descriptions NonLocating


{--------- operations on progressions -----------------------}
class (Actions progression) => Progressions dp dir progression | dir->progression, dp -> progression   where
        makeProgressAtLM :: dp -> dir -> Maybe progression
        makeProgress :: dp -> dir -> Maybe progression
        isProgAtLM, isProgNoLM :: progression -> Bool

instance Progressions DecisionPoint Direction (Progression) where
        makeProgressAtLM (DP _ NoLM) _ = Nothing
        makeProgressAtLM dp dir | dir == Str = Just $ ProgressionAtLM (lmToSimpleLm $ getLM dp) (getNode dp)
                                | otherwise = Nothing
        makeProgress dp dir | dir == Str = Just $ ProgressionNoLM (getNode dp)
                            | otherwise = Nothing
        isProgAtLM (ProgressionAtLM _ _) = True
        isProgAtLM _ = False
        isProgNoLM (ProgressionNoLM _) = True
        isProgNoLM _ = False

{-- instance Show Progression where
        show (ProgressionAtLM lm _) = "Continue at the " ++ show lm  --}

{----------------------------------------------------------------}
{----- operations on reorientations -------------------------}

class (Actions reorientation) => Reorientations dp dir reorientation | dp -> reorientation, dir -> reorientation where
        makeReorientAtLM :: dp -> dir -> Maybe reorientation
        makeReorient :: dp -> dir -> Maybe reorientation
        isReorAtLM, isReorNoLM :: reorientation -> Bool

instance Reorientations DecisionPoint Direction Reorientation where
        makeReorientAtLM (DP _ NoLM) _ = Nothing
        makeReorientAtLM dp@(DP _ lm) dir | dir == Str = Nothing
                    | otherwise = Just $ ReorientationAtLM lm (getNode dp) dir
        makeReorient dp dir | dir == Str = Nothing
                    | otherwise = Just $ ReorientationNoLM (getNode dp) dir
        isReorAtLM (ReorientationAtLM _ _ _) = True
        isReorAtLM _ = False
        isReorNoLM (ReorientationNoLM _ _) = True
        isReorNoLM _ = False
```

```
instance Show Reorientation where
        show (ReorientationAtLM lm _ dir) = "Turn " ++ show dir ++ " at the " ++ show lm
        show (ReorientationNoLM _ dir) = "Turn " ++ show dir
{----------------------------------------------------------------}


{------------operations on Locatings---------------------------}
class (Descriptions locating) => Locatings dp dir locating | dp -> locating, dir -> locating
      where
        makeLocating :: dp -> dir-> Maybe locating

instance Show Locating where
        show (LocatingLM lm n) = "There is " ++  show lm

instance Locatings DecisionPoint Direction Locating where
        makeLocating (DP _ NoLM) _ = Nothing
        makeLocating dp _ =  Just  $ LocatingLM (getLM dp) (getNode dp)

{---------------------------------------------------------------}


{--------operations on nonLocatings---------------------------}

class (Descriptions nonLocating) => NonLocatings dp dir nonLocating where
        makeNonLocating :: dp -> dir-> Maybe nonLocating

instance NonLocatings DecisionPoint Direction NonLocating where
        makeNonLocating (DP _ NoLM) _ = Nothing
        makeNonLocating dp@(DP _ lm) _ = Just $ NonLocatingLM (getAttributes lm) (getNode dp)


              {---------------------------------------------------------------}

class  CombInstrs dp dir combInstr | dp -> combInstr where
        makeProgAtLMNonLoc :: dp -> dir -> Maybe combInstr
        makeReorAtLMNonLoc :: dp -> dir -> Maybe combInstr
        makeLocNonLoc :: dp -> dir -> Maybe combInstr

instance CombInstrs DecisionPoint Direction CombType where
        makeProgAtLMNonLoc (DP _ NoLM) _ = Nothing
        makeProgAtLMNonLoc dp@(DP _ lm) _ = Just $ ProgAtLMNonLoc (makeLM (getID lm) (
              getLMCat lm) (getAttributes lm)) (getNode dp)
        makeReorAtLMNonLoc (DP _ NoLM) _ = Nothing
        makeReorAtLMNonLoc  dp@(DP _ lm) dir = Just $  ReorAtLMNonLoc (makeLM (getID lm) (
              getLMCat lm) (getAttributes lm)) (getNode dp) dir
        makeLocNonLoc (DP _ NoLM) _ = Nothing
        makeLocNonLoc dp@(DP _ lm) dir = Just $ LocNonLoc (makeLM (getID lm) (getLMCat lm) (
              getAttributes lm)) (getNode dp)

----- lists with all possible functions to generate instructions --}
progList :: [(DecisionPoint -> Direction -> Maybe Progression)]
progList =  [makeProgress, makeProgressAtLM]
reorList :: [(DecisionPoint -> Direction -> Maybe Reorientation)]
reorList = [makeReorient, makeReorientAtLM]
locList :: [(DecisionPoint -> Direction -> Maybe Locating)]
locList = [makeLocating]
nonLocList :: [(DecisionPoint -> Direction -> Maybe NonLocating)]
nonLocList = [makeNonLocating]
{------------End of Modul Instruction-------------------------}
```

A.6 MODULE CONVERSATION

```
{-- Copyright: Paul Weiser, Oct 2014 --}

module Conversation where

import Control.Concurrent
import Safe (headMay, lastMay, tailMay)
import qualified Data.Maybe as Maybe
import qualified Data.List as List

import Signal
import Agent
import Route
import Instruction

-- a message consists of the instructions and the signals attached

data Message = Message {instr :: Maybe Instr, signals ::Signals} | EmptyMessage deriving (Show
      , Eq)
type CVar a b = (MVar a, -- source message container
                              MVar b) -- target signal container

-- create a new buffered channel variable

newCVar :: IO (CVar Message Signals) newCVar = do
          m <- newEmptyMVar
          s <- newEmptyMVar
          return (m,s)
```

```haskell

-- The source produces messages according to T's assumed/confirmed prefs and T's signals
putMsg :: CVar Message Signals -> Role r -> IO (Maybe Message, Role r)
putMsg (sMsg, tSig) r = do
        sig <- takeMVar tSig
        let track1_sig = track1 sig
        let track2_sig = track2 sig
        case track1_sig of
                StartConversation -> do
                        let destination = getDestination $ getRoute $ unRole r
                        print $ "T: " ++ show track1_sig ++ show destination
                _ -> do
                        if track1_sig == NoSignal1 then print $ "T: " ++ show track2_sig else print $ "T: " ++ show track1_sig ++ " and " ++ show track2_sig
        -- create Message based on Target preferences and signals
        let (msg, role) = cMessage sig r
        putMVar sMsg (Maybe.fromMaybe EmptyMessage msg) -- put only if there is a signal
        return (msg, role)

-- The target consumes messages and produces signals whether the message matches preferences

getMsg :: CVar Message Signals -> Role r -> IO (Maybe Signals, Role r) --Message
getMsg (sMsg, tSig) r = do
        -- get msg, then evaluate and produce according signal
        msg <- takeMVar sMsg
        let instr = Maybe.fromMaybe (EmptyInstr) $ getInstr msg
        let sig = getSignals msg
        print $ ("S: " ++ (show $ track1 sig) ++ " " ++ show instr ++ ". " ++ (show $ track2 sig))
        let (sig, role) = eMessage msg r
        putMVar tSig (Maybe.fromMaybe EmptySignals sig)
        return (sig, role)

--The agents converse by initiating the conversation and then negotiating the route
converse :: Role Source -> Role Target -> Signals -> IO ()
converse src trg startSig = do
        c <- newCVar
        startConversation c startSig
        negotiate c src trg >>= print
        return ()

startConversation :: CVar Message Signals -> Signals -> IO ()
startConversation (_, tSig) = putMVar tSig

negotiate :: CVar Message Signals -> Role Source -> Role Target -> IO (Role Source, Role Target)
negotiate c src trg = do
        (msg, Source nSrc) <- putMsg c src
        --let route = getRoute nSrc
        -- print $ "this is the route " ++ show route
        let sSig = signals (Maybe.fromJust msg)
        --print $ "S signals" ++ show sSig
        --print $ "S has instrList " ++ (show $ getInstrList nSrc
        case sSig of
                Signals EndConv NoSignal2 -> do
                        (sig, nTrg) <- getMsg c trg
                        print $ "T signals " ++ show sig
                        return (Source nSrc, nTrg)
                _ -> do
                        (_, nTrg) <- getMsg c trg
                        negotiate c (Source nSrc) nTrg

-- create message according to T's preferences and signals
-- route is empty, reached end or no instr. possible -> send endConv Signal

cMessage :: Signals -> Role r -> (Maybe Message, Role r)
cMessage _ role@(Source (Agent [] _ _)) = (Just $ Message Nothing endSignal, role)
cMessage sig (Source agent@(Agent (r:_) _ _))  = if i == Nothing then (msg' endSignal, Source a) else (msg standardSignal, Source a)
                where (i,a) = evalTSig sig agent
                                msg sig' = Just Message {instr = i, signals = sig'}
                                msg' sig' = Just Message {instr = endLoc, signals = sig'}
                                endLoc = Just $ Lo $ Maybe.fromJust $ makeLocating (getEndDP r) (getDir r)
-- T evaluates messages according to their preferences (is T capable of understanding?)
eMessage :: Message -> Role r -> (Maybe Signals, Role r)
eMessage msg (Target agent@(Agent _ _ _)) = (s, Target a)
                where (s, a) = evalSMsg msg agent

evalSMsg :: Message -> Agent -> (Maybe Signals, Agent)
evalSMsg (Message (Nothing) (Signals EndConv NoSignal2)) agent = (Just $ Signals NoSignal1 Acknowledge, agent)
evalSMsg (Message inst (Signals EndConv NoSignal2)) agent = (Just $ Signals NoSignal1 Acknowledge, nAgent)
        where nAgent = agent{getRoute = route'}
                                seg = instrToRouteSeg inst
                                route' = getRoute $ unRole ( integrateRS (Maybe.fromJust seg) (Target agent))
evalSMsg (Message inst (Signals Assert RConfirm)) agent = if isCapable then (Just acceptSignal, nAgent) else rejectOrLOD
        where rejectOrLOD = if isIndeterminate (Target agent) then (Just rejectLoDSignal, agent) else (Just rejectSignal, agent)
        isCapable = checkInstr (Target agent) inst
                                seg = instrToRouteSeg inst
```

```
                                  nAgent = agent{getRoute = route'}
                                  route' = getRoute $ unRole (integrateRS (Maybe.fromJust seg) (Target agent))
-- S evaluates T's signals
evalTSig :: Signals -> Agent -> (Maybe Instr, Agent)
evalTSig (Signals StartConv NoSignal2) agent   = (Just p, agent {getInstrList = (Su [], Tr [p], Po (ps))})
        where (p:ps) = map Maybe.fromJust $ createInstr (Source agent)
evalTSig (Signals Ratify Acknowledge) agent@(Agent [] _ _) = (Nothing, agent)
evalTSig (Signals Ratify Acknowledge)  agent@(Agent (_:rs) (Su _, Tr (ts), Po []) _) = (p, nAgent{getInstrList = (Su [],Tr ts',Po $ ps)})
        where
                -- create new instructions for next route segment
                nRoute = rs
                nAgent = agent{getRoute = nRoute}
                nInstr = if null nRoute then [Nothing] else createInstr (Source nAgent)
                -- next instr
                p = head nInstr
                ts' = if p == Nothing then ts else ts ++ [(Maybe.fromJust p)]
                -- possible instrs = rest of new instructions for next segment
                ps = map Maybe.fromJust $ tail nInstr
evalTSig (Signals Ratify Acknowledge)  agent@(Agent _ (Su ss, Tr tried@(_:ts), Po ((p:ps))) _) = (p', agent {getInstrList = (Su $ ss', Tr $ ts', Po ps)})
        where  ss' = ss ++ ts
                        ts' = ts ++ [p]
                        p' = if elem p tried then Nothing else Just p

evalTSig (Signals NoSignal1 Reject) agent@(Agent (r:_) (Su ss, Tr ts, Po []) _) = (p, nAgent {getInstrList = (Su ss, Tr $ ts', Po ps')})
        where p = if triedLoc then Nothing else headMay $ allLocs r
                        ts' = if p == Nothing then ts else ts ++ [Maybe.fromJust p]
                        triedLoc = (any (\x -> x == True) $ map isLoc ts)
                        triedAct = (any (\x -> x == True) $ map (\x -> isProg x || isReor x) ts)
                        advance =  (any (\x -> x == True) $ map (\x -> isProg x || isReor x) ss)
                        ps' = if advance || triedAct then [] else  ts
                        nAgent =  (modPrefMap $ switchPrefs . instrToStr2 $ lastMay ts) agent

evalTSig  (Signals NoSignal1 Reject) agent@(Agent _ (Su ss, Tr ts, Po (ps)) _) = (p, nAgent{getInstrList = (Su ss, Tr $ ts', Po ps')})
        where newTPrefs = modPrefMap $ switchPrefs . instrToStr2 $ lastMay ts
                        nAgent = newTPrefs agent
                        ps' = (List.\\) ps ts
                        p = if null ps' then Nothing else Just $ head ps'
                        ts' = if (Maybe.isNothing p) then ts else ts ++ [Maybe.fromJust p]

evalTSig (Signals LoDChange Reject) agent@(Agent (r:_) (ss, Tr ts, Po (ps)) _) = (p, nAgent {getInstrList = (ss, Tr $ ts', Po $ ps)})
        where p = if null allPoss' then Nothing else head allPoss'
        -- create all possible alternatives based on last tried instr
        allPoss = map (Just$) $ allInstrs (instrToStr $ lastMay ts) r
        -- remove the one that was tried before
        allPoss' = filter (/= lastMay ts) allPoss
        --update tried
        ts' = if p == Nothing then ts else ts ++ [Maybe.fromJust p]
        nAgent = modPrefMap2 [switch, add] agent
        -- switch pref of last tried instr
        switch = switchPrefs . instrToStr2 $ lastMay ts
        -- if pref of new instr exists update else add
        add = if t then (switchPrefs $ instrToStr2 p) else addPrefs (instrToStr2 $ p, True)
        t = isIn (instrToStr2 $ p) (getPrefMap agent )
evalTSig _ agent@(Agent _ (Su [], Tr [], Po []) _ ) = (Nothing, agent)
{----------end of Module Conversation-----------}
```

A.7 MODULE: GRAPHS

```
{-- code of Module Graphs was adapted from Andrew U. Frank --}
module Graphs where
import Data.List

class Nodes n where
        zeroNode :: n
class Edges e n where
        isAB :: n -> n -> e n -> Bool
        costE :: e n  -> Cost
        makeE :: n -> n -> Cost -> e n
        nodesE :: e n -> [n]
class Graphs g e n where
        insertG :: e n -> g e n -> g e n
        nodes :: g e n -> [n]
        cost :: n -> n -> g e n -> Cost
------------------------------------------------------------------------
data data G e n = G [ e n]
data E n = E n n Cost deriving (Show)
data DE n = DE n n Cost deriving (Show)
data EDE n = ED n n Cost | EUD n n Cost deriving (Show) -- directed and undirected edge in one graph

```

```haskell
instance Eq (E Node) where
        (E n m c) == (E u v w) = n==u && m==v || n==v && m==u -- cost is not relevant for equality
instance Eq (DE Node) where
        (DE n m c) == (DE u v w) = n==u && m==v -- must be same direction, cost is not tested
instance Eq (EDE Node) where
        (EUD n m c) == (EUD u v w) = n==u && m==v || n==v && m==u
        (ED n m c) == (ED u v w) = n==u && m==v     _ == _ = False
-----------------------------------------------------------------------instances
instance (Eq n) => Edges E n where
    isAB n m (E a b c) = (a==n && b==m) || (b==n && a==m) --    makeE n m c = E n m c
    costE (E a b c) = c      nodesE (E a b c) = [a,b]

instance (Eq n) => Edges DE n where
isAB n m (DE a b c) = (a==n && b==m)      -- only one direction --    makeE n m c = DE n m c
costE (DE a b c) = c      nodesE (DE a b c) = [a,b]

instance (Eq n) => Edges EDE n where
    isAB n m (EUD a b c) = (a==n && b==m) || (b==n && a==m)
    isAB n m (ED a b c) = (a==n && b==m)      -- only one direction
    costE (ED a b c) = c
    costE (EUD a b c) = c
    nodesE (ED a b c) = [a,b]
    nodesE (EUD a b c) = [a,b]

instance (Eq n, Edges e n) => Graphs G e n where
        insertG e (G es) = G (e:es)
        nodes (G es) = nub (concat [ nodesE e | e <- es])
        cost n m (G es) = cost' n m es
                where cost' n m (e:es) = if isAB n m e then costE e else cost' n m es
                                                      cost' n m [] = maxCost
----------------------------------------------------------------------
maxCost = Cost (999999.9 ::Float)
data Cost  = Cost Float deriving (Eq, Ord, Show)
data Node = N Int | NoNode deriving (Eq, Ord, Show)
```

A.8 MODULE: TESTCASES

```haskell
{-- Copyright: Paul Weiser, Oct 2014 --}

module TestCases where

import Conversation
import Agent
import Route
import Landmarks
import Signal
import Graphs

--information S has available (complete)
routeS :: Route routeS = [(DP (N 1) newEI, Str, DP (N 2) restaurant),
        (DP (N 2) restaurant, L, DP (N 3) bikeShop),
        (DP (N 3) bikeShop, Str, DP (N 11) cafe)]

-- information T has available
partialRouteT :: Route
partialRouteT = [(DP (N 1) newEI, EmptyDir,DP (NoNode) NoLM),
        (DP (NoNode) NoLM, EmptyDir, DP (NoNode) cafe)]
{-----------------------------------------------------}
--test case 1a (target agent accepts any type of instruction and S starts with an empty "theory of mind")

agentSource1a :: Role Source
agentSource1a = Source $ Agent routeS emptyInstrList emptyPrefMap

-- assumed preferences of T
theoryOfMindS :: PrefMap
theoryOfMindS = PM [("ProgressionAtLM", True),
        ("ProgressionNoLM", True),
        ("ReorientationAtLM", True),
        ("ReorientationNoLM", True),
("Locating", False)]

-- test case 1b (target agent accepts any type of instruction and S starts with a matching "theory of mind")

agentSource1b :: Role Source
agentSource1b = Source $ Agent routeS emptyInstrList theoryOfMindS
agentTarget1 :: Role Target agentTarget1 = Target $ Agent partialRouteT emptyInstrList $ PM []

--actual preferences of T (set in module Agent)
--instance BasicPrefs Progression Reorientation Locating NonLocating (Role Target)
{-----------------------------------------------------------}

{------------------------------------------------------------}
--test case 2 (target agent always want landmark information, source does think no landmark information is prefered)

agentSource2 :: Role Source
```

```
agentSource2 = Source $ Agent routeS emptyInstrList theoryOfMindS2 -- (PM [])
theoryOfMindS2 = PM [("ProgressionAtLM", False),
        ("ProgressionNoLM", True),
        ("ReorientationNoLM", True),
        ("ReorientationAtLM", False)]

--actual preferences of T (set in module Agent)
{-- instance BasicPrefs Progression Reorientation Locating NonLocating (Role Target)  where
        prefersProg _ (ProgressionAtLM _ _) = True
        prefersProg _ _ = False
        prefersReor _ (ReorientationAtLM _ _ _) = True
        prefersReor _ _ = False --}
{-------------------------------------------------------------}

{---------------------------------------------------------------------------------}
--test case 3 (S has only a partial route)
routeSPartial :: Route routeSPartial = [(DP (N 1) newEI, Str, DP (N 2) NoLM),
        (DP (N 2) NoLM, L, DP (N 3) bikeShop),
        (DP (N 3) bikeShop, Str, DP (N 11) cafe)]

agentSource3 = Source $ Agent routeSPartial emptyInstrList emptyPrefMap
{---------------------------------------------------------------------------------}
{---------------------------------------------------------------------------------}
-- test case 4 (S has no route - trivial case)
routeSEmpty :: Route
routeSEmpty = [] agentSource4 = Source $ Agent routeSEmpty emptyInstrList emptyPrefMap
{---------------------------------------------------------------------------------}
testCase1aConv = converse agentSource1a agentTarget1 startSignal
testCase1bConv = converse agentSource1b agentTarget1 startSignal
testCase2Conv = converse agentSource2 agentTarget1 startSignal
testCase3Conv = converse agentSource3 agentTarget1 startSignal
testCase4Conv = converse agentSource4 agentTarget1 startSignal
```

B
LANGUAGE DATA

The following pages present the transcripts from the Klein dataset [1979]. Tags within the text refer to the transcripts in this appendix via the following schema:

```
DESTINATION TRANSCRIPT_NO. AGENT_ROLE UTTERANCE_NO.
```

There are two destinations, i.e., O = Opera and G = Goethehaus. The transcripts are numbered in ascending order. An agent can either have the role of a source (S = instruction giving) or a target (T = instruction receiving). Utterances are also numbered in ascending order.

For example, the tag O4 refers to the whole of the fourth transcript that was collected to the destination opera. The tag G3T1 refers to the first utterance by the target made during the third conversation with destination Goethehaus. In a similar fashion, tag G5S4 refers to the fourth utterance by the source made during the fifth conversation that exchanged instructions to the Goethehaus.

BIBLIOGRAPHY

[1] Gary L. Allen. From knowledge to words to wayfinding: Issues in the production and comprehension of route directions. In Stephen C. Hirtle and Andrew U. Frank, editors, *Spatial Information Theory A Theoretical Basis for GIS*, volume 1329 of *Lecture Notes in Computer Science*, pages 363–372. Springer Berlin Heidelberg, 1997. ISBN 978-3-540-63623-6. doi: 10.1007/3-540-63623-4_61. URL http://dx.doi.org/10.1007/3-540-63623-4_61. (Cited on pages 49, 60, and 61.)

[2] Gary L. Allen. Principles and practices for communicating route knowledge. *Applied Cognitive Psychology*, 14 (4):333–359, 2000. doi: 10.1002/1099-0720(200007/08)14:4\%3C333::AID-ACP655\%3E3.0.CO;2-C. URL http://dx.doi.org/10.1002/1099-0720(200007/08)14:4%3C333::AID-ACP655%3E3.0.CO;2-C. (Cited on pages 49, 57, and 58.)

[3] John L. Austin. *How to do things with words - The William James Lectures delivered in Harvard University in 1955*. 1955. (Cited on pages 34, 35, and 37.)

[4] Jodie A. Baird and Dare A. Baldwin. *Intentions and intentionality: Foundations of social cognition*, chapter Making sense of human behavior: Action parsing and intentional inference, pages 193–206. MIT Press, 2001. (Cited on page 40.)

[5] Alan Both, Werner Kuhn, and Matt Duckham. Spatiotemporal braitenberg vehicles. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'13, pages 74–83, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2521-9. doi: 10.1145/2525314.2525344. URL http://doi.acm.org/10.1145/2525314.2525344. (Cited on page 29.)

[6] Valentino Braitenberg. *Vehicles - Experiments in Synthetic Psychology*. MIT Press, 1986. (Cited on page 29.)

[7] Susan E. Brennan, Alexia Galati, and Anna K. Kuhlen. *Two Minds, One Dialog - Coordinating Speaking and Understanding*, volume 53, pages 301–344. Elsevier, 2010. ISBN 9780123809063. doi: 10.1016/s0079-7421(10)53008-1. URL http://dx.doi.org/10.1016/s0079-7421(10)53008-1. (Cited on pages 23, 28, and 34.)

[8] Rodney A. Brooks. Intelligence without representation. *Artificial intelligence*, 47(1):139–159, 1991. (Cited on pages 29, 30, and 42.)

[9] Rodney A. Brooks. Intelligence without reason. *The artificial life route to artificial intelligence: Building embodied, situated agents*, pages 25–81, 1995. (Cited on pages 29 and 30.)

[10] David Brosset, Christophe Claramunt, and Eric Saux. A location and action-based model for route descriptions. In *Proceedings of the 2Nd International Conference on GeoSpatial Semantics*, GeoS'07, pages 146–159, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-76875-0, 978-3-540-76875-3. URL http://dl.acm.org/citation.cfm?id=1778502.1778516. (Cited on page 53.)

[11] David Brosset, Christophe Claramunt, and Eric Saux. Wayfinding in natural and urban environments: A comparative study. *Cartographica*, 43(1):21–30, 2008. (Cited on page 59.)

[12] Gary Burnett. "turn right at the traffic light" : The requirement for landmarks in vehicle navigation systems. *The Journal of Navigation*, 53:499–510, 9 2000. ISSN 1469-7785. URL http://journals.cambridge.org/article_S0373463300001028. (Cited on page 6.)

[13] Jerome R. Busemeyer and Peter D. Bruza. *Quantum models of cognition and decision*. Cambridge University Press, 2012. (Cited on page 18.)

[14] Angelo Cangelosi. The grounding and sharing of symbols. *Pragmatics and Cognition*, 14(2):275–285, 2006. (Cited on pages 30 and 31.)

[15] Daniel Chandler. *Semiotics - The Basics*. Routledge, 2007. (Cited on pages 27, 28, and 31.)

[16] Herbert H. Clark. *Using Language*. Cambridge University Press, 1996. (Cited on pages 7, 9, 24, 32, 34, 36, 37, 38, 39, 41, 42, 43, 72, 74, 75, 77, and 120.)

[17] Herbert H. Clark and Susan E. Brennan. Grounding in communication. In Lauren B. Resnick, John M. Levine, and Stephanie D. Teasley, editors, *Perspectives on Socially Shared Cognition*, pages 127–149. American Psychological Association, 1991. URL http://psych.stanford.edu/~herb/. (Cited on page 77.)

[18] Herbert H. Clark and Edward F. Schaefer. Collaborating on contributions to conversations. *Language and Cognitive Processes*, 2(1):19–41, 1987. (Cited on pages 74 and 80.)

[19] Herbert H. Clark and Edward F. Schaefer. Contributing to discourse. *Cognitive Science*, 13(2):259–294, 1989. ISSN 1551-6709. doi: 10.1207/s15516709cog1302_7. URL http://dx.doi.org/10.1207/s15516709cog1302_7. (Cited on pages 74 and 80.)

[20] Herbert H. Clark and Deanna Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22:1–39, 1986. (Cited on pages 32, 33, and 72.)

[21] Gillian Cohen and Deborah M. Burke. Memory for proper names: A review. *Memory*, 1(4):249–263, 1993. (Cited on page 4.)

[22] Silvia Coradeschi, Amy Loutfi, and Britta Wrede. A short review of symbol grounding in robotic and intelligent systems. *Kuenstliche Intelligenz*, 27:129–136, 2013. (Cited on page 30.)

[23] Helen Couclelis. Verbal directions for way-finding: Space, cognition, and language. In Juval Portugali, editor, *The Construction of Cognitive Maps*, volume 32 of *GeoJournal Library*, pages 133–153. Springer Netherlands, 1996. ISBN 978-0-7923-3949-6. doi: 10.1007/978-0-585-33485-1_7. URL http://dx.doi.org/10.1007/978-0-585-33485-1_7. (Cited on pages 19, 60, 61, and 62.)

[24] Francis. Crick and Christof Koch. Consciousness and neuroscience. *Cerebral Cortex*, 8:97–107, 1998. (Cited on page 17.)

[25] Andrew Crompton. Perceived distance in the city as a function of time. *Environment and behavior*, 38(2):173–182, 2006. (Cited on page 4.)

[26] Marie-Paule Daniel and Michel Denis. The production of route directions: investigating conditions that favour conciseness in spatial discourse. *Applied Cognitive Psychology*, 18(1):57–75, 2004. ISSN 1099-0720. doi: 10.1002/acp.941. (Cited on page 59.)

[27] Morton D. Davis. *Game Theory - A nontechnical Introduction*. Dover Publications Inc, 1997. (Cited on page 24.)

[28] Giorgio De Felice, Paolo Fogliaroni, and Jan Oliver Wallgrün. A hybrid geometric-qualitative spatial reasoning system and its application in gis. In *Proceedings of the 10th International Conference on Spatial Information Theory*, COSIT'11, pages 188–209, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-23195-7. URL http://dl.acm.org/citation.cfm?id=2040205.2040219. (Cited on page 6.)

[29] Daniel Delling, Peter Sanders, Dominik Schultes, and Dorothea Wagner. Algorithmics of large and complex networks. chapter Engineering Route Planning Algorithms, pages 117–139. Springer-Verlag, Berlin, Heidelberg, 2009. ISBN 978-3-642-02093-3. doi: 10.1007/978-3-642-02094-0_7. URL http://dx.doi.org/10.1007/978-3-642-02094-0_7. (Cited on page 3.)

[30] Michel Denis. The description of routes: A cognitive approach to the production of spatial discourse. *Current Psychology of Cognition*, 16(4):409–458, 1997. (Cited on pages 49, 58, 59, 60, 67, 68, and 69.)

[31] Michel Denis, Francesca Pazzaglia, Cesare Cornoldi, and Laura Bertolo. Spatial discourse and navigation: an analysis of route directions in the city of venice. *Applied Cognitive Psychology*, 13 (2):145–174, 1999. doi: 10.1002/(SICI)1099-0720(199904)13:2\%3C145::AID-ACP550\%3E3.0.CO;2-4. URL http://dx.doi.org/10.1002/(SICI)1099-0720(199904)13:

2%3C145::AID-ACP550%3E3.0.CO;2-4. (Cited on pages 58 and 67.)

[32] Roger M. Downs and David Stea. *Cognitive Maps and Spatial Behaviour: Process and Products*, chapter Image and Environment: Cognitive Mapping and Spatial Behavior, pages 8–26. John Wiley & Sons, 1973. (Cited on page 18.)

[33] Umerto Eco. *A Theory of Semiotics*. Midland Book, 1979. (Cited on page 27.)

[34] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998. (Cited on page 30.)

[35] Peter C Fishburn. Utility theory. *Management science*, 14(5):335–378, 1968. (Cited on page 24.)

[36] Paolo Fogliaroni. *Qualitative Spatial Configuration Queries - Towards Next Generation Access Methods for GIS*. PhD thesis, University of Bremen, 2013. (Cited on page 6.)

[37] Sylvie Fontaine and Michel Denis. The production of route instructions in underground and urban environments. In Christian Freksa and David M. Mark, editors, *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science*, volume 1661 of *Lecture Notes in Computer Science*, pages 83–94. Springer Berlin Heidelberg, 1999. ISBN 978-3-540-66365-2. doi: 10.1007/3-540-48384-5_6. URL http://dx.doi.org/10.1007/3-540-48384-5_6. (Cited on page 59.)

[38] A. U. Frank and D. M. Mark. *Language issues for GIS*, volume 1, pages 147–163. Wiley, New York, 1991. (Cited on page 9.)

[39] Andrew U. Frank. Qualitative spatial reasoning about distances and directions in geographic space. *Journal of Visual Languages and Computing*, 3:343–371, 1992. (Cited on pages 6 and 58.)

[40] Andrew U. Frank. Qualitative spatial reasoning: Cardinal directions as an example. *International Journal of Geographical Information Science*, 10(3):269–290, 1996. (Cited on page 6.)

[41] Andrew U. Frank. Spatial communication with maps: Defining the correctness of maps using a multi-agent simulation. In Freksa et al., editor, *Spatial Cognition*, volume II, 2000. (Cited on page 21.)

[42] Andrew U. Frank. Pragmatic information content how to measure the information in a route description. In *Perspectives on Geographic Information Science*, pages 47–68. Taylor & Francis, 2003. (Cited on pages 23, 52, 55, and 124.)

[43] Andrew U. Frank. Chapter 2: Ontology for spatio-temporal databases. In Timos K. Sellis, Manolis Koubarakis, Andrew Frank, Stéphane Grumbach, Ralf H. Güting, Christian Jensen,

Nikos A. Lorentzos, Yannis Manolopoulos, Enrico Nardelli, Barbara Pernici, Babis Theodoulidis, Nectaria Tryfona, Hans-Jörg Schek, and Michel O. Scholl, editors, *Spatio-Temporal Databases*, volume 2520 of *Lecture Notes in Computer Science*, pages 9–77. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-40552-8. doi: 10.1007/978-3-540-45081-8_2. URL http://dx.doi.org/10.1007/978-3-540-45081-8_2. (Cited on page 31.)

[44] Susan R. Fussell and Robert M. Krauss. Coordination of knowledge in communication: Effects of speakers' assumptions about what others know. *Journal of Personality and Social Psychology*, 62 (3):378–391, 1992. (Cited on pages 34 and 41.)

[45] Bruno Galantucci. An experimental study of the emergence of human communication systems. *Cognitive Science*, 29:737–767, 2005. (Cited on page 31.)

[46] Ioannis Giannopoulos, Peter Kiefer, Martin Raubal, Kai-Florian Richter, and Tyler Thrash. Wayfinding decision situations: A conceptual model and evaluation. In *Geographic Information Science*, pages 221–234. Springer International Publishing, 2014. (Cited on page 124.)

[47] James J. Gibson. *The Ecological Approach to Visual Perception*. Psychology Press, 1979. (Cited on pages 21 and 31.)

[48] Arthur M. Glenberg and David A. Robertson. Symbol grounding and meaning: A comparison of high-dimensional and embodied theories of meaning. *Journal of Memory and Language*, 43: 379–401, 2000. (Cited on page 30.)

[49] Paul Grice. *Studies in the Way of Words*. Harvard University Press, 1989. (Cited on pages 23, 24, 28, 42, 43, 59, 76, and 80.)

[50] Thomas R. Gruber. A translation approach to portable ontology specifications. *KNOWLEDGE ACQUISITION*, 5:199–220, 1993. (Cited on page 50.)

[51] Christopher Habel. Prozedurale aspekte der wegplanung und wegbeschreibung. In Helmut Schnelle and Gert Rickheit, editors, *Sprache in Mensch und Computer*, Psycholinguistische Studien, pages 107–133. VS Verlag für Sozialwissenschaften, 1988. ISBN 978-3-531-11919-9. doi: 10.1007/978-3-322-86251-8_5. URL http://dx.doi.org/10.1007/978-3-322-86251-8_5. (Cited on pages 51 and 52.)

[52] Juergen Hahn and Paul Weiser. A quantum formalization for communication coordination problems. In *8th International Conference on Quantum Interaction, Filzbach Switzerland*, 2014. (Cited on pages 7, 41, and 56.)

[53] Stefan Hansen, Kai-Florian Richter, and Alexander Klippel. Landmarks in openls - a data structure for cognitive ergonomic

route directions. In *Geographic Information Science - Fourth International Conference, GIScience 2006*, pages 128–144. Springer, 2006. (Cited on page 53.)

[54] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346, 1990. (Cited on page 29.)

[55] Stephen C. Hirtle, Kai-Florian Richter, Samvith Srinivas, and Robert Firth. This is the tricky part: When directions become difficult. *J. Spatial Information Science*, 1(1):53–73, 2010. (Cited on pages 7 and 59.)

[56] Stephen C. Hirtle, Sabine Timpf, and Thora Tenbrink. The effect of activity on relevance and granularity for navigation. In *Proceedings of the 10th international conference on Spatial information theory*, COSIT'11, pages 73–89, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-23195-7. URL http://dl.acm.org/citation.cfm?id=2040205.2040211. (Cited on pages 5, 7, and 59.)

[57] Christoph Hoelscher, Thora Tenbrink, and Jan M. Wiener. Would you follow your own route description? cognitive strategies in urban route planning. *Cognition*, 121(2):228–247, 2011. ISSN 0010-0277. doi: 10.1016/j.cognition.2011.06.005. (Cited on pages 5 and 60.)

[58] Edwin Hutchins. *Cognition in the Wild*. The MIT Press (A Bradford Book), 1995. (Cited on pages 3 and 32.)

[59] Ellen A. Isaacs and Herbert H. Clark. References in conversation between experts and novices. *Journal of Experimental Psychology: General*, 116:26–37, 1987. (Cited on page 41.)

[60] Toru Ishikawa and Daniel R. Montello. Spatial knowledge acquisition from direct experience in the environment: Individual differences in the development of metric knowledge and the integration of separately learned places. *Cognitive Psychology*, 52: 93–129, 2006. (Cited on page 22.)

[61] ISO. Open systems interconnection (osi) –basic reference model: The basic model, iec 7498-1. *ISO standard ISO/IEC*, pages 7498–1, 1994. (Cited on pages 25 and 123.)

[62] Mark Johnson. *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*. University of Chicago Press, 1987. (Cited on pages 19, 31, and 42.)

[63] Philip N. Johnson-Laird. *Foundations of Cognitive Science*, chapter Mental Models, pages 469–499. MIT Press, 1989. (Cited on page 19.)

[64] Philip N. Johnson-Laird. *The history of mental models*, chapter Psychology of Reasoning, pages 179–212. 2004. (Cited on pages 21 and 22.)

[65] Simon Peython Jones. *Haskell 98 Language and Libraries: The Revised Report*. Cambridge University Press, 2003. (Cited on pages 11 and 91.)

[66] David Jonietz and Sabine Timpf. An affordance-based simulation framework for assessing spatial suitability. In Thora Tenbrink, John Stell, Antony Galton, and Zena Wood, editors, *Spatial Information Theory*, volume 8116 of *Lecture Notes in Computer Science*, pages 169–184. Springer International Publishing, 2013. ISBN 978-3-319-01789-1. doi: 10.1007/978-3-319-01790-7_10. URL http://dx.doi.org/10.1007/978-3-319-01790-7_10. (Cited on pages 5 and 60.)

[67] Daniel Kahneman. *Thinking, Fast and Slow*. Allen Lane, 2011. (Cited on pages 17 and 42.)

[68] Peter Kiefer, Ioannis Giannopoulos, and Martin Raubal. Where am i? investigating map matching during self-localization with mobile eye tracking in an urban environment. *Transactions in GIS*, 2013. (Cited on page 123.)

[69] Wolfgang Klein. Wegauskünfte. *Zeitschrift für Literaturwissenschaft und Linguistik*, 9(33):9–57, 1979. (Cited on pages 6, 11, 60, 61, 62, 67, 71, 72, and 143.)

[70] Wolfgang Klein. Local deixis in route directions. In R. J. Jarvella and W. Klein, editors, *Speech, Place, and Action*, pages 161–182. Wiley, Chichester, 1982. (Cited on page 60.)

[71] Alexander Klippel. Wayfinding choremes. In Werner Kuhn, Michael Worboys, and Sabine Timpf, editors, *COSIT*, number LNCS 2825, pages 301–315, 2003. (Cited on pages 52, 58, and 116.)

[72] Alexander Klippel. *Wayfinding choremes: Conceptualizing wayfinding and route direction elements*. PhD thesis, SFB/TR 8 Spatial Cognition, 2003. (Cited on pages 6, 52, 55, and 68.)

[73] Alexander Klippel, Heike Tappe, Lars Kulik, and Paul U. Lee. Wayfinding choremes-a language for modeling conceptual route knowledge. *J. Vis. Lang. Comput.*, 16(4):311–329, August 2005. ISSN 1045-926X. doi: 10.1016/j.jvlc.2004.11.004. URL http://dx.doi.org/10.1016/j.jvlc.2004.11.004. (Cited on pages 9 and 52.)

[74] Alexander Klippel, Stefan Hansen, Kai-Florian Richter, and Stephan Winter. Urban granularities - a data structure for cognitively ergonomic route directions. *GeoInformatica*, 13:223–247, 2009. ISSN 1384-6175. doi: 10.1007/s10707-008-0051-6. URL http://dx.doi.org/10.1007/s10707-008-0051-6. (Cited on page 52.)

[75] Joachim I. Krueger. From social projection to social behaviour. *European Review of Social Psychology*, 18(1):1–35, 2007. (Cited on page 40.)

[76] Tobias Kuhn. A survey and classification of controlled natural languages. *Computational Linguistics*, 40(1):121–170, 2014. (Cited on page 32.)

[77] Werner Kuhn. Modeling the semantics of geographic categories through conceptual integration. In *Geographic Information Science*, pages 108–118. Springer Berlin Heidelberg, 2002. (Cited on page 100.)

[78] Werner Kuhn. Geospatial semantics: Why, of what, and how? 3534:1–24, 2005. (Cited on page 30.)

[79] Benjamin Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2(2):129–153, 1978. ISSN 1551-6709. doi: 10.1207/s15516709cog0202_3. (Cited on page 50.)

[80] Benjamin Kuipers. The map in the head metaphor. *Environnment and Behavior*, 14(2):202–220, 1982. (Cited on page 18.)

[81] Benjamin Kuipers. Modeling human knowledge of routes: Partial knowledge and individual variation. In *In Proceedings, AAAI-83 Conference*, pages 216–219. Morgan Kaufmann, 1983. (Cited on pages 50, 51, and 53.)

[82] Benjamin Kuipers. The spatial semantic hierarchy. *Artifical Intelligence*, 119:119–233, 2000. (Cited on page 19.)

[83] George Lakoff. *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*. University of Chicago Press, 1987. (Cited on pages 31, 33, 42, and 56.)

[84] George Lakoff and M. Johnson. *Metaphors We Live By*. University of Chicago Press, 1980. (Cited on page 42.)

[85] George Lakoff and Rafael E. Nunez. *Where Mathematics Comes From: How the embodied mind brings Mathematics into Being*. Basic Books, 2001. (Cited on pages 6, 18, 27, and 58.)

[86] Aleksei N. Leontiev. The problem of activity in psychology. *Soviet Psychology*, 13(2):4–33, 1974. URL http://www.comnet.ca/~{}pballan/Leontiev%281979%29.htm. (Cited on page 56.)

[87] W. J. Levelt. The speaker's linearization problem. *Philosophical Transactions of the Royal Society of London*, B 295:305–315+, 1981. (Cited on page 22.)

[88] Willem J. Levelt. Linearization in describing spatial networks. In Stanley Peters and Esa Saarinen, editors, *Processes, Beliefs, and Questions*, volume 16 of *Synthese Language Library*, pages 199–220. Springer Netherlands, 1982. doi: 10.1007/978-94-015-7668-0_7. URL http://dx.doi.org/10.1007/978-94-015-7668-0_7. (Cited on page 57.)

[89] Stephen C. Levinson. *Pragmatics*. Cambridge University Press, 1983. (Cited on pages 9, 32, 33, 34, 35, 36, and 58.)

[90] Stephen C. Levinson. On the human "interaction engine". In Nicholas J. Enfield and Stephen C. Levinson, editors, *Roots of Human Sociality: Culture, Cognition and Interaction*, pages 39–69. Berg Publishers, October 2006. URL http://www.worldcat.org/isbn/1845203941. (Cited on pages 23 and 33.)

[91] Charlotte Linde and William Labov. Spatial networks as a site for the study of language and thought. *Language*, 51(4):924–939, 1975. (Cited on pages 22 and 60.)

[92] Kristin L. Lovelace, Mary Hegarty, and Daniel R. Montello. Elements of good route directions in familiar and unfamiliar environments. In *Proceedings of the International Conference on Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science*, COSIT '99, pages 65–82, London, UK, UK, 1999. Springer-Verlag. ISBN 3-540-66365-7. URL http://dl.acm.org/citation.cfm?id=646127.681068. (Cited on pages 53, 58, and 68.)

[93] Kevin Lynch. *The Image of the City*. The MIT Press, 1960. (Cited on pages 19, 20, 21, 53, and 94.)

[94] Bertram Malle. *NOBA Textbook Series: Psychology*, chapter Theory of Mind. DEF Publishers, 2013. (Cited on page 40.)

[95] David M. Mark and Michael D. Gould. Wayfinding as discourse: A comparison of verbal directions in English and Spanish. *Multilingua*, 11(3):267–291, 1992. (Cited on page 58.)

[96] David Marr. *Vision*. W.H. Freeman and Company, 1982. (Cited on pages 18 and 21.)

[97] Andrew N. Meltzoff. Understanding the intentions of others: re-enactment of intended acts by 18-month-old children. *Developmental psychology*, 31(5):838, 1995. (Cited on page 40.)

[98] Pierre-Emmanuel Michon and Michel Denis. When and why are visual landmarks used in giving directions? In Daniel R. Montello, editor, *Spatial Information Theory*, volume 2205 of *Lecture Notes in Computer Science*, pages 292–305. Springer Berlin Heidelberg, 2001. ISBN 978-3-540-42613-4. doi: 10.1007/3-540-45424-1_20. URL http://dx.doi.org/10.1007/3-540-45424-1_20. (Cited on page 58.)

[99] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63(2):81–97, March 1956. URL http://www.musanim.com/miller1956/. (Cited on page 9.)

[100] Marvin Minsky. A framework for representing knowledge. Technical report, Cambridge, MA, USA, 1974. (Cited on page 31.)

[101] D. R. Montello. *The Cambridge Handbook of Visuospatial Thinking*, chapter Navigation. Cambridge University Press, 2005. (Cited on page 3.)

[102] Daniel R. Montello. Scale and multiple psychologies of space. In AndrewU Frank and Irene Campari, editors, *Spatial Information Theory A Theoretical Basis for GIS*, volume 716 of *Lecture Notes in Computer Science*, pages 312–321. Springer Berlin Heidelberg, 1993. doi: 10.1007/3-540-57207-4_21. URL http://dx.doi.org/10.1007/3-540-57207-4_21. (Cited on page 21.)

[103] Daniel R. Montello. A new framework for understanding the acquisition of spatial knowledge in large-scale environments. *Spatial and temporal reasoning in geographic information systems*, pages 143–154, 1998. (Cited on page 22.)

[104] Daniel R. Montello and Paul C. Sutton. *An Introduction to Scientific Research Methods in Geography*. Sage Publications, 2006. (Cited on pages 11 and 72.)

[105] Daniel Morrow, Alfred Lee, and Michelle Rodvold. Analysis of problems in routine controller-pilot communication. *The International Journal of Aviation Psychology*, 3(4):285–302, 1993. doi: 10.1207/s15327108ijap0304_3. URL http://www.tandfonline.com/doi/abs/10.1207/s15327108ijap0304_3. (Cited on page 77.)

[106] Daniel Morrow, Michelle Rodvold, and Alfred Lee. Nonroutine transactions in controller-pilot communication. *Discourse Processes*, 17(2):235–258, 1994. doi: 10.1080/01638539409544868. URL http://www.tandfonline.com/doi/abs/10.1080/01638539409544868. (Cited on page 77.)

[107] Joe Navarro and Marvin Karlins. *What every body is saying*. William Morrow Paperbacks, 2014. (Cited on page 123.)

[108] Donald A. Norman. *Cognitive Engineering*. User Centered System Design. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986. (Cited on pages 3 and 48.)

[109] Clemens Nothegger, Stephan Winter, and Martin Raubal. Selection of salient features for route directions. *Spatial cognition and computation*, 4(2):113–136, 2004. (Cited on page 53.)

[110] Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994. (Cited on page 24.)

[111] Martin J. Pickering and Simon Garrod. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–226, 2004. (Cited on page 41.)

[112] Steven Pinker. *The Language Instinct*. Harper Perennial Modern Classics, 1994. (Cited on page 31.)

[113] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *The Behaviroal and Brain Sciences*, 4:515–526, 1978. (Cited on page 40.)

[114] Hilary Putnam. The meaning of 'meaning'. *Minnesota Studies in the Philosophy of Science*, 7:131–193, 1975. (Cited on page 33.)

[115] Martin Raubal and Stephan Winter. Enriching wayfinding instructions with local landmarks. In *GIScience 2002*, pages 243–259. Springer, 2002. (Cited on pages 6, 53, and 92.)

[116] Kai-Florian Richter, Tomko Martin, and Stefan Winter. A dialog-driven process of generating route directions. *Computers, Environment and Urban Systems*, 32:233–245, 2008. (Cited on page 57.)

[117] Christopher K. Riesbeck. You can't miss it!: Judging the clarity of directions. *Cognitive Science*, 4(3):285–303, 1980. ISSN 1551-6709. doi: 10.1207/s15516709cog0403_3. URL http://dx.doi.org/10.1207/s15516709cog0403_3. (Cited on page 48.)

[118] Giacomo Rizzolatti and Michael A. Arbib. Language within our grasp. *Trends in Neuroscience*, 21:188–194, 1998. (Cited on page 42.)

[119] Eleanor Rosch. Natural categories. *Cognitive Psychology*, 4:328–350, 1973. (Cited on page 19.)

[120] Eleanor Rosch. Principles of categorization. In E. Rosch and B. B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum Associates, Hillsdale (NJ), USA, 1978. Reprinted in *Readings in Cognitive Science. A Perspective from Psychology and Artificial Intelligence*, A. Collins and E.E. Smith, editors, Morgan Kaufmann Publishers, Los Altos (CA), USA, 1991. (Cited on page 94.)

[121] Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382 – 439, 1976. ISSN 0010-0285. doi: http://dx.doi.org/10.1016/0010-0285(76)90013-X. URL http://www.sciencedirect.com/science/article/pii/001002857690013X. (Cited on page 56.)

[122] Tracy Ross, Andrew May, and Simon Thompson. The use of landmarks in pedestrian navigation instructions and the effects of context. In Stephen Brewster and Mark Dunlop, editors, *Mobile Human-Computer Interaction - MobileHCI 2004*, volume 3160 of *Lecture Notes in Computer Science*, pages 300–304. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-23086-1. doi: 10.1007/978-3-540-28637-0_26. (Cited on page 58.)

[123] L. Tiina Sarjakoski, Pyry Kettunen, Hanna-Marika Flink, Mari Laakso, Mikko Rönneberg, and Tapani Sarjakoski. Analysis of verbal route descriptions and landmarks for hiking. *Personal*

Ubiquitous Comput., 16(8):1001–1011, December 2012. ISSN 1617-4909. doi: 10.1007/s00779-011-0460-7. URL http://dx.doi.org/10.1007/s00779-011-0460-7. (Cited on page 59.)

[124] Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals, and Understanding*. BF 311 S 378, 1977. (Cited on page 31.)

[125] Manuela Schmidt and Paul Weiser. Web mapping services: Development and trends. In *Online Maps with APIs and WebServices*, pages 13–21. Springer, 2012. (Cited on page 3.)

[126] Angela Schwering, Rui Li, and Vanessa Joy A. Anacta. Orientation information in different forms of route instructions. In *Short Paper at the 16th AGILE Conference on Geographic Information Science*, 2013. (Cited on pages 50, 68, and 69.)

[127] John R. Searle. *Speech Acts: An essay in the philosophy of language*. Cambridge University Press, 1969. (Cited on pages 35, 36, and 120.)

[128] John R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3:417–424, 1980. (Cited on pages 18, 28, and 29.)

[129] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948. (Cited on pages 23, 54, and 120.)

[130] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008. (Cited on page 24.)

[131] Alexander W. Siegel and Sheldon H. White. The development of spatial representations of large-scale environments. *Advances in Child Development and Behavior*, 10:9–55, 1975. (Cited on page 21.)

[132] Herbert Simon. *The Sciences of The Artificial*. MIT Press, 1998. (Cited on pages 32, 41, 54, 97, and 111.)

[133] Dan Sperber and Deidre Wilson. *Relevance : Communication and Cognition*. Basil Blackwell Oxford, 1986. (Cited on pages 42 and 43.)

[134] Samvith Srinivas and Stephen C. Hirtle. Knowledge based schematization of route directions. In Thomas Barkowsky, Markus Knauff, Gérard Ligozat, and Daniel R. Montello, editors, *Spatial Cognition*, volume 4387 of *Lecture Notes in Computer Science*, pages 346–364. Springer, 2006. ISBN 978-3-540-75665-1. URL http://dblp.uni-trier.de/db/conf/spatialCognition/spatialCognition2006.html#SrinivasH06. (Cited on page 53.)

[135] Robert Stalnaker. Common ground. *Linguistics and Philosophy*, 25:701–721, 2002. (Cited on page 39.)

[136] Luc Steels. *The Talking Heads Experiment Volume 1*. Antwerpen: Laboratorium, 1999. (Cited on pages 31 and 39.)

[137] Luc Steels. *Symbols and embodiment: Debates on meaning and cognition*, chapter The symbol grounding problem has been solved. So what's next?, pages 223–244. Oxford University Press, 2008. (Cited on page 30.)

[138] Albert Stevens and Patty Coupe. Distortions in judged spatial relations. *Cognitive Psychology*, 10(4):422 – 437, 1978. ISSN 0010-0285. doi: http://dx.doi.org/10.1016/0010-0285(78)90006-3. URL http://www.sciencedirect.com/science/article/pii/0010028578900063. (Cited on page 18.)

[139] T. Tenbrink and S. Winter. Variable granularity in route directions. *Spatial Cognition and Computation*, 9(1):64 – 93, 2009. doi: 10.1080/13875860902718172. (Cited on pages 7, 54, and 55.)

[140] Paul Thagard. *Mind - Introduction to Cognitive Science*. MIT Press, 2005. (Cited on page 18.)

[141] Sabine Timpf. Abstraction, levels of detail, and hierarchies in map series. In Christian Freksa and David M. Mark, editors, *COSIT*, volume 1661 of *Lecture Notes in Computer Science*, pages 125–139. Springer, 1999. ISBN 3-540-66365-7. URL http://dblp.uni-trier.de/db/conf/cosit/cosit99.html#Timpf99. (Cited on page 54.)

[142] Sabine Timpf. Ontologies of wayfinding: a traveler's perspective. *Networks and Spatial Economics*, 2(1):9–33, 2002. ISSN 1566-113X. doi: 10.1023/A:1014563113112. (Cited on pages 8 and 56.)

[143] Sabine Timpf and Werner Kuhn. Granularity transformations in wayfinding. In *In Spatial Cognition*, pages 77–88. Springer, 2003. (Cited on page 55.)

[144] Sabine Timpf, Gary S. Volta, David W. Pollock, and Max J. Egenhofer. A conceptual model of wayfinding using multiple levels of abstraction. In *Proceedings of the International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning on Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, pages 348–367, London, UK, UK, 1992. Springer-Verlag. ISBN 3-540-55966-3. URL http://dl.acm.org/citation.cfm?id=647939.741251. (Cited on page 55.)

[145] Ariane Tom and Michel Denis. Referring to landmark or street information in route directions: What difference does it make? In Walter Kuhn, Michael F. Worboys, and Sabine Timpf, editors, *Spatial Information Theory. Foundations of Geographic Information Science*, volume 2825 of *Lecture Notes in Computer Science*, pages 362–374. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-20148-9. doi: 10.1007/978-3-540-39923-0_24. URL

http://dx.doi.org/10.1007/978-3-540-39923-0_24. (Cited on page 59.)

[146] Ariane Tom and Michel Denis. Language and spatial cognition: comparing the roles of landmarks and street names in route instructions. *Applied Cognitive Psychology*, 18(9):1213–1230, 2004. ISSN 1099-0720. doi: 10.1002/acp.1045. URL http://dx.doi.org/10.1002/acp.1045. (Cited on pages 48 and 67.)

[147] Martin Tomko. *Destination Descriptions in Urban Environments*. PhD thesis, Department of Geomatics, The University of Melbourne, 2007. (Cited on page 56.)

[148] Martin Tomko and Stephan Winter. Recursive construction of granular route directions. *Journal of Spatial Science*, 51:101–115, 2006. (Cited on page 56.)

[149] Martin Tomko and Stephan Winter. Pragmatic construction of destination descriptions for urban environments. *Spatial Cognition and Computation*, 9:1–29, 2009. (Cited on pages 56 and 57.)

[150] Barbara Tversky. Cognitive maps, cognitive collages, and spatial mental models. In *COSIT*, pages 14–24, 1993. (Cited on pages 19, 21, and 71.)

[151] Paul Weiser and Andrew U. Frank. Cognitive transactions - a communication model. In Thora Tenbrink, John Stell, Anthony Galton, and Zena Wood, editors, *Conference on Spatial Information Theory*, number 8116, pages 129–148, 2013. (Cited on pages 5, 6, 7, 9, 11, 52, 67, 71, and 72.)

[152] Paul Weiser, Andrew U. Frank, and Amin Abdalla. Process composition and process reasoning over multiple levels of detail. In *GI Science 2012 Extended Abstract*, 2012. (Cited on page 56.)

[153] Norbert Wiener. *The Human Use of Human Beings - Cybernetics and Society*. Free Association Books, 1989. (Cited on pages 23, 28, and 34.)

[154] Anna Wierzbicka. *Semantics: Primes and Universals*. Oxford University Press, 1996. (Cited on page 30.)

[155] Stephan Winter and Yunhui Wu. Intelligent spatial communication. In Gerhard Navratil, editor, *Research Trends in Geographic Information Science*, Lecture Notes in Geoinformation and Cartography, pages 235–250. Springer Berlin Heidelberg, 2009. ISBN 978-3-540-88243-5. doi: 10.1007/978-3-540-88244-2_16. URL http://dx.doi.org/10.1007/978-3-540-88244-2_16. (Cited on pages 9 and 12.)

[156] Dieter Wunderlich and Rudolph Reinelt. *Speech, Place, and Action*, chapter How to Get There From Here, pages 183–201. John Wiley & Sons Ltd., 1982. (Cited on pages 48, 60, and 61.)

[157] George Yule. *Pragmatics*. Oxford University Press, 1996. (Cited on pages 9 and 36.)

[158] Hubert Zimmermann. Osi reference model - the iso model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):425–432, 1980. (Cited on page 25.)