2016-1

Resolving Perception Based Problems in Human-Computer Dialogue

Niels Schütte
niels.schutte@student.dit.ie

Recommended Citation

Schutte, N. (2016) *Resolving perception based problems in human-computer dialogue*. Thesis submitted for the degree of Doctor of Philosophy to Technological University Dublin, School of Computing, January 2016. doi:10.21427/D74K56

Resolving Perception Based Problems in Human-Computer Dialogue

by

Niels Schütte

Supervisors: John Kelleher
Brian Mac Namee

School of Computing

Dublin Institute of Technology

A thesis submitted for the degree of

Doctor of Philosophy

January, 2016

Declaration

I certify that this thesis which I now submit for examination for the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate study by research of the Dublin Institute of Technology and has not been submitted in whole or in part for an award in any other Institute or University.

The work reported on in this thesis conforms to the principles and requirements of the institute's guidelines for ethics in research.

The Institute has permission to keep, to lend or to copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature__________________________ Date__________________________

Acknowledgements

I would like to thank my supervisors John Kelleher and Brian Mac Namee for their advice and support during this PhD. I would equally like to thank my parents for the support and patience.

Furthermore I would like to thank a number of people who I met along the way. The other members of the original Lok8 team: John, Mark and Slav. The other members of the AI Group office: Colm, Paddy, Ken and Yan.

I would also like the thank the members of the DIT Judo Club and the Aquatec Sub Aqua Club.

I also need to thank the other people from DIT and Focas and many people from the outside who helped and supported me, but they are too numerous too mention. I promise to try to catch up once things have settled down a bit.

And of course, I need to thank the countless volunteers who participated in the experiments :-) .

Abstract

We investigate the effect of sensor errors on situated human-computer dialogues. If a human user instructs a robot to perform a task in a spatial environment, errors in the robot's sensor based perception of the environment may result in divergences between the user's and the robot's understanding of the environment.

If the user and the robot communicate through a language based interface, these problems may result in complex misunderstandings. In this work we investigate such situations. We set up a simulation based scenario in which a human user instructs a robot to perform a series of manipulation tasks, such as lifting, moving and re-arranging simple objects. We induce errors into the robot's perception, such as misclassification of shapes and colours, and record and analyse the user's attempts to resolve the problems.

We evaluate a set of methods to alleviate the problems by allowing the operator to access the robot's understanding of the scene. We investigate a uni-directional language based option, which is based on automatically generated scene descriptions, a visually based option, in which the system highlights objects and provides known properties, and a dialogue based assistance option. In this option the participant can ask simple questions about the robot's perception of the scene. As a baseline condition we perform the experiment without introducing any errors.

We evaluate and compare the success and problems in all four conditions. We identify and compare strategies the participants used in each condition. We find that the participants appreciate and use the information request options successfully. We find that that all options provide an improvement over the condition without information.

We conclude that allowing the participants to access information about the robot's perception state is an effective way to resolve problems in the dialogue.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The core idea of this thesis is to investigate the effect of perception errors on situated human-computer dialogue. In particular, we are interested in scenarios in which a human user interacts with a robot that experiences perception problems, and in how the users resolve the problems.

In our basic application scenario, a human user interacts with a robot that is in a remote location through a language based dialogue interface. The robot uses a video camera to perceive the environment and sends a live feed of the video to the user. The robot and the user therefore have a shared perspective on the world. A set-up like this may be useful in environments that are accessible to a robot, but inaccessible to humans. This may be due to environmental hazards (such as fire, radioactivity or a danger of collapsing structures in emergency situations), due to restricted physical accessibility (e.g. in the exploration of the caves or archaeological structures) or generally extreme environments (such as under water or in outer space) (Summers-Stay *et al.*, 2014).

In scenarios such as these it may be advantageous to allow the human user to instruct the robot with goals at a high level, while leaving low level details of the implementation of the instructions to the robot itself. The robot needs to be able to perceive the environment, detect and recognize objects, and it needs to be able to establish a correspondence between the discussion with the user and the contents of the environment. In order to communicate successfully, the robot and the participant need to achieve a shared understanding of the environment (illustrated in Figure 1.1). If they do not have a shared understanding, e.g. because the robot has errors in its object detection mechanism, problems may arise in the communication (illustrated in Figure 1.2). Any statement the robot makes that involves the environment, or information derived from the environment, is potentially problematic to the dialogue partner, because the partner has a diverging idea of the environment.

We are interested in this scenario for two reasons. First it presents a problem. If the robot's understanding diverges from the user's understanding, the potential for **misunderstandings** arises. These misunderstandings can affect the quality of the dialogue and make it difficult or impossible to solve tasks that the dialogue partners are meant to co-operate on.

Second, it also presents an opportunity. If the robot has access to a dialogue partner with "better" perception, it may communicate with this partner in order to improve its own understanding of the world and improve

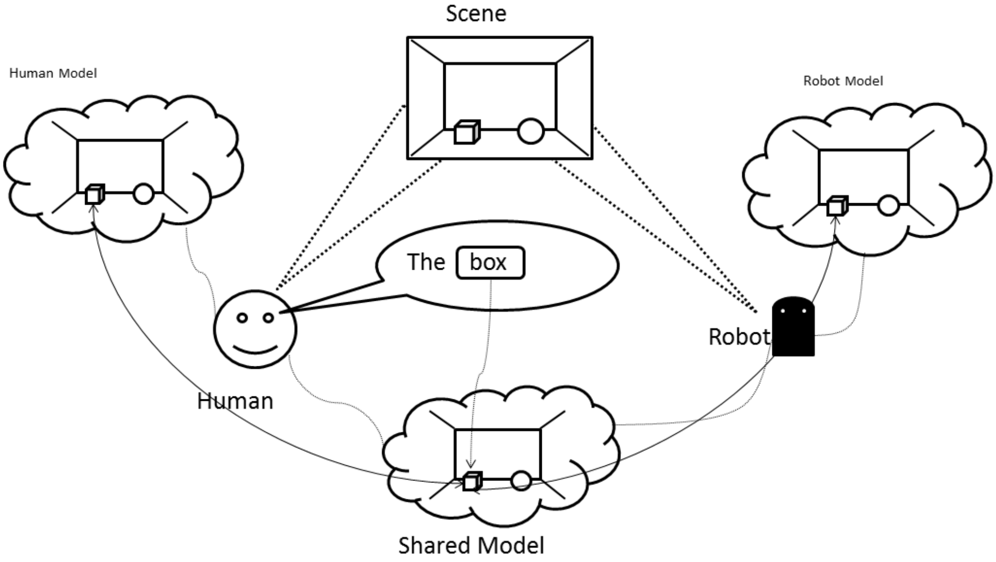


Figure 1.1: The user and the robot have a shared understanding of the environment. The user refers to the box in the scene, and the robot is able to correctly resolve the reference. Communication is successful.

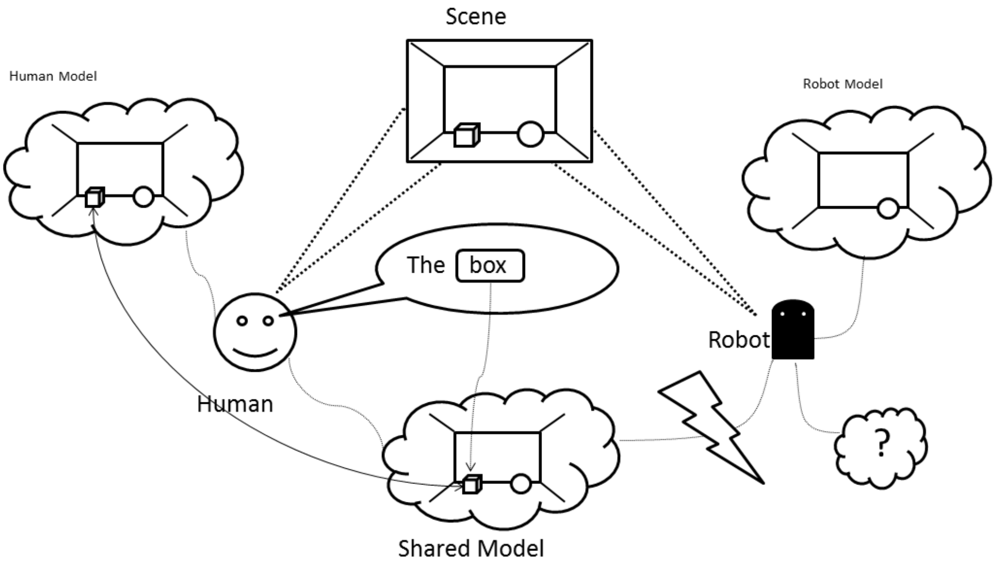


Figure 1.2: The user and the robot do not have a shared understanding of the environment. The robot does not perceive the box and is not able to understand the user's reference. Communication is unsuccessful.

its perception for the future, e.g. by re-training its perception classification models. An alternative option would be to, rather than having the robot adapt to the user, enable the user to understand the problems experienced by the robot, and resolve them by adapting to the robot. Adaptation, or alignment, (Pickering & Garrod, 2006) is an important part of dialogue between human speakers, and utilizing the human ability to adapt to ameliorate problems appears an approach worth investigating.

In this work we present a series of experiments in which the human participants co-operate through a dialogue interface with a robot that is affected by perception errors. We investigate how participants react to problems that arise due perception errors and how they resolve them. We also offer a number of different ways to request information about the robot's understanding of the environment to the participants, and observe how the different options affect the problems in the dialogues and the resolution strategies.

1.1 Contributions

The contributions of this thesis arise from the Toy Block experiment and the evaluation of its results. The main contributions are as follows:

1. The experiment itself and the dialogue system that was implemented to perform the experiment. The experiment design describes a set-up in which a human user interacts with a robot to resolve an object manipulation task. In the different phases of the experiment, the robot experiences perception errors, and offers different ways to request in-

formation about the robot's perception of the world.

The dialogue system simulates a robot with a dialogue interface that is able to manipulate objects in a virtual world. Errors can be introduced into the robot's perception and the robot is able to provide information about its perception of the world through descriptions, visual markup and through dialogue.

2. We show through the experiment that if a robot's perception is affected by errors, this makes it harder to complete tasks in cooperation with a human user, and that the user experiences an increase in frustration.

3. We show that if users are given information about the robot's perception of the world, this increases the users' confidence and their ability to complete tasks in cooperation with the robot.

4. We show that, to resolve problems arising from perception errors, users tend to align their descriptions to the robot's understanding of the world if they can access information about it.

5. We furthermore show that if users have no direct information about the robot's perceptions, they tend to avoid using descriptions that can be affected by perception errors, and use descriptions that are robust to them instead.

6. We show that users request information about the robot's perception of the world particularly often after they encounter a problem in the dialogue.

7. We present models based on the data from the experiments that describe the sequences of actions participants perform to resolve problems that arise in dialogues due to perception errors experienced by the robot.

We see **Contribution 2**, **4** and **5** as the most important contributions of this thesis. With **Contribution 2** we show that perception errors have a negative impact on situated dialogue, and thereby highlight the need to address this issue. With **Contribution 4**, we show that if we give participants information about the problem-affected understanding the robot dialogue partner has of what it perceives, they use that information to facilitate the interaction. With **Contribution 5** we show that if participants do not get access to this type of information, they develop strategies that avoid unreliable information and instead utilize information that is robust. We believe that our findings may be generalized to other problems in dialogue that arise from non-shared information. Table 1.1 contains an overview of the contributions of this thesis, the chapters they are discussed in and the research questions and publications related to them.

Contribution	Chapter	Research Questions	Publications
1. The Toy Block experiment, an experiment for investigating dialogue between a human user and a robot that experiences perception problems.	Chapter 3 and Chapter 4.		(Schütte *et al.*, 2014c)
Experimental findings showing that: • 2. Perception errors cause problems in human-robot interaction. • 3. Giving users access to information about the robot's perceptions helps them to resolve the problems.	Chapter 5	RQ 5.1, RQ 5.2, RQ 5.4	(Schütte *et al.*, 2014b)
Experimental findings showing that: • 4. Users align to the robot's perception if they can access information about it. • 5. Users avoid descriptions that are prone to perception errors.	Chapter 8	RQ 8.2, RQ 8.3	(Schütte *et al.*, 2014a),(Schütte *et al.*, 2015)
6. Experimental findings showing that users request information about the robot's perception after they encounter a problem in the dialogue.	Chapter 6	RQ 6.3	
7. Models of actions human users performed to resolve perception based problems.	Chapter 7	RQ 7.3	

Table 1.1: The contributions presented in this thesis.

1.2 List of Research Questions

In the following we present a list of the research questions addressed in this thesis. The were addressed through a set of experiments in the Toy Block experiment setup.

- In Chapter 5 we evaluate the high-level results of the Toy Block experiment about perception errors in situated human-computer dialogues and address the following questions. Our main goal here is to show that perception errors have an impact on a situated dialogue, and that giving participants information about the robot helps reduce problems.
  - **Research Question 5.1**: *How did the participants experience the task and the problems in the dialogues?*
  - **Research Question 5.2**: *Do perception errors as experienced by the robot have an impact on the difficulty of the task?*
  - **Research Question 5.3**: *If participants are offered the option to request information about the robot's understanding of the scene, do they use it?*
  - **Research Question 5.4**: *Does the ability to request information about the robot's understanding of the scene have an impact on the participants' ability to solve the task?*
  - **Research Question 5.5**: *How do the information request options compare to each other in terms of effectiveness?*

- After investigating the effect of the errors on the dialogues, we investigate in Chapter 6 how and under what circumstances participants request information about the robot's perception of the world.

  - **Research Question 6.l**: *How often did the participants request information?*
  - **Research Question 6.2**: *Did the way the participants requested information evolve during the course of the experiment?*
  - **Research Question 6.3**: *Under what circumstances did the participants request information?*
  - **Research Question 6.4**: *What were the effects of sequences of queries?*

- After we showed in the previous chapters that perception errors have an impact on the dialogues, and that participants use information request options to resolve the problems, we then focus in Chapter 7 in more detail on the participants' reaction to perception based problems in the dialogue. We investigate what actions the participants performed to resolve perception based problems. We address the following questions:

  - **Research Question 7.1**: *How successful were the outcomes of the resolution attempts?*
  - **Research Question 7.2**: *How long did the resolution attempts take?*

  - **Research Question 7.3**: *What structures can be observed in the resolution attempts?*

- After we investigated what actions participants performed after they encountered a perception based problem in Chapter 8, we investigate how participants modified references after they had encountered a perception based problem in the dialogue. In a sense, we focus in this chapter on the content of what is being said in the resolution attempts, as compared to the previous chapter where we focused on the sequences of actions participants performed. We address the following questions:
  - **Research Question 8.1**: *What attributes did the participants include in their initial and final reference?*
  - **Research Question 8.2**: *How did the participants modify their expressions between the initial and the final reference?*
  - **Research Question 8.3**: *What effect did information requests have on how the participants modified the references?*

1.3 Structure of the Thesis

The thesis is structured as follows:

- In Chapter 2 we discuss technologies and literature related to dialogue, dialogue systems and understanding in dialogue.
- In Chapter 3 we present the Toy Block experiment. We discuss the experiment setup, the different phases of the experiment.

- In Chapter 4 we present the dialogue system that was used to perform the experiment.

- In Chapter 5 we provide a general overview of the results of the experiment. In particular we quantify the effect of the introduction of perception errors on task success and user satisfaction and the effect of allowing the participants to request information about the robot's understanding of the world.

- In Chapter 6 we investigate how often the participants used the different information request options, and under what circumstances.

- In Chapter 7 and Chapter 8 we investigate how participants reacted after they encountered a problem in the dialogue that was due to a perception error, and how they resolved the problems. In Chapter 7 we investigate the sequence of dialogue acts that occurred after a problem occurred in the dialogue due to a perception error and describe differences between the different phases of the experiment. In Chapter 8 we focus on the choice of attributes in referring expressions that were used in resolution attempts. In particular we focus on how the participants reformulated referring expressions in repeated instructions and on how this choice was influenced by the information request options that were available.

- Finally, in Chapter 9 we conclude the thesis and provide an overview of possible future directions for this work.

1.4 Other Relevant Publications

The following publications were produced in the context of this thesis but do not directly contribute to it:

- In (Schütte *et al.*, 2010) we investigated reference in situated instruction giving dialogues and used visual salience to disambiguate ambiguous exophoric references.
- in (Schütte *et al.*, 2011) we suggested a method to automatically annotate references in a situated instruction giving situated dialogue based on the actions of the participants.

Chapter 2

Background

In this thesis we investigate dialogues between a human user and a robot that is experiencing perception problems. We are particularly interested in the effects the errors in the perception have on the dialogues, and in how the participants resolve the problems. This thesis therefore involves the research fields of dialogue, computational approaches to dialogues.

A dialogue is a conversation involving two (or more) participants. Some topics of research into dialogue are:

1. The structure of dialogue
   - What are the elements of a dialogue and how do they relate to each other?
2. The content of the dialogue
   - What information is exchanged in a dialogue, and how do the

participants form an understanding of what they discuss?

3. The relation of the dialogue with the world in general and the environment the dialogue takes place.

    - How do participants involve the environment in the dialogue, and how can we attempt interpret a dialogue given the environment?

The application scenario in this work presents a case of a dialogue that involves a human user and a computer system that controls a robot in an environment. It therefore touches on all of these areas of interest.

The term **human-computer dialogue** is used to describe situations in which a human user interacts with a computer system through a natural language based dialogue interface. A computer system that is capable of engaging in dialogue with a human participant is called a **spoken dialogue system** (McTear, 2002). Dialogue systems have been a topic of research for a number of years and have also found some practical applications, for example in telephone based services. More recently, dialogue system technology has begun to be integrated into mass market products such as Apple's Siri[1] on the iPhone and Microsoft's Cortana [2] in the Windows 10 operating system.

In this chapter we are going to discuss dialogue systems and topics from dialogue that are related to our research. In particular we are going to focus on reference and understanding in situated dialogue, and on how problems in perception can lead to problems in the dialogues. Since this work focuses on problems that may arise from errors in visual perception, we are briefly

[1] http://www.apple.com/ios/siri/

[2] http://www.windowscentral.com/cortana

going to address computer vision and why it can potentially be a source of problems.

We, however, will not attempt a general discussion of these topics but focus on their aspects that are related to our research.

2.1 Situated Dialogue Systems

A dialogue system may serve as an interface for an embodied agent. For example:

- A virtual character in a video game world for example in the games described by Chernova *et al.* (2010) or in the video game Façade[1].
- A character that exists in a virtual world but also perceives the physical world (Thórisson, 2002)
- A robot that performs tasks in an actual physical environment (Hawes *et al.*, 2012; Petrick & Foster, 2013).

Such dialogue systems are referred to as **situated dialogue systems** because they are frequently faced with situations in which the environment or objects from the spatial context are referenced in the dialogue. This is a particular challenge because the use of spatial language introduces a number of challenges that are not present in dialogue that is not situated. For example, some of the following challenges are cited by Byron in the context of reference in situated dialogue:

[1] `http://www.interactivestory.net/`

- The system needs to create a model of mutual knowledge, and in particular keep track of knowledge which may be accessible from the physical environment.

- It needs to keep track of the attentional prominence of entities. Again, while this is also a challenge in regular (not situated) dialogue, a situated dialogue needs to account for the salience of objects in the environment, and how it is influenced, e.g. by visual properties, gestures or interaction.

- The system needs to be able to understand spatial predicates that are used in the dialogue.

In the following we are going to discuss situated human-computer dialogue and the different topics related to it.

2.1.1 Elements of a Situated Dialogue Systems

As stated earlier, dialogue has been extensively studied in a number of disciplines. We will therefore not attempt to provide a general discussion of all aspects of dialogue but focus on those aspects that are related in practical terms to the issue of perception based problems in human-computer dialogue.

Dialogue systems in their early development were often understood as interfaces to database driven applications such as travel information (Seneff & Polifroni, 2000) or banking services (Melin *et al.*, 2001). Their main tasks consisted in interacting with the user to formulate valid requests to a database, and then to present the results of the requests. Therefore they

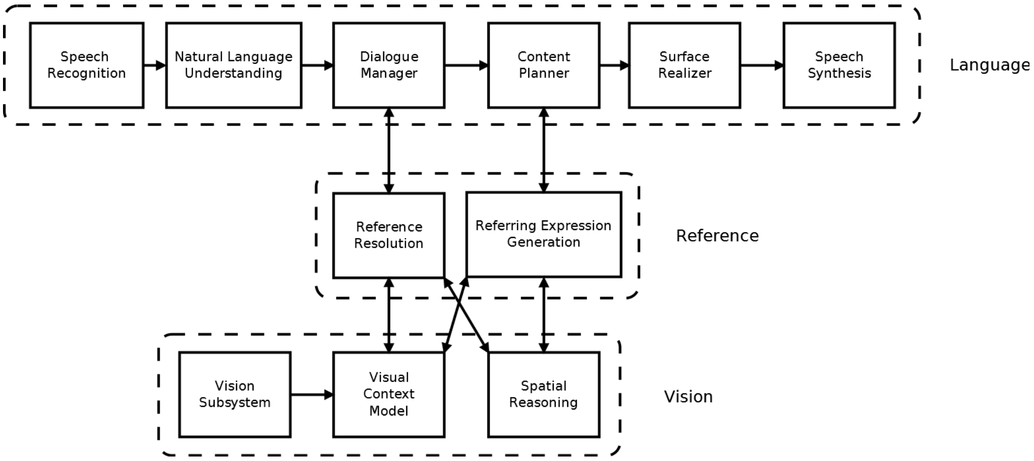


Figure 2.1: Architecture of a visually situated dialogue system.

typically were only concerned in relating the information discussed with the participant to information that was already available in a purely symbolic format which was closely related to the way information is presented in language. A situated dialogue system on the other hand needs to be able to deal with information that is available through the environment and that is not necessarily a-priori available in a symbolic format.

Figure 2.1 presents the possible architecture of situated dialogue system (based on an architecture presented in (Kelleher & Costello, 2009)). The boxes represent modules of the system and the arrows represent flow in information between modules.

It can roughly be divided into three areas: The modules at the top row of the image (labelled as the *Language* part) are responsible for handling the dialogue based interaction with the user (this type of architecture is sometimes referred to as a pipeline architecture for spoken dialogue systems (Dzikovska *et al.*, 2014)). The architecture of a classic dialogue system that is only concerned with access to data typically consists of this pipeline with an additional module that mediates interaction with the application database. The following modules are part of it:

- The **Speech Recognition** module accepts waveforms representing the sound of spoken utterances as input and produces a transcript of the text of the utterance (or some sort of probabilistic hypothesis of the text of the utterance such as a ranked list of possible results or a word lattice).

- The task of the **Natural Language Understanding** module is to produce an abstraction of the content of the utterance at a level the system can reason about. This module produces some sort of logical expression or some other frame-based representation that represents the content of the user's utterance and the associated intention.

- The task of the **Dialogue Manager** is to control the actions of the dialogue system. To do this, it accepts the output produced by the Natural Language Understanding module, interprets it in the context of the current state of the interaction, and then to decides what to do next. It produces an abstract specification of the next action the system is to perform. We will discuss the problem of dialogue management in the Section 2.2.1.2 in more detail after we have introduced more dialogue concepts that are useful for this topic.

- The **Content Planner** then takes the action produced by the dialogue manager and fleshes it out an abstract specification of a response. In particular it has to decide which information needs to be presented so that the user can understand the contribution.

- The **Surface Realizer** translates this specification into a text. This is a Natural Language Generation task (Reiter & Dale, 2000). The text is then presented to the user through the **Speech Synthesis** module.

The elements in the bottom row of Figure 2.1 represent the part of the system that is related to processing vision. They are labelled as the *Vision* part of the system. The task of the **Vision Subsystem** is to provide a model that can serve as a basis for the dialogue system to discuss the environment. It accepts images from the world (e.g. from a video camera), and detects objects of interest and their properties in these images. Based on this, the system then constructs and maintains a model of the world in the **Context Model**.

The elements between these two groups represent modules (labelled as the *Reference* part) that provide interfaces between the speech pipeline and the vision system. The **Referring Expression Generation** module helps the system decide how to talk about objects in the environment, while the **Reference Resolution** module helps the system decide what objects the user is talking about. Both of these modules provide a link between the dialogue and the contents of the **Context Model**. The **Spatial Reasoning** module assists the referring expression generation and resolution modules by relating qualitative spatial expressions from language (e.g. "to the left of") to quantitative concepts in the spatial context model. We will discuss the generation and resolution of referring expressions later in this chapter in more detail in Section 2.3.

In the remainder of this chapter we are going to discuss different topics that relate to the subject of this thesis. First we are going to discuss how dialogue can be modelled for spoken interactions. Then we will address reference and understanding in dialogue, and how problems can result from this. Finally we are going to address topics related to perception in situated dialogue systems.

2.2 Modelling Dialogue

In order to enable a system to engage in dialogue, it is necessary to investigate how humans interact in dialogue. One avenue of research has focused on the **structure** of dialogues. The goal of this type of research is to identify typical sequences in dialogues, determine how these sequences combine and interact, and to identify when and how the participants acted the way they did. An ideal outcome would be to develop a sort of syntax of dialogues that, analogously to grammars of natural languages, describes how elements fit together to construct a meaningful dialogue.

While much research was originally based on a natural interest to understand the workings of language and interaction, some of the resulting theories can be useful in the context of human-computer dialogues. A theory that accurately describes how participants act under given circumstances can be useful to predict future actions of human dialogue partners and can provide clues for the interpretation of the actions of the human participants. On the other hand, it can also be useful to plan the actions of the dialogue system.

The first step in discovering a structure in dialogue would be to determine what the building blocks of this structure might be. The second step would then be to describe how these building blocks fit together and to create a theory about how these structures come about, and how these elements relate to the task of the dialogue.

The work in this thesis is concerned with analysing the behaviour of participants of a dialogue when perception based errors lead to problems in the dialogue. We will therefore discuss in Section 2.2.1 the concept of **dialogue acts** as a description of actions in dialogue. In Section 2.2.2 we are going to discuss aspects **dialogue structure**.

2.2.1 Dialogue Acts

A concept that is frequently used in the analysis of dialogues is that of **dialogue acts** (or **speech acts** in a more general sense). A dialogue act describes a contribution to a dialogue not in terms of its literal content, but in terms of its effect as action in the sense that the speaker intends to achieve an intended effect (e.g. to make a listener believe a proposition by informing them of it) when they produce the utterance — in the same way they intend to achieve an effect when they perform a physical action (e.g. to sweeten a cup of tea by putting a lump of sugar in it). Most discussions of dialogue acts begin with the classification of speech acts presented by Searle (1975). In this theory each utterance is associated with an **illocutionary act** that represents the type of action the speaker performs with the utterance. Searle describes five types of speech acts:

- **Assertives:** Utterances in which the speaker asserts that some fact is true.
- **Directives:** Utterances by which the speaker requests the listener to perform some actions.
- **Commissives:** Utterances by which the speaker commits themself to some action.
- **Expressives:** Utterances by which the speaker expresses their psychological state of mind.
- **Declarations:** Utterances by which the speaker directly brings about some change in the world (e.g. by giving a name to an object.).

This classification describes at a general level the different types of actions that can be performed through the use of speech in general. For the area of dialogue, more specific theories and formalizations have been investigated. Bunt (1994) describes dialogue acts in terms of their effects on different aspects of the context of a dialogue. Under this perspective, dialogue acts become similar to physical actions in that they have well defined effects (that become effective when the act is successfully performed) and conditions that need to hold for a dialogue act to be effective. Cohen & Perrault (1979) explicitly formalize the effects and preconditions of dialogue acts in terms of the belief states of the participants in the framework of a

planning theory. The notion of dialogue acts is therefore particularly interesting for dialogue systems because it provides a level of abstraction over the actions in a dialogue.

Depending on the task of the dialogue system, different actions can be relevant. Therefore specific dialogue applications often feature specific dialogue act tag sets. Traum (2000) discusses some of the considerations that go into the design of a dialogue act tag set.

Dialogue acts have often been used as a tool to analyse dialogues after the fact by manually annotating corpus data. In order to use them in an actual human-computer dialogue scenario, the computer system needs to be able to determine dialogue acts expressed in the contributions by the human participant. In the next section we are going to describe approaches towards the automatic classification of the dialogue acts of utterances.

2.2.1.1 Dialogue Act Classification

In order to understand the dialogue acts performed by its human partner, a dialogue system needs to able to determine the dialogue act of each utterance. The process of automatically determining the dialogue act expressed in an utterance is called **dialogue act classification** (or **dialogue act tagging**). Unfortunately recognizing dialogue acts is not a simple task. While some aspects of the intention of an utterance can be determined directly from the grammatical form of the utterance (e.g. the mood is a good indicator), other aspects may heavily depend on inference, context and convention. For example, if a speaker produces the utterance “Can you pass me the salt?”

they normally do not intend this as a question about whether the listener is capable of providing them with salt, but as a request.

Keizer (2001) present an approach to dialogue act classification that uses a belief state and linguistic features extracted from the utterance and uses a Bayesian network. Surendran & Levow (2006) present an approach that uses contextual knowledge about the preceding dialogue act and combines this with textual and acoustic features through a machine based approach using support vector machines.

2.2.1.2 Dialogue Manager

As stated in our review of dialogue system architecture, the primary task of the dialogue manager is to direct the behaviour of the dialogue system. In order to do this, it has to relate the contributions by the user to the current context, update the context with new information, and decide when and how to act. There are different approaches towards this issue.

Three classic approaches are listed by McTear (2002) – the finite state based approach, the frame based approach and the agent based approach.

The finite state based approach is relatively strictly structured. Dialogues follows pre-defined scripts. The information structure underlying the dialogues is a discrete set of states that represents all possible situations in the dialogues. For example, one state may represent a greeting to initialize the interaction, while other states represent requests for specific bits of information. Each state represents a specific situation in an interaction script and sets up specific expectations for the following actions. Actions by the user and the system trigger transitions between the states. Finite state based dialogue managers support strongly directed interactions and are somewhat inflexible.

The **frame based approach** represents is more flexible. It is particularly appropriate for dialogues in which the system needs to elicit specific information from the user in order to formulate a query to a data base or to formulate an order. The state of the dialogue is represented by a frame structure that the system fills out using keyword spotting and pre-defined questions. It is more flexible in that the interaction is not driven by predefined scripts but is able to use information that is provided without prior prompt. This approach is less restrictive than the finite state approach. but appears to be primarily focused on information retrieval domains. It also poses higher demands on e.g. the natural language understanding components of the system since it provides fewer restrictions for the user's inputs and fewer expectations for the interpretation of the input.

The term **agent based approach** is used to describe approaches towards dialogue management in which the actions of the system are based on some sort of psychologically inspired model of the dialogue system and the human user as agents, where goals and intentions of the participants are explicitly modelled. The dialogue manager can for example use automatic planning to determine the actions of the dialogue system. The information state based approach to dialogue management (Traum & Larsson, 2003) presents a model of dialogue management in which the state of the system is modelled as the belief state of the dialogue agent, the agent's agenda, and a representation of the agents model of the state of the information shared with the other participants. Actions in this theory correspond to modifications of these structures. Agent based systems can be quite complex and pose high demands on the other components of the system. They are therefore not particularly widely used in practical applications.

More recently, approaches that are based on **statistical methods** have been applied to dialogue management. In the POMDP (Partially Observable Markov Decision Process) based approach (Young *et al.*, 2013) the dialogue is modelled as a stochastic process. Unlike in the finite state based approach, where the dialogue at each point in time has one fixed state, in the POMDP approach the system maintains a probability distribution over a set of states, which it updates based on a user's actions using the Bayes' theorem. The fact that this update is probabilistic makes it particularly robust towards errors related to the recognition of the user's intention that, for example, often arise from errors in the speech recognition component. One interesting aspect of POMDP based dialogue managers is that they can be trained using reinforcement learning with data from observed interactions.

Common to all the dialogue models is that they need to be informed about how interactions in the domain are performed. In the finite state based approach this is coded directly into the structure of the states and the transitions between them — interactions following pre-specified scripts. In the frame based approach dialogue is mostly controlled by the needs of the underlying frame structure — for example, some slots may specify that other slots need to be completed before them. In the agent based approach, examples from observed dialogues can be used to formulate the operators of the planning system. In statistical approaches, observed interactions can be used to train the system or to inform user simulations that are used for training. One part of our research is therefore focused on identifying sequences of actions that were used in the dialogues to successfully resolve problems in the dialogue that occurred due to errors in perception.

2.2.2 Dialogue Structure

The idea behind dialogue structure is that dialogue is not a linear sequence of separate utterances, but that each utterance relates to the utterances preceding it and the discourse so far in a specific way. Furthermore, dialogue also appears to have a structure in that interaction does not just proceed in a linear fashion, but may include interruptions and digression, yet still ultimately return to its original purpose. If a computer system engages in dialogue with a human user, it needs be able to understand how the actions by the user and its own actions relate to the context of the dialogue. It therefore needs to be able to not only understand the content of the contributions but also their relation to the overall structure of the dialogue.

At a low level, there are relationships between individual pairs of utterances that are adjacent (the second utterance follows directly after the first). Schegloff & Sacks (1973) use the term **adjacency pairs** to describe pairs of utterances where an utterance of a specific type by one speaker normally induces the second speaker to respond with an utterance of a corresponding type. For example, utterances in which the speaker expresses a *question* are typically followed by an *answer* type utterance by the other speaker.

Coulthard & Brazil (1981) describe a structure for information exchange which consists of three actions: an initiating move by the first speaker, a response move by the second speaker and a follow-up move by the first speaker that comments on the information provided in the response. Adjacency pairs and the structure for information exchange describe coherence between contributions to a dialogue at a **local** perspective.

Dialogue structure can also be understood from a more **global** perspective. At a global level, the structure of dialogues depends strongly on the purpose of the dialogue, and in task-oriented dialogues, strongly on the structure of the task. Orkin & Roy (2007) describes an online-experiment in which they use crowd sourcing to observe and learn structures of interactions in restaurant scenario in a video game engine. Their experiment primarily focuses on discovering typical sequences of actions in a given domain in order to inform an automated system.

The focus stacks model presented by Grosz & Sidner (1986) attempts to describe the structure of discourse at a more abstract level by relating the emergence and satisfaction of goals in the task with shifts in attention. Pieces of discourse (called **discourse segments**) give rise to **focus spaces**. Focus spaces represent the set of elements that are under discussion during a segment of the discourse. Each discourse segment has a purpose that represents the intentions the speaker intends to realize with the discourse segment. Critically, these purposes are interrelated, in particular the purpose of one segment may contribute to the completion of the purpose of another segment (and conversely, a segment purpose may depend on the completion of the purpose of another segment). These relations therefore describe how the elements of the discourse relate to each other in terms of the intentions expressed through them (the *intentional structure* of the discourse). If a new discourse segment brings up a new purpose that has to be completed before the purpose that is currently under discussion can be completed, the focus space associated with this segment takes precedence over the space of the current segment. After the purpose of the new segment has been completed, the discussion returns to the old discourse segment. The structure that stores the focus spaces is therefore modelled as a stack. A stack represents the introduction of a new focus space that supersedes the current space as the *push* operation. If the task that is associated with the topmost space is completed, the space is removed through the *pop* operation.

This structure where the current purpose of the dialogue is temporarily put aside and returned to after another problem has been solved, is often referred to as to as a **sub-dialogue**. Sub-dialogues are relevant to our work because they can be triggered by problems in the dialogue that prevent the dialogue from proceeding as intended. Sub-dialogues that serve to resolve a problem or to clarify an issue are referred to as **clarification dialogues**.

Sub-dialogues have previously been investigated in general dialogue and for human-computer dialogue systems. Purver *et al.* (2003) discuss different forms and interpretations of clarification questions in corpus data, under the perspective of clarification questions a dialogue system might be asked by a human speaker. Gabsdil (2003) discusses clarification questions in a frame based dialogue system and in a system that involves deeper analysis of the spoken content. They propose that the system ask the participant clarification questions to achieve a better grounding of ambiguous information (mostly arising from acoustic problems).

Clarification sub-dialogues arise from problems in the language based interaction. Such problems can arise due to divergences of the participants' understanding of the environment. One reason for such divergences can be perception errors. We therefore believe that perception errors may trigger clarification dialogues, and that clarification dialogues may be used to resolve problems that arise from perception errors. We therefore plan to identify and analyse clarification dialogues in this work.

2.3 Reference in Dialogue

One particularly interesting aspect of language in situated dialogue is the problem of **reference**. The term reference is used to describe the phenomenon that expressions in language represent concepts or objects. The expression that is used in an utterance to make a reference is called the **referring expression**, and the target of the reference is called the **referent** of the expression. References can be distinguished into three categories:

- A reference to an object that has previously been discussed in the dialogue is called an **anaphoric** reference. In the following example, the first sentence introduces a new reference (a tall man). In the second sentence, the **pronoun** "he" is used to refer back to this referent.

"A tall man walked into a shop. *He* bought a newspaper."

The pronoun "he" represents an anaphoric reference. The expression "a tall man" is called the **antecedent** of the anaphor.

- A reference that introduces a new object into the discourse (either from general background knowledge of the world the participants share or as an imaginary object, as in the previous example) is called an **evoking** reference. For example, if a participant mentions the name "Barack Obama", both participants are likely to know that the reference refers to the current president of the USA, even if he has not been discussed in the dialogue recently and is not physically present.

- A reference to an object that has not been discussed previously in the dialogue, but that is available in the spatial context of dialogue is called an **exophoric** reference. References of this type are particularly common in situated dialogues as discuss them in this work. For example, Figure 2.2 presents a scene from the Toy Block experiment (Chapter 3) in which a dialogue takes place. In this scene, the human user may give the following instruction to the robot:

  "Pick up *the green box.*"

  In this utterance, the term *the green box* is an example of an exophoric reference. In order to understand the instruction, the robot has to identify the object in the world that matches the description given by the user in the referring expression.

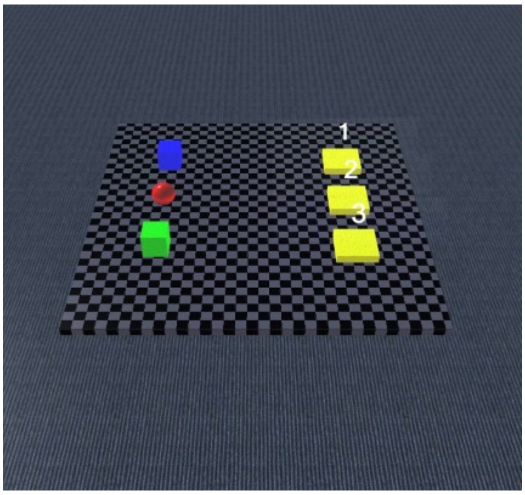


Figure 2.2: A scene in which an exophoric referring expression is interpreted.

A referring expression can be understood as a description that the speaker believes is appropriate for the listener to identify the intended referent object. In practical terms the description provided by a referring expression is often modelled as a set of attributes with given values. The expression “the green box” can, for example, be notated as follows:

$$\{\langle colour : green \rangle, \langle type : box \rangle\}$$

Referring expressions often involve descriptions that describe the referent in terms of other objects. These descriptions are called **relational descriptions**. They can be modelled as attributes whose values contain referring expressions themselves. The expression “the ball near the box” can be represented in the following way:

$$\{\langle type : ball \rangle, \{\langle near : \langle type : box \rangle\rangle\}\}$$

An attribute-value pair can also be represented as an attribute value matrix (AVM). Complex referring expressions that involve relations between objects are often more legible in this notation. The AVM for the expression “the ball near the box” looks as follows:

$$\begin{bmatrix} \text{type} & \text{ball} \\ \text{rel} & \begin{bmatrix} \text{reltype} & \text{near} \\ \text{relatum} & \begin{bmatrix} \text{type} & \text{box} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

We will use AVMs where space permits. Relational descriptions are often used to express **spatial relations** that describe the relative location of objects in relation to a landmark object (such as in previous example).

In the context of referring expressions the following terms are often used:

- **Target:** The intended referent of the reference.
- **Context set:** The set of all objects in the scene.
- **Contrast set:** The set of objects the target has to be distinguished from.

In this work referring expressions occur in specific roles in instructions and actions:

- **Patient:** An object that is being affected by an action (or, depending on the context, a referring expression that is used to describe the object that is affected by an action).
- **Destination:** A description of the targeted end point of a motion (or the referring expression that is used to describe the location).

The two main tasks a dialogue system that involve referring expressions are the **resolution of referring expressions** and the **generation of referring expressions**. In the following section we discuss them and review different approaches.

2.3.1 Resolution of Referring Expressions

Referring expressions in a situated dialogue can either be resolved anaphorically (i.e. as a reference to the prior discourse) or exophorically (i.e. with respect to the visual context). We do not discuss evoking references because they are rarely of relevance in the type of dialogue we investigate in this thesis.

We say that a referring expression is **distinguishing** if it can be resolved to one and only one only referent. We call it **ambiguous** if it can be resolved to more than one object, and **unresolvable** if it cannot be resolved to any object at all.

2.3.1.1 Anaphoric References

To resolve an anaphoric reference, the system has to identify the antecedent of the reference. A possible candidate as antecedent has to be *compatible* with the referring expression in terms of its attributes – the description provided in the referring expression has to match the antecedent. A first step towards determining the antecedent would therefore be to identify all previously mentioned entities with compatible attributes.

In order to do this, the system needs to keep track of all objects that are mentioned during the discourse. In a more general sense, the representation of the contents of a discourse is called a **discourse model**. However, the discourse may contain several references to objects that fit the description of the referring expression. The expression would therefore be ambiguous if only its attributes are considered. The system therefore needs to be able to distinguish which object is most likely to be the intended reference among all the possible referents based on other criteria.

One approach towards this problem uses syntactic properties of the discourse. In their **centering theory** Grosz *et al.* (1995) describe local coherence of sentences in short texts based on the co-reference relations between the entities mentioned in the sentences. Probable co-references are predicted based on the syntactic roles of the words denoting the entities under discussion. For example, entities that occur in the *subject* position of a sentence are seen as more likely to co-refer to the focused object of the preceding sentence than entities that occur in the *object* position. Based on the work by Grosz *et al.*, Brennan *et al.* (1987) present an approach to anaphora resolution by using the rankings produced in the centering approach to predict the referents of pronouns.

Hajičová *et al.* (1992) present a method for pronoun resolution that is similar to the centering approach in that it is also based on a notion of linguistic prominence or salience. The salience in this case however is not directly calculated based on the syntactic role of the expressions, but on the focus status of the objects.

In general, early approaches towards reference resolution focus on reference resolution in written text, and not on dialogue in particular. Byron (2002) present rule based resolution of pronouns in dialogue. One particular problem of pronoun resolution in dialogue is that pronouns more often than in regular discourse have antecedents that are not objects that are introduced as noun phases. Instead pronouns are often used to refer to entire sentences or verb phrases. Strube & Müller (2003) present an approach towards pronoun resolution in spoken dialogue that is based on machine learning (decision trees).

The **reference domain** approach (Salmon-Alt & Romary, 2009) is a cognitively inspired approach towards reference that models a number of phenomena related to reference well. It is also suitable for anaphoric references and has been used in dialogue systems (Kelleher *et al.*, 2005).

2.3.1.2 Exophoric References

The resolution of exophoric referring expressions poses different challenges than the resolution of anaphoric references. In the anaphoric domain, the references are resolved against the so called **discourse model**. Its actual implementation depends on the specific theory, but in general it is a symbolic structure that is constructed based on information that is recorded during the dialogue. It is therefore relatively unproblematic to compare a referring expression with a record of the information available about an object that has been discussed.

In the exophoric domain, this is not possible since the objects are not present as language-derived symbolic information, but as phenomena the system observes in the environment. In order to resolve referring expressions, it is therefore necessary to establish a link between the symbolic representation of the language-based attributes in the discourse and the phenomena observed in the environment. This problem is known as the **symbol grounding** problem (Harnad, 1990).

Coradeschi & Saffiotti (2000, 2003) describe the process of establishing and maintaining a link between the symbolic contents of language and the perception of a robot as the **anchoring problem**. They describe a *predicate grounding relation* that associates symbol from the discourse representation and planning with attributes that can be quantitatively measured in the percepts of the system. The attributes of the symbolic object then determine values the percept may have that would be acceptable for the concept to hold. Roy *et al.* (2002) describe a vision system that learns to associate vision based attributes with words it receives from descriptions generated by human, and also learns a grammar. Gorniak & Roy (2004) describe a system in which reference resolution is based on grounded expressions, and in which the composition of expressions is based on visual processes as well.

In most approaches towards visual reference resolution the context for the resolution of an expression is taken to be the current visible scene. Determining the referent therefore is mostly based on selecting an object from the scene that fits the attributes contained in the referring expression. As in the anaphoric domain, this filtering approach alone is often not sufficient to single out the intended referent because multiple fitting objects may exist in the scene.

One possible strategy is to identify which objects in the context are particularly **visually salient** (Itti, 2007) (an approach towards computational modelling of visual salience is discussed in (Itti & Koch, 2001)) in their own right or in the focus of attention of the speaker. Kelleher & van Genabith (2004) present an approach to resolve ambiguous references based on the visual salience of objects based on the (apparent) size of objects and on their proximity to the center of attention (the center of the screen).

In a situated dialogue, often a trade-off between linguistic salience and visual salience has to be found. Kelleher (2006) presents an approach towards reference resolution that integrates both visual and linguistic salience. Some early systems (e.g. (Wachsmuth & Cao, 1995)) employ a notion related to linguistic salience, but do not use visual salience. Instead they ask the human speaker to clarify their references through a clarification dialogue.

While the task of resolving referring expressions is one problem, the complimentary problem is the production of referring expressions. We discuss this problem in the next section.

2.3.2 Generation of Referring Expressions

As discussed earlier, a spoken dialogue system presents its reactions to the user as spoken language. It therefore needs to be able to translate the internal representation of intentions and content into natural sounding text. This is the problem that is addressed in the field of Natural Language Generation (NLG) (Reiter & Dale, 2000).

There are a number of different approaches towards NLG, and dialogue systems have also used a number of different approaches. For example, Becker (2006) use a so-called template based approach, where surface utterances are composed from pre-formulated pieces of text in the context of the SmartKom dialogue system (Wahlster, 2006). Foster *et al.* (2005) use a grammar based approach, where text is generated through a generator that uses structural language rules to create a text based on a high-level semantic specification in the COMIC dialogue system[1].

One important sub-problem of NLG that is particularly relevant to the work presented in this thesis is the decision of how to refer to objects in the domain. The system has to decide whether to use a pronoun to refer to an object (and which pronoun), which attributes to include in a noun-phrase based description, and – in a multi-modal domain – whether to use additional visual cues and how to combine them with the description. Following the Gricean maxims (Grice, 1975), Reiter & Dale (2000) propose three requirements that need to be taken into account when generating a referring expression:

- **Adequacy:** The expression has to provide enough information such that the listener is able to identify the intended referent.
- **Efficiency:** The expression should not contain unnecessary information.
- **Sensitivity:** The expression should make use of properties the addressee can evaluate, and avoid properties they cannot evaluate.

[1] `http://groups.inf.ed.ac.uk/comic/`

The last point is particularly interesting for the work in this thesis. Whether or not a speaker believes that a listener is able to interpret an expression depends on the speaker's understanding of the listener's understanding of the scene. While in most cases speakers may assume that they have a shared understanding, this may lead to issues if the perception of one of the participants is affected by errors.

The **full brevity algorithm** (Dale, 1989) presents an approach that always aims to generate the shortest distinguishing description as the referring expression for an object. It thereby maximizes the efficiency of the expression, but the calculation may turn out quite expensive in more complex scenes.

The **incremental algorithm** (Dale & Reiter, 1995) abandons this strict efficiency goal and instead bases its selection of attributes on a psychologically based order of preference. Dale & Haddock (1991) and (Kelleher & Kruijff, 2005) are examples of algorithms that generate descriptions for attributes that contain references to other objects. This is potentially useful for situated dialogue, because it enables the system to generate descriptions that involve (spatial) relations between the objects in the scene. Van Der Sluis (2005) presents an algorithm that includes the use of pointing gestures in multimodal referring expressions.

One common problem of these approaches is that they assume that both participants have a shared understanding of the environment. The incremental algorithm technically includes a provision to ensure that attributes used are actually shared between the speaker and the listener, but this aspect is not particularly elaborated upon.

Horacek (2005) presents an algorithm for the generation of referring expressions under conditions of uncertainty. The algorithm is an extension of the incremental algorithm that attempts to model the probability that a candidate expression would be resolved to the intended referent. It also keeps track of potential distractor objects. Three sources of uncertainty are explicitly identified:

1. **Uncertainty about knowledge:** Whether or not the addressee knows the terms that could be used in a referring expression.

2. **Uncertainty about perception capabilities:** Whether or not the addressee perceives the intended referent.

3. **Uncertainty about conceptual agreement:** This source of uncertainty arises from vague attributes.

The problems we address in this work are best represented in the second point and the third point (in a more general sense). However, as Krahmer & Van Deemter (2012) discusses, it is not clear how these uncertainties could be estimated.

2.4 Understanding in Dialogue

A dialogue is not simply a sequence of statements put forward by its participants, but an interactive and constructive process in which information is exchanged and negotiated. Each contribution has to be interpreted in the

context of the dialogue, and it also contributes to the context. Jurafsky & Martin (2000) use the term **discourse context** to describe a body of information that the participants share and construct during the course of the dialogue through their discussions. This is contrasted with the **situational context** which is used to describe information that is available through the environment in which the dialogue takes place. An important property of contextual information in a dialogue is its **sharedness**. If a speaker intends to formulate an utterance that presupposes a certain piece of background information in order to be understood, they need to be reasonably certain that the addressee possesses all the background information necessary to correctly understand the contribution.

The term **common ground** is used to describe information that is shared between the participants. The process of entering information into the common ground is called **grounding**. In the following we are going to discuss grounding in Section 2.4.1. We then are going the discuss misunderstandings as a result of grounding problems in Section 2.4.2.

2.4.1 Grounding

The term **grounding** is used to describe the process that is used to ensure that a piece of information that one participant shared with another participant in a dialogue is accepted by both participants, and that it is mutually known to participants that the other participant knows that they have accepted the information as shared.

Information can enter a *grounded* state in different ways. Dialogue research is primarily concerned with grounding that is achieved through dialogue. For example, Clark & Schaefer (1989) describe grounding in a dialogue as a sequence of two phases. In the first phase, the **presentation** phase, the speaker presents their contribution. In the second phase, the **acceptance** phase, the addressee reacts to the contribution by displaying evidence of understanding or evidence of non-understanding. The speaker then may present evidence that they accepted the addressee's acceptance.

Apart from grounding through dialogue, grounded information can be derived from other sources. For example, Clark (1996) discuss the **communal common ground** – a set of information members of groups *a priori* assume as shared. For example, people living in New Zealand are likely to have mutually shared knowledge of local geography and politics, while rock musicians are likely to have shared knowledge of music and bands. Importantly for the work presented in this thesis, common ground can also have **perceptual basis**. According to (Clark, 1996) an event becomes perceptually shared if the participants both perceive it and attend to it. This demonstrates that if objects are present in the context, and participants are likely to believe that the other participant is attending to them, they enter a grounded state. If this is not the case, and the other participants does not attend to the object in question, or perceives it differently from the first participant (e.g. due to perception errors), this can lead to situations where the participants incorrectly assume to have a common ground. This can lead to misunderstandings in the dialogue.

2.4.2 Misunderstandings

The goal of the process of grounding is to make sure that both participants of the dialogue mutually understand each other. Problems in this process can lead to situations in which communication does not directly result in a shared understanding. If participants act on a false assumption of shared understanding problems can arise in the dialogue.

Paek (2003) discusses different types of communication errors and their possible effects on the communication. He proposes a classification of errors based on the "joint action ladder" described by Clark (1996), and classifies different communication problems presented by different authors in it. Some of the problems are explicitly referred to as misunderstandings in the original works, while others can be seen as misunderstandings in a wider sense. The joint action ladder describes coordination between participants in performing joint actions. While it refers to joint actions in general, which may comprise complex actions such as ordering beverages or making a bet, in the context of dialogue it can refer to actions as simple as suggesting a piece of information to someone.

A **communicative act** is modelled as a process that requires coordination at 4 levels. Successful coordination at each level presupposes coordination at the next lower level. Conversely, evidence for coordination at one level can be seen as evidence for coordination at all lower levels.

The lowest level is the **channel level**. Success at this level is achieved if the sender (A) produces a behaviour and the addressee (B) attends to this behaviour. The behaviour can be any communicative behaviour such as a hand gesture, a directed gaze or, in the case of dialogue, speech. If coordination fails at this level B simply does not notice A's actions as an attempt to communicate.

The second level is the **signal level**. Success at this level is achieved if B recognizes A's behaviour as a signal. In the case of speech based communication this means that B recognizes the sounds A produces in speaking as speech. If coordination fails at this level, a very fundamental type of communication error occurs where B notices that A tried to communicate, but was not able to extract a message. In this case B is clearly aware that A tried to communicate, but that the attempt was unsuccessful.

The third level is the **intention level**. At this level B has to extract the intention A tried to communicate. If B is unable to extract the intention A attempted to communicate, B will either be unable to extract a sensible intention or extract an intention that is sensible to B but different from the one intended by A. In this case, B may be aware or unaware that the process has failed.

The fourth level is the **conversation level**. Communicative success at this level entails that B has recognized A's proposal and is able to consider it. Errors at this level generally involve problems involving mismatched understanding of concepts or inference over the contents of the discourse.

Errors during the interpretation of the intention and the consideration of the proposal can lead to situations in which the listener achieves a different understanding of the joint activity than the speaker (possibly without noticing).

We argue that an error that is based on false perception by B can be understood as a breakdown at the intention level of the joint action ladder. In the next section we motivate this position.

2.4.3 Misunderstandings and Perception

As stated previously, the environment presents a context and a possible source of information for the dialogue. Misunderstandings pose a particular problem in this context. While the process of grounding new information in a dialogue is naturally included in normal conversation as a cooperative action, the process of perceiving information from the environment is necessarily separate for each participant.

We believe that an error that is based on false perception can be located at the intention level. In particular we believe that these errors match the descriptions of misunderstandings given by Hirst *et al.* (1994). The listener believes that they have correctly understood the speaker's intention. In fact they actually have, but only as far as the speaker's intention as it is contained in the spoken utterance. However, since the participants, due to diverging perception, have different perception of the situational context, they actually may arrive at different interpretations of the proposed activity.

2.5 Computer Vision

If a situated dialogue system is to be able to discuss the visual context of the dialogue it first needs to be able to perceive and understand this context. The area of research that is concerned with computer based visual perception is called **computer vision**. In this section we will attempt to briefly highlight which topics in computer vision are relevant for this work, why we believe that errors in computer vision may be a problem worth addressing, and the different types of visual contexts that are used in situated dialogue systems. For this work we are interested in two major topics of computer vision:

1. The system detects if objects are present and then determines their location (**Object detection**).
2. The system then needs to detect the properties of the objects (**Object classification**).

We focus on two possible errors that may arise from the outcome of these tasks:

1. The system fails to detect that an object is present.
2. The system correctly detects the object but does not recognize its properties correctly.

The first type of error may occur if the system does not perform the object detection step correctly. For example, the system may be unable to distinguish an object from the background, or it may be unable to correctly detect two objects as separate objects if they are placed adjacent to each other. The second type of error may occur if the system experiences problems related to object classification. Other types of errors are possible. For example, a system may erroneously detect an object where there actually is none. However, in this work we focus on the two errors mentioned here.

Errors in computer vision can lead to problems in the reference resolution process, which in turn can lead to problems in dialogue. If a speaker intends to refer to an object that the system does not perceive due to a perception error, the system will not be able to resolve the reference this may lead to a problem in the dialogue where, e.g. the system will not be able to perform an instruction involving the object, and it will not be able to answer questions related to it. In fact, it is possible that it resolves the reference to an object other than the one the speaker intended. In this case the system would not even be aware that there is a problem, and complex misunderstandings can arise.

Reference resolution algorithms typically involve comparing a set of attributes that was used in a referring expressions to a set of attributes that the computer vision system believes are valid to an object it perceives. If the vision system classifies one or more of the attributes incorrectly, problems can arise in the reference resolution process and subsequently in the dialogue.

In order to avoid problems of this type, a situated dialogue system requires a vision system that minimizes object detection errors and property classification errors.

Computer vision has made good progress in the recent years, but is still not perfect. In fact, Mitchell (2012) states "it basically does not work" (unless restricted to specific domains).

To determine the state of the art, we looked at the results of a shared task challenge for computer vision, namely the PASCAL Visual Object Challenge (Everingham *et al.*, 2015). In this challenge participants were encouraged to compare their computer vision systems against each other by comparing their performance on a given set of example images. Three tasks were of interest for our work:

- **Classification:** To determine whether or not an object of a given class was contained in a given image
- **Detection:** To determine the rough location of a given object in an image via its bounding box
- **Segmentation:** To determine the location of objects in the image on a pixel basis

The up-to-date leaderboards for the currently most successful systems are available online[1]. At the time of writing[2], the most successful system in the classification category achieved an average precision of 85.4% using the default training data. The most successful system in the detection category achieved an average precision of 42.2%, and the most successful system in the segmentation category achieved an average precision of 47.5%.

[1] `http://host.robots.ox.ac.uk:8080/leaderboard/main_bootstrap.php`

[2] 15.09.2015

Liu *et al.* (2012) investigate reference resolution in human-computer dialogues in a scenario that is not dissimilar to the one addressed in this thesis. They use a computer vision system that misclassifies the type of an object in 84.7% of the cases and produces a segmentation error in 10.2% of the cases[1].

Based on these figures, we believe it is reasonable to conclude that computer vision is not necessarily perfect in all instances, and that errors in perception may therefore occur in situated dialogue systems and lead to problems in the interactions.

2.5.1 Visual Context in Dialogue Systems

Situated dialogue may take place in different kinds of environments, ranging from actual physical spaces to purely virtual simulations. Depending on the type of the environment, the visual context poses different sets of challenges to the system.

If the visual context of the interaction is simulated by the dialogue system, the system developer can assign fixed attributes to the objects that populate the world (e.g. the developer can determine which objects may be referred to as *boxes*, and which objects the system believes have the colour *green*). We identify three levels of abstraction:

Symbolic world: The world is rendered by the system based on a symbolic definition. The system's understanding of the world is directly based on this representation. An example for system is (Winograd, 1971).

[1] This is a very high number of misclassifications. It nevertheless shows that other researchers have addressed dialogue in the context of very poor perception.

Synthetic vision: The world is simulated, and some aspects of the vision (e.g. visibility detection or visual salience) are based on a simulation of actual vision using techniques such as false colouring or ray casting. Other aspects, such as object recognition or the recognition of properties, are abstracted away ((Noser *et al.*, 1995) (Kelleher, 2003, 2006) (Kuffner & Latombe, 1999)) and the system is directly supplied with information from the world definition.

Artificial vision: The world is perceived purely in a visual way (e.g. through a camera), and the system has only access to what information it can extract from the image ((Kruijff *et al.*, 2006a; Sjöö, 2011)).

2.6 Spatial Attributes

Spatial descriptions are particularly important in situated dialogue. Liu & Chai (2015) report that in an object naming task, spatial attributes were the third most frequently used description strategy after the *type* and the *colour* attributes of an object (443 expressions contained a spatial relation, while 686 contained a type attribute and 747 contained a colour attribute).[1] Summers-Stay *et al.* (2014) report that in their human-robot navigation experiment "much of the discussion involved spatial language pertaining to objects configurations".

This indicates that participants do use spatial attributes in situated human computer dialogue. In order to be able to successfully participate in

[1] Unfortunately they do not state the proportion of referring expressions in the total data set for each attribute type.

situated dialogue, a dialogue system therefore has to be able to deal with spatial attributes. In order to understand referring expressions with spatial attributes used by the human speaker, and in order to be able to use spatial descriptions in a felicitous way, the dialogue system must be able to decide when a spatial relation holds. In the terminology used in the symbol grounding area, the system needs a predicate grounding relation for spatial attributes.

One major class of spatial attributes are **spatial relations**. They are expressed in language as **prepositions** such as *to the left*, *in front of*, or *between*, and they are used to describe the location of one object in relation to another object. The object that serves as the point of reference is often referred to as the **landmark**. A predicate grounding relation for spatial relations can be implemented through **spatial templates**. Spatial templates are geometric objects that are anchored to landmark objects and which then project a set of fields that describe how well objects in each field are suitable to describe the landmark with the given relation.

For example, Figure 2.3 shows a spatial template for the preposition *above* (Kelleher & Costello, 2005). This template distinguishes three types of regions:

- **Good:** Objects in this region are likely to be perceived as *above* the landmark.

- **Acceptable:** Objects in this region may also be perceived as *above* the landmark.

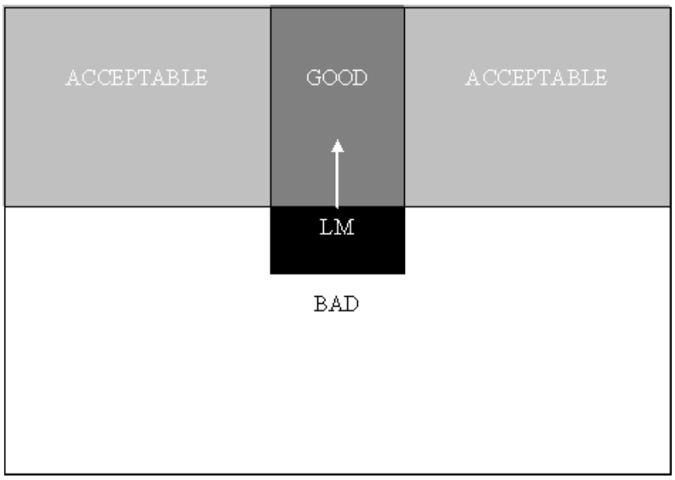


Figure 2.3: A spatial template for the spatial relation *above*.

- **Bad:** Objects in this region will probably not be considered *above* the landmark.

There are also approaches that use templates that work with a continuous notion of appropriateness rather than a discrete one (Costello & Kelleher, 2006; Logan & Sadler, 1996; Regier & Carlson, 2001).

One important aspect of relations and their associated template is the **frame of reference**. The frame of reference describes from what perspective the relation is to be interpreted, which determines in what way the spatial template is attached to the landmark object. Following (Levelt, 1996) we can distinguish three frames of reference:

- **Deictic:** (or the **viewer-centered perspective**) The relation is interpreted from the point of view of the observer.

- **Intrinsic:** The relation is interpreted from the orientation of the landmark object. This perspective is relevant for objects that have an intrinsic front and back such as chairs, cars or dogs, but less so for

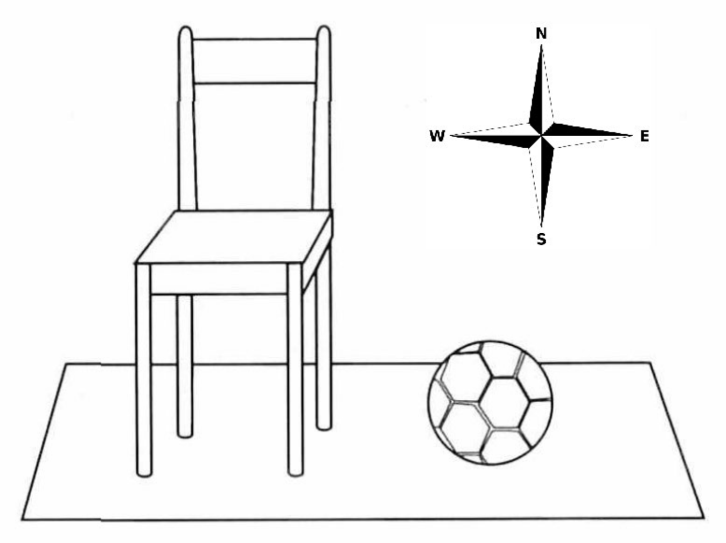


Figure 2.4: A chair and a ball.

objects that do not (such as balls or apples).

- **Absolute:** The relation is interpreted in terms of some global frame of reference that is independent of the perspective of the view or the landmark. An example of this would be cardinal directions on a map.

Figure 2.4 contains an example image (adapted from (Levelt, 1996)) that illustrates the three perspectives. From the deictic perspective the ball would be *to the right* of the chair. From the intrinsic perspective, the ball would be *to the left* of the chair (the chair is facing the viewer, the ball is therefore on its left). A relation from an absolute frame of reference might specify that the ball is *to the east* of the chair (as indicated by the compass in the upper right corner).

A discussion of the use of spatial templates for relational expressions can be found in (Kelleher & Costello, 2009).

2.7 Perception Based Uncertainty in Dialogue

Uncertainty in dialogue has been addressed by other authors previously. For example, the POMDP based approach to dialogue management discussed earlier represents an approach towards integrating uncertainty about the user's intention into the dialogue management. We were not able to find any instances of POMDPs being used to explicitly address the problem of vision based uncertainty. It would appear consistent to model uncertainty about the referent of an expression that may arise due to problems in robot perception, as part of the uncertainty about the intention of the user. There is also work on monitoring processes in videos using POMDPs (Hoey *et al.*, 2010) which seems applicable.

A number of authors have addressed uncertainty in dialogue that arises from uncertainty in visual perception, in particular in relation to reference. Liu and his colleagues performed a number of experiments about unreliable computer vision in dialogue. In (Liu *et al.*, 2012) they performed an experiment where two participants perceived the same scene and where asked to perform an object naming task. The second participant's view of the scene was filtered through an object recognition system and therefore inaccurate. They performed a number of different analyses based on this data that mostly involved some probabilistic matching between graphs representing the output of a computer vision system and the contents of the dialogue. Particularly Liu *et al.* (2013) is interesting because they investigate collaborative structures in the dialogues to enhance reference resolution. In (Liu & Chai, 2015) they discuss an experiment in which they used the same graph based approach to enable a robot to learn weights for a word grounding model in an object naming task. After they manipulated the robot's vision mechanisms to produce errors, the robot learnt which concepts were reliable and which were not.

Mast & Wolter (2013) present an architecture for grounded reference. In addition to the discriminatory power of expressions, they model the notion of **acceptability** that describes in a probabilistic manner whether or not the human user would accept a description for an object. In (Mast *et al.*, 2014) they describe the probabilistic grounding of concepts in the context of a dialogue system. The system observes the behaviour and uses clarification questions to dynamically determine how the user interprets attributes in the current context.

(Kruijff *et al.*, 2006b) describe a system for *human-augmented mapping* (a task where a robot creates a map of an environment and interacts with human partner to add additional information to the map) that attempts to detect problems in its geometric model of the world, and uses a dialogue interface to ask a human partner clarification questions to resolve the problems.

2.8 This Work

As discussed in the previous sections, there has been some work on uncertainty in dialogue and also on mismatched perception in dialogue. What we are planning to address with this work, and what we believe has not been addressed by other researchers so far, is the question of how human participants in a dialogue with a robot react when a perception based problem exists, what strategies they employ to resolve these problems, and how these resolution strategies are affected by different types of information that is available.

We believe that the findings of this work may be useful in the design of dialogue systems in that they may inform ways a system may assist a dialogue partner if it believes that perception errors have occurred and in that it may inform strategies that may be employed by the system itself to resolve problems at a later stage.

2.9 Summary

In this chapter we discussed topics related to dialogue and in particular situated human-computer dialogue that relate to the issue investigated in this work. We are particularly interested in how perception influences the content of dialogue, and in errors in perception that lead to problems in the dialogue. We will however not attempt to modify the robot's understanding of predicates, or the robot's model of the world, but focus on investigating how the users attempt to resolve the problems. We see this as an alternative approach that may complement other methods.

We discussed situated dialogue systems because the main experiment discussed in this thesis centres around a situated dialogue system. We discuss this system in Chapter 3.

We discussed dialogue acts and dialogue structures. In Chapter 7 we describe how we identify situations in which participants encounter a problem that is due to perception errors in the dialogue, and describe the actions by the participants at a dialogue act level, and attempt to describe dialogue structures.

We discussed reference and referring expressions. In Chapter 8 we investigate the referring expressions that were used when participants resolved perception based problems.

In Chapter 8 we investigate the referring expressions in unsuccessful reference at the beginning of problem resolution sub-dialogues and at their successful conclusion.

Chapter 3

The Toy Block Experiment

In an earlier set of experiments (Schütte *et al.*, 2012) we investigated the effect of diverging perception in a situated human-human dialogue. This experiment was based on data from the Map Task corpus (Anderson *et al.*, 1992). In the Map Task experiment one participant, the instruction giver gave navigation instructions to a second participant, the instruction follower. Both participants were given separate maps of the same territory. The instruction giver's map contained a route, and their task was to instruct the follower to recreate the route on their map. The maps contained a set of landmarks. Crucially, there were differences between the maps that were given to the instruction giver and the maps given to the instruction follower. For example, landmarks present on one map were missing on the other map, or landmarks were named differently between the maps. We investigated how the follower reacted to new landmarks the instruction giver introduced, and investigated how they managed to navigate landmarks that were not mutually visible. We extracted the actions the instruction giver and follower performed after such an introduction, and presented a model that could be used as a basis to model such dialogues in a dialogue system.

However, we ultimately found that the Map Task data was of limited use for our investigation because the number of relevant examples was smaller than expected and the interactions did not transfer particularly well to the current domain. We therefore set out to perform an experiment that was more strongly focused on the issue of perception problems in dialogue — the **Toy Block experiment**. In this chapter we describe the set-up of the experiment. In Chapter 4 we describe the dialogue system that was used in the experiment. In Chapter 5 we analyse the results of the experiment at a high level. In Chapter 6, Chapter 7 and Chapter 8 we analyse different aspects of the experiment in more detail.

3.1 The Experiment

The goal of the experiment was to observe how human users reacted to problems in a dialogue between a human robot operator and a robot that was prone to perception errors, and to observe if and how the users were able to resolve the problems caused by these perception errors. We had three main concerns while developing the scenario for the experiment:

- The scenario needed to feature opportunities for the robot to interact with objects in the environment.

- The scenario needed to be plausible, i.e. it needed to be clear to the participants that they were interacting with a robot, and that it was possible that the robot had problems with perception.

- In addition to introducing problems the scenario should also include options to resolve the problems.

We decided to use a scenario in which one participant instructs a robot to rearrange a set of objects into a given configuration. Situated dialogue about manipulating objects is a well-established domain. In a sense the scenario in this experiment is similar to the classic SHRDLU system (Winograd, 1971). Since then there have, of course, been many more recent approaches towards dialogue based object manipulation. For example, Kelleher *et al.* (2005) present an experiment in which participants interact with a dialogue system to modify the properties of objects in a simulated world and Knoll *et al.* (1997) present a system in which a human user instructs an actual robot to assemble a toy airplane.

We prototyped the scenario using a set of toy blocks (Figure 3.1). We then iteratively refined the task and implemented a simulation based experiment system that uses a dialogue system for interaction. As an early step in the implementation we compiled a corpus of instructions through an online survey. Participants of the survey were presented with two images that showed configurations of objects and were asked to produce instructions for a robot to transform the first configuration into the second configuration. The instructions collected in this corpus informed our implementation by

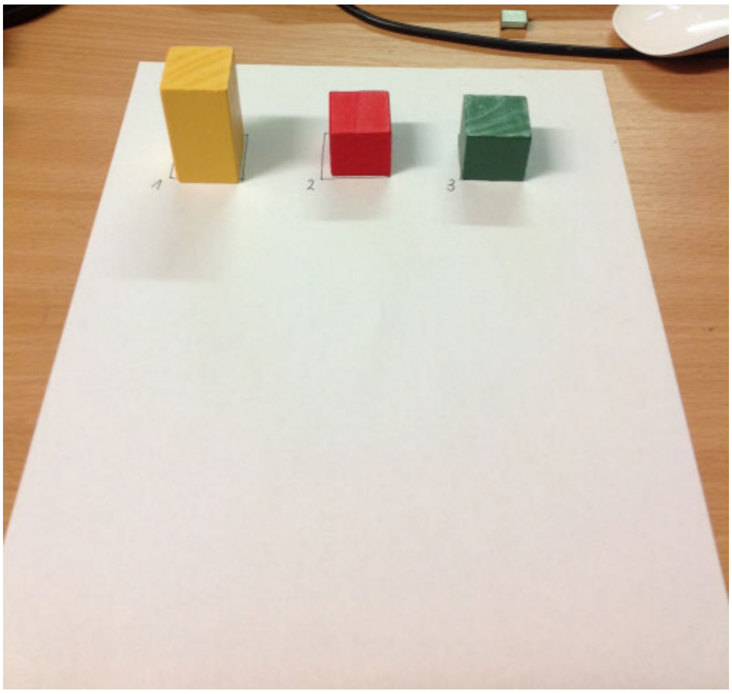
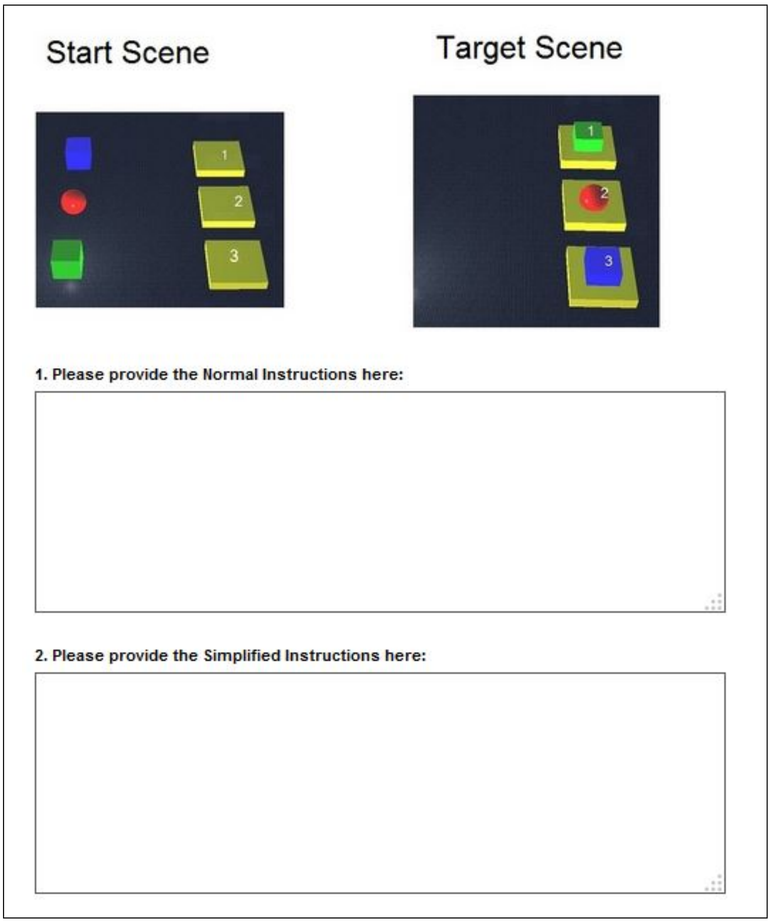


Figure 3.1: The prototype experiment setup.

determining the scope of linguistic expressions the system had to cover and by informing the types of spatial concepts the system would be likely to encounter. An example of one of the tasks from the online survey is shown in Figure 3.2.

After the first version of the system had been implemented, we performed a small and informal pilot study using close colleagues as participants. The purpose of this study was to test for any bugs that had not been discovered so far, and to collect suggestions for further improvements. The participants were asked to take notes during the experiments and interviewed afterwards. Based on the pilot study we integrated further spatial expressions into the system and clarified the instructions that were given to the participants.

In final version of the experiment, the participants were presented with a set of objects that were arranged on a surface in a simulation world (called the



1. Please provide the Normal Instructions here:

2. Please provide the Simplified Instructions here:

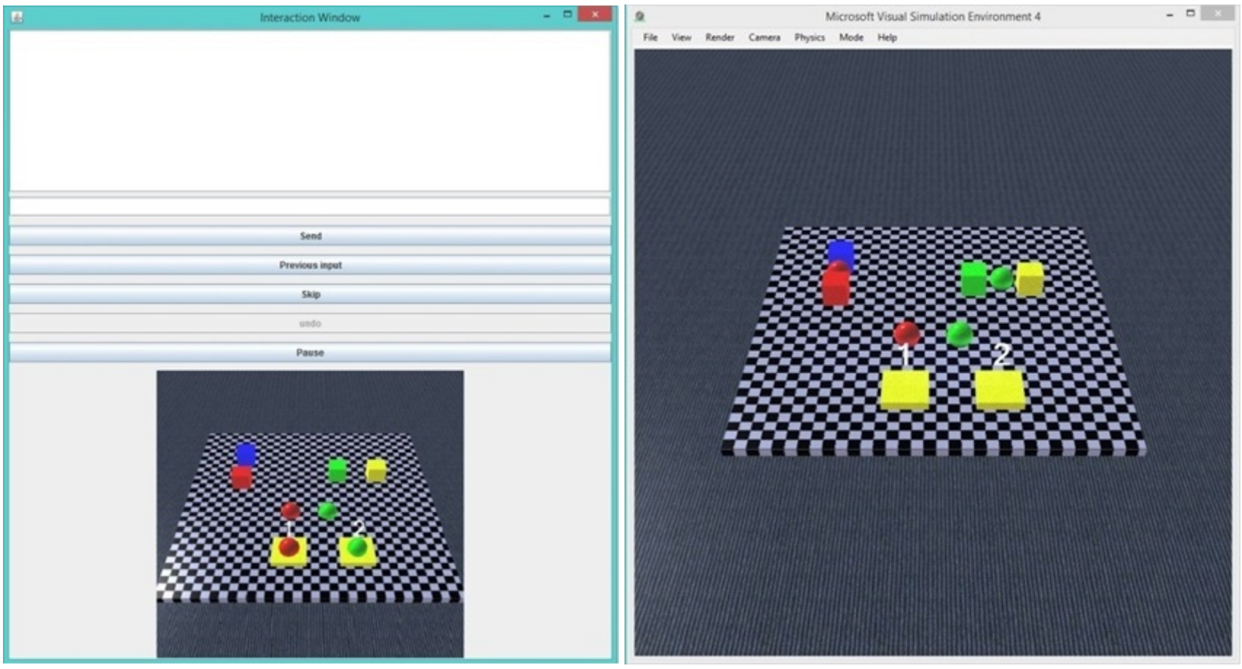
Figure 3.2: A task from the online survey.

scene). The scene is presented through the **simulation window** (Figure 3.3b). The robot was described to the participants as a manipulator arm that could move objects around the scene, but that was not visible in the simulation.

The participants interacted with the robot through a text based natural language interface, shown in Figure 3.3a. The upper part of the interface contained a text based chat interface and a set of buttons. The lower part of the interface showed an image of the **target scene**, i.e. the configuration of objects the participants was asked to recreate in the scene. Participants sent instructions to the robot by entering them in the chat interface. The robot then responded through a text reply in the interface and an audible reply in synthesized speech.

Participants were presented with a series of scenes. After a participant had successfully transformed the scene into the target scene, the system loaded the next scene. In order to investigate different possible approaches to resolving the problems in the dialogue, the experiment was split up into five different **phases**. In four of them, errors were introduced into the robot's perception. In three of these phases, the system offered different ways to the participant to access the robot's understanding of the scene.

Overall the experiment was composed of 5 phases. In the first phase, the **No Error Phase**, the robot works as intended and, unlike the later phases, no errors are introduced into the robot's perception of the scene. The purpose of this phase was to establish a baseline for the difficulty of the task and to record the behaviour of the participants in the absence of errors.


Interaction Window
Send
Previous input
Skip
undo
Pause
1
2
Microsoft Visual Simulation Environment 4
File View Render Camera Physics Mode Help
1
2

(a) The interaction window.

(b) The simulation window.

Figure 3.3: The user interface.

In the **Error Phase** the participants attempt the same set of scenes as in the No Error Phase. This time however, errors are introduced into the robot's perception of the world. The purpose of this phase was to establish the effect of the errors on the difficulty of the task and to record the behaviour of the participants when problems occurred due to the robot's perception and the strategies they use to try to mitigate the problems.

In the remaining phases the participants attempted the same scenes as in the Error Phase with the same errors, but were provided with different options that allowed them to access the robot's understanding of the scene. In the **Description Phase**, the participants can request a **scene description**. The system then verbally describes how it believes the objects are arranged in the scene. The Description Phase represents a uni-directional linguistic approach towards helping the users to resolve the problems in the dialogue.

Phase	Condition	Purpose
No Error Phase	No errors	Baseline
Error Phase	Errors	Impact of errors
Description Phase	Errors + Description	Verbal assistance
Markup Phase	Errors + Markup	Visual assistance
Querying Phase	Errors + Querying	Interactive language based assistance

Table 3.1: An overview of the phases.

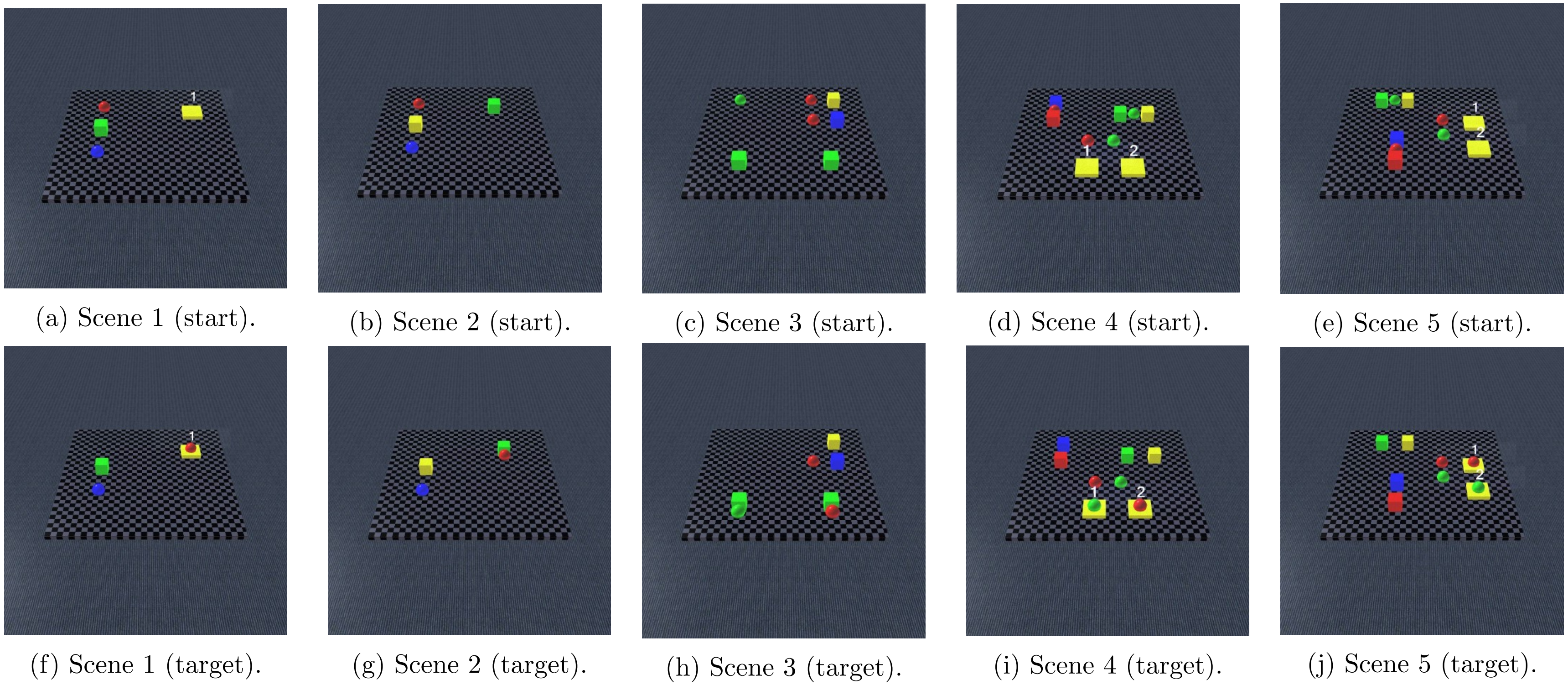
In the **Markup Phase** the participants could ask the system to **mark up** its understanding of the scene in the simulation view. The system then highlights every object it is aware of and annotates what type and colour it believes the object to be. The Markup Phase represents a visual approach towards resolving the problems in the dialogue.

In the **Querying Phase**, the participants could ask the system questions about its understanding of the scene. The Querying Phase therefore represents an interactive, dialogue-based approach. A summary of the different phases is given in Table 3.1.

3.2 Scenes

In this section we describe the scenes that were used in the experiment, what the design considerations were, and how we reached the set of scenes that was used in the experiments. All phases in the experiment used the same set of scenes. In total there were 20 scenes. 14 of the scenes contained errors in the Error Phase, the Description Phase, the Markup Phase and the Querying Phase. The scenes were each manually designed to elicit specific types of referring expressions and to accommodate perception errors. Figure 3.4 to

Figure 3.7 contain images of the start scenes and the corresponding target scenes.



(a) Scene 1 (start).

(b) Scene 2 (start).

(c) Scene 3 (start).

(d) Scene 4 (start).

(e) Scene 5 (start).

(f) Scene 1 (target).

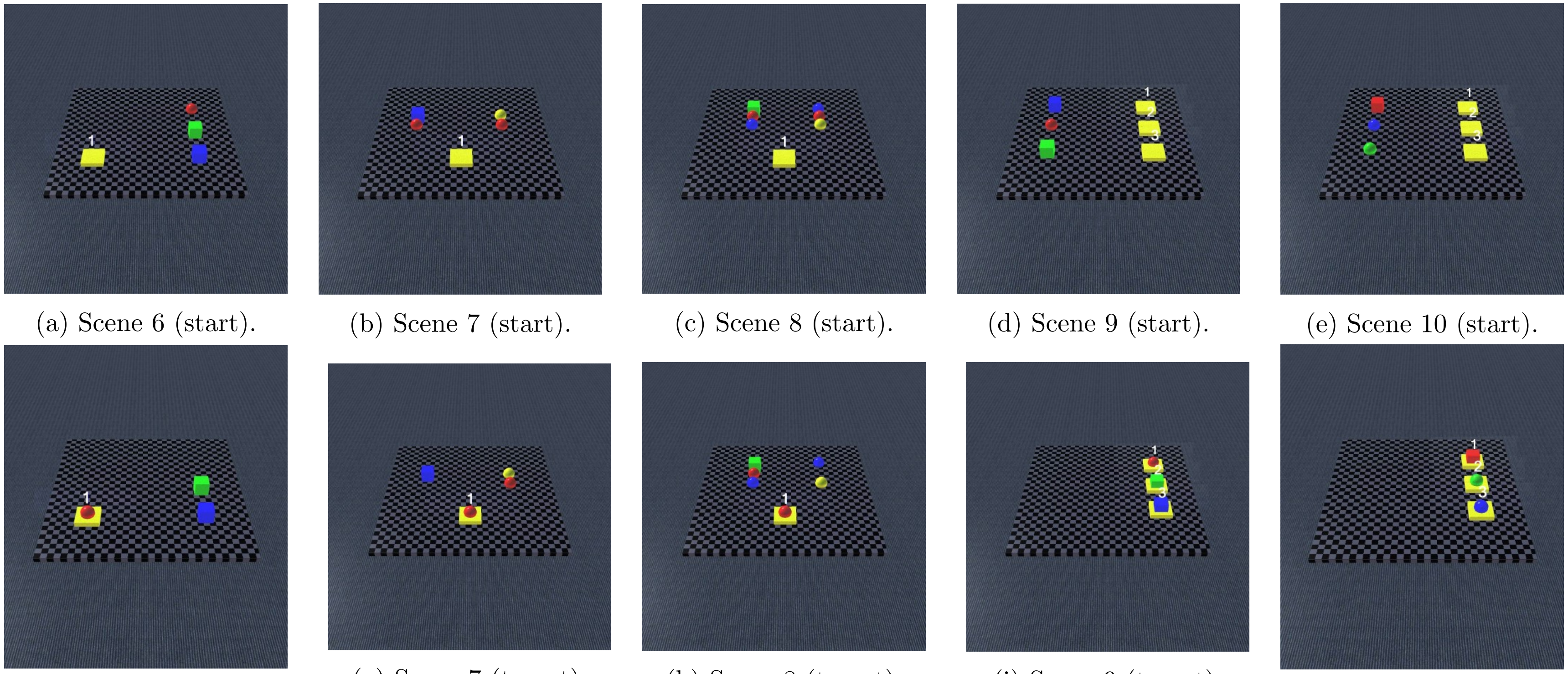
(g) Scene 2 (target).

(h) Scene 3 (target).

(i) Scene 4 (target).

(j) Scene 5 (target).

Figure 3.4: Start and target scenes for Scene 1 to Scene 5.



(a) Scene 6 (start).

(b) Scene 7 (start).

(c) Scene 8 (start).

(d) Scene 9 (start).

(e) Scene 10 (start).

(f) Scene 6 (target).

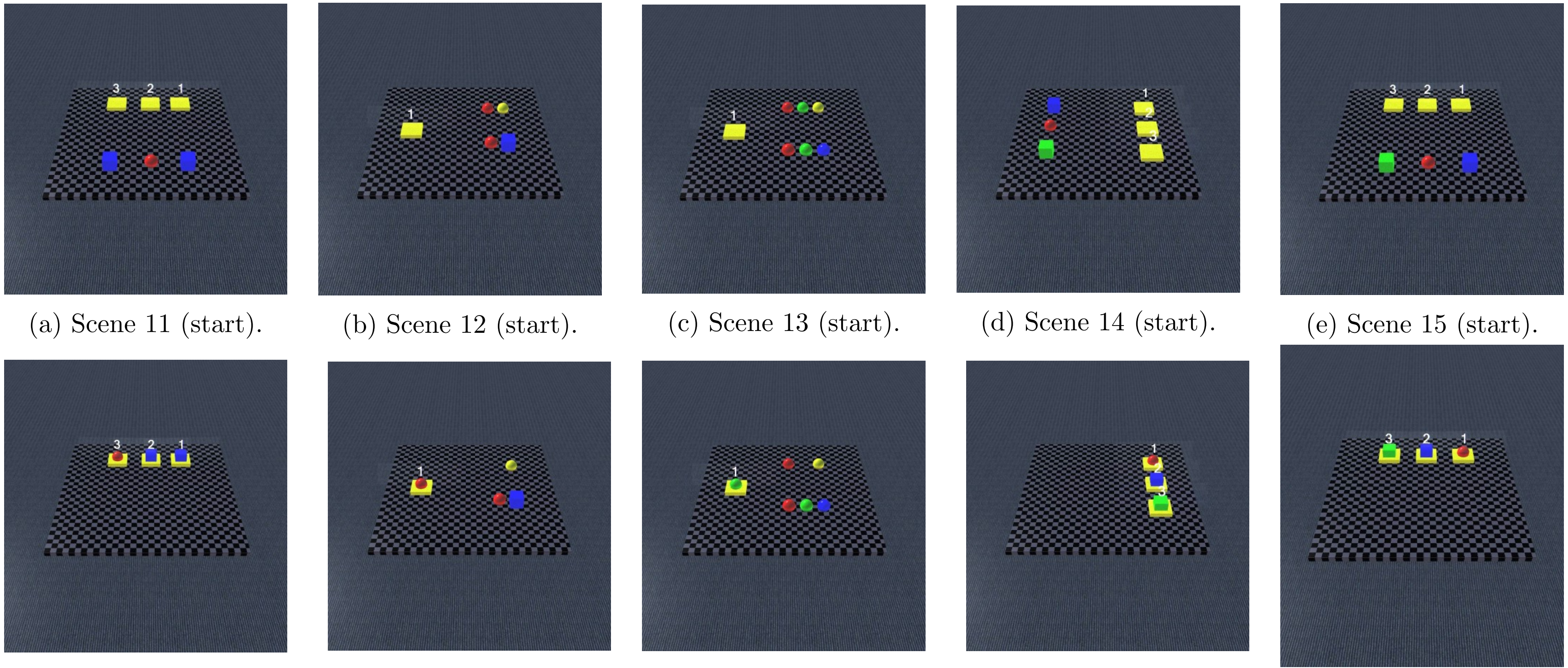
(g) Scene 7 (target).

(h) Scene 8 (target).

(i) Scene 9 (target).

(j) Scene 10 (target).

Figure 3.5: Start and target scenes for Scene 6 to Scene 10.



(a) Scene 11 (start).

(b) Scene 12 (start).

(c) Scene 13 (start).

(d) Scene 14 (start).

(e) Scene 15 (start).

(f) Scene 11 (target).

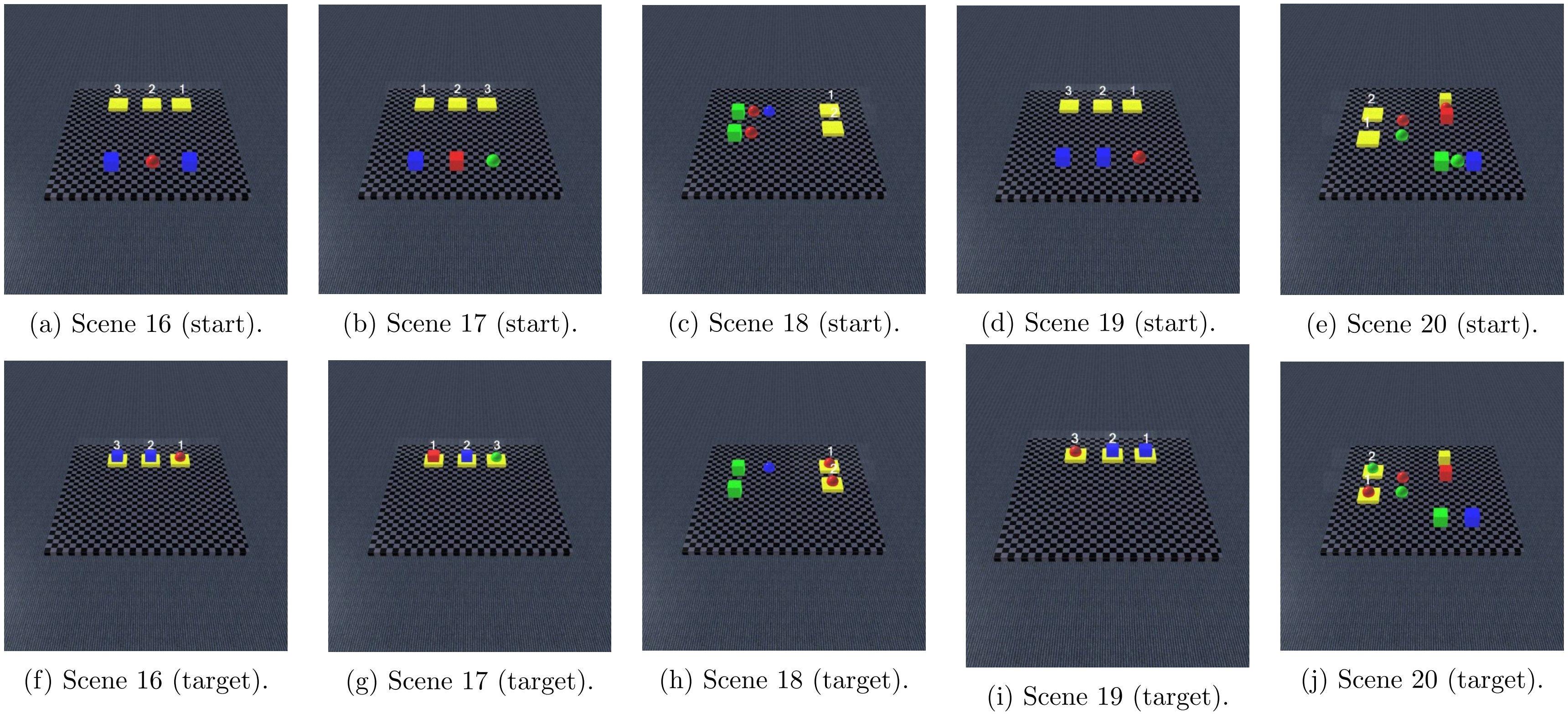
(g) Scene 12 (target).

(h) Scene 13 (target).

(i) Scene 14 (target).

(j) Scene 15 (target).

Figure 3.6: Start and target scenes for Scene 10 to Scene 15.



(a) Scene 16 (start). (b) Scene 17 (start). (c) Scene 18 (start). (d) Scene 19 (start). (e) Scene 20 (start).

(f) Scene 16 (target). (g) Scene 17 (target). (h) Scene 18 (target). (i) Scene 19 (target). (j) Scene 20 (target).

Figure 3.7: Start and target scenes for Scene 16 to Scene 20.

3.2.1 Object Roles

Each scene contained a number of objects. Objects could be either *boxes* or *balls*. They could either be red, green, blue or yellow. A third category of objects, called *places* was used to mark locations. Some of these objects needed to be re-arranged to create the target scene while others were likely to be referenced to describe an object that has to be moved. We use the following terms:

- **Critical object:** An object that needs to be moved to successfully complete a scene.
- **Non-critical object:** An object that does not need to be moved to successfully complete a scene.
- **Landmark object:** An object that does not need to be moved itself but that is in locations where they are likely to be used as a landmark in a description of a critical object

3.2.2 Design of the Scenes

To successfully complete a scene the users had to instruct the robot to move the objects that are present in the scene so that they matched the target scene. To move an object the users had to complete 3 steps. They needed to

1. Specify which object to pick up.
2. Specify where to move the object.

3. Tell the robot to set down the object.

In order to specify which object to pick up and where to move it, the user has to produce a referring expression that uniquely describes the object to pick up, and the location to which to move the object. We were specifically interested in these expressions because they represent the primary point where the users' perception and the system's perception come into contact.

We tried to design scenes which would cause the users to produce specific types of referring expressions. In general we were interested in

1. Basic referring expressions (i.e. referring expressions that use only basic attributes)

   The blue box.

2. Landmark based referring expressions

   The blue box near the yellow ball.

3. Direction based referring expressions

   The blue box on the left.

Figure 3.8 provides examples of scenes that were designed with specific expressions in mind. The white arrows indicate which object needs to be moved to which location to successfully complete the scene. Figure 3.8a shows an example of a scene in which all objects can be identified with basic referring expressions.

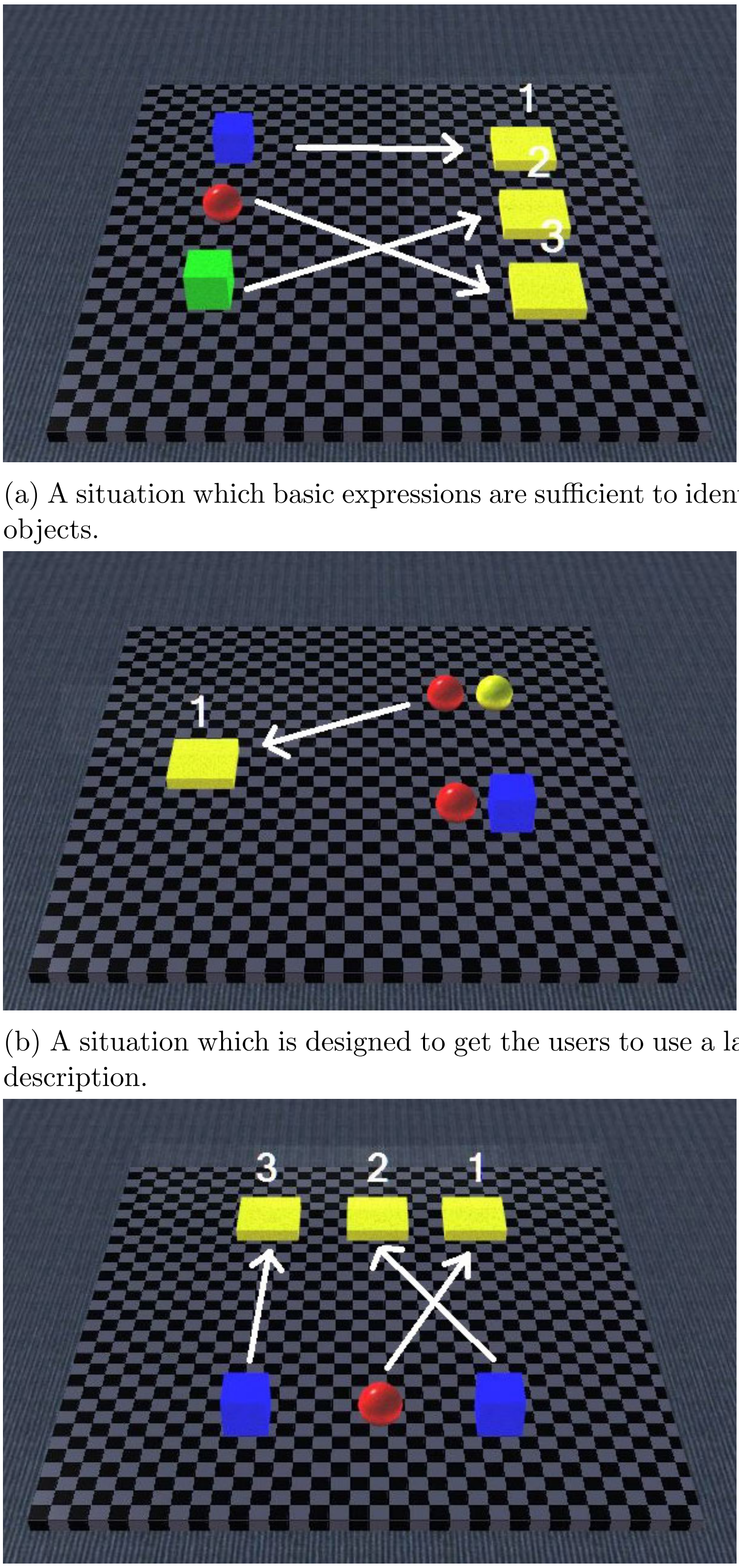
Figure 3.8b shows an example of a scene that was designed to elicit landmark based referring expressions. The arrow in the image indicates that the red ball in the upper right corner needs to be moved to Place 1 on the left side. Since there are two red balls in the scene, the user needs to use a referring expression more specific than "the red ball". We placed a yellow ball directly next to the red ball in question in order to encourage users to use the yellow ball as the landmark in a landmark based referring expression such as "the red ball near the yellow ball".

Figure 3.8c shows an example of a scene that was designed to elicit a direction based referring expression. The critical part is that the participants need to find an expression that distinguishes the blue box on the left side from the blue box on the right side. Since we did not introduce any distinguishing landmark, we expect that users are going to use a directional expression to identify the target object, such as "the blue box on the left".

We also designed the scenes to be conducive to the introduction of perception errors. We developed possible perception errors for each scene that would lead to problems in the dialogue. We simulated situations in which the robot failed to detect an object, recognized the type of an object incorrectly or recognized the colour of an object incorrectly.

3.3 Phases of the experiment

As discussed earlier, the experiment contained a series of sub-phases. Each phase was designed to investigate a specific combination of errors and ways for



(a) A situation which basic expressions are sufficient to identify the critical objects.

(b) A situation which is designed to get the users to use a landmark based description.

(c) A situation which is designed to get the users to use a direction based description.

Figure 3.8: Example start scenes.

the user to obtain information about the robot's understanding of the world. In the following we introduce and define the five phases of the experiment.

3.3.1 No Error Phase

No errors were introduced in the No Error Phase. This phase represents a baseline version of the system that is free from perception errors, but also does not provide any of the information request options that are offered in the other phases.

3.3.2 Error Phase

In the Error Phase errors were introduced. The errors were intended to model typical errors that may occur in an artificial vision system. There were three types of errors:

- **Missing object:** The system failed to detect an object.
- **Wrong colour error:** The system determined the colour of an object incorrectly. For example, the system recognized a blue box as a green box.
- **Wrong type error:** The system determined the type of an object incorrectly. For example, the system recognized a ball as a box.

The **missing object** error represents a situation in which the object detection mechanism of the computer vision system failed to detect an object. The **wrong colour** error and the **wrong type** error represent situations in which the object classification mechanism of the computer vision system classified properties of a detected object incorrectly.

Each scene contained at most one error. The effect of errors depends on the role that the object that is being affected by the error has in the task. Generally speaking errors may affect critical objects or non-critical objects. If a critical object is affected, the users are likely to experience problems when they attempt to move the object. If a non-critical object is affected, the users are unlikely to notice the error unless they need to refer to the object as a landmark to create a reference to a critical object. We therefore applied errors to both critical objects and landmark objects. We distinguish between three **error situations**:

- **Critical object error:** The error affects a critical object.
- **Landmark error 1:** The error affects a landmark object that is the only available landmark for a critical object.
- **Landmark error 2:** The error affects an object that is one of multiple possible landmarks for a critical object.

The *missing object error* required special consideration because if the robot did not perceive an object, the robot was not able to interact with it. For the experiment this meant that if an a critical object was affected by a missing object error, the users would not be able to complete the scene successfully. On the one hand we were interested to observe how the participants would react to such a situation, and what effect this frustrating

	Missing Object	**Wrong Colour**	**Wrong Type**
Landmark Error 1	Scene 3	Scene 7	Scene 12
Landmark Error 2	Scene 4, 5	Scene 8	Scene 13
Critical Object Error	Scene 6	Scene 9,10	Scene 11,14,15,16

Table 3.2: The combination of perception error type and error situation for each scene.

problem would have on their behaviour. On the other hand, our main interest was in observing how the participants actually solved problems. We therefore decided to include only one scene in which a critical object was affected by a missing object error[1].

Errors were introduced so that each type of error occurred at least once for each possible error situation. An overview of the errors that were introduced in each scene is given in Table A.1 in Appendix A. Table 3.2 shows which scene contained which combination of perception error and error situation.

3.3.3 Description Phase

In the Description Phase the interaction window contained an additional button labelled *Description.* If the user clicked on this button, the system would generate a description of the scene. For example, for the scene shown in Figure 3.9, the system generated the description:

S: There is a red ball on the top left. There is a green box on the left. There is a blue ball on the bottom left. There is a place named place 1 on the top right.

[1] The affected scene is Scene 6 (Figure 3.5a).

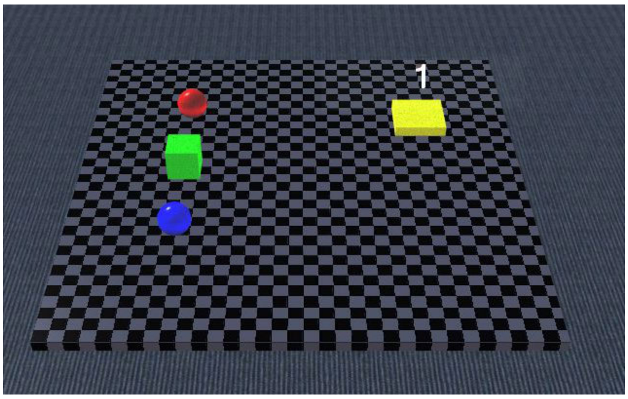


Figure 3.9: An example scene without a perception error.

This scene was not affected by a perception error. The description therefore matches the scene. The scene shown in Figure 3.10 did contain a perception error (the robot perceived the yellow ball as a yellow box). If the participant requested a description, the following description would be generated:

> **S:** There is a red ball to the left of a blue box on the bottom right. There is a red ball to the left of a yellow box on the top right. There is a place named place 1 on the left centre.

By comparing the description to their own view of the scene, the participant could then determine how their view of the scene differed from the robot's. We describe the algorithm used to generate the descriptions in Section 4.3.5.2.

The intention underlying this approach is to provide the user with a linguistic description of the system's model of the world. The markup is displayed only for the current view of the scene, i.e. if the robot performs an action, the markup would become outdated and is removed.

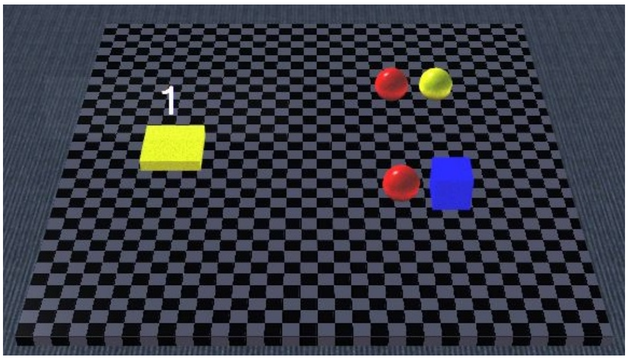


Figure 3.10: An example scene with a perception error.

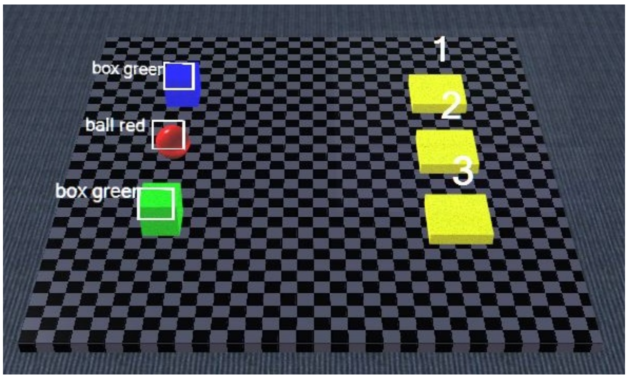


Figure 3.11: A scene with activated markup.

3.3.4 Markup Phase

In the Markup Phase the interaction window contained an additional button labelled *Markup*. If the user clicked on this button, the system would overlay the view in the simulation window with visual markup that explained its understanding of the scene. Figure 3.11 shows an example. The markup shows that the system correctly identified the green box as a green box and the red ball correctly as a red ball. The blue box in the back however was incorrectly identified as a green box.

The intention underlying this approach is to provide the user with information about the system's understanding of the scene in a visual way.

3.3.5 Querying Phase

In the Querying Phase the participants were informed that they could ask the robot simple questions about whether or not it perceived a given object. In the following we provide examples of questions and answers in relation to the scene presented in Figure 3.10.

U: Do you see a blue box?

S: Yes, I see a blue box.

.

U: Do you see a green box?

S: Sorry, I don't see any green boxes.

.

U: Do you see three red balls?

S: I only see 2 red balls.

.

U: Do you see a red ball on the left?

S: Sorry, I don't see any red balls that fit your description.

.

U: Do you see a yellow ball?

S: Sorry, I don't see any yellow balls.

The robot then answers the questions based on its understanding of the scene. The system then attempts to generate for each possible situation an informative reply. In particular, the system informs the user about whether there are more objects in the scene than fit the given description, but that are not in the location specified in the question. We provide a more detailed discussion of the responses in Section 4.4.

The intention underlying this approach is to allow the user to query the system's understanding of the scene through language based interaction. This phase was intended to allow interaction similar to the interaction we investigated in an earlier work (Schütte *et al.*, 2012) in the Map Task corpus (Anderson *et al.*, 1992).

3.4 Experiment Execution

Participants were brought into a quiet room. In the first step the experimenter gave a description of the experiment at an abstract level and gave the participants a brief demonstration of the experiment system. The participants were then given an instruction sheet appropriate for the experiment phase they participated in. The instructions sheets for all phases of the experiment are included in Appendix B. After reading the instruction sheet, the participants were shown a video that provided some general examples of interaction with the system. After the video, the experimenter demonstrated to the participants the aspects of the system that were related to the phase of the experiment they were participating in (e.g. in the Description Phase the experimenter activated the description button). After this the system was restarted and the participants were allowed to work by themselves.

The experiment always started with the first two scenes (Scene 1 and Scene 2 in Figure 3.4). They contained simple tasks and were intended to allow the participants to familiarize themselves with the system. The remaining scenes were presented in random order.

Participants were asked to complete a post experiment questionnaire after they had completed the final scene. The questionnaires for all phases of the experiment are included in Appendix Section B.

3.4.1 Participants

Participants were recruited on a voluntary base. Before volunteering, each participant was given an information sheet that outlined the experiment, requirements of the participants and an estimate of the duration of the experiment. It is included in Appendix Section B as Figure B.1. Potential participants were contacted through a mixture of personal correspondence, flyer distribution and email. As an incentive to participants, each participant was offered a chocolate bar or piece of fruit and a cup of tea or coffee while participating in the experiment.

The participants were recruited in a college environment. In total there were 55 participants. About half were computer science undergraduates, the other half were postgraduate researchers in Computer Science or Physics and Chemistry. A few were lecturers in Computer Science. The participants were between 20 and 50 years old, and about 20 of the participants were female (36.5%). All participants were either native English speakers or competent (self-assessed) second language English speakers. All participants used computers in their daily activities but were, except for two or three, not familiar with computer dialogue systems or natural language processing. We believe that the participants were therefore well suited for this experiment because they were able to naturally interact with the system. However, we believe that the level of general computer experience is not essential for the higher level outcomes of the experiment, since the experiment mostly involves natural language based interaction which we expect should be similar for all levels of computer experience.

Five of the participants who had participated in the No Error Phase agreed to participate in one of the later phases. This was not considered problematic because there was a considerable amount of time between the phases, and because the No Error Phase did not contain the errors that were the chief source of problems in the later phases. Table 3.3 provides an overview of the number of participants in each phase, the number of scenes that were attempted and the total length of the interactions recorded.

Phase	Number of participants	Scenes attempted	Total length
No Error Phase	10	200	04:16:08
Error Phase	17	338	09:03:36
Description Phase	11	220	08:08:03
Markup Phase	11	220	06:13:01
Querying Phase	11	220	06:12:38

Table 3.3: Overview of the recorded data.

3.4.2 Collected Data

While a participant was interacting with the system during the experiment, all events that occurred were logged and stored. The system logged input by the participant and the response from the system. For each input-response pair the system's analysis of the user input, the results of the reference resolution process and a description of the actions performed by the robot (if any were performed) were stored as well. In addition to that, for each action the resulting state of the simulated world was recorded. Apart from data directly related to the interaction, the system also stored data related to the state of the experiment system (such as events denoting the beginning or conclusion of scenes) and actions related to the markup and description requests and queries.

We also determine the following measures that are related to the success and costs of the interactions as well as the quality of the system as experienced by the users:

- **Abandon rate:** The percentage of scenes that were abandoned. This measure is related to the *task success rate* (any scene that was not abandoned was successfully completed). It is calculated as the quotient of the total number of scenes abandoned in a phase divided by the total number of scenes attempted.

- **Reference problem rate:** The percentage of user inputs that contained a reference that the system found ambiguous or could not resolve to any object at all. This is calculated as the number of instructions that contained a reference problem divided by the total number of instructions.

- **Undo rate:** The percentage of the actions by the system that the participants undid by using the undo-button. This measure is calculated as the total number of uses of the undo button divided by the total number of actions in a phase.

- **Number of actions per scene:** The number of text inputs a participant produced until they finished a scene.
- **Completion time:** The time the participants spent on a scene before they completed it or abandoned it (in seconds).
- **Number of information requests per scene:** How often the participants used the information request option that was available to them.
- **Speak time proportion:** The percentage of the total completion time that was filled with the robot speaking. This is calculated as the sum of the amount of time it took the system to present all utterances by the robot in a scene divided by the overall completion time for the scene.

These measures are typical of those generally recorded in experiments involving dialogue systems. If we take the perspective from (Walker *et al.*, 1997), the Abandon rate is a measure of **task success**. The completion time and number of actions represent efficiency measures of the **cost of the dialogue**. The Reference problem rate and the Undo rate correspond to cost measures that capture **qualitative aspects** of the interaction. The number of information requests and the Speak time proportion do not correspond to any success or cost measures, we instead use them to interpret and to provide context for other observations in the experiment.

3.5 Summary

In this chapter we presented the **Toy Block** experiment. It is designed to investigate interactions between a human participant and a robot that is affected by perception problems. In the different phases of the experiment we investigate different combinations of sensor errors and ways for the participant to access the robot's understanding of what it perceives. We gave an overview of how the experiment was performed and of the data recorded. We implemented a dialogue system for the users to interact with during the experiment. We describe this system, and the way it implements the different conditions of the experiment in the next chapter and analyse the data collected in the experiment from Chapter 5 onward.

Chapter 4

The Experiment System

In the previous chapter we described the set-up and the goals of the Toy Block experiment. In this chapter we describe the experiment system that was built to perform the experiments. It contains two distinct sub-systems: the **interaction system** and the **world simulation system**. The interaction system handles the interaction with the user, i.e. it interprets input by the user, formulates responses and plans the action the robot performs in order to fulfil the user's instructions. We describe the interaction system in Section 4.2. The world simulation system provides a visualisation of the world the robot acts in and presents the robot's actions to the user. We describe the world simulation system in the next section.

The experiment was performed using an actual dialogue system. Alternatively, the experiment could have been performed using a **Wizard-of-Oz** set-up (Kelley, 1984). In a Wizard-of-Oz based experiment, part of the verbal and non-verbal behaviour of the robot is provided by a human confederate . Wizard-of-Oz set-ups are frequently employed in human-robot interaction experiments to simulate functionality that the robot is not capable of performing (Riek, 2012).

We decided not to use a Wizard-of-Oz setup in this experiment. The main reason for this was that the main purpose of the experiments was to investigate the behaviour of participants when faced with errors in the robot's perception, and the attempts of the participants to resolve the problems that arose from these errors in the dialogues. Since the participants were interacting with the system through a dialogue, the participants' actions depended strongly on the behaviour shown by the system. Therefore, in order to produce useful results, the system had to act in a very consistent manner, especially across the different problem conditions. We were concerned that this would be difficult to ensure in a Wizard-of-Oz scenario.

One problem with the set-up of the experiment system was that experiments could only be performed in an offline fashion, i.e. only one participant could perform the experiment at a time, while physically using a computer on which the experiment system was set up. A number of experiments contributing towards human-computer and human-robot interaction have been performed in online and **crowd-sourcing** scenarios (e.g. (Orkin, 2013)). Using an online experiment would have made the process of recruiting participants and having them perform the experiment much easier and enabled us to collect a larger set of data. Unfortunately some of the core components of the system made it impossible to deploy the system in an online set-up without major changes to the system. On the other hand, we found during the pilot study that participants, if the experiments were not performed in a controlled environment, tended to get distracted from the experiment, which had a major negative impact on their performance in the experiment. By performing the experiment in an offline scenario, we were able to control the environment, and minimize distractions.

4.1 The World Simulation System

We decided that it was not practical to use actual robot hardware in the experiment because that would introduce a number of additional sources of problems that would detract from this work's focus on problems that arise from errors in perception. We therefore decided to use a simulation based alternative. The primary requirements for this simulation environment were as follows:

- It needed to be reasonably realistic looking, i.e. it needed to be able to produce a scene that could be easily interpreted as a three dimensional representation of a world that contained different objects.
- It had to be possible to modify the world in real time to visualise actions performed by the robot.

We decided to use the simulation environment provided by Microsoft Robotics Studio [1] as the basis of the simulation system. It provided a reasonably realistic looking simulation environment that allowed us to model

[1] `www.microsoft.com/robotics/`

a wide range of objects. It is also used for robot simulations and therefore appeared as an appropriate tool. We additionally used the SPL[1] software package ((Kim, 2015)) that allowed us to specify and modify scenes containing geometric objects in a simple way to develop a server based simulation programme that we could use to render and display the simulated world of our system in real time.

We were faced with a number of decisions while we designed the simulation world. First of all it was decided to choose a perspective that would remain fixed during the course of experiment and always show the complete scene. The perspective was chosen in a way to minimize the problem of obstruction. The presence of obstruction would have introduced a number of problems. If an object is completely obstructed, it is not visible to the participant at all, and they will not be able to account for it when they produce references to the world. If an object is partially obstructed, it will be less visually salient, which has an impact on reference resolution (Kelleher *et al.*, 2005), (Schütte *et al.*, 2010). This creates the possibility that the participant does not notice the object (or judges it as irrelevant) and omits it from their reference planning. Finally, even if the participant notices a partially obstructed object, it is still possible that they identify some of it properties incorrectly. For example, an observer may be able to tell the colour of a rectangle that is partially obstructed from view, but may not be able to determine the actual shape of the object. This is illustrated in Figure 4.1.

[1] *Simple Programming Language*

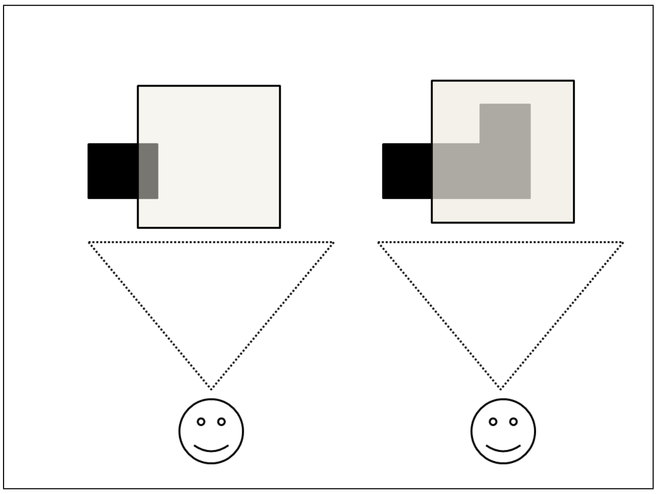
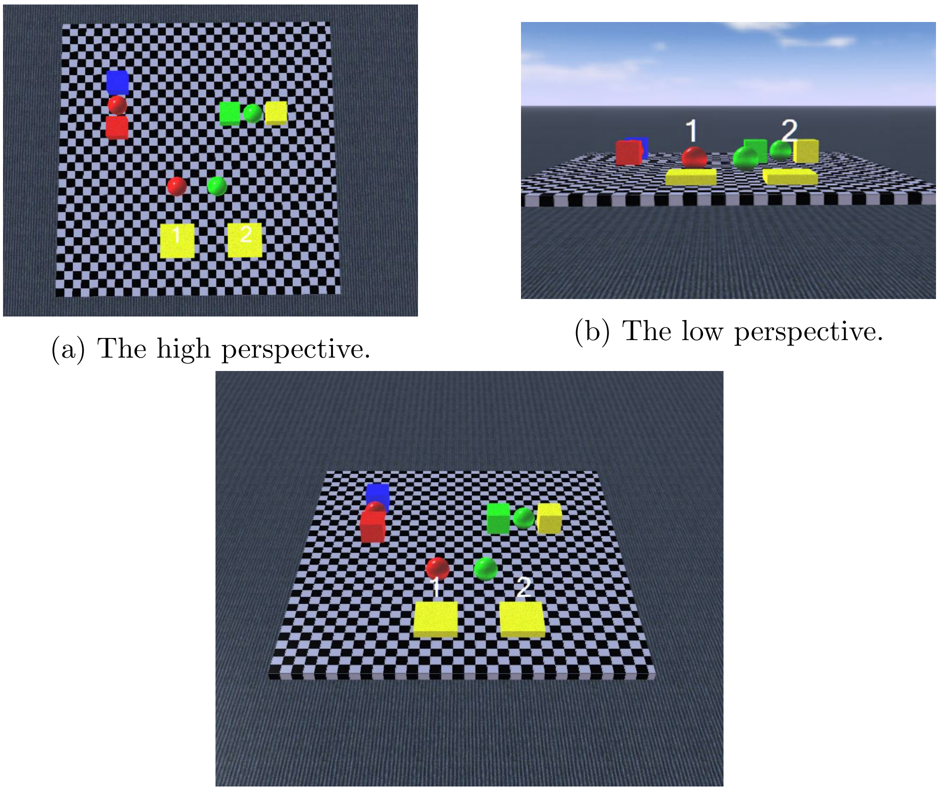


Figure 4.1: A demonstration of the problem of obstruction. While both participants see the same scene, the obstructed object is actually quite different.

Finding the right perspective involved a trade-off. A high camera perspective (Figure 4.2a) would approximate the perspective in the Map Task experiment (Anderson *et al.*, 1992). It would also make sure that no objects could be obstructed by other objects. However, it had the drawback that the elevation of objects became difficult to perceive, which was critical when objects were being lifted and held over the surface. A low perspective (Figure 4.2b) made it easy to recognize the elevation of objects, but introduced the problem that objects could be obstructed by other objects. A low perspective is used in (Kelleher, 2006). In their work obstruction is not a problem due to the scenario and the design of the scenes.

In the end a medium-high perspective was decided upon, where the size of the objects and the design of the scenes was chosen in a way to minimize the possibility of obstruction (Figure 4.2c). It is similar to the perspective used in (Gorniak & Roy, 2004). In their scenario a certain amount of obstruction occurs but is not enough to cause problems in the task.



(a) The high perspective.

(b) The low perspective.

(c) The final perspective.

Figure 4.2: The different perspectives.

The objects were arranged on a surface similar to a game board to give the participants a clearer sense of the space the experiment took place in. The game board was covered with a checker board style texture that clearly communicated the board's spatial orientation and assisted the participants in determining the relative position of the objects on the board.

We did not intend the participants to use the board itself or the tiles of the board as a means of describing the location of objects (e.g. by using expressions such as "The ball on the left edge of the board") or to define movement targets (e.g. with instructions like "move the green ball three tiles forward"). We therefore made the board large enough that the objects were well clear of the edges of the board, and the tiles of the surface small enough to make it impractical to use them as reference.

The worlds contained geometric objects that were intended to represent toy building blocks similar to the ones from the original physical prototype. The world contained two types of objects: boxes and balls. alls, which were sphere-shaped.

The objects within each scene were designed to be roughly the same size, i.e. it was not intended that **size** be used a criterion to distinguish objects.

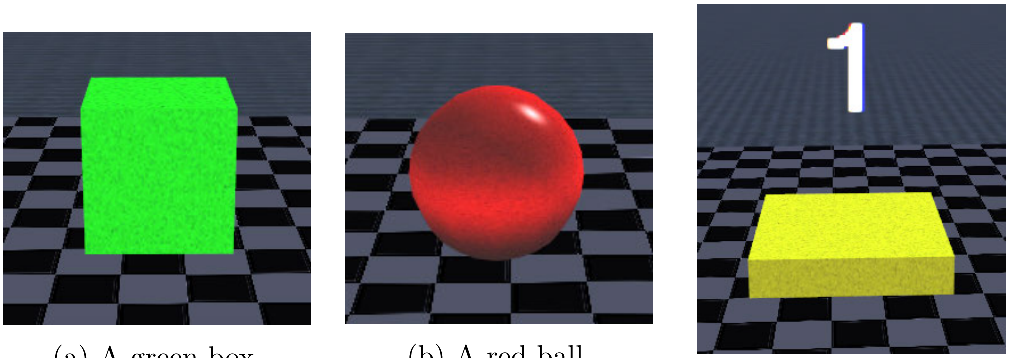
Each object was assigned a **colour** (green, red, blue or yellow). Each colour was resolved to an RGB colour value[1] during the rendering process, which was then applied to the rendered object. The world was illuminated by diffuse white light which provided slight shading on the objects but did not alter the colour otherwise.

Each object was defined by its **basic attributes**, namely its type and its colour. Each attribute had exactly one value. A green ball for example would be described as the tuple:

$$\langle type : ball, colour : green\rangle.$$

There was a special type of object called **place**. It was specifically designed to serve as a marker for locations on which objects could be placed. Places are modelled as flat squares and are yellow. Places were labelled with a number that allowed the participants to easily identify them and refer to them. Places were the only kind of object that was expected to have other objects placed on top of them.

[1] The colour values were: *green*: 0-1-0, *red*: 1-0-0, *blue*: 0-0-1 and *yellow*: 1-1-0.



(a) A green box.

(b) A red ball.

(c) "Place 1"

Figure 4.3: Some objects from the simulation.

4.2 The Interaction System

While the world simulation simulates the world the interaction takes place in and the robot's actions, the interaction system handles the actual interaction with the user. The interaction system is a dialogue system for situated language and is based on the general architecture for situated spoken dialogue systems discussed in Section 2.1.1. An overview of the architecture of this system is provided in Figure 4.4. We segmented the modules slightly differently from this architecture and introduced new modules to account for the introduction of perception errors. Additional modules were introduced that handle the simulation of the environment and the actions of the robot in the world, and the information request options for the later phases of the experiment. Like the general architecture for spoken dialogue systems discussed in Section 2.1.1, this architecture can be divided into one part that relates to vision, and one part that relates to language based interaction and the reference part that establishes a link between the vision and the language part. We discuss the vision system in the next section and the remainder of

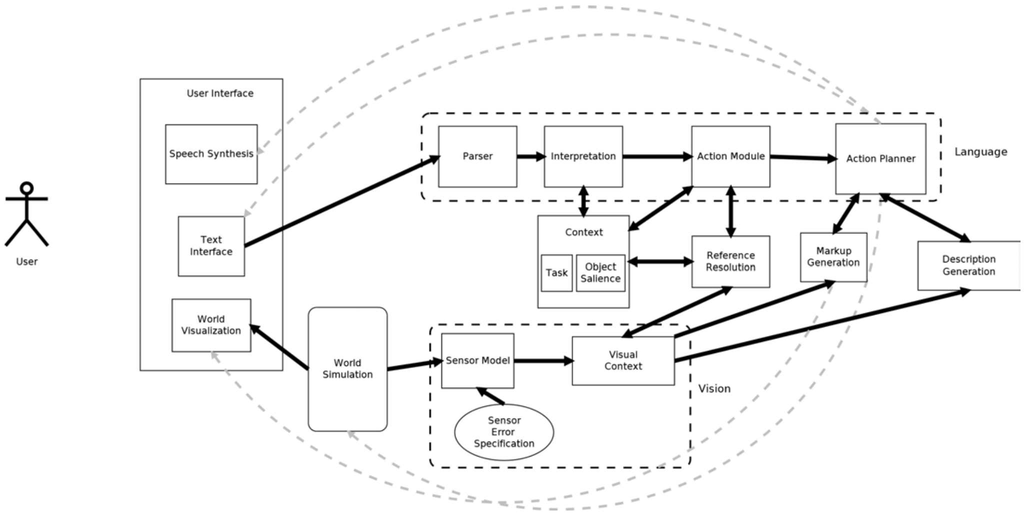


Figure 4.4: The architecture of the Toy Block system.

the architecture in Section 4.3.

4.2.1 The Vision System

The vision system simulates the perception of the robot. The system maintains a representation of the robot's perception of the world in the **Visual Context** module. It is regularly updated with input from the **World Simulation** that is mediated through the **Sensor Module**. The purpose of the sensor module is to simulate the computer vision based perception of the robot. In general it provides the system with an accurate representation of the world. However, through the **Sensor Error Specification** interface errors can be introduced into the robot's perception. Errors are manually specified. This is illustrated in Figure 4.5. The sensor model mediates the robot's perception of the scene and thereby the model of the world the robot forms. In this example, the sensor model is specified to produce a sensor error which causes the robot to not perceive a specific object in the scene. It is consequently missing in the robot's model of the world. This is an instance

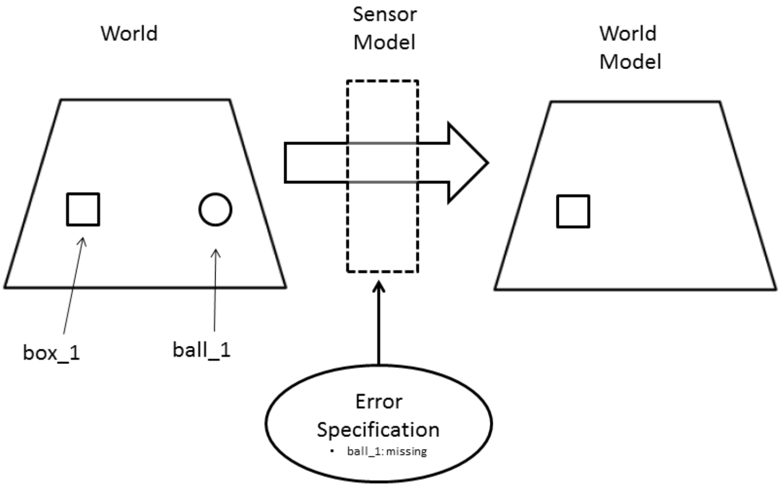


Figure 4.5: The sensor model mediates the robot's perception of the world and may be used to introduce errors through manual error specifications.

of a *missing object error*.

4.3 Language based interaction system

The interaction system forms the second half of the experiment system. In the following sections we describe the system and its functionality. First we describe how the systems processes the input by the user (Section 4.3.1). We then describe how the system plans and performs actions requested by the user (Section 4.3.3) and how it generates responses (Section 4.3.4). Finally we give a description of the implementation of the different information request options (Section 4.3.5).

4.3.1 Language Understanding

The task of the language understanding component is to analyse the inputs by the user and to extract a representation of the user's intention that enables the system to produce an appropriate reaction. In the first step the language understanding component analyses the input text produced by the user by *parsing* it. In this system we use the NLTK parser (Manning *et al.*, 2014). The parser takes the input text and identifies grammatical dependency structures between pairs of words.

The interpretation module accepts the results of the parse and attempts to construct a representation of the intention of the utterance. If the parser is not able to find a parse of the input or the system is not able to derive a useful interpretation of the input, the system simply responds with a response asking the user to reformulate the input. The content of instructions are represented as a set of frames that may contain representations of referring expressions. In the following sections we describe the frames and the referring expressions, and then discuss the process of producing frames based on the output produced by the parser.

4.3.1.1 Frames

Each frame represents one of the possible actions the user may ask the system to perform. We follow the terminology used in VerbNet[1] and use the term *patient* to refer to objects that are being affected by an action and

[1] `https://verbs.colorado.edu/~mpalmer/projects/verbnet.html#thetaroles`

destination to describe the end point of motions. There are four frames in our system which can be described as follows.

Action: Pick-up

Patient: A description of the object to pick up.

The *pick-up* frame represents an instruction to pick up an object. The patient slot is filled with a representation of the referring expression used by the speaker to describe the object they want the robot to pick up.

Action: Move

Patient: A description of the object to move.

Destination: A description of where to move the object.

The *move* frame represents an instruction to move an object to a given point. The destination slot contains a combination of a landmark and a relation that specifies where to move the object in relation to the landmark (e.g. *in front of* it, *to the right of* or *on* it).

Action: Put

Patient: A description of the object to put down.

Destination: A description of the location where the object should be placed.

The *put* frame represents an instruction to put an object down. Such an instruction can either contain a location where to put the object (e.g. *“Put the ball on Place 1.”*) or no location (e.g. *“Put the ball down.”*. The destination slot in this frame is therefore optional. If it is not provided, the frame represents an instruction to put the object down in the place where it is currently being held.

Action: See

Patient: A description of the object(s) the user is asking about.

The *see* frame is only used in the Querying Phase. In this phase the user is able to ask the system whether or not it perceives an object (or multiple objects) that match a given description. The patient and destination slots in the frames are filled with representations of referring expressions used by the participants.

Referring expressions are represented using a feature structure. In general, a referring expression is modelled as a list of attribute-value pairs. We distinguish between **basic attributes** (type and colour) that apply only to the referent in isolation and **spatial attributes** that need to be interpreted in relation to the context. We provide examples of referring expressions in this system in the following section.

4.3.1.2 Interpretation Process

The interpretation module selects and instantiates a frame based on the main verb of the text. The following verbs were assigned to each action frame based on our initial corpus analysis:

- **pick-up:** pick, take, select, grab, grasp
- **move:** move
- **put:** put, drop, place, set
- **see:** see

The interpretation module then fills the slots of the appropriate frame by interpreting the grammatical relations identified in the utterance as semantic relationships depending on the type of the verb. For the sentence *"Pick up the green ball near the blue box."* the parser produces the following output:

```
root(ROOT-0, pick-1),
prt(pick-1, up-2),
det(ball-5, the-3),
amod(ball-5, green-4),
dobj(pick-1, ball-5),
det(box-9, the-7),
amod(box-9, blue-8),
prep_near(ball-5, box-9)
```

The output consists of a list of binary relation predicates. The name of the relation indicates the type of the grammatical relation, while the arguments denote words in the input text.[1] The output can also be presented in a graphic format as shown in Figure 4.6.

[1] The meaning of the relations is defined in (de Marneffe & Manning, 2008)

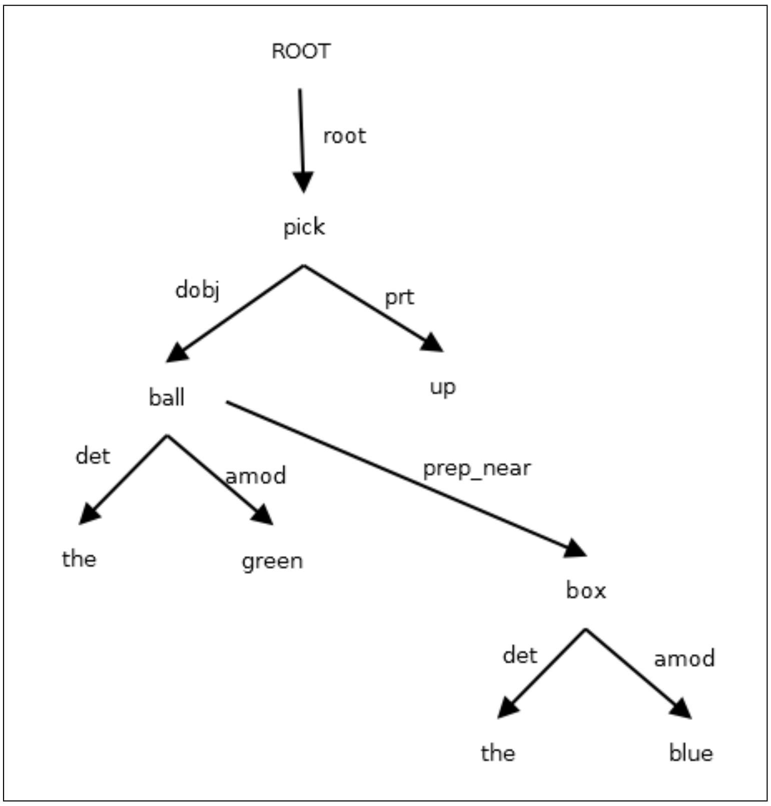


Figure 4.6: The dependency graph produced for the sentence *Pick up the green box near the blue box.*

In this example, the parser identifies the word *pick* as the root verb of the input text. The interpretation process therefore instantiates a new pick-up frame:

$$\begin{bmatrix} \text{action} & \text{pick-up} \\ \text{patient} & \text{(empty)} \end{bmatrix}$$

It further finds that the word *ball* is the direct object (*dobj*) of verb. The direct object of a verb *"functions as the patient or beneficiary upon which the verb acts"* (Trask & Stockwell, 2007). It therefore represents an object that is being affected by the root verb. The interpretation therefore begins to fill the *patient* slot of the frame with the properties associated with this object. In this example, there is an *amod* relationship between word *ball* and the word *green*. An *amod* relationship represents an adjectival modification, and it indicates that *green* modifies the meaning of *ball*.

To enable to system to decide which attribute modifications represented colour attributes, we created a list of colour terms[1]. Using this list, the system classified *green* as a *colour* attribute.

$$\begin{bmatrix} \text{type} & \text{ball} \\ \text{colour} & \text{green} \end{bmatrix}$$

Furthermore we find that the word *ball* is in a prepositional relationship with the word *box* (indicated by the *prep_near* relationship in the dependency graph. This relationship is lexicalised with the word *near*, which the system recognizes as a spatial relation term. We therefore extract the attributes of *box*

$$\begin{bmatrix} \text{type} & \text{box} \\ \text{colour} & \text{blue} \end{bmatrix}$$

and then add it as the value of a relational attribute to the attributes of the representation of *box*. The type of the relation is represented with the *reltype* attribute in the frame.

$$\begin{bmatrix} \text{type} & \text{ball} \\ \text{colour} & \text{green} \\ \text{rel} & \begin{bmatrix} \text{reltype} & \text{near} \\ \text{relatum} & \begin{bmatrix} \text{type} & \text{box} \\ \text{colour} & \text{blue} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

[1] *Red*, *blue*, *green* and *yellow*

This frame represents the referring expression that was used. In the next step it is inserted into a frame for a pick-up action:

$$\begin{bmatrix} \text{action} & \text{pick-up} \\ \text{patient} & \text{(empty)} \end{bmatrix}$$

The completed frame then looks as follows:

$$\begin{bmatrix} \text{action} & \text{pick-up} \\ \text{patient} & \begin{bmatrix} \text{type} & \text{ball} \\ \text{colour} & \text{green} \\ \text{rel} & \begin{bmatrix} \text{reltype} & \text{near} \\ \text{relatum} & \begin{bmatrix} \text{type} & \text{box} \\ \text{colour} & \text{blue} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

If the interpretation module is not able to derive a complete interpretation of the output produced by the parser, the system stops the interpretation process and produces a reply that asks the user to reformulate their input:

S: Sorry, can you please reformulate this?

The output of the interpretation process is used to update the **task context**. The task context is a representation of the most recent utterance produced by the user, i.e. it contains a representation of the requested action and the referring expressions used to describe the object involved in the instruction. Together with the **Object Salience Context** it represents the state of the dialogue. The Object Salience Context represents the salience of objects in the scene. Salience in this context refers to a notion of semantic salience, where objects that have recently been interacted with are more salient. It contains a list of the objects the system has directly interacted with (i.e. objects it has moved, or in the Querying Phase spoken about). However, in practical terms, only the object the robot most recently interacted with is relevant.

4.3.2 Acting

Once the input has been successfully interpreted into a task frame, the action module then attempts to perform the desired action. This is done in three steps:

1. In the first step, the action module attempts to ground all references in the object salience context by resolving the **referring expressions** in the utterance (we discuss the resolution of referring expressions separately in Section 4.3.6).

2. If the first step succeeds, the action module then constructs a **plan** by specifying an action or a sequence of actions for the robot to perform. We describe the action planning process in Section 4.3.3.

3. If a plan could successfully be constructed, the plan is then performed, and a response message to the user is generated.

This process is illustrated in Figure 4.7. Problems may occur during reference resolution and during planning. The robot can only perform *pick-up*, *move* and *put* actions if it can identify unique patients and destinations. If it finds that a referring expression is unresolvable or ambiguous, it responds with a reply that communicates this fact and that indicates the referring expression that is causing the problem:

U: Pick up the green box.

(The robot does not see a green box.)

S: Sorry, I don't see any green boxes.

U: Pick up the red ball.

(The robot sees multiple red balls.)

S: Sorry, there is more than one red ball and I can't figure out which one you mean.

To increase the informativeness of the replies, they are slightly modified if an unresolvable referring expression contained a spatial description. We distinguish between two cases:

1. No referent that matched the basic attributes was found anywhere in the scene.

2. A referent was found that matched the basic attributes, but not the spatial attributes (i.e. a matching referent was found but not at the location specified in the expression).

In the first case, the system simply states that it does not see any object that expression described:

S: Sorry, I don't see any green boxes.

However, this would not be an appropriate reply for the second case. The system instead replies with a reply that states that it did not find any objects that fit the given description:

U: Pick up the red ball near the blue box.
(The robot does not find any objects that match the description.)

S: Sorry, I don't see any red balls that fit your description.

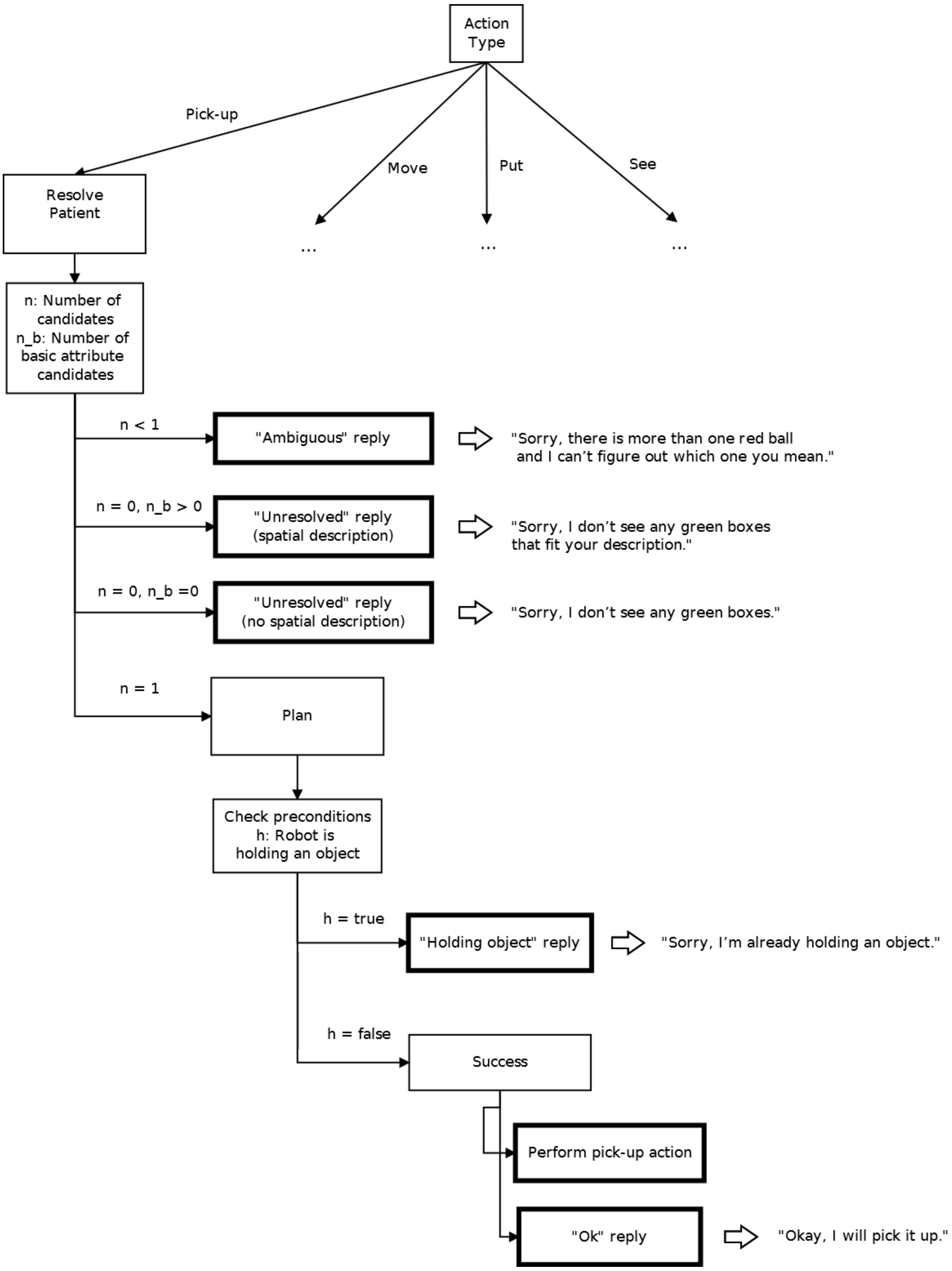

Action Type
Pick-up
Move
Put
See
Resolve Patient
...
...
...
n: Number of candidates
n_b: Number of basic attribute candidates
n < 1
"Ambiguous" reply
"Sorry, there is more than one red ball and I can't figure out which one you mean."
n = 0, n_b > 0
"Unresolved" reply (spatial description)
"Sorry, I don't see any green boxes that fit your description."
n = 0, n_b =0
"Unresolved" reply (no spatial description)
"Sorry, I don't see any green boxes."
n = 1
Plan
Check preconditions h: Robot is holding an object
h = true
"Holding object" reply
"Sorry, I'm already holding an object."
h = false
Success
Perform pick-up action
"Ok" reply
"Okay, I will pick it up."

Figure 4.7: Reacting to an instruction. We only show the process for pick-up instructions. The process for the other cases is similar, but involves resolving the expression for the destination and additional reactions if this fails.

4.3.3 Action Planning

The simulated robot is capable of performing three types of actions:

- **Pick-up:** The robot picks up an object. As a precondition, the robot must not be already holding an object.
- **Move:** The robot moves an object to a specified location. This represents a horizontal movement of the robot arm. As a precondition, the robot must be holding an object.
- **Put-down:** The robot sets down an object it is holding. As a precondition, the robot must be holding an object.

To enable a more flexible interpretation of instructions, the system was able to plan actions to fulfil the preconditions of other actions. For example, if the user gave the instruction "*Move the green ball to Place 1.*" while the robot was not actually holding an object, the system would make a plan to identify and pick up an object that fit the description "the green ball" first. Since the domain was rather limited, the system did not perform full planning but used a small library of pre-formulated plans.

If the planning process fails, the system produces a response that communicates the problem. For example, if the user instructs the robot to pick up an object while it is already holding an object, the system will produce the following response:

S: Sorry, I'm already holding an object.

The robot is also limited in that it cannot stack objects on top of other objects. If a user issues an instruction that would result in the robot stacking one object on top of another object, the planning process is abandoned and the following response is produced:

S: Sorry, I can't put objects on top of other objects.

Every time the robot successfully completes an action, the object salience context is updated by making the object that was last interacted with the most salient object.

4.3.4 Response Generation

Utterances by the system are based on pre-formulated templates for each possible reply that, where necessary, are combined with automatically generated referring expressions for the objects under discussion. The responses are displayed in the interaction window. They are also presented as spoken language that is produced using the Mary Text-to-Speech system (Schröder & Trouvain, 2003). The robot therefore both speaks and also replies in written text. Action plans are passed on to the world simulation, where they are put into action.

4.3.5 Information Request Modules

The information request options in the Description Phase and the Markup Phase of the experiment are implemented in the Markup Generation module and the Description Generation module. The **Description Generation** module is active in the Description Phase of the experiment. Its purpose is to access the Visual Context and to produce a description of the scene as it is perceived by the robot. The **Markup Generation** module is active in the Markup Phase of the experiment. Its purpose is to provide the user with a visual indication of the robot's understanding of the scene. In the Querying Phase the participants were able to ask the system simple questions. We discuss the generation of responses to these questions in Section 4.4.

4.3.5.1 Markup Generation

The Markup Generation Module accesses the Visual Context module. It retrieves the location of each object the robot perceives and the value of the *colour* and *type* attribute. Based on this, it produces a specification of a set of labels that are passed on to the **Markup Generation** module. The Markup Generation module then specifies a set of markup labels that are overlayed over the view of the scene presented to the user. For each object one markup label is generated that consists of the following:

- A rectangular frame that encompasses the object, the purpose of which is to highlight the object to show that it is perceived by the robot.

- A text label showing the type attribute value that is perceived for the object (e.g. *box*) and the colour value that is perceived for the object (e.g. *green*). The position of the text label relative to the object (e.g. above or to the right) is specified during the design of the scene to

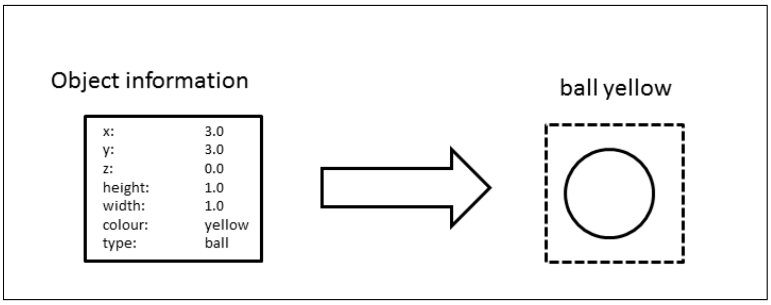


Figure 4.8: An object with a label.

avoid problems with a label that might become hard to read because it overlaps with an object).

Figure 4.8 shows an example. The left side represents the information that the robot has for an example object (in this case a yellow ball). The right side shows the markup label that would be generated based on this information (overlayed over the object in question).

After the markup button has been activated, the markup is produced and overlayed over the scene. If the markup button is pressed a second time, the markup is removed. If an object in the scene is moved, the markup is also removed.

4.3.5.2 Description Generation

In the Description Phase of the experiment the interaction window contained an additional button labelled *Description*. If the user clicked on this button, the system would generate a description of the scene. The system generates descriptions by clustering objects that are located close to each other into groups, and then describing each group in terms of spatial configuration, and the group's overall location in the scene. The process comprises the following

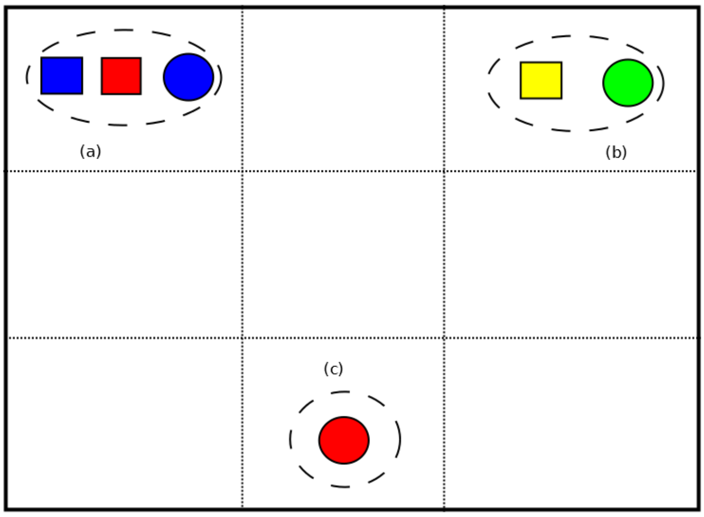


Figure 4.9: An example scene containing three groups of objects.

steps:

1. It clusters objects that are close to each other into groups.

2. It selects a relation that describes the objects in each group in relation to each other. Each relation is associated with a text template that describes the objects and the relation between them. The type of the relation is chosen based on the number of objects in the group:

   - For groups of three[1] objects, the *between* relation was chosen. It describes the object that was most central in the group in relation to the other objects. Group (a) in Figure 4.9 is an example of such a group.

   - For groups of two objects, the *left of* relation is used to describe one of the objects in relation to the other object. Group (b) in Figure 4.9 is an example of this (in this example we decide to describe the green ball as being to the right of the yellow box).

[1] Scenes were designed such that at most three objects would form a group.

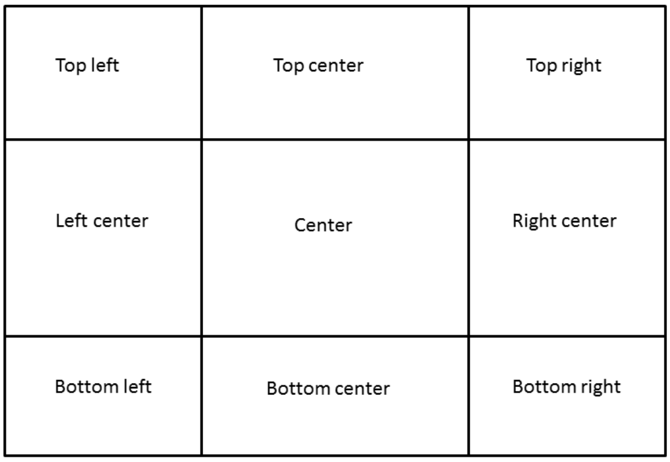
- No relation was needed for groups containing a single object (for example Group (c) in Figure 4.9

3. A directional attribute that describes the location of the group in a global frame of reference is added to the group. For this purpose we divided the scene into nine regions as shown in Figure 4.10, and associated each region with an attribute value. For example, Group (a) is assigned the attribute *back left*, Group (b) is assigned the attribute *top right*, and Group (c) is assigned the attribute *bottom centre*.

4. For each group a sentence is generated. This is done by instantiating the templates that are associated with the group relations and filling them out with descriptions of the objects involved in the groups and adding a lexicalisation of the directional attribute. In our example the following sentences are generated:

   - Group (a): There is a red box between a blue ball and a blue box on the top left.
   - Group (b): There is a green ball to the right of a yellow box on the top right.
   - Group (c): There is a red ball on the bottom centre.

The sentences are then simply concatenated and presented to the user. Overall, the system would present the following description:



Top left	Top center	Top right
Left center	Center	Right center
Bottom left	Bottom center	Bottom right

Figure 4.10: The spatial regions that are used for description generation.

S: There is a red box between a blue ball and a blue box on the top left. There is a green ball to the right of a yellow box on the top right. There is a red ball on the front centre.

This approach to generating descriptions is not intended as a general solution. Some scenes were specifically designed to guarantee that the system chooses groupings that are plausible to a human user. The intention underlying the Description Generation approach is to provide the user with a linguistic description of the system's model of the world, and to examine how users resolve perception based problems using the descriptions. We however do not claim that the approach used here is a general solution for scene descriptions.

4.3.6 Reference Resolution

The system can resolve references to objects in the visual context, or to objects that have recently been discussed or interacted with. The process of reference resolution is influenced by the role of the referring expression in the task and the state of the robot.

If the referring expression is used in the patient slot for an action that presupposes that an object is being held (for example in a *move*-instruction), and the robot is actually holding an object, the process attempts an **anaphoric** interpretation by checking whether the object that is being held is compatible with the given expression. If this is the case, the expression is resolved to the object that is being held.

If the referring expression is a pronoun, the system also attempts an anaphoric resolution based on salience. If an object is currently being held, the expression is resolved to this object because we assume that this object is always the most salient. In the following example, the robot correctly resolves the pronoun *it* in the third utterance to the object that is currently being held:

U: Pick up the green box.

(The robot picks up the box.)

S: Ok.

U: Move it to Place 1.

S: Ok.

(Robot moves the box to Place 1.)

If no object is being held, the object salience context is queried for the most salient object. In general, this is the most recent object the robot has interacted with. In the following example, the system resolves the pronoun *it* in the third utterance the object it has just put down in the previous step:

(Robot is holding a box.)

U: Put the box down.

S: Ok.

U: Pick it up.

S: Ok.

(The robot picks up the box it just put down.)

In the Querying Phase, an object that has been discussed through a query can also become salient:

U: Do you see a green box?

(The robot sees exactly one green box.)

S: Yes, I see a green box.

U: Pick it up.

S: Ok.

(The robot picks up the box.)

In this example, the system is asked whether it sees a green box. It confirms that it sees one green box. The box in question thereby becomes the most salient object. The system subsequently resolves the pronoun *it* to it.

If no pronoun is used, the process attempts an **exophoric** interpretation by finding a matching object in the visual context. The visual context is a representation of the state of the world that is being maintained in the system. It is updated after every action based on the sensor model (as discussed earlier). The resolution process proceeds as follows:

1. In the first step, the system filters all visible objects based on the basic properties provided in the expression. The result forms the set of candidate referents.

2. If the referring expression contained no spatial attributes, the set of candidates is returned as the result.

3. Otherwise the set of candidates is filtered based on the spatial attributes.

In the following section we describe the evaluation of basic attributes and the different types of spatial attributes.

4.3.7 Basic Attributes

A basic attribute provided in a referring expression matches the attribute of an object if the attributes values are identical or **synonymous**. We annotated for each possible attribute value a set of synonyms. For example, the objects of the type *ball* can be referred to as *spheres* and *circles* and *balls*. The sets of synonyms were determined based on our corpus analysis as described in Section 3.1.

We also introduced the type expressions *thing* and *object*. They are interpreted as hypernyms for *ball* and *box*. For example, a green ball and a green box can both be referred to as a *green object* (whether the expression is distinguishing depends on the context of the scene).

4.3.8 Spatial Attributes

If the expression contained a spatial attribute, the candidates are again filtered based on the spatial attribute. We distinguish between two types of spatial attributes: directional attributes and relational attributes.

4.3.8.1 Directional Attributes

Directional attributes describe the position of an object in terms of the global frame of reference by specifying a location in the scene. They are expressed as prepositional phrases (e.g. "the green box *on the left*") or as adjectives (e.g. the *bottom right* green box"). We segmented the scene into three horizontal regions (front, back and center) and three vertical regions (left, right and center), and associated them with the corresponding directional attribute values.

If an expression contained a directional attribute (e.g. "the box on the left"), the reference resolution was restricted to the region specified by the attribute. By combining a vertical and a horizontal directional attribute, the target region could be restricted to the intersection of the two regions (e.g. "the box on the bottom left"). The regions are illustrated in Figure 4.11. The set of regions used for reference resolution is different from the set

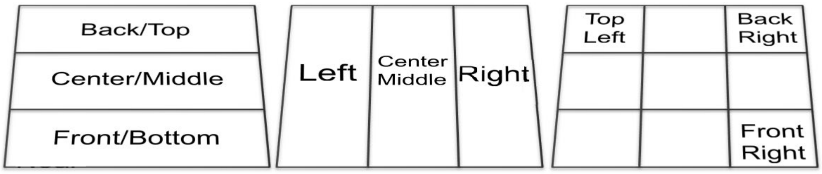


Figure 4.11: The spatial regions that are used for referring expression resolution.

of regions used in the generation of descriptions (presented in Figure 4.10) in that during interpretation general expressions such as *on the left* or *in the front* were acceptable, while for the descriptions always the most specific region was used (e.g. an object located anywhere on the left side of the world was either described as being in the *top left* region, the *left center* region or the *bottom left* region.

Direction attributes such as *near*, *far* or *furthest* form a special case of directional attributes. They denote an object that is in an extreme position along a direction. For example, the expression "the nearest box" is generally intended to refer to a box that is has the smallest distance to the speaker. The expressions "the leftmost box" is generally understood to refer to the box that has no other box to the left of it. To interpret an attribute of this type, all objects in the visual context are filtered by the basic attributes in the expression and then ordered by their position along the x-axis (for *rightmost* and *leftmost*), or the y axis (for *far* and *near*). The object that has the most extreme position is then chosen as the referent. This is illustrated in Figure 4.12.

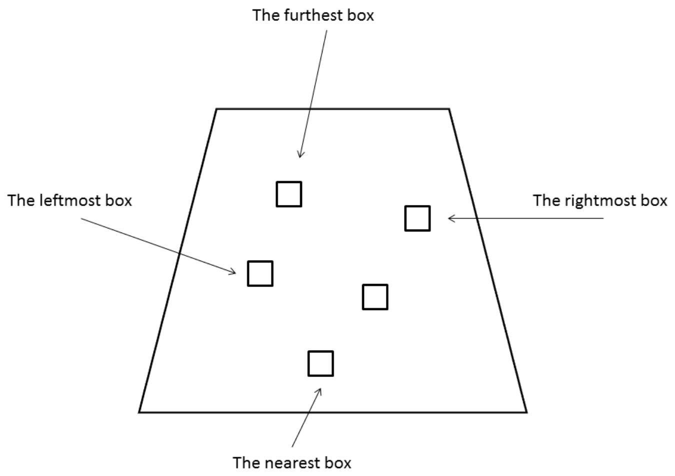


Figure 4.12: Interpretation of directional attributes that describe objects in extreme positions.

4.3.8.2 Relational Attributes

Relational attributes describe the position of an object in relation to a landmark object. They are expressed as prepositional phrases, for example "the green box *near the blue ball*" or "the box between the green balls". The following prepositions are covered by the system:

- to the right of
- to the left of
- behind
- in front of
- above
- next to
- near

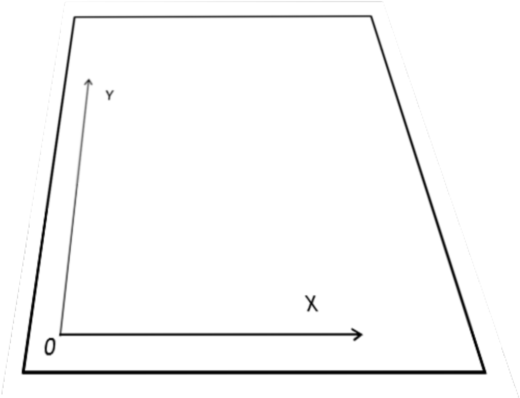


Figure 4.13: The coordinate system underlying the scene.

- between

They represent projective relations and in our system are interpreted using spatial templates. A relational attribute applies if the system is able to identify an object inside the template that matches the referring expression given for the landmark. The template for the preposition *to the right* is presented in Figure 4.14. The template defines a rectangular region that stretches away from the landmark. It has a length of 2 times the width of the bounding box of an object and the height of one width (the bounding box of all objects is the same size). The type of the relation determines in what direction the template is projected from the landmark:

- **to the right of:** The template stretches horizontally from the landmark in the positive direction of the x-axis.
- **to the left of:** The template stretches horizontally from the landmark in the negative direction of the x-axis.

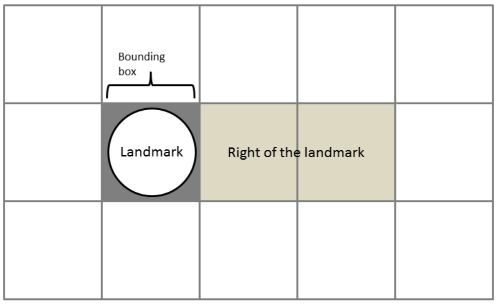


Figure 4.14: The template for the relation *right of* the landmark.

- **behind:** The template stretches vertically from the landmark in the positive direction of the y-axis.
- **in front of:** The template stretches vertically from the landmark in the negative direction of the y-axis.
- **above:** This is interpreted as a synonym of *behind*.[1]
- **next to:** This relation holds if any of the previously mentioned relations hold.
- **near:** This is treated as a synonym of *next to*.

The *between* relation describes a relation between the target object and a group of landmark objects. This group can either be described as a list of objects ("the ball between the yellow box and the blue box") or a set of objects ("the ball between the boxes"). To evaluate this relation, the system identifies a group of objects that are *close* together (two objects are considered *close* if they are no further apart than the width of two bounding

[1] While this interpretation may appear counterintuitive, we found that users did use this relation to refer to objects that were behind other objects. We believe that this is because users interpreted the image of the world in the simulation view as a two dimensional surface when they used this reference where the further object appeared to be *above* the nearer object.

boxes) and satisfies the description of the landmark group. The relation then holds for any candidate objects that are in the space between the landmark objects. This is illustrated in Figure 4.15.

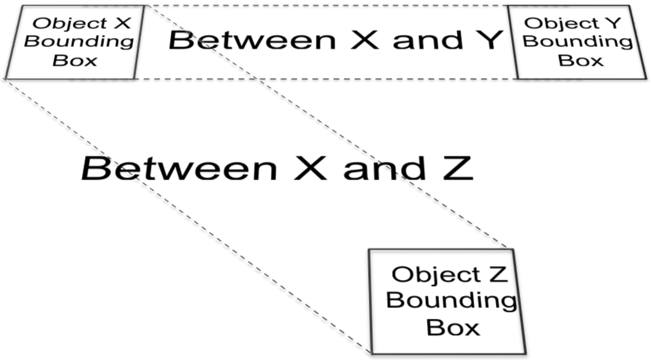


Figure 4.15: An illustration of the interpretation of *between*.

4.4 Querying

In the Querying Phase of the experiment, the users were able to ask the robot simple questions about whether or not the robot perceived an object, such as:

U: Do you see a green box?

U: Do you see a green box on the right?

U: Do you see a green box near a blue ball?

In addition to this, the user could also ask about the number of objects the robot perceived that matched a given description:

U: Do you see two green boxes?

U: Do you see three balls?

The users were instructed to only ask questions that could be answered by a yes or no answer. The responses always contained a reformulation of the description the speaker used in the original query. The purpose of this to emphasize to the speaker that the system had correctly interpreted the query (and to allow the speaker to detect if the system had incorrectly interpreted an expression). The answering process is an extension of the reference resolution process where the chief differences are that the users could ask about arbitrary numbers of objects, while the reference resolution for the normal instructions was focused on identifying unique referents. In addition to that we aimed to maximize the amount of information contained in the responses by the system.

If the user did not specify an explicit number of objects in the question (e.g. "Do you see a green ball?"), the response is straightforward. If an object that matches the expression is found, the systems replies with a positive reply:

S: Yes, I see a green ball

If the expression contained a spatial attribute, the system responds with a positive response that acknowledges that the expression contained a spatial attribute:

S: Yes, I see a green ball that fits your description.

If no fitting objects are found, the system gives a negative response:

S: Sorry, I don't see any green balls.

If the expression contained a spatial attribute and no candidates were found, the system responds with a response that acknowledges the spatial attribute:

> **S:** Sorry, I don't see any green balls that fit your description.

The replies become more involved if the user asks about a specific number of objects. For example, a user may ask *"Do you see three boxes?"*. If the user asks about a specific number of objects in a specific location, the system answers the question based on the number of object that fit the description, but also mentions other objects, that fit the basic attributes of the description, but are in a different location. We refer to the number of objects the user asks about (in this example *three*) as the **queried number** in the following. We differentiate the reply based on the following features:

1. Did the number of objects that matched the description match the queried number?

2. Did the expression contain a spatial attribute?

3. Were there candidate referents that matched the basic attributes of the expression, but not the spatial attributes (i.e. were there objects that had matching type and colour, but that were not where the participant described them)?

Table 4.1 shows all possible combinations of these factors. After excluding cases that were logically impossible (e.g. if no spatial attribute was used in

Number of candidates matches queries number	**Was a spatial attribute used?**	**More candidates elsewhere?**	**Case**
Yes	No	No	1
Yes	Yes	No	2
Yes	Yes	Yes	3
No	Yes	Yes	4
No	No	Yes	-
No	No	No	5
Yes	No	Yes	-
No	Yes	No	6

Table 4.1: Possible outcomes of number based queries.

the expression, it was not necessary to investigate whether there were objects "elsewhere" in the scene since no location was specified in the first place), we were left with six possible outcomes. We formulated a specific response for each of these outcomes as follows:

Case 1: The user asked about a specific number of objects without using a spatial attribute, andthat number of objects was found.

U: Do you see three green balls?

S: Yes, I see three green balls.

Case 2: The user asked about a specific number of objects with a description that contained a spatial attribute. A matching number of objects was found, and no object that matched the basic description was found elsewhere in the scene.

U: Do you see 2 green balls on the left?

S: Yes, I see 2 green balls that fit your description.

Case 3: The user asked about a specific number of objects with a description that contained a spatial attribute. A matching number of objects was found, but there are more objects in the scene that match the basic description.

U: Do you see 2 green balls on the left?

S: Yes, I see 2 green balls that fit your description, and 1 green ball somewhere else.

Case 4: The user asked about a specific number of objects with a description that contained a spatial attribute. The system did find some objects that matched the description, but not the number queried. Objects that matched the basic expression were found somewhere else in the scene.

U: Do you see 2 green balls on the left?

S: I only see 1 green ball that fits your description and I see 1 green ball somewhere else.

Case 5: The user asked about a specific number of objects without using a spatial attribute, and the number of matching objects did not match the queried number.

U: Do you see 2 green balls?

S: I 1 green ball.

Case 6: The user asked about a specific number of objects with a description that contained a spatial attribute. A number of objects were found that matched the expression, but not the queried number. No objects that matched the basic description were found elsewhere in the scene.

U: Do you see 2 green balls on the left?

S: I see one green ball that matches your descriptions.

If the user asked a question, the system resolved the expression used exophorically in the scene. It then determined the case based on the features discussed earlier, and chose the template associated with the case to formulate the reply.

4.5 Summary

In Chapter 3 we described the Toy Block experiment. In this chapter we described the dialogue system that was used to perform the experiment. We described the components of the system and how the system interprets commands and performs actions. Furthermore we described how errors are introduced into the system's perception, and we described the information request options.

In the following Chapter we analyse the data collected during the experiments. In Chapter 5 we analyse how well the participants were able to solve the tasks in the experiment, how big the impact of the perception errors was, and whether the information request options were useful in resolving the problems arising from perception errors. In Chapter 6 we investigate how participants used information requests. In Chapter 7 and Chapter 8 we investigate how participants reacted when the robot encountered a perception error, and how they resolved the problems arising from this.

Chapter 5

Effect of Perception Errors on Task Performance and User Experience

In this chapter we examine the headline results of the Toy Block experiment. In the experiment, the participants interacted with a simulated robot to complete a series of tasks. During four of the five phases of the experiment the robot experienced artificially induced perception problems. In three of these phases, the users were offered different ways to request information about the robot's perception. In this chapter we investigate how the participants experienced the task at a subjective level and investigate the objective success and cost measures that were recorded during the experiment to provide a quantitative description of the difficulties encountered in each phase. We do not investigate the content of the interactions in the experiment in much detail in this chapter, but aim to provide a general overview of the relative characteristics of the phases of the experiment. We focus on the structure and content of problem resolution dialogues in the following chapters.

5.1 Research Questions

In this chapter we address the following research questions:

Research Question 5.1: *How did the participants experience the task and the problems in the dialogues?* – For this question we focus on how the participants experienced the experiment by evaluating answers provided on the post-experiment questionnaire.

Research Question 5.2: *Do perception errors as experienced by the robot have an impact on the difficulty of the task?* – We investigate whether introducing perception errors had an impact on the measurable aspects of the task difficulty such as the participant's likelihood to successfully complete scenes, their likelihood to encounter reference problems and the effort necessary to complete scenes.

Research Question 5.3: *If participants are offered the option to request information about the robot's understanding of the scene, do they use it?* – We investigate whether or not the participants used the description option, the markup option and the querying option when they were available.

Research Question 5.4: *Does the ability to request information about the robot's understanding of the scene have an impact on the participants'* *ability to solve the task?* – For this question we investigate whether the ability to request information from the robot decreased the measurable task difficulty as discussed in Research Question 5.2.

Research Question 5.5: *How do the information request options compare to each other in terms of effectiveness?* – For this question we investigate whether one of the ways to request information is superior to the other alternatives in general, and whether one of them is particularly suitable to optimizing certain aspects of task difficulty, e.g. whether one of the options reduces the Abandon Rate particularly strongly.

In the remaining sections of this chapter we address each one of these research questions.

5.2 RQ 5.1: The Participants' Experience

How did the participants experience the task and the problems in the dialogues?

The participants were asked to complete an questionnaire after the experiment. The questionnaires contained three questions that referred to all phases of the experiment. The questionnaires for the later phases also contained additional questions that referred to the specific phase. The three general questions were:

Question 1:	*"How well would you say the robot understood you? (1 = very poor, 5 = very good")*
Possible responses:	1,2,3,4,5

Question 2:	*"Interacting with the system was frequently frustrating."*
Possible responses:	Strongly disagree, Disagree, Neutral,Agree,Strongly agree

Question 3:	*"When the robot misunderstood something, I was often able to figure out what its problem consisted in."*
Possible responses:	Strongly disagree, Disagree, Neutral,Agree,Strongly agree

Question 1 to Question 3 were included in the questionnaires for all phases. In the Error Phase, the Markup Phase and the Querying Phase two additional questions were introduced:

Question 4:	*"I found it easy to accomplish the tasks."*
Possible responses:	Strongly disagree, Disagree, Neutral,Agree,Strongly agree

Question 5 (Description Phase):	"I found the descriptions offered by the system helpful."
Possible responses:	Strongly disagree, Disagree, Neutral,Agree,Strongly agree

Question 5 (Markup Phase):	"I found the markup-option offered by the system helpful."
Possible responses:	Strongly disagree, Disagree, Neutral,Agree,Strongly agree

Question 5 (Querying Phase):	"I found it helpful that I was able to ask the system questions."
Possible responses:	Strongly disagree, Disagree, Neutral,Agree,Strongly agree

While at the surface Question 1 only refers to language based communication we hope to determine whether this impression is affected by the presence of perception errors. The intention behind Question 2 was to determine whether the participants' felt more frustrated with the task when perception errors were present. In Question 3 the participants were asked to describe how well they were able to resolve problems they encountered in the dialogue. This is aimed at the problems that arise due to perception errors. The purpose of Question 4 was to determine how difficult the participants found it to complete the task. It somewhat overlaps with Question 1, but is not focused on the robot but only the task. With Question 5 we attempt to determine how helpful the participants found the different ways of accessing the robot's understanding of the world.

5.2.1 Responses

An overview of the distribution of the responses to the questions is given in Table 5.1 (Question 1), Table 5.2 (Question 2), Table 5.3 (Question 3), Table 5.4 (Question 4) and Table 5.5 (Question 5). They are presented as bar charts in Figure 5.1 and Figure 5.2. The graphs in this figure are are arranged so that all graphs in one column refer to the same question and all graphs in one row refer to the same phase of the experiment.

Response	**No Error Phase**	**Error Phase**	**Description Phase**	**Markup Phase**	**Querying Phase**
1	0.00%	0.00%	0.00%	0.00%	0.00%
2	0.00%	5.88%	0.00%	0.00%	0.00%
3	20.00%	52.94%	18.18%	45.45%	36.36%
4	30.00%	23.53%	54.55%	36.36%	45.45%
5	50.00%	17.65%	27.27%	18.18%	18.18%

Table 5.1: The responses for Question 1 (*“How well would you say the robot understood you?”, where 1 represented “Very poor” and 5 represented “Very good’*).

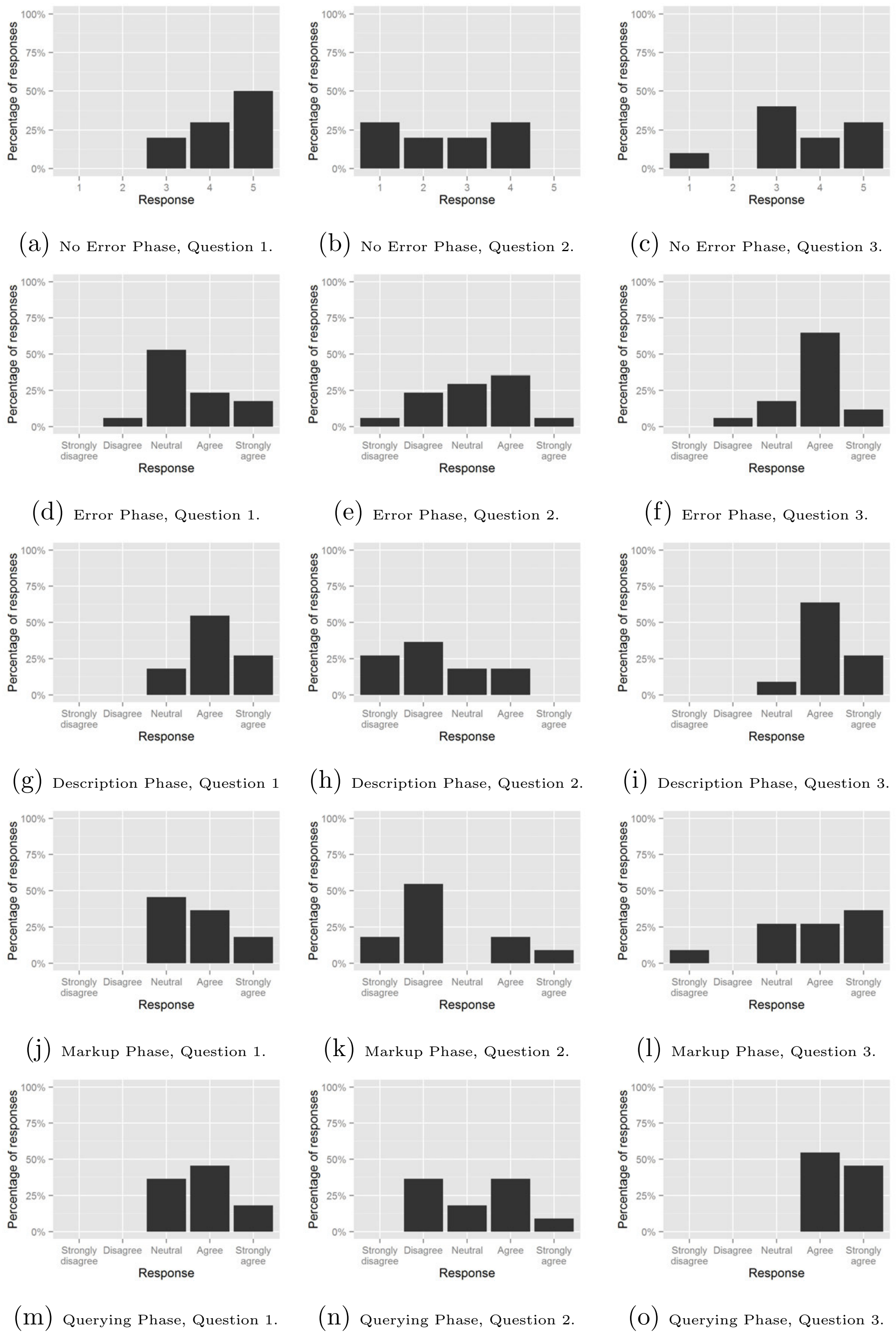
Response	**No Error Phase**	**Error Phase**	**Description Phase**	**Markup Phase**	**Querying Phase**
Strongly disagree	30.00%	5.88%	27.27%	18.18%	0.00%
Disagree	20.00%	23.53%	36.36%	54.55%	36.36%
Neutral	20.00%	29.41%	18.18%	0.00%	18.18%
Agree	30.00%	35.29%	18.18%	18.18%	36.36%
Strongly agree	0.00%	5.88%	0.00%	9.09%	9.09%

Table 5.2: The responses for Question 2 (*“Interacting with the system was frequently frustrating.”*).

Response	**No Error Phase**	**Error Phase**	**Description Phase**	**Markup Phase**	**Querying Phase**
Strongly disagree	10.00%	0.00%	0.00%	9.09%	0.00%
Disagree	0.00%	5.88%	0.00%	0.00%	0.00%
Neutral	40.00%	17.65%	9.09%	27.27%	0.00%
Agree	20.00%	64.71%	63.64%	27.27%	54.55%
Strongly agree	30.00%	11.76%	27.27%	36.36%	45.45%

Table 5.3: The responses for Question 3 (*“When the robot misunderstood something, I was often able to figure out what its problem consisted in.”*).

The responses to Question 1 suggest that the participants were generally satisfied with the robot’s communicative capabilities. The responses for the Error Phase tend to be lower than for the other phases. This indicates that the perception errors had a negative effect on the participants’ impression of



(a) No Error Phase, Question 1.

(b) No Error Phase, Question 2.

(c) No Error Phase, Question 3.

(d) Error Phase, Question 1.

(e) Error Phase, Question 2.

(f) Error Phase, Question 3.

(g) Description Phase, Question 1

(h) Description Phase, Question 2.

(i) Description Phase, Question 3.

(j) Markup Phase, Question 1.

(k) Markup Phase, Question 2.

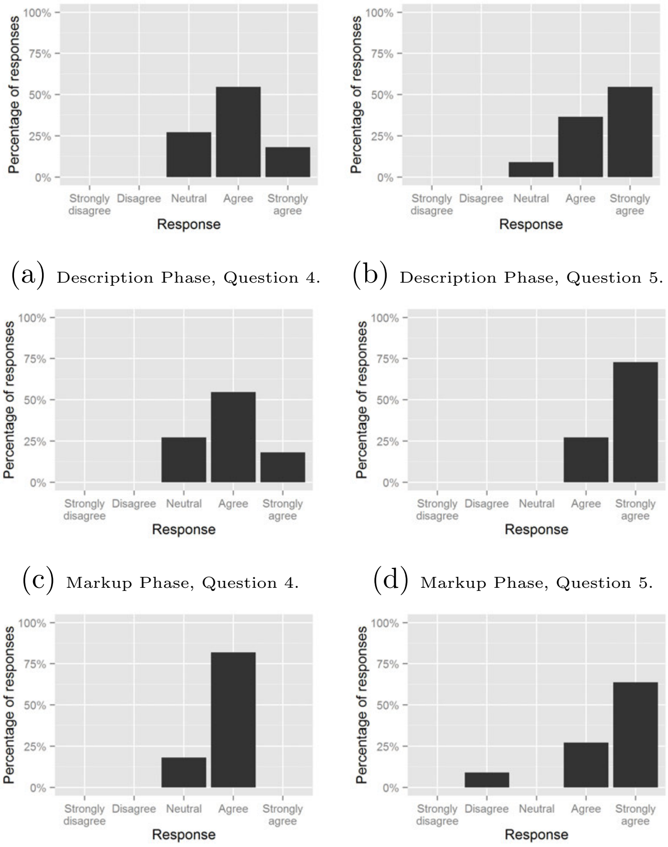
(l) Markup Phase, Question 3.

(m) Querying Phase, Question 1.

(n) Querying Phase, Question 2.

(o) Querying Phase, Question 3.

Figure 5.1: The responses to Question 1, Question 2 and Question 3.



(a) Description Phase, Question 4.

(b) Description Phase, Question 5.

(c) Markup Phase, Question 4.

(d) Markup Phase, Question 5.

(e) Querying Phase, Question 4.

(f) Querying Phase, Question 5.

Figure 5.2: The responses to Question 4 and Question 5.

Response	**Description Phase**	**Markup Phase**	**Querying Phase**
Strongly disagree	0.00%	0.00%	0.00%
Disagree	0.00%	0.00%	0.00%
Neutral	27.27%	27.27%	18.18%
Agree	54.55%	54.55%	81.82%
Strongly agree	18.18%	18.18%	0.00%

Table 5.4: The responses for Question 4 (*"I found it easy to accomplish the task."*).

Response	**Description Phase**	**Markup Phase**	**Querying Phase**
Strongly disagree	0.00%	0.00%	0.00%
Disagree	0.00%	0.00%	9.09%
Neutral	9.09%	0.00%	0.00%
Agree	36.36%	27.27%	27.27%
Strongly agree	54.55%	72.73%	63.64%

Table 5.5: The responses to Question 5 (*"I found the markup-option offered by the system helpful."*, *"I found the descriptions offered by the system helpful."*, *"I found it helpful that I was able to ask the system questions."*).

the robot. However, it should be noted that the replies for the Error Phase are still mostly neutral. We believe that this may be due to the formulation of the question. It is possible that participants did not attribute all of the problems arising from perception errors to the robot's capability to understand them, but may have perceived them as a separate issue, and therefore have been less inclined to issue a negative response for this question.

The responses to Question 2 are surprisingly evenly distributed for the No Error Phase. However, the majority of the responses disagree or are neutral with respect to the statement (*"Interacting with the system was frequently frustrating."*). This indicates that the participants were not overly frustrated. For the Error Phase a tendency towards neutral and agreeing responses appears to exist, indicating that participants found the interaction with the robot more frustrating if errors were present. It is confounding though that at least a few participants did not find that the task was frequently frustrating. In personal discussions with participants we found that at least some participants perceived the problems in the dialogues not so much as a source of frustration, but more as an interesting game or puzzle, and that they felt challenged rather than frustrated. It is possible that the positive responses were due to this phenomenon.

In the Description Phase and the Markup Phase a majority of the participants did not find the task frustrating. The distribution of the responses for the Querying Phase is almost as strongly positive as it is negative. This indicates that in while some participants in the Querying Phase had a positive experience, another group had a negative experience. A possible reason for this may be that the querying option set up expectations in some participants it could not fulfil. While the participants could ask any question they could think of, the system could only interpret and answer simple yes-or-no questions. It is plausible that one group of participants found it easier than others to accept the limitations of the querying option, and work within them, while others found it difficult and kept experiencing disappointment when the system was not able to handle more complicated questions.

Another problem that had an impact on the responses could be that the statement in Question 2 was formulated in a somewhat biased manner ("Interacting with the system was *frequently* frustrating."). It is possible that responses were skewed towards neutral and negative responses because of this.

As stated earlier, Question 3 was aimed at discovering how well participants felt they were able to resolve problems that arose in the dialogue due to perception errors. In the No Error Phase, no errors had been introduced, and participants gave mostly neutral and agreeing responses. This may appear somewhat confounding since the question presupposes that perception errors do occur. On the other hand, participants were not aware of the fact that in later phases errors would be introduced, and they probably interpreted the question as referring to other problems encountered by the system such as parser errors or mismatches about the interpretation of spatial attributes. We therefore do not attribute much importance to these responses. In the Error Phase the participants gave mostly agreeing responses, indicating that they were able to resolve the problems they encountered. In the Description Phase, the Markup Phase and Querying Phase the responses are also agreeing, but with a stronger tendency towards "Strongly agree" rather than just "Agree". This indicates that, while participants felt that they were able to resolve the problems without being able to explicitly request information from the system, they were more confident when they had access to additional information. Similar to Question 2, the statement in Question 3 was formulated in a slightly positive manner ("When the robot misunderstood something, I was *often* able to figure out what its problem consisted in."). This may have skewed the responses slightly negatively.

The responses to Question 4 show that in general most participants found it easy to solve the tasks. In addition to this, the responses to Question 5 show that almost all participants found the information options they were offered useful. Overall, the markup option appears to have the most positive response, while the other options have (very few) neutral and disagreeing responses mixed in.

5.2.2 Summary

The introduction of perception errors gives the participants the impression that the robot understands them less well. This shows that problems in perception are registered by the participants as problems in the robot's communication. If participants can access the robot's understanding of the scene, their impression of the robot's understanding capabilities improves.

The introduction of errors leads to increasing frustration, and giving participants the ability to request information from the robot decreases frustration. We believe that this is because the additional information about the robot's understanding of the scene allows the participants to construct a mental model of the robot's perception which allows them to construct a model the robot's understanding and identify the underlying problem. This observation is consistent with the participants' impression that they were able to identify and resolve problems in the dialogue.

Conclusions:

1. The introduction of errors decreased the participants' impression of the robot's communicative capabilities and increased frustration.

2. The participants were more confident that they could resolve problems

the robot had if they could access information about the robot's understanding of the scene.

While the overall responses are plausible, we were still surprised that the responses for the Error Phase were not more clearly negative. As we stated earlier, a possible explanation for this may be found in the fact that some participants perceived the experiment as an interesting and challenging game rather than a frustrating task. It is unlikely that users in a real world scenario, who may have to use a robot as part of their job, or under time constraints, would be as tolerant. It is also possible that the participants of the experiment were, due to the relatively low age, familiar with computer puzzle games, and therefore related the task to one of those games rather than to a work-related task. A different group might therefore have shown more frustration.

5.3 Task Success and Dialogue Cost Measures

In the first part of this chapter we evaluated the subjective experience of the participants during the experiment. In the remainder of this chapter, we are going to investigate the dialogues and in particular the difficulties the participants experienced in completing the tasks in terms of the recorded task success and the costs of the dialogues that were measured. We presented a list of the recorded measures earlier in Chapter 5. We will reiterate them here and discuss how they relate to the difficulty of the task.

We take the **Abandon Rate** as a high-level indicator of the difficulty of the task. The instructions given to the participants[1] stated that they were free to abandon scenes if they thought they would not be able to complete them. For this measure we assume that a higher value indicates that the task was more difficult.

We see the **Reference Problem Rate** as a second indicator of the difficulty of the task that is directly related to the presence or absence of perception errors. It indicates how often the system was not able to resolve a reference a participant used in an instruction to a unique referent. If the system could not find a unique resolution to a reference in an instruction, it was not able to perform the instruction and had to ask the participant for clarification. While a certain baseline amount of reference problems must be attributed to general deficiencies in the system's understanding capabilities or mismatches between the participant's interpretation of spatial expressions and the system's, reference resolution problems are also a likely symptom of divergences between the participants' and the robot's understanding of the scene (the results from the No Error Phase provide a baseline for the performance of the system without perception errors). For this measure higher values indicate that the participants encountered more problems, and that the task was therefore more difficult.

The **Undo Rate** gives an impression of the relation between the number of instructions the participants entered and the number of times participants decided to undo an action performed by the system. There are a number of possible reasons why a participant might decide to undo a system action. For

[1] The instruction sheets for each phase are provided in Appendix B

example, a participant might notice that after an action has been completed, that they had actually misinterpreted the goals of the task (e.g. they might have moved a box not to Place 1 as shown in the target image, but to Place 2), and then decide to undo the action, and perform it differently. However, and critically for this experiment, another reason why a participant might have to undo an action would be that the system interpreted an instruction in an unexpected way due to perception errors. For example, participants might give the instructions "Pick up the green box". In this situation the system might, due to a colour perception error, resolve the expression to a red box, and pick it up, and thereby perform an unintended action. In this situation the user would be likely to use the undo-button to undo this action. In this category a higher value indicates that the task was more difficult.[1]

The **Number of Actions** describes how many instructions the participants sent to the system to complete a scene. For this measure we counted every instruction that was sent to the system by the participant. It does not account for actions that were performed outside of the dialogue proper such as activating the undo-button or actions related to requesting information. It can be understood as a representation of the amount of work the participants spent on a scene. We propose that the presence of perception errors increases the number of actions necessary to solve a scene for the following reasons:

[1] However, we do not believe that this measure is a particularly strong indicator since, rather than use the undo button to revert an action, the participants could also give instructions to the robot that reverted the effects of the previous action. For example, if the robot picked up an object that the participants did not expect it to pick up, they could just tell the robot to *"Put it down."*.

1. Participants gave instructions that the system misunderstood so that the participants had to reformulate the instructions, thereby producing more instructions.

2. Participants had to try a strategy other than the direct one to solve the task. We assume that these strategies would involve more instructions.

The **Completion Time** indicates how much time participants on average spent on a scene until they had either finished it or until they abandoned it. Like the number of actions, it relates to the amount of work the participants had to invest to solve the scenes. However, the two measures are not directly interchangeable. While the number of actions counts only the instructions that a participant sent to the system, the completion time is also affected by other factors, such as the typing speed of the participants. It is also influenced by the amount of time which the participants spent planning their approach, as well as time that was spent using and interpreting the information provided by the robot. Therefore the value of this measurement is somewhat more difficult to interpret than the previous ones. Intuitively, faster completion times indicate a lower difficulty than slower completion times. On the other hand, the fact that participants were able to abandon scenes means that shorter completion times can also indicate that participants got frustrated frequently and abandoned scenes early. The value of this measurement therefore has to be interpreted in the context of the Abandon Rate and information about the use of information requests.

The **Number of Information Requests** describes how often the participants requested information (i.e. how many times they requested a scene description or markup and how many queries they asked). Unlike the other measures, we do not think that there is a direct relation between this measure and the difficulty of the task. On the one hand, we expect that participants request information more frequently if they find the task difficult. On the other hand, we expect that the use of the information requests would make the task easier, and therefore result in a reduction of the measurable task difficulty.

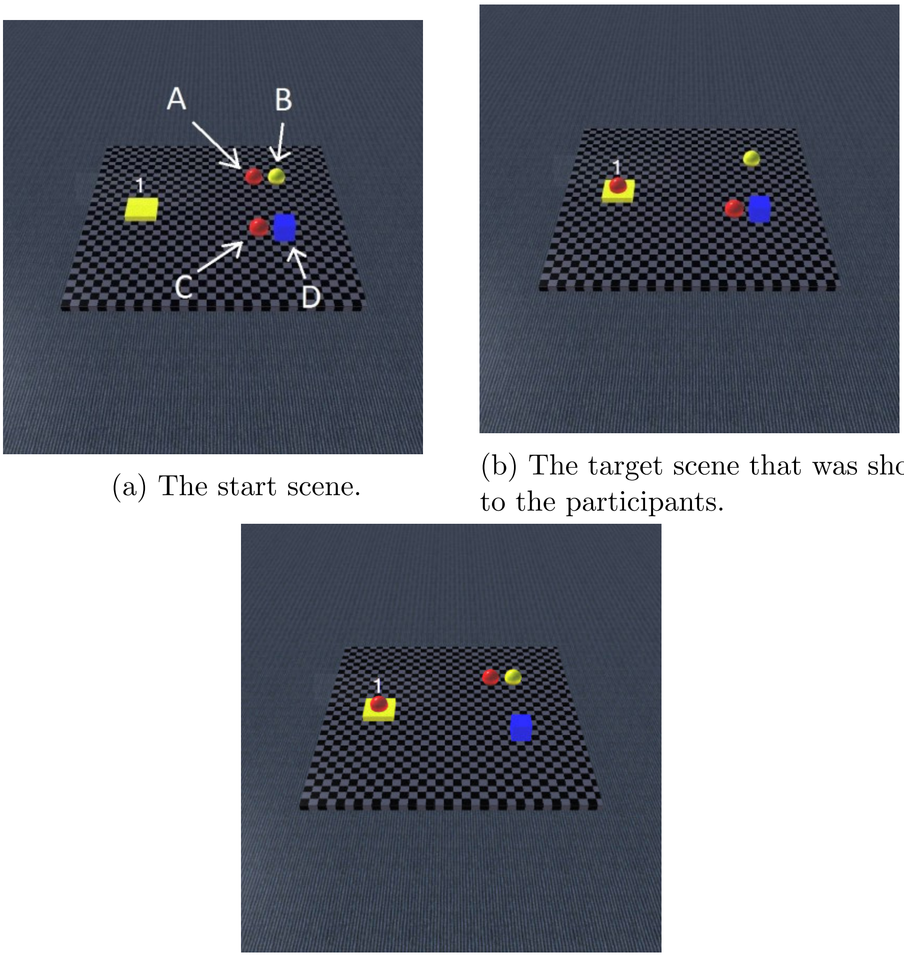
5.3.1 Irregular Outcomes

During the evaluation we discovered a problem in the recordings. Due to an oversight in the formulation of the target conditions, participants were able to complete some of the scenes with configurations that did not match the actual target scenes.

An example of a scene in which an irregular outcome was possible is Scene 12. The start configuration of this scene is shown in Figure 5.3a. The target scene that was shown to the participants is shown in Figure 5.3b. To complete the scene, the participants had to instruct the system to pick up the red ball that is labelled as 'A' put it on Place 1. The second red ball (labelled 'C') was introduced to provide a distractor object so that the participants had to use a referring expression other than “the red ball” in order to pick up the intended ball. The scene was designed so that participants were likely to use the yellow ball that is to the right of the red ball (labelled 'B') as a landmark, e.g. in an instruction like “pick up the red ball near the yellow ball”. The yellow ball however was affected by a perception error and appeared to the robot as a yellow box. Therefore, if a participants attempted to use a referring expression that involved a description of the yellow ball as a yellow ball, the robot would be unable to resolve the reference. We had intended that the participants would resolve the problem by finding an alternative way to refer to the red ball. They could, for example, figure out how the robot perceived the landmark by using the information request options, or use a direction based description that avoided the landmark.

Unfortunately the system accepted scenes such as the one in Figure 5.3c as valid solutions. In this scene, the participant had not moved the intended object ('A') but the distractor object ('C') to Place 1. This was much easier because the landmark next to the object ('D') was not affected by a perception error. The reason for this was that the target conditions that were used to specify whether or not a scene was completed only checked if there was any red ball on Place 1, but did not check whether the distractor object was still in place.

It is not completely obvious how to account for these examples in the measures. On the one hand the participants did not actually complete the scene successfully. It would therefore not be fair to count them as a successful completion. On the other hand, they also did not abandon the scene. We therefore excluded the affected scenes from the calculation of the Abandon Rate. We assume that the other measures would not be influenced by the outcome in a meaningful way and therefore used the affected scenes in their calculation. In total 24 out of 1198 recorded scenes (about 2%) were affected.



(a) The start scene.

(b) The target scene that was shown to the participants.

(c) A scene that was erroneously accepted as a valid target configuration.

5.3.2 Research Questions and the Measures

To address Research Question 5.2 (*Do perception errors as experienced by the robot have an impact on the difficulty of the task?*), we look primarily at the Abandon Rate and see how it compares between the No Error Phase and the Error Phase. We also look at the Reference Problem Rate and the Undo Rate to determine whether more or fewer problems occurred. As a secondary measure we also attempt to interpret the number of actions and the completion time. To address Research Question 5.3 (*If participants are offered the option to request information about the robot's understanding of the scene, do they use it?*), we look at the number of information requests. We will investigate the use and the effect of information requests in Chapter 6. To address Research Question 5.4 (*Does the ability to use request information about the robot's understanding of the scene have an impact on the participants' ability to solve the task?*), we compare the Error Phase and the Description Phase, Markup Phase and Querying Phase based on the measures used in our investigation of Research Question 5.2. If the measures show a decrease in difficulty this would indicate that the information requests have a positive impact. To address Research Question 5.5 (*How do the information request options compare to each other in terms of effectiveness?*), we compare the results from the Description Phase, Markup Phase and Querying Phase in more detail and determine whether one of the phases stands out from the other ones. Table 5.6 contains a summary of this discussion.

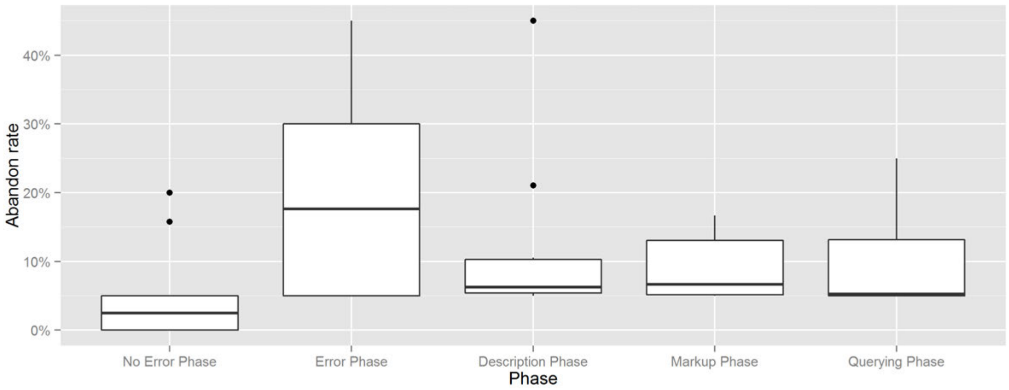


Figure 5.4: Distribution of the rate of abandoned scenes per user in each phase.

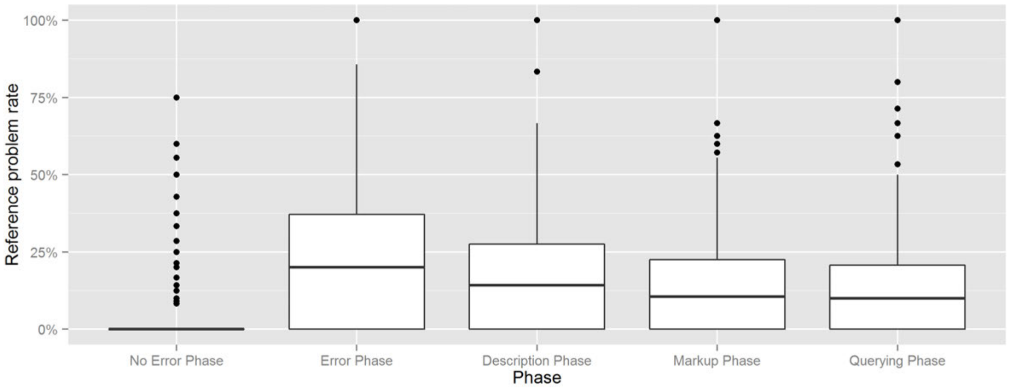


Figure 5.5: Distribution of the rate of reference problems per user in each phase.

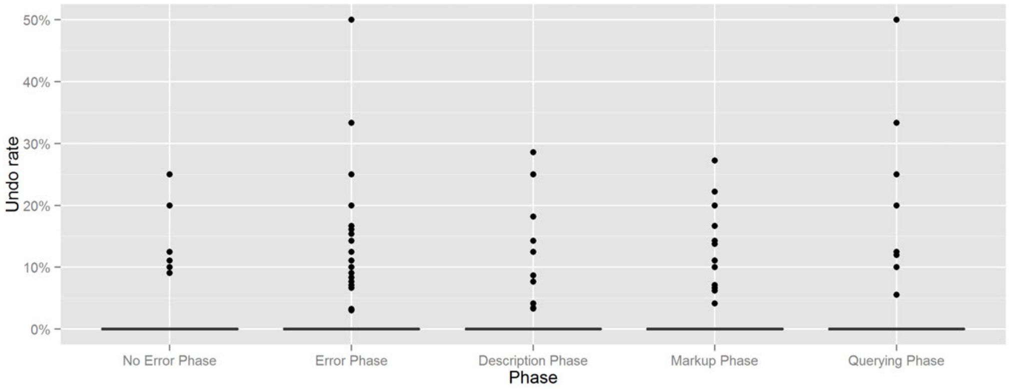
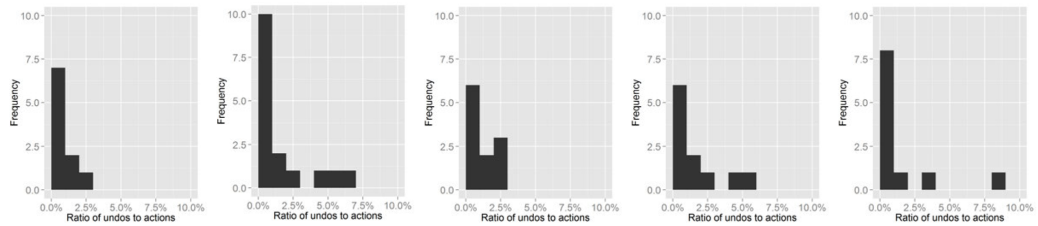


Figure 5.6: Distribution of the rate of actions that were reverted with the undo-function per user in each phase.



(a) No Error Phase. (b) Error Phase. (c) Description Phase. (d) Markup Phase. (e) Querying Phase.

Figure 5.7: Histograms of the percentage of actions that were reverted with the undo function.

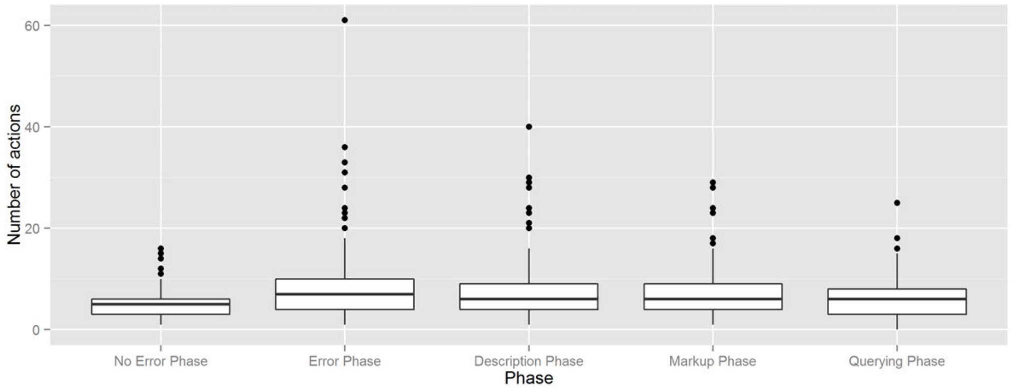


Figure 5.8: Distribution of the number of actions per scene for each phase.

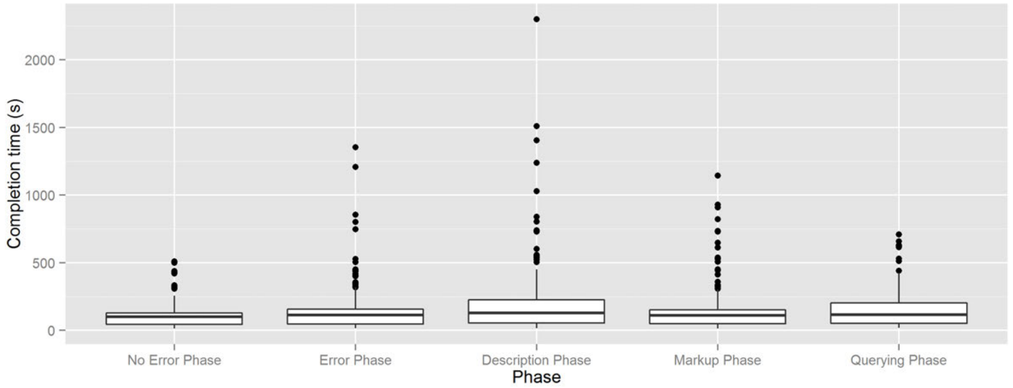


Figure 5.9: Distribution of the time per scene for each phase.

Research Question	**Phases**	**Measures**
RQ 5.2: Impact of perception errors	No Error Phase vs Error Phase	Abandon Rate, Completion Time, Reference Problem Rate
RQ 5.3: Information requests	Error Phase, Markup Phase, Querying Phase	Number of Information Requests
RQ 5.4: Effect of information requests	Error Phase vs Description Phase, Markup Phase and Querying Phase	Abandon Rate, Completion Time, Reference Problem Rate
RQ 5.5: Relative effectiveness of information request options	Description Phase vs Markup Phase vs Querying Phase	Abandon Rate, Reference Problem Rate, Undo Rate, Number of Actions, Completion Time

Table 5.6: Research questions and the relevant phases and measures for their evaluation.

Phase	**Abandon rate**	**Reference problem rate**	**Undo rate**	**Average number of actions**	**SD**	**Average completion time (s)**	**SD**	**Average number of assistance requests**	**SD**
No Error Phase	5.05%	7.82%	0.58%	5.18	2.83	108.64	94.12	0.00	0.00
Error Phase	19.16%	29.23%	1.80%	7.57	5.74	140.28	146.89	0.00	0.00
Description Phase	12.08%	18.86%	0.93%	7.30	5.08	201.47	256.41	0.75	0.83
Markup Phase	9.13%	16.37%	1.70%	6.69	4.33	149.39	164.83	1.05	1.02
Querying Phase	9.72%	15.55%	0.88%	6.20	3.69	150.14	133.10	1.53	2.56

Table 5.7: Measure values for the different phases.

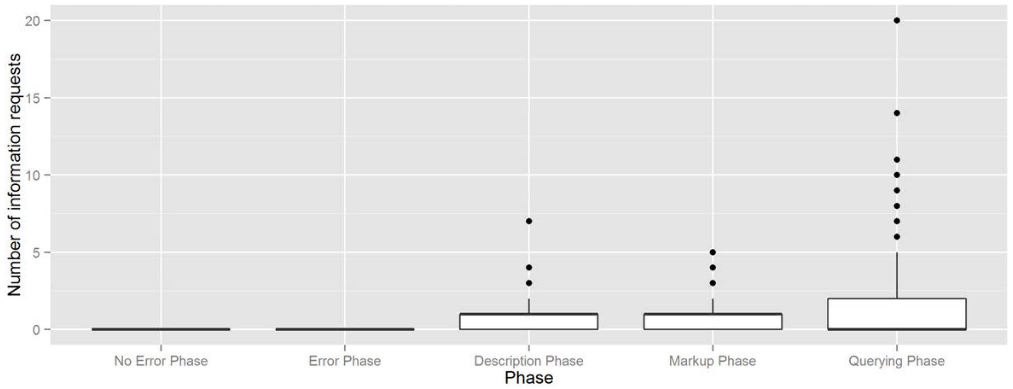


Figure 5.10: Distribution of the number of information requests per scene for each phase.

5.4 RQ 5.2: Impact of Perception Errors

Do perception errors as experienced by the robot have an impact on the difficulty of the task?

Table 5.7 contains a summary of the measures recorded during five phases of the experiment.

- The Abandon Rate shows the percentage of the scenes that was abandoned in a phase (scenes that had an irregular outcome as described in Section 5.3.1 were excluded from the calculation of this value). Figure 5.4 provides a boxplot for the distribution of the percentage of scenes each participants abandoned in each. The figure in the table refers to the total percentage i.e. the number of abandoned scene by the participants divided by the number of all scenes attempted. The boxplot shows the distribution of the percentage of abandoned scenes for each participant.

- The Reference Problem Rate shows the percentage of instructions that contained a problematic reference. Figure 5.5 shows the distribution of the percentage of instructions by a participant that contained a problematic reference in each phase.

- The Undo Rate shows the percentage of the instructions that were reverted with the undo function. The boxplot in Figure 5.6 shows the distribution over all participants. Since the boxplot appears not very informative, we also present the data as histograms in Figure 5.7.

- The Average Number of Actions shows the average number of actions a participant performed in a scene. Figure 5.8 provides a boxplot for the corresponding distribution.

- The Average time shows the average number of seconds per scene, and Figure 5.9 shows the corresponding distribution.

- The Average Number of Information Requests shows how often the participants requested information from the robot per scene on average. Figure 5.10 shows the corresponding distribution.

If we compare the values and the box plots for the No Error Phase and the Error Phase we find that participants on average abandoned more scenes in the Error Phase. They also produced more reference errors on average, reverted more actions with the undo button and spent more actions and more time on scenes. While the differences are clear from the overall averages and distribution visualizations, we also compared the distributions with a Welch Two-Sample t-test (the results are presented in Table 5.8)[1]. We found that the differences were, in fact, all statistically significant at the 95% confidence level except for the Undo Rate and the Average Completion Time.[2] If we compare the values and the box plots for the No Error Phase and the Error Phase we find that participants on average abandoned more scenes in the

[1] More precisely, we compared the distribution of the Abandon Rate for each user, the distribution of the Reference Problem Rate, the Undo Rate and the distribution of the average number of actions and average completion time for each user.

[2] The values for the means of Abandon Rate, the Reference Problem Rate and the Undo Rate in the t-tests are not the same as the values provided in Table 5.7. The values in Table 5.7 refer to global percentages over all scenes that were attempted by all participants, while the values in the t-tests refer to means over the percentage for each individual participant.

	No Error Phase		Error Phase				
Measure	**Mean**	**SD**	**Mean**	**SD**	**df**	**t-value**	**p-value**
Abandon Rate	0.05	0.07	0.19	0.15	24.51	-3.34	< 0.01
Reference Problem Rate	0.08	0.04	0.28	0.10	22.16	-7.31	< 0.01
Undo Rate	0.01	0.01	0.02	0.03	20.09	-1.79	0.09
Average Number of Actions	103.60	25.25	150.53	45.90	24.96	-3.43	< 0.01
Average Completion Time	2172.80	954.98	2789.18	1144.55	21.86	-1.50	0.15

Table 5.8: Data for Welch Two-Sample t-tests between the measures for the No Error Phase and Error Phase.

Error Phase. They also produced more reference errors on average, reverted more actions with the undo button and spent more actions and more time on scenes. While the differences are clear from the overall averages and distribution visualizations, we also compared the distributions with a Welch Two-Sample t-test (the results are presented in Table 5.8)[1]. We found that the differences were, in fact, all statistically significant at the 95% confidence level except for the Undo Rate and the Average Completion Time.[2]

Conclusion:

1. The introduction of perception errors makes the task more difficult.

Aside from the measures related to performance, we also recorded the number of times the system was not able to parse or interpret an instruction

[1] More precisely, we compared the distribution of the Abandon Rate for each user, the distribution of the Reference Problem Rate, the Undo Rate and the distribution of the average number of actions and average completion time for each user.

[2] The values for the means of Abandon Rate, the Reference Problem Rate and the Undo Rate in the t-tests are not the same as the values provided in Table 5.7. The values in Table 5.7 refer to global percentages over all scenes that were attempted by all participants, while the values in the t-tests refer to means over the percentage for each individual participant.

by the user. Overall we found that across all phases, 6.07% of the instructions could not be interpreted. In the No Error Phase, 8.01% could not be interpreted, in the Error Phase 7.39%, in the Description Phase 4.98%, in the Markup Phase 6.71% and in the Querying Phase 6.07%. Overall the differences were not statistically significant except the different between the No Error Phase and the Querying Phase. We do therefore not believe that the number of parser errors was related to the presence or absence of perception errors and the information request options. In order to investigate the effect of adaptation to the system, we ordered the scenes each user attempted by the order in which they were presented, and added up the number of parser errors for each scene with the same index (i.e. the first data point represents the number of errors the participants encountered in the first scene they attempted, the second data point represents the number of errors in the second scene and so on). The results are shown as a scatter plot in Figure 5.11. The index of the scene and the number of errors have a weak negative correlation of about -0.21. This indicates that the number of parser errors the participants encountered slightly decreased the longer they worked on the experiment. This is likely due to the participants learning which inputs the system was able to interpret, and them successfully adapting to them.

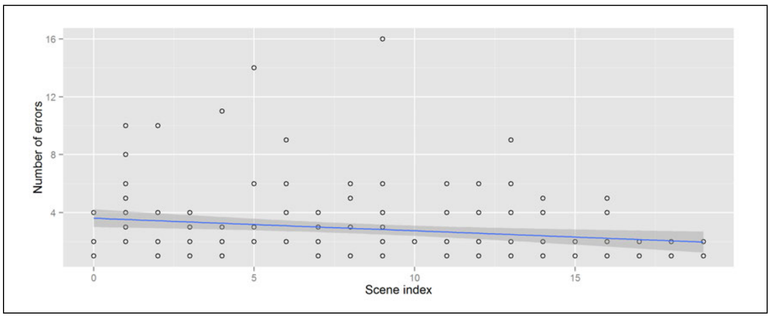


Figure 5.11: A scatter plot of the index of the scenes and the number of parser errors.

5.5 RQ 5.3: Information Request Use

If the participants are offered the option to request information about the robot's understanding of the scene, do they use them?

The average number of information requests per scene shows that the participants tended to request information approximately once per scene (we investigate the distribution in more detail in Chapter 6). The Description Phase has overall the lowest average number of uses while the Querying Phase has the highest average number of uses.

The descriptions and the markup provided a complete description of the robot's understanding of the scene. They also always produced the same output for a given configuration. Participants could therefore not gain additional information by requesting information multiple times in a row. The query information requests on the other hand only provided one piece of information per request. The participants could therefore gain additional information by posing multiple queries about different aspects of the scene. It is therefore quite plausible that participants in the querying condition had to

	Error Phase		Description Phase				
Measures	**Mean**	**SD**	**Mean**	**SD**	**df**	**t-value**	**p-value**
Abandon Rate	0.19	0.15	0.12	0.12	24.40	1.45	0.16
Reference Problem Rate	0.28	0.10	0.19	0.06	25.40	2.84	0.01
Undo Rate	0.02	0.03	0.01	0.01	21.46	1.40	0.18
Average Number of Actions	150.53	45.90	146.09	50.75	19.90	0.23	0.82
Average Completion Time	2789.18	1144.55	4029.36	3215.11	11.66	-1.23	0.24

Table 5.9: Data for Welch Two-Sample t-tests between the measures for the Error Phase and Description Phase.

ask multiple queries to achieve the same level of information the participants in the Description Phase and the Markup Phase achieved with one request.

Table 5.7 shows that the description option was used fewer times than the markup option. The evaluation of the questionnaires responses in Section 5.2 showed that the participants – while still overall satisfied with the description option – were more likely to state that they found the markup option useful. It is possible that these two observations are related and that participants used the description options less because they did not find them as useful.

Conclusions:

1. The participants did use the options available to request information.

2. Participants requested markup more often than descriptions.

3. Participants posed queries more often than they requested markup or descriptions (this is probably related to the fact that queries provide only limited information).

	Error Phase		Markup Phase				
Measure	**Mean**	**SD**	**Mean**	**SD**	**df**	**t-value**	**p-value**
Abandon Rate	0.19	0.15	0.09	0.05	20.58	2.63	0.02
Reference Problem Rate	0.28	0.10	0.17	0.05	25.12	3.80	< 0.01
Undo Rate	0.02	0.03	0.02	0.02	25.75	0.58	0.57
Average Number of Actions	150.53	45.90	133.82	39.84	23.65	1.02	0.32
Average Completion Time	2789.18	1144.55	2987.82	1584.75	16.70	-0.36	0.72

Table 5.10: Data for Welch Two-Sample t-tests between the measures for the Error Phase and Markup Phase.

	Error Phase		Querying Phase				
Measure	**Mean**	**SD**	**Mean**	**SD**	**df**	**t-value**	**p-value**
Abandon Rate	0.19	0.15	0.10	0.07	24.78	2.26	0.03
Reference Problem Rate	0.28	0.10	0.16	0.04	21.31	4.50	< 0.01
Undo Rate	0.02	0.03	0.01	0.03	24.49	0.69	0.49
Average Number of Actions	150.53	45.90	123.91	34.58	25.27	1.75	0.09
Average Completion Time	2789.18	1144.55	3002.82	869.11	25.20	-0.56	0.58

Table 5.11: Data for Welch Two-Sample t-tests between the measures for the Error Phase and Querying Phase.

5.6 RQ 5.4: Effect of Information Requests

Does the ability to request information about the robot's understanding of the scene have an impact on the participants' ability to solve the task?

To determine what effect the information request options had on the task performance, we compare the values presented in Table 5.7. We also performed a t-test to compare the distribution of the values. The results are presented in Table 5.9 (the Error Phase and the Description Phase), Table 5.10 (the Error Phase and the Markup Phase) and the Table 5.11 (the Error Phase and the Querying Phase).

We notice in Table 5.7 that the overall Abandon Rate is higher in the Error Phase than in the Description Phase, the Markup Phase and the Querying Phase (i.e. all the phases in which information request options were available).

As shown in Table the difference is statistically significant at the 95% confidence level for the Markup Phase and the Querying Phase. The p-value for the Description Phase is about 0.16 and falls below the 95% confidence level. The Reference Problem Rate is lower in the Description Phase, the Markup Phase and the Querying Phase than in the Error Phase. The difference between the distributions is statistically significant at the 95% confidence level for all phases.

Similarly, the Undo Rate is also lower in the Description Phase, the Markup Phase and the Querying Phase than in the Error Phase. However, the difference is not statistically significant at the 95%. As we stated earlier, we do not believe that the Undo Rate is a particularly good indicator of difficulties in the task since participants could also manually revert actions without using the undo function (as discussed in Section 5.3).

The average number of actions required to complete a scene is lower in the Description Phase, the Markup Phase and the Querying Phase than in the Error Phase. The differences are not statistically significant at the 95% level however.

Contrary to the average number of actions, the average completion time is overall longer in the phases where information requests are available. However, the differences are not statistically significant.

Conclusion:

1. Participants tend to be more successful at the task if they are able to request information, they encounter fewer problems, and need fewer actions to complete scenes.

The results for the completion time are somewhat surprising because the completion time for the Description Phase, the Markup Phase and the Querying Phase is overall higher on average than the completion time for the Error Phase. We would expect that access to information would make the task easier and enable the participants to solve the tasks more quickly. On the other hand, the average number of actions per scene does not mirror the increase in completion times. In the Markup Phase, for example, the average completion time is higher than in the Error Phase, but the average number of actions is lower. We will investigate this in the next section in more detail.

5.7 RQ 5.5: Relative Effectiveness of Information Request Options

How do the information request options compare to each other in terms of effectiveness?

In the Description Phase of the experiment, the participants were able to ask the system to generate a verbal description of the scene. In the Markup Phase the participants were able to ask the system to visually present its understanding of the scene by marking up objects with their properties in the simulation view. In the Querying Phase, the participants were able to ask the system simple questions about whether or not the robot perceived an object of their description. We were interested in finding out whether one of the options was generally superior to the other ones. A second aspect would be to find out whether one of the options was superior to the other ones in specific aspects of the task.

When we compare the values for the Abandon Rate, the Reference Problem Rate and the Undo Rate in Table 5.7, we find that Description Phase, the Markup Phase and the Querying Phase are close to each other. The Markup Phase and the Querying Phase appear to do slightly better in the Abandon Rate and Reference Problem Rate measures than the Description Phase. In addition to that, the Description Phase has the longest average

	Completed		**Abandoned**	
Phase	**Average time (s)**	**(SD)**	**Average time (s)**	**(SD)**
No Error Phase	94.95	71.75	333.30	126.68
Error Phase	113.23	107.33	257.88	219.26
Description Phase	158.69	149.43	540.08	523.65
Markup Phase	130.44	133.61	318.16	259.68
Querying Phase	130.11	105.2	333.24	194.21

Table 5.12: The average completion times for scenes that were successfully completed and scenes that were abandoned.

completion time with the highest standard deviation.

Conclusions:

1. The description option appears to be the least effective information option.
2. The markup option and the querying option are of similar effectiveness.

In the previous section we noted that the phases in which the participants were able to request information from the robot had longer average scene completion times. Our first hypothesis was that this might be due the fact that the participants abandoned more scenes in the Error Phase than in the other phases. If participants abandoned scenes early on, this might decrease the overall average time spent on scenes. We calculated the average completion times for scenes that were successfully completed and scenes that were abandoned separately for each phase. The results are presented in Figure 5.12. It appears that the participants did in fact abandon scenes earlier in the Error Phase. This, combined with the higher abandon rate probably contributed to the longer average completion times in the phases in which the participants could request information.

Apart from the difference between the Error Phase and the later phases, there is a noticeable difference between the completion times for the Description Phase and the Markup Phase and the Querying Phase. It appears that the participants took longer when they used the descriptions. We were interested to find out if this difference in completion time can actually be attributed to the information request options and in particular whether the longer completion times in the Description Phase can be attributed to the fact that the system had to read out the descriptions — especially as some of the descriptions can become somewhat lengthy.[1] We therefore extracted from the logs for each utterance produced by the system the time it took for the system to read the message out. In Table 5.13 we present for each phase

- The total completion time for each phase (over all participants and all scenes).
- The sum of the length of all utterances produced by the system.
- The average completion time per scene.
- The average of the sum of the length of the system utterances per scene.
- The percentage of the total completion time that was filled with the robot speaking.

[1] We discuss the process of how the descriptions were generated in Section 4.3.5.2 and provide examples of long descriptions there.

Phase	Completion time (s)	Sum of time spoken (s)	Average completion time (s)	(SD)	Average time spoken (s)	(SD)	Average number of information requests	(SD)	Speak time proportion
No Error Phase	21,728.00	1,835.28	108.64	94.12	9.18	5.89	0.00	0	8.45%
Error Phase	47,416.00	5,296.47	140.28	146.89	15.67	13.53	0.00	0	11.17%
Description Phase	44,323.00	5,005.74	201.47	256.41	22.75	19.30	0.75	0.83	11.29%
Markup Phase	32,861.00	2,781.51	149.37	164.83	12.64	9.94	1.05	1.02	8.46%
Querying Phase	33,038.00	3,173.54	150.17	133.10	14.43	10.13	1.53	2.56	9.61%

Table 5.13: Data related to time filled with system speech.

The data shows that in fact, of all the phases, the Description Phase has the longest total time spoken. The contrast is particularly clear in comparison to the Markup Phase. The difference is maintained in the average speech duration per scene. The final column shows what percentage of the total time spent on the task was filled with the system speaking. Again, the Description Phase has the highest value. The value for the Markup Phase on the other hand is roughly equal for the value for the No Error Phase. It is therefore possible, that the amount of time that was taken up by the descriptions contributed to the longer completion times in the Description Phase. Another possible explanation could be that the descriptions were more difficult to use than the other options. The participants needed to listen to the descriptions, relate them to their perception of the scene, and then act upon this. It is plausible that this contributed to the completion times as well.

Conclusion:

1. The longer completion times in the Description Phase may be due to

the fact that the system spent more time presenting the descriptions.

5.8 Summary

In this chapter we investigated the reported experience of the participants in the experiment analysed the performance measures recorded during the experiment. Through analysis of both the subjective reported experience of participants and the objective measures recorded during the experiment we found that both in the experiences and in the measures the introduction of perception errors made it more difficult for the participants to complete the tasks. In particular we found that the errors introduced into the robot's perception of the world were registered by the participants as problems in the robot's communicative capabilities. The option to request information from the robot about its perception of the world helped to ameliorate the effect of the perception errors. We also found that there were differences between the effectiveness of the different information request options. The Markup option and the Querying option appear mostly similarly effective and popular, but the Description Generation option appears to be the least favourite option as well as the least effective one.

One general issue we encountered was that the participants appeared to be less frustrated by the problems in the dialogues than we had expected. As stated earlier, we found that some participants reported that they experienced the problems that arose in the dialogues not so much as frustrating but rather as a challenge or puzzle. This means that the effect of the errors and the information requestion options on the user experience is not as clear as expected. The objective measures nevertheless show that the participants had more difficulties completing the tasks and encountered more problems in general — even if this is not directly reflected in the user experience.

Chapter 6

Information Request Use

In the previous chapter we investigated how perceptions errors affected the experience of the participants in the Toy Block experiment and their ability to complete the tasks. We also investigated whether participants used the information request options that were available in the later phases of the experiment and found that they were indeed used and appeared to increase the participants' satisfaction as well as their ability to complete the tasks.

In the Description Phase they were able to ask the robot to generate a verbal description of its perception of the scene. In the Markup Phase, they could ask the robot to mark up its understanding of the scene on the display in the simulation view. In the Querying Phase they could ask the robot simple questions about whether or not it perceived an object of a given description. In this chapter we investigate how and when the participants made use of these options to request information.

6.1 Research Questions

In this chapter we focus on the following research questions:

Research Question 6.1: *How often did the participants request information?* We investigate how often the participants requested information in each scene, and in the experiment overall. We also compare the number of uses between the different phases to determine whether some of the information request options were used more frequently than the others.

Research Question 6.2: *Did the way the participants requested information evolve during the course of the experiment?* For this question we investigate whether the participants requested information uniformly throughout the experiment, or whether they requested information more or less often in the later scenes of the experiment as compared to the early scenes.

Research Question 6.3: *Under what circumstances did the participants request information?* In particular we identify what happened in the dialogue before the participants requested information.

Research Question 6.4: *What were the effects of sequences of queries?* We found that in the Querying Phase participants tended to ask multiple queries one after another. We investigate whether the success of instructions that followed after multiple queries was related to the number of the preceding queries.

6.2 RQ 6.1: Frequency of Information Requests

How often did the participants request information?

We investigated how often the participants used the information request options in each phase. There were two aspects to this question:

1. How often each participant requested information during the course of an experiment.

2. How often each participant requested information per scene.

6.2.1 Data

We counted the number of times each participant requested a description in the Description Phase, turned on the markup in the Markup Phase and posed a query in the Querying Phase. The results are presented in Table 6.1. The first column contains the mean and median number of times participants requested information during the course of one complete experiment (i.e. the full set of 20 scenes). The second column shows the mean, standard deviation and median number of information requests per individual scene. The third column shows the same information per individual scene, where only scenes are taken into account where the participant requested information at least once are taken into account.

Figure 6.1 contains three boxplots that represent the distribution of the

	Per experiment			Per scene			Per scene (≥ 0)		
Phase	**Mean**	**SD**	**Median**	**Mean**	**SD**	**Median**	**Mean**	**SD**	**Median**
Description Phase	14.91	8.04	13	0.75	0.83	1	1.26	0.72	1
Markup Phase	20.91	10.28	16	1.05	1.02	1	1.55	0.88	1
Querying Phase	30.64	20.54	30	1.53	2.56	0	3.21	2.89	2

Table 6.1: The mean and median number of information requests per scene and per experiment.

total number of times the participants requested information across the full experiment. The plots show that the median number of uses in the Description Phase is close to the median number of uses in the Markup Phase, but slightly higher there. The higher upper quartile also indicates that the participants tended to request more assistance in the Markup Phase. The plot for the Querying Phase indicates that the participants asked queries more often than they requested assistance in the other phases.

Of course it is difficult to compare the number of queries with the number of description and markup requests. The descriptions and the markup provided a complete description of the systems understanding of the scene, while the querying option answered one specific question. It is therefore plausible that participants had to ask multiple queries to obtain the same amount of information they could obtain through one description request or one query.

A second aspect of the frequency of use of the information options is the question of how often the participants used them in each scene, e.g. whether they tended to use them once in each scene to get an overview, or whether they used them more often. Figure 6.3 show histograms of the distribution

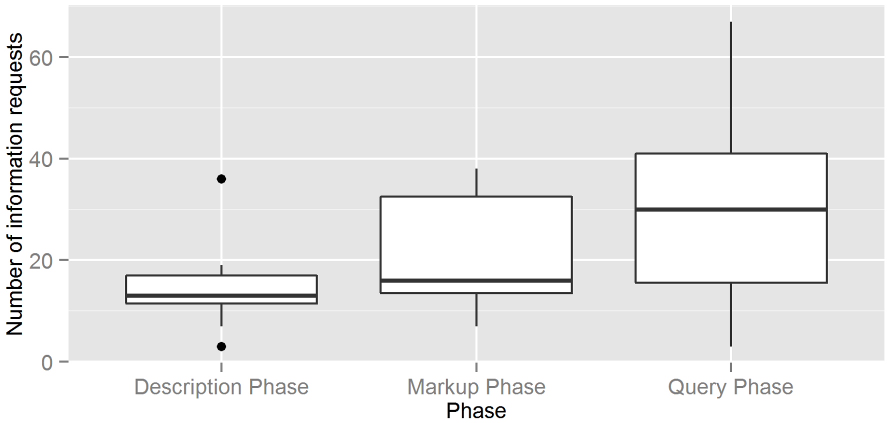
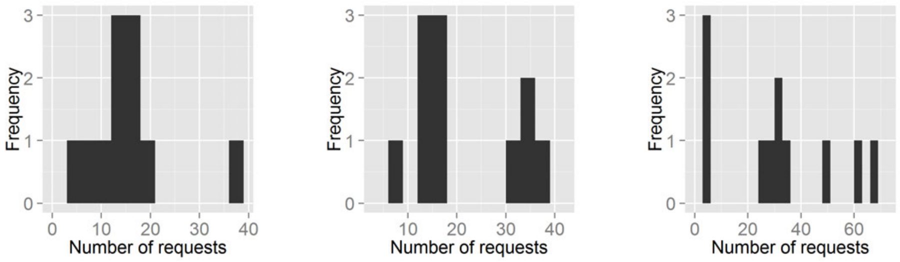


Figure 6.1: Distribution of the total number of information requests in the Description Phase, the Markup Phase and the Querying Phase.

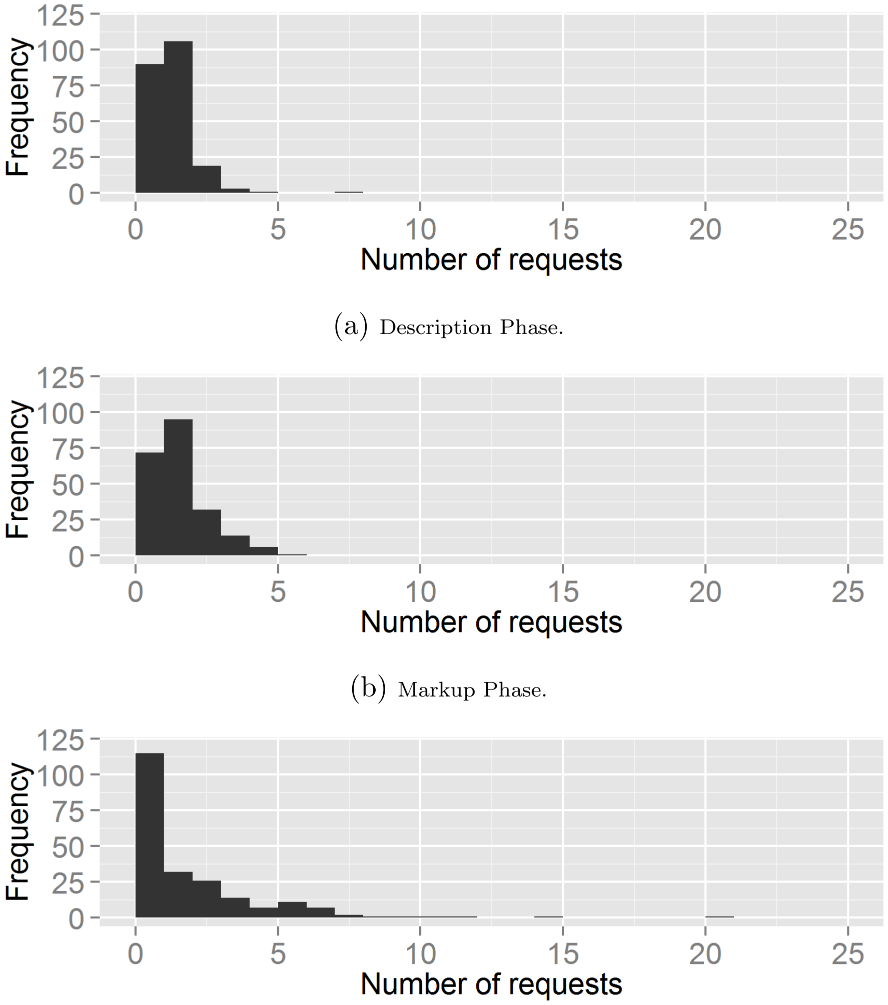


(a) Description Phase (b) Markup Phase (c) Querying Phase

Figure 6.2: Data from Figure 6.1 as histograms.

of the number of uses in each scene.

The plots show that participants tended to request information about 1 time in the Description Phase and the Markup Phase and less often 2 or more times. In the Querying Phase on the other hand, they either did not ask queries or, if they did, they tended to ask multiple queries in one scene. This suggests that there was a higher barrier associated with asking a query. While the description option and the markup option only required the participants to activate a button and then evaluate the output of the information request option, the querying option required them to formulate a targeted question, which arguably required more effort. It is possible that



(a) Description Phase.

(b) Markup Phase.

(c) Querying Phase.

Figure 6.3: Histograms of the number of information requests per scene.

participants therefore were likely to shy away from formulating a query if they did not have a specific problem. However, once they asked a query, they were then likely to ask further queries to work out the problem.

6.2.2 Summary

We investigated the number of times participants requested information either in a single scene or across the entire experiment. We found that the markup option was used more frequently than the description option, and the querying option more frequently than the markup option. Interestingly, the participants tended to ask queries in sequences of multiple queries, while the other options were mostly used only one time in a scene.

It is plausible that the markup option was used more often than the description option because it was faster. The higher variation in the use of the markup option might be also be explained by this. While in the description condition the participants only activated it when necessary because they did not want to sit through the description too often, in the markup condition they could activate it as often as they wanted without any real negative effects. The fact that participants asked multiple queries in a row suggests that participants needed multiple queries to compile enough information to be confident about formulating another instruction.

Conclusions:

1. The participants requested markup more often than descriptions.

2. They asked more queries than they requested markup or descriptions.

3. The participants generally tended to use the markup option and the description option about one time per scene.

4. Participants tended to either not use queries in a scene, or if they did, pose more than one query.

6.3 RQ 6.2: Evolution of the Use of Information Requests

Did the way the participants requested information evolve during the course of the experiment?

For Research Question 6.2 we investigated how often the participants requested information in one scene or in the experiment in total. Another aspect of the frequency of information requests was the question of whether the participants tended to use more information requests in the later scenes of experiments or fewer than in earlier scenes. If they increased the use, this could indicate that they found the information request option useful and chose to use it more frequently. The opposite development could imply that participants, after an initial phase where they tried out the information request option, preferred to not use it later on.

Figure 6.4 shows a plot of the total number of times the participants used the respective information request option in each phase up to a given scene. The x-axis refers to the order in which the scenes were presented, i.e. the point in the plot for the x-axis position 5 describes how often the

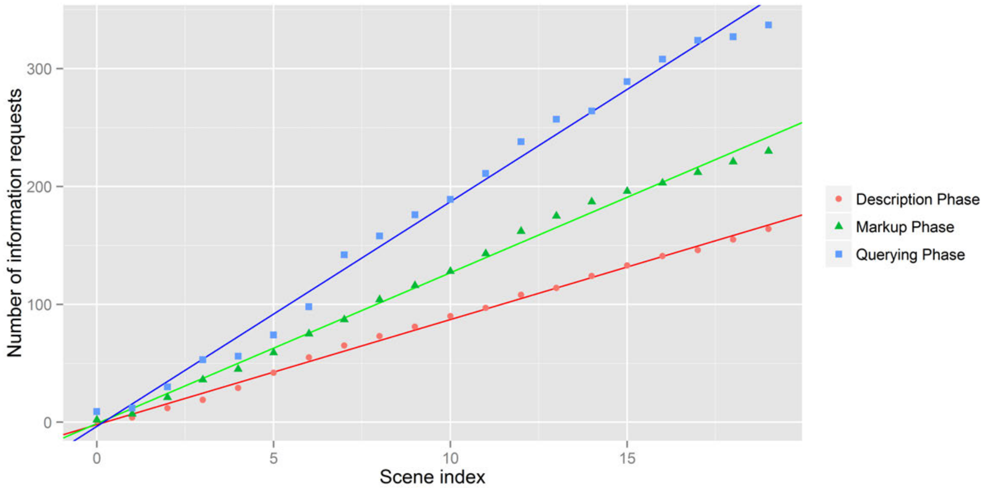


Figure 6.4: Total number of information requests over the course of the experiment. For each phase a regression line is fitted in.

participants had requested information up to (and including) the 5th scene that was presented to them. The order in which the scenes were presented was randomized for each participant (except for the two introduction scenes which were always presented first). Small variations in the graph are therefore not related to the difficulty of individual scenes. Separate series are included for each phase in the experiment. A line representing a linear model that was based on the points is fitted into the graph.

Overall the curves are more or less linear. This suggests that the participants on average did not change their propensity to use the information request options over the course of the experiment much, but requested information uniformly throughout the scenes.

Conclusion:

1. The participants request information uniformly throughout the course of the experiment.

6.4 RQ 6.3: Circumstances of Information Requests

Under what circumstances did the participants request information?

We were interested to find out when and why participants requested information. If we identify points in the interaction at which participants are particularly likely to request information, this could provide clues as to what situations are particularly difficult for the participants. These points could then serve as opportunities at which the robot could pre-emptively offer assistance. On the other hand, it would also be interesting to determine whether there are differences between the circumstances under which the participants request information between the different information options. This could indicate that participants find some types of information particularly helpful in specific situations, while other types of information are more appropriate for other circumstances.

To investigate this issue, we constructed the immediate context of each information request. We extracted for each instance of an information request the interaction event that immediately preceded it (its *predecessor*). We consider as events the most recent action performed by the user and the robot (e.g. an instruction and whether the robot was able to perform the instruction), and other events related to the experiment (e.g. the beginning of a scene). We compare the distribution of the events that preceded information requests with the distribution of events in the data set in general. We also calculate the conditional probability for each event that the following event would be a information request.

6.4.1 Data

We retrieved for each information request (i.e. each description request, markup request and query) the event that immediately preceded it. We call the set of events that preceded the information requests the **predecessor set**. We distinguish 12 event categories.

Resolution problem: The information request was preceded by an instruction which the system could not complete because the instruction contained a reference that was either ambiguous or that the system could not resolve to an object.

Successful instruction: The information request was preceded by an instruction the system was able to interpret and that did not result in an interpretation error.

Beginning: The information request was preceded by the start of the scene (i.e. the information request was the first action in the scene).

No parse: The information request was preceded by an instruction that the system was not able to parse.

Undo: The previous action was undone by the user through the undo-button.

Markup on: The participant turned the markup-assistance on.

Markup off: The participant turned the markup-assistance off.

Description: The participant requested a scene description.

Query: The participant asked the system a query

Pause: The participant activated a pause using the pause button.

Abandon: The participant abandoned the scene.

We then calculated for each event category its proportion in the total set. We created one *predecessor set* for each information request option. As a comparison baseline we determined for each event category how often it was observed in the total data set (i.e. for the "Resolution problem" category we counted how often a reference was made that the robot could not successfully resolve). We call this set of events the **general set**. Table 6.2 shows the distribution of the events in the *predecessor set* for the Description Phase, the Markup Phase and the Querying Phase. It also contains the distribution of the *general set* across all three phases.

We provide a separate summary for each set in Figure 6.5 (for the Description Phase), Figure 6.6 (for the Markup Phase) and Figure 6.7 (for the Querying Phase). We present the most frequent events in the *predecessor set* ordered by their frequency along with the probability that an information request would be the next action after each type of event.

If we assume that the participants' use of the information requests was independent of preceding events in the dialogue, we would expect to observe a distribution of the events over the categories in the *predecessor set* that is

	Description Phase				**Markup Phase**				**Querying Phase**			
	Predecessor		**General**		**Predecessor**		**General**		**Predecessor**		**General**	
Category	**Prop.**	**Count**	**Prop.**	**Count**	**Prop.**	**Count**	**Prop.**	**Count**	**Prop.**	**Count**	**Prop.**	**Count**
Beginning	18.90%	31	10.92%	220	31.74%	73	11.04%	220	7.72%	26	11.35%	220
Description request	1.22%	2	8.14%	164	0.00%	0	0.00%	0	0.00%	0	0.00%	0
Markup on	0.00%	0	0.00%	0	0.00%	0	11.54%	230	0.00%	0	0.00%	0
Markup off	0.00%	0	0.00%	0	4.35%	10	2.06%	41	0.00%	0	0.00%	0
No parse	4.27%	7	2.28%	46	2.17%	5	2.36%	47	4.45%	15	2.48%	48
Pause	0.61%	1	0.10%	2	0.43%	1	0.10%	2	0.00%	0	0.10%	2
Query	0.00%	0	0.00%	0	0.00%	0	0.00%	0	56.97%	192	17.38%	337
Resolution problem	59.76%	98	15.04%	303	38.70%	89	12.09%	241	22.55%	76	10.93%	212
Successful instruction	14.63%	24	62.76%	1264	22.17%	51	59.56%	1187	8.01%	27	57.14%	1108
Undo	0.61%	1	0.74%	15	0.43%	1	1.25%	25	0.30%	1	0.62%	12
Total	100.00%	164	100.00%	2014	100.00%	230	100.00%	1993	100.00%	337	100.00%	1939

Table 6.2: The *general* set and the *predecessor* sets for the Description Phase, the Markup Phase and the Querying Phase.

	Predecessor		General		
Category	Proportion	Count	Proportion	Count	p(request|event)
Resolution problem	59.76%	98	15.04%	303	32.34%
Beginning	18.90%	31	10.92%	220	14.09%
Successful instruction	14.63%	24	62.76%	1264	1.90%
No parse	4.27%	7	2.28%	46	15.22%
Description request	1.22%	2	8.14%	164	1.22%
Pause	0.61%	1	0.10%	2	50.00%
Undo	0.61%	1	0.74%	15	6.67%

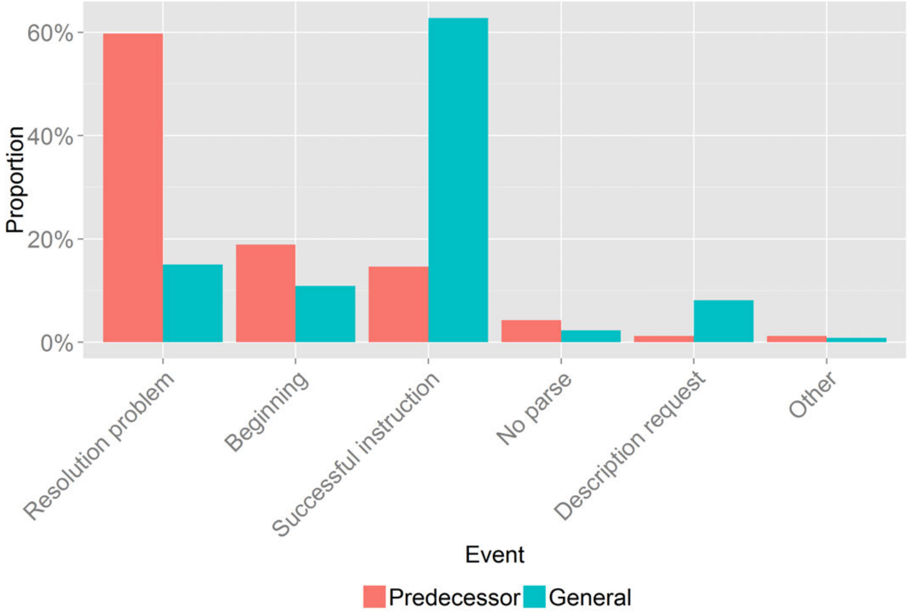


Figure 6.5: Overview of the Predecessor set and the General set for the Description Phase.

	Predecessor		General		
Category	**Proportion**	**Count**	**Proportion**	**Count**	**p(request|event)**
Resolution problem	38.70%	89	12.09%	241	36.93%
Beginning	31.74%	73	11.04%	220	33.18%
Successful instruction	22.17%	51	59.56%	1187	4.30%
Markup off	4.35%	10	2.06%	41	24.39%
No parse	2.17%	5	2.36%	47	10.64%
Pause	0.43%	1	0.10%	2	50.00%
Undo	0.43%	1	1.29%	25	4.00%

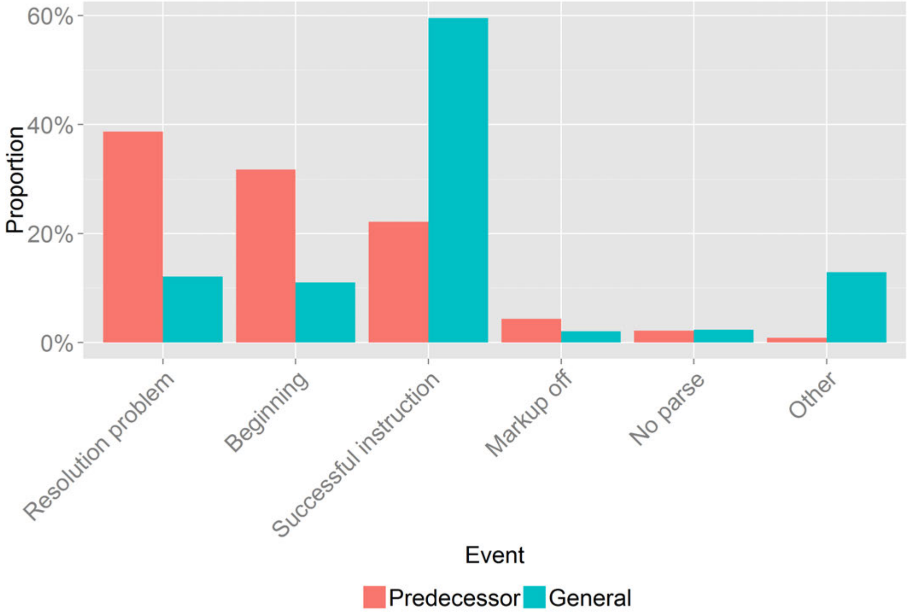


Figure 6.6: Overview of the Predecessor set and the General set for the Markup Phase.

	Predecessor		**General**		
Category	**Proportion**	**Count**	**Proportion**	**Count**	**p(request|event)**
Query	56.97%	192	17.38%	337	56.97%
Resolution problem	22.55%	76	10.93%	212	35.85%
Successful instruction	8.01%	27	57.14%	1108	2.44%
Beginning	7.72%	26	11.35%	220	11.82%
No parse	4.45%	15	2.48%	48	31.25%
Undo	0.30%	1	0.62%	12	8.33%
Pause	0.00%	0	0.10%	2	0.00%

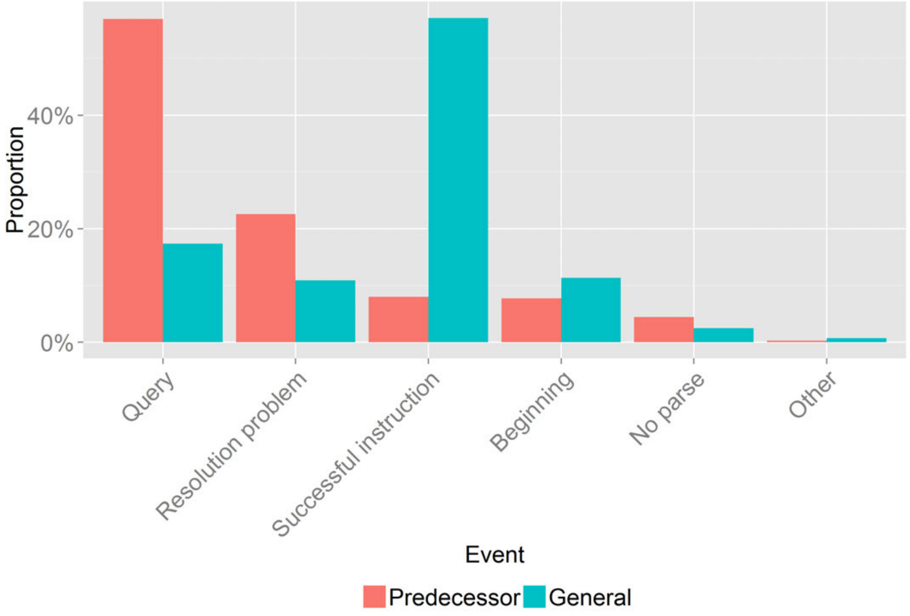


Figure 6.7: Overview of the Predecessor set and the General set for the Querying Phase.

equivalent to the distribution over the categories in the *general* set. We would expect the values for each category to more or less match up. If the values for the *predecessor set* are higher than the ones for the *general* set, this would indicate that participants tended to request information more often after this type of event. The opposite case, if the values for the *predecessor* set are lower than the values in the *general* series, would indicate that participants request information less often after events of the category.

Based on the data from the *general* set and the *predecessor set*, we calculated for each event the conditional probability that the next event would be an information request. The results are presented in Table 6.3. The final row contains the prior probability of an information request in each phase (highlighted in bold in the table).

6.4.2 Analysis

We investigate the distribution of the *predecessor set* in each phase and compare it to the distribution of the *general* set. We also investigate the conditional probability for information requests following the events in question.

6.4.2.1 Description Phase

Almost 60% of the events preceding a description request were instructions that resulted in a resolution problem. In the *general set*, this type of event amounted to only about 15%. This means that resolution problems occurred more frequently before description requests than in general. Table 6.3 shows that in the Description Phase a description request followed resolution prob-

Preceding event	**Description request**	**Markup request**	**Query**
Beginning	14.09%	33.18%	11.82%
Description request	1.22%	0.00%	0.00%
Markup on	0.00%	0.00%	0.00%
Markup off	0.00%	24.39%	0.00%
No parse	15.22%	10.64%	31.25%
Pause	50.00%	50.00%	0.00%
Query	0.00%	0.00%	56.97%
Resolution problem	32.34%	36.93%	35.85%
Successful instruction	1.90%	4.30%	2.44%
Undo	6.67%	4.00%	8.33%
Baseline	**8.14%**	**11.54%**	**17.38%**

Table 6.3: The conditional probability that each event would be followed by an information request.

lems in 32.34% of the cases. Description requests only made up 8.14% of the observed events. This indicates that participants were more likely to request a description after a resolution problem.

Successful instructions made up more than 60% of the events in the *general set*, but they only made up about 15% of the events that preceded description requests. Only in 1.90% of the cases participants requested a description after a successful instruction. This indicates that participants were less likely to request a description after a successful instruction.

For the *beginning* category we observe that it was more likely to precede description requests than expected from the general distribution, and conversely that description requests more frequently followed the beginning of scenes than other actions (14.09% vs 8.14%).

One value stands particularly out in Table 6.3. Participants requested a description after they activated the pause button in half the cases. While it is quite plausible that some participants requested a pause when they were faced with a confounding problem in the task and subsequently requested a description to address the problem, pauses were requested only 2 times during this phase of the experiment. We should therefore not attribute too much importance to this observation.

6.4.2.2 Markup Phase

The distribution of the *predecessor set* in the Markup Phase is overall similar to the distribution in the Description Phase. The three most frequent events in the *predecessor set* are in the same order. Again, resolution problems are frequent before markup requests. Also, successful instructions are less frequent before markup requests than we would expect based on the *general* set.

There is an interesting difference for the *beginning* category between the Description Phase and the Markup Phase. While it occurs more frequently before information requests than we would expect in the Description Phase (18.90% vs 10.92%), the difference is more marked in the Markup Phase (31.74% vs 11.04%). We also find that the probability that a markup was requested at the beginning of a scene is higher than the probability that a description is requested at the beginning of scene (33.18% vs 14.90%). This might indicate that participants were willing to briefly activate markup at the beginning of scenes, to check for problems before they commenced to work, while they preferred to use the description when they were actually faced with a problem.

Overall we find that the circumstances in which the participants requested information are similar in the Description Phase and the Markup Phase.

6.4.2.3 Querying Phase

The distribution of the *predecessor set* in the Querying Phase is similar to the distributions in the Description Phase and the Markup Phase in that resolution problems precede queries more often than we would expect from the *general set*, and less frequent after successful instructions. However, a major interesting difference is that the most frequent event before a query was another query. This is supported by the fact that the probability that participants asked a query after they had already asked a query was 56.97%.

This indicates that participants often requested multiple queries after another. This is markedly different from the other phases. The probability that participants requested a description after another description was only 1.22%.

6.4.3 Summary

We find in general that participants tend to request information after the robot could not perform an instruction due to a resolution problem and at the beginning of scenes. After a successful instruction they are less likely to request information. Both observations are intuitively plausible. When the robot encountered a resolution problem, the participants were faced with the possibility that their view of the world was different from the robot's view of the world. They therefore had a reason to request information about how the robot perceived the scene. The beginning of scenes generally appears as a useful point in the task to request information before taking any actions.

One particular observation was that participants tended to ask multiple queries after another in the Querying Phase. We will investigate sequences of queries in the next section.

Conclusions:

1. Participants tended to request information after a resolution problem and at the beginning of scenes.
2. Participants tended not to request information after an instruction had been successfully completed.
3. Participants often pose multiple queries in a row.

6.5 RQ 6.4: Query Sequences

What were the effects of sequences of queries?

As discussed in the previous section, we found that participants in the querying condition were particularly likely to ask multiple queries one after another. This is in contrast with the other phases. For example, participants rarely requested another scene description after they had requested a description in the Description Phase. We therefore investigate instances where participants asked a sequence of queries. The fact that the participants asked multiple queries indicates that they were faced with situations in which they believed there to be a divergence between their understanding of the scene and the robot's, and that they requested information in order to be able to formulate a successful instruction. We therefore investigate the instruction that follows after a query sequence and examine whether or not it was a successful instruction.

6.5.1 Data

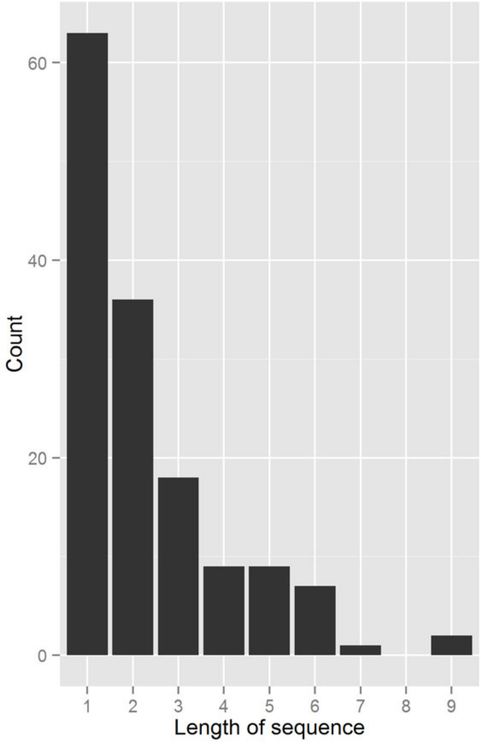
We extracted **query sequences** from the corpus in the following way:

1. Each query action formed a query sequence of length 1.
2. If multiple queries were contiguous we combined them into a query sequence with a length equal to the number of sequences that were combined.

In total there were 337 queries in the data set for the Querying Phase. The queries formed 145 sequences. The sequences had an average length of 2.3 and a median length of 2. Figure 6.8 shows the distribution of the length of the detected sequences.

We select the first instruction the participants issued after a query sequence and investigate whether the robot was able to perform the action requested in the action. We distinguish four possible outcome situations:

OK: The participant was able to formulate a valid instruction (This does not necessarily imply that the system performed the action perfectly the



Length	Proportion	Count
1	43.45%	63
2	24.83%	36
3	12.41%	18
4	6.21%	9
5	6.21%	9
6	4.83%	7
7	0.69%	1
8	0.00%	0
9	1.38%	2
Total		145

Figure 6.8: Length of query sequences.

way it was intended by the participant). This also includes instructions that were valid, but could not be performed due to reasons outside of reference resolution (e.g. in one case the robot rejected a valid pick-up instruction because it was already holding an object).

Resolution problem: The participant formulated an instruction, but the system was not able to perform the requested action, e.g. because the instruction contained an ambiguous or unresolvable expression.

Abandon: The participant abandoned the scene after the query sequence. This should not necessarily be interpreted as an indication that the information request options did not provide sufficient information. Instead it can actually be the case that the participants decided based on the information provided that they would not be able to complete the scene and moved on.

No parse: The system was not able to derive an interpretation for the input.

Table 6.4 shows the distribution of the outcomes for query sequences of length 1, 2, 3 and 4. For comparison, Table 6.5 contains the equivalent information for the Description Phase and Markup Phase (i.e. the outcome of the action following either description requests or markup requests). The participants very rarely requested two or more descriptions in a row (or turned the markup on again after they had turned it off just before). We therefore report only the outcomes after sequences of length 1.

We state for each sequence length the number of times each outcome was observed. Based on this we calculated the proportion of successful instructions and instructions that could not be performed due to resolution problems.

	Querying Phase							
	Length 1		**Length 2**		**Length 3**		**Length 4**	
Outcome	**Count**	**Proportion**	**Count**	**Proportion**	**Count**	**Proportion**	**Count**	**Proportion**
OK	43	68.25%	44	61.11%	39	72.22%	28	77.78%
Resolution problem	13	20.63%	6	8.33%	3	5.56%	0	0.00%
Abandon	4	6.35%	6	8.33%	6	11.11%	4	11.11%
No parse	3	4.76%	16	22.22%	6	11.11%	4	11.11%
Total	63	100%	72	100%	54	100%	36	100%

Table 6.4: Outcomes of instructions after queries.

	Description Phase		Markup Phase	
	Length 1		Length 1	
Outcome	**Count**	**Proportion**	**Count**	**Proportion**
OK	109	68.13%	133	57.83%
Resolution problem	36	22.50%	36	15.65%
Abandon	4	2.50%	4	1.74%
No parse	11	6.88%	57	24.78%
Total	160	100%	230	100%

Table 6.5: Outcomes of instructions after one information request in the Description Phase and the Markup Phase.

6.5.2 Analysis

Overall we find that the likelihood that participants formulated a successful instruction appears to increase as the length of the query sequences increases. As we discussed in Chapter 5, the Reference problem rate for the Querying Phase was 15.55%. This means that if a participant gave an instruction to the system, and the system was able to parse the instruction, the robot encountered a resolution problem in 15.55% of the cases (or, inversely, was able to perform the requested action in 84.45% of the cases). If we compare this value to the proportion of successful instructions after query sequences, we find that after sequences of length 1, the participants appear slightly less successful[1]. However, the success rate is at a level that is comparable to the success rate after one description requests or after the markup has been turned on.

[1] It may appear counter-intuitive that instructions after one query should be less successful than instructions in general. However, this becomes clear if we take into account that queries were particularly often used after instructions that resulted in resolution errors, which are an indication of the presence of a perception error. Instructions after queries were therefore likely to also be affected by perception errors

As the sequence length increases however, the success rate increases as well, until it approaches the general rate. This suggests that participants accumulated information over multiple queries and used the combined information to formulate successful instructions.[1]

6.5.3 Summary

We investigated sequences of queries in the Querying Phase. We found that as the length of query sequences increased, the success of instructions following the sequences increased as well.

Conclusions:

1. Participants were more successful after posing multiple queries.

6.6 Summary

In this chapter we investigated how participants used the information request options. We found that the participants tended to use the markup option more often than the description option. The participants tended to ask queries even more often than they requested markup.

We then found that participants tended to use information requests uniformly throughout the experiment, i.e. we observed no tendency that the participants increased or decreased the use of information requests in the later stages of the experiment.

[1] If we exclude the *Abandon* and *No parse* outcome from the calculation, the 76.79% of the references are successful after one query, 88.00% after a sequence of two queries, 92.86% after three queries and 100% after four.

When we investigated the circumstances under which the participants requested information, it was noticeable that participants tended to request information after the robot encountered a resolution problem. They also often asked multiple queries in sequence. We investigated these sequences of queries and found that the length of a query sequence appears to contribute towards the success of an instruction following the sequence. This suggests that participants used the querying option to accumulate information about the robot's perception of the world and gradually build a model of the robot's understanding of the scene, and that the more complete the model is, the more likely participants are to be able to formulate a successful instruction.

In the previous chapter we investigated the effect of errors in perception and the information request option on user satisfaction and task success. We elaborated this investigation in this chapter by investigating how often and under what circumstances the participants used the information request options.

In the following chapters we are going the investigate in more detail how participants acted when they encountered problems that were due to perception errors. In the first step we are going to investigate in Chapter 7 what actions participants performed after they encountered a problem.

Chapter 7

Dialogue Structures in Problem Resolution Sequences

In this section we examine data from the Toy Block experiment that was discussed in the previous chapters at a structural level to investigate how the participants reacted to perception based problems in the dialogue, and how they resolved the problems. We were particularly interested to observe how participants reacted when they encountered situations in which the robot experienced resolution problems. In the context of the dialogue system used in the experiment, a resolution failure could present itself as either an ambiguous reference (i.e. the robot could not resolve a referring expression in an instruction to a unique reference) or an unresolvable reference (i.e. the robot could not find any objects that fit the given referring expression).

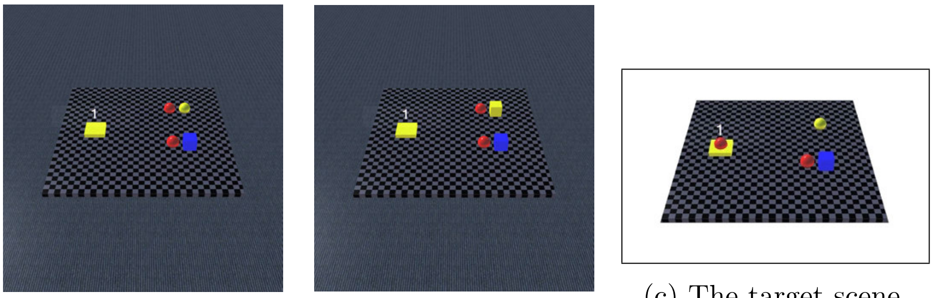
To gain a better understanding of how the participants react to these problems, and to understand how they attempted to resolve them, we analyse the actions the participants performed after they encounter a problem in the dialogue that is due to a a reference resolution problem caused by one of the perception errors that we deliberately introduced into the system. We extract sequences of actions that occur after a perception error occurred and analyse whether they resulted in a resolution of the problem, i.e. whether the user was able to fulfil their original intent eventually despite the problem. We subsequently attempt to identify recurring structures in the actions the participants used in their resolution attempts.

7.1 Research Questions

In this section we address the following research questions:

Research Question 7.1: *How successful were the outcomes of the resolution attempts?* – With this question we attempt to discover whether the resolution attempts were more or less successful depending on the method in which the participants could request information from the robot. This question will inform us about the effectiveness of the different information options.

Research Question 7.2: *How long did the resolution attempts take?* – With this question we attempt to discover whether there were differences between the phases of the experiment in terms of the length of the resolution sequences. This is another aspect of the effectiveness of the information options.



(a) The scene as it is presented to the user.

(b) The scene as it is perceived by the system.

(c) The target scene.

Figure 7.1: The user and system view of a scene ((c) shows the target scene).

Research Question 7.3: *What structures can be observed in the resolution attempts?* – With this question we attempt to discover structures that are frequent or shared between the observed resolution attempts. These structures would form common approaches or strategies towards solving the types of problems the participants encountered in the dialogues.

7.2 Resolution Sequences

In this part of the analysis we focus on problems in the dialogue that arise from perception errors we introduced. Since we deliberately designed the errors we can anticipate the conditions under which they can lead to problems in the dialogue, and we can detect whether or not the participant was able to resolve the problem and achieve their original intention. To do this, we undertake the following steps:

1. We detect utterances in which a participant attempted to refer to an object affected by a perception error in a way which resulted in a problem when the robot attempted to resolve the reference.

2. We then attempt to identify in the subsequent interaction the utterance through which the participant managed to achieve their initial intention, indicating that the reference resolution problem had been resolved.

3. We collect all actions (i.e. utterances by the user and information requests) that take place between the first occurrence of the problem and its resolution and evaluate them.

We call the sequence of actions between the occurrence of a problem and its resolution a **resolution sequence**. We call the reference used in the action that started the sequence the **initial reference**. If the sequence was ended successfully, we call the reference in the final action the **final reference**.

To further illustrate this we present an example of the process of extracting resolution sequences. Figure 7.1 shows an example scene from the experiment. Figure 7.1a shows the scene as it was presented to the user at the beginning of the experiment. Figure 7.1b shows the scene as it was perceived by the system after the introduction of an error. In this example the robot makes a perception error by perceiving the yellow ball in the top right-hand corner as a yellow box. The target scene that was shown to the user is presented in Figure 7.1c. To successfully complete the scene the user had to instruct the system to pick up the red ball that is next to the yellow ball in the user scene (and next to the yellow box in the robot's perception of the scene), and then to put it on Place 1. Since there were two red balls present in the scene, the user would not be able to uniquely specify the ball using an instruction with a referring expression that included only basic attributes such as *"Pick up the red ball"*. If they did attempt this, the system would respond that it found the instruction ambiguous:

U: Pick up the red ball

S: Sorry, there is more than one red ball, and I'm not sure which one you mean.

To uniquely identify the ball, the user would have to use other attributes (namely spatial attributes such as *relative attributes* or *directional attributes*) to describe the ball. We designed the scene so that the most obvious approach would be to use the yellow ball right next to the target red ball as a landmark in an expression, for example: *"Pick up the red ball near the yellow ball"*.

The yellow ball however was affected by a perception error and perceived as a yellow box by the system. Attempts to use the ball as a landmark would therefore have to fail:

U: Pick up the red ball near the yellow ball

S: Sorry, I can't find any red balls that fit your description.

A resolution sequence then consisted of all actions the user performed after they encountered the problem until they either resolved it or abandoned the scene. For example, a participant could use the spatial location of the object in the scene to describe it:

U: Pick up the red ball near the yellow box.

S: Sorry, I can't find any red balls that fit your description.

U: Pick up the red ball in the back

S: Okay, I will pick it up

A participant with access to an information request option could also request information from the robot, and base their resolution attempt upon this information:

U: Pick up the red ball near the yellow box

S: Sorry, I can't find any red balls that fit your description.

(User requests a description.)

S: There is a red ball to the left of a blue box on the bottom right. There is a red ball to the left of a yellow box on the top right. There is a place named place 1 on the left centre.

U: pick up the red ball near the yellow box.

S: Okay, I will pick it up.

We interpret each successful resolution sequence as an example of a user discovering and resolving a perception based problem in the dialogue.

7.2.1 Resolution Sequence Extraction

In the first step towards extracting resolution sequences we needed to determine the beginning of the resolution sequences. To do this, we needed to be able to identify instances in the dialogues where a participant attempted to refer to an object that was affected by a perception error in such a way that the robot would encounter a reference resolution problem.

For each object that was affected by a perception error we created a list of referring expressions that a participant who is not aware of the robot's perception problems would use to describe the object and that also includes the attributes of the object that are affected by the perception error.

For each of these expressions we then created an **expression template**. The templates specified the properties that were used in the original expression and could be matched against the expressions that were recorded in the participant's utterances during the experiment. Figure A.5 and Figure A.6 in Appendix A contain the expression templates that were used.

To detect the successful end of a resolution sequence we needed to determine when the robot picked up the object intended in the instruction that started the sequence. In order to do that we associated with each expression the *id* of the referent object in order to identify it in the following interaction. To find the resolution of the problem we therefore searched instruction-action pairs in which the system's action plan contained a pickup-action that affected the object with the id associated with the expression. In the following we provide an example for the scene that was used in the example in the previous section (shown in Figure 7.1).

To complete the scene, the participants were required to pick up the red ball next to the yellow ball and then put it on Place 1. The scene was designed in such a way as to induce the participants to describe the red ball with a referring expression that contained the yellow ball as a landmark (e.g. “Pick up the red ball left of the yellow ball” or “Pick up the red ball near the yellow ball”). We introduced a perception error that made the yellow ball appear as a yellow box to the robot. To detect the beginning of resolution sequences, we therefore had to look for instances where the participants attempted to pick up the ball using an expression that contained the yellow ball as a landmark. We therefore set up an expression template that matched any referring expression that contained a landmark reference that involved the yellow ball. Presented as a feature structure in the same format we use for referring expressions (Section 2.3) they appear as follows:

$$\begin{bmatrix} \text{type} & \text{typeT} \\ \text{colour} & \text{colourT} \\ \text{rel} & \begin{bmatrix} \text{reltype} & \text{relation} \\ \text{relatum} & \begin{bmatrix} \text{type} & \text{typeLM} \\ \text{colour} & \text{colourLM} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The attribute values represent the following:

- **typeT:** The type of the target object of the expression.

- **colourT:** The colour of the target object of the expression.

- **useLM:** Whether or not a landmark is used.
- **relation**: The relation to the landmark (if one is used)
- **typeLM:** The type of the landmark object (if one is used).
- **colourLM:** The colour of the landmark object (if one is used).
- **objID:** The ID of the target object.

An expression template is compared to an observed referring expression by matching the entry in each slot to the corresponding slot in the referring expression. A slot matches if the observed value is compatible with the value specified in the slot. A template matches if all slots match. Any of the slots can be filled with a wild card character (*). It represents a value that matches all possible values in the expression. For the example scene we set up the following expression template:

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & \text{ball} \\ \text{colour} & \text{yellow} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball1} \end{bmatrix}$$

In the compact notation it appears as follows:

$$\langle *, * \rangle \ \textit{Rel: * } \langle \textit{ball}, \textit{yellow} \rangle\textit{,true,ball1}$$

7.3 RQ 7.1: Resolution Sequence Outcomes

How successful were the outcomes of the resolution attempts?

7.3.1 Data

We used our set of expression templates to detect beginnings of resolution sequences. In total 128 instances of expressions that were affected by a perception error were found in the Error Phase; 87 instances were found in the Description Phase; 58 were in found in the Markup Phase; 74 were in found in the Querying Phase.[1]

We subsequently determined for each sequence how it was resolved. We collected the actions following the problem until one of the following occurred:

1. The robot picked up the object that was the original target. We denote this outcome as **success**.

2. The user abandoned the scene. We denote this outcome as **abandon**.

3. The user managed to fulfil the success conditions of the scene without actually moving the target object. We denote this outcome as **other**.

The **other** outcome refers to scenes where the system erroneously accepted configurations as complete scenes that did not actually match the target scene (we discuss this in Section 5.3.1). As discussed previously, we did not count these cases as successfully completed scenes or abandoned scenes

[1] In the No Error Phase no perception errors had been introduced therefore no resolution sequences occurred.

	Success		**Abandon**		**Other**		**Total**
Phase	**Prop.**	**Count**	**Prop.**	**Count**	**Prop.**	**Count**	**Count**
Error Phase	64.1%	82	29.7%	38	6.3%	8	128
Description Phase	80.5%	70	13.8%	12	5.7%	5	87
Markup Phase	69%	40	15.5%	9	15.5%	9	58
Querying Phase	74.3%	55	20.3%	15	5.4%	4	74

Table 7.1: The number of resolution sequences found in each phase and the percentage of their outcomes.

but as a third outcome that did not belong into either category. Table 7.1 provides a summary of the number of sequences found and their outcomes.

7.3.2 Summary

The results show that there were more successful resolution sequences in the Description Phase and the Querying Phase than in the Error Phase. This is consistent with the observations from Chapter 5 where we showed that interactions were generally more successful in the Markup Phase and the Querying Phase. In that section we showed that the introduction of information request options increases task success and reduces the effort that was necessary to complete a scene. The results in the current section suggest that one of the reasons that interactions were more successful in the Description Phase, the Markup Phase and the Querying Phase was that if the participants encountered problems, they were more likely to be able to resolve them if information request options were available.

It is noticeable that the Markup Phase has a lower success rate than the Description Phase and the Querying Phase. This can be explained by the higher rate of *Other* outcomes, because the *Abandon* rate is not much higher. It is hard to determine what the reason for this higher rate of irregular outcomes is. We speculate that is might be related to how the markup option presented the information.

It showed all objects equally, including the objects that some participants used to complete scenes instead of the intended objects (as discussed in Section 5.3.1. It is possible that they therefore appeared more available than in other conditions.

Conclusion:

1. Dialogue problems arising from perception problems are more easily resolved in the conditions where the participants could request information about the robot's perception of the scene.

7.4 RQ 7.2: Resolution Sequence Length

How long did the resolution attempts take?

The results in the previous section show that being able to request information from the robot increases the participants' ability to resolve perception based problems. In this section we investigate whether they also decrease the amount of effort necessary to resolve the problems by measuring the length of each resolution sequence.

Phase	Success	(SD)	Abandon	(SD)	Other	(SD)	All	(SD)
Error Phase	4.52	3.31	8.13	7.99	6.13	3.14	5.70	5.41
Description Phase	3.10	1.65	5.75	7.53	3.60	0.80	3.49	3.30
Markup Phase	2.63	1.35	3.89	2.60	6.44	7.57	3.41	3.62
Querying Phase	3.71	2.51	5.27	2.79	6.25	4.02	4.16	2.79

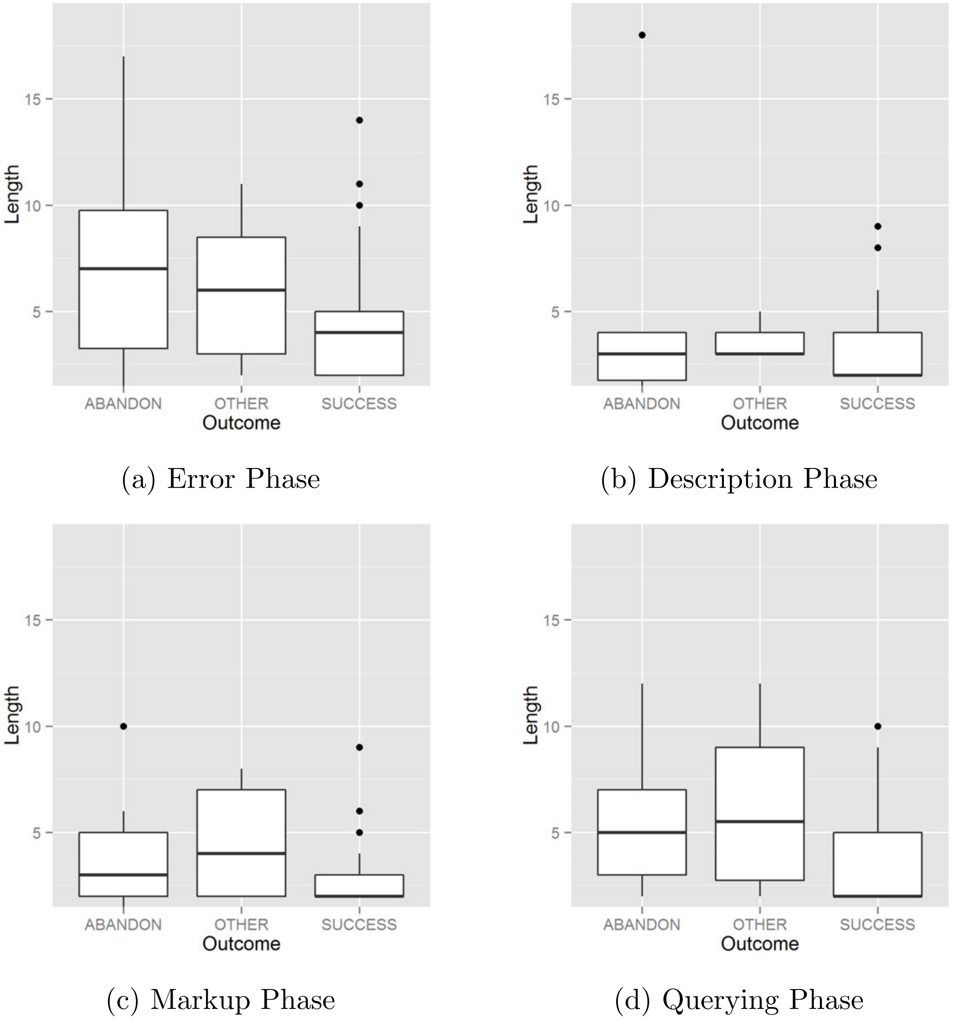
Table 7.2: Average lengths and standard deviations of the resolution sequences.

7.4.1 Data

We counted for each sequence the number of user instructions it included. We only counted instructions in which actions were requested. Information requests or actions like clicking the undo-button were not included. The length of each sequence therefore represents how much effort the participants had to spend before they reached the sequence's conclusion. For example, for a **success** sequence, it shows how much work the participants spent to successfully create the target configuration. For an **abandon** sequence on the other hand it indicates how much effort the participants spent on the problem until they decided that they would not be able to solve it. Table 7.2 presents the average length of each the resolution sequences for each outcome. Figure 7.2 contains box plots showing the distribution of the lengths for each phase and outcome.

7.4.2 Summary

The results in Table 7.2 and Figure 7.2 show that successful resolution sequences were the longest in the Error Phase on average. The Error Phase



(a) Error Phase

(b) Description Phase

(c) Markup Phase

(d) Querying Phase

Figure 7.2: The distribution of the lengths of the resolution sequences.

also has the highest standard deviation. The average lengths are shorter if any kind of information option is available. Among the phases where participants were able to request information, the Querying Phase has the longest average resolutions sequences as well as the highest standard deviation. The results appear plausible. We performed ANOVA tests for each outcome, and found that only for the successful sequences statistically significant differences exist with an F value of 6.78 and a p value below 0.01. A post-hoc Tukey test showed that there were statistically significant differences between the sequences from the Error Phase and the Description Phase and the Error Phase and the Markup Phase p values below 0.01 in both cases. This indicates that if participants could request markup or descriptions, they were able to resolve problems quicker at a statistically significant level.

In the Error Phase the only way to resolve a problem was to use a trial-and-error approach. In the other phases the participants could request information and attempt more informed approaches. In the Description Phase and the Markup Phase the descriptions and the markup provided all the information that was available in one action. In the Querying Phase however, the queries only provided information that was explicitly requested. It is therefore plausible, that participants had to ask multiple queries to form a sufficient idea of the robot's understanding of the scene. This would explain why Querying Phase has longer successful resolution sequences as well as a higher standard deviation.

Conclusion:

1. Resolution sequences tend to be shorter if participants can request information about the robot's perception options are available.

7.5 Dialogue Act Sequences

Each resolution sequence describes the actions one participant took to resolve one particular problem. We investigate the sequences under two perspectives:

1. From a structural perspective we investigate the sequences as sequences of actions and identify common structures.
2. From a content perspective we investigate the referring expressions the participants chose in their attempts to resolve the problem, and how they modified the expressions when they encountered problems.

In this section we perform a structural analysis of the sequences. The content based analysis will be performed in the following chapter.

We form an abstraction over the dialogues by abstracting them into **dialogue acts**. We define the set of dialogue acts as follows:

pickup: The participant successfully instructs the system to pick up an object.

move: The participant instructs the system to move an object the robot is holding to a given location.

put: The participant instructs the system to put down an object the robot is holding.

description: The participant requests a description of the scene (this was only possible in the Description Phase).

markupon: The participant turns the markup information on (this was only possible in the Markup Phase).

markupoff: The participant turns the markup off (this was also only possible in the Markup Phase).

query: The participant makes a query (this was only possibly in the Querying Phase).

pause: The user clicks on the pause button.

Apart from these dialogue acts, we define three further elements that represent the beginning and end of resolution sequences:

init: This refers to the action that initializes the resolution sequence (i.e. an attempt to pick up an object that is affected by a perception error, that fails due to reference resolution problem).

SUCCESS: This denotes a pickup instruction that successfully completes the resolution sequence.

ABANDON: The participant abandoned the scene.

OTHER: The participant finished the scene with an invalid solution (as defined in Section 5.3.1).

As discussed in Section 4.3.3 the system was capable of planning sequences of actions to fulfil instructions. For example, if the user instructed the system to move an object to a given location while the robot was not actually holding an object, the system would create a plan to pick up an object that fit the instruction given by the user. While the instruction on a surface level is a *move* instruction, it is therefore interpreted by the system as a combination of an implicit *pickup* instruction and a *move* instruction. In order to make it possible to understand in this analysis which actions the system actually performed, we decided to represent actions of this kind as complex instructions.

pickup_move_put: An instruction in which requested that the robot pick up an object, move it to a given location, and put if down there.

move_put: An instruction to move an object to a given place and put it down there.

pickup_move: An instruction to pick up an object and move it to a given place.

In this analysis we focus only on the actions by the participants. The reactions by the system will therefore not be explicitly included as separate dialogue acts. Since it is important whether the robot was able to perform an instruction, we represent the system's response by adding the following suffixes to action tags that represent instructions:

- **ok:** The system successfully performed the instruction.
- **not_ok:** The system was not able to successfully complete the instruction.

For example *pickup_ok* represents a *pickup* instruction the system was able to perform, while *pickup_not_ok* represents a *pickup* instruction the system was not able to perform.

7.5.1 RQ 7.3: Dialogue Structure

What structures can be observed in the resolution attempts?

To visualize the structure of the collected sequences we construct a series of graphs. The first set of graphs, the **full sequence graphs** provide an overview over all the sequences of actions that were observed in each phase. Since they are very large, we then present a second set of graphs, the **Markov graphs**. They provide a more abstract view and show for each action the probability of other possible actions being performed after it. The third set of graphs, the **most frequent sequences graphs** provide an overview of the most frequently observed sequences in each phase.

7.5.2 Full Sequence Graphs

We created a graph for each phase of the experiment that contains all the resolution sequences that were observed in that phase. To do this, we created a list of all resolution sequences. We then created a graph in which all nodes with the same name were merged and all edges that ran between nodes of the same name were merged as well. Each edge is labelled with a figure that represents the number of times the edge was found in the sequence set. To highlight the importance of each connection, the thickness of each edge is determined by its relative frequency. The graph for the Error Phase is presented in Figure 7.3. The graphs for the Description Phase, the Markup Phase and the Querying Phase are presented in Figure 7.5, Figure 7.7 and Figure 7.9.

Since the graphs contain all observed sequences, they naturally are quite large and not particularly legible given the restricted size of the page. They are particularly stretched out by a few branches of the graph that are unusually long but not particularly frequent. Figure 7.4, 7.6, 7.8 and 7.10 focus on the upper parts of the graphs that contain the more frequent events. However, these graphs still contain a high number of nodes, making it difficult to interpret them.

This highlights the need to focus in on particularly frequent resolution sequences. The investigation of the length of resolution sequences in Section 7.4 showed that in all phases resolution sequences with successful outcomes tend to be shorter than sequences that ended when the participant abandoned the scene. The graphs appear to be consistent with this observation in that the paths leading to the *ABANDON* nodes appear longer and more convoluted than the ones leading to the *SUCCESS* nodes. If we focus on individual edges, it appears that there are a few paths that appear much stronger than the remaining paths. This is encouraging because it suggests that structures exist that are particularly frequent. However, this is only based on a first visual impression.

Before we focus on individual sequences we take an alternative look at the data and construct and analyse a Markov graph for each set of resolution

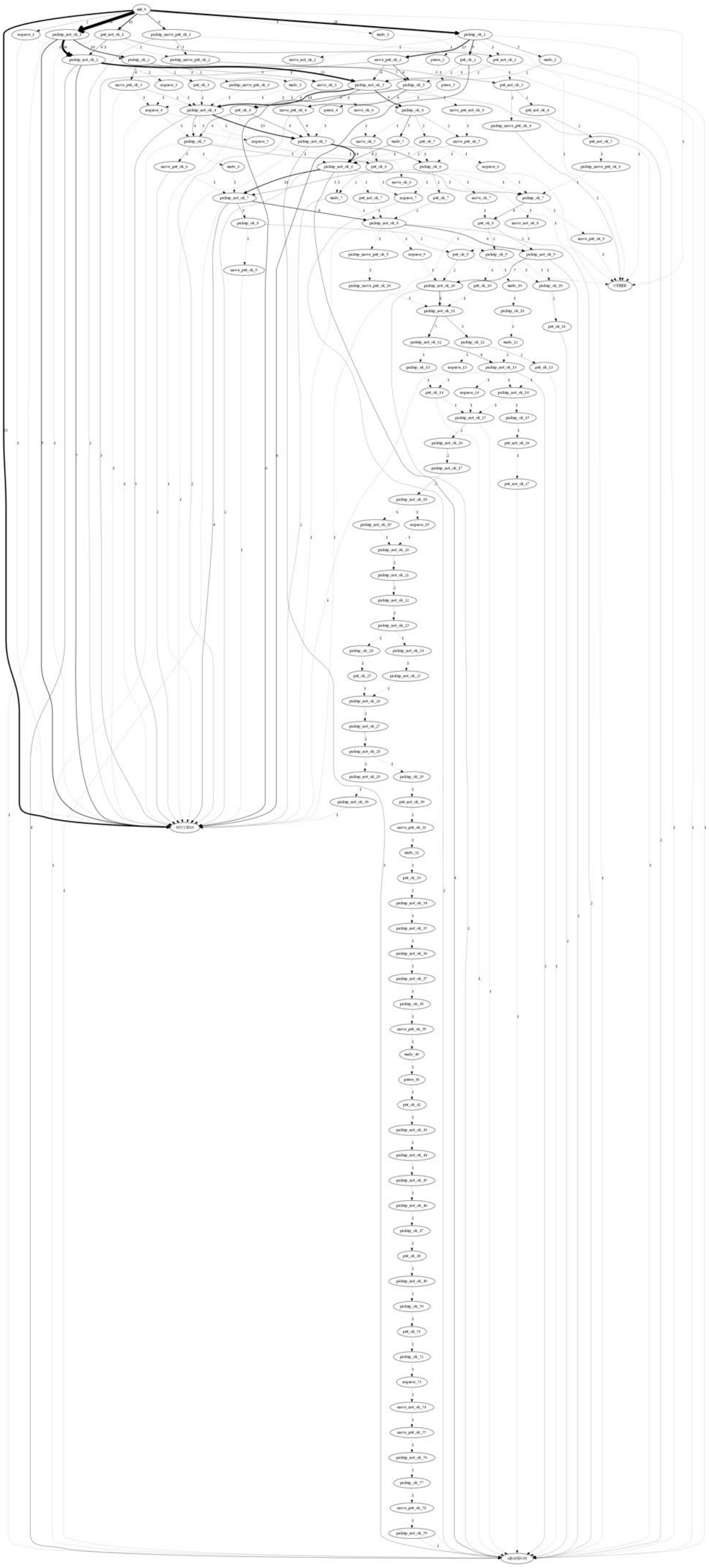


Figure 7.3: The sequence graph for all sequences from Error Phase.

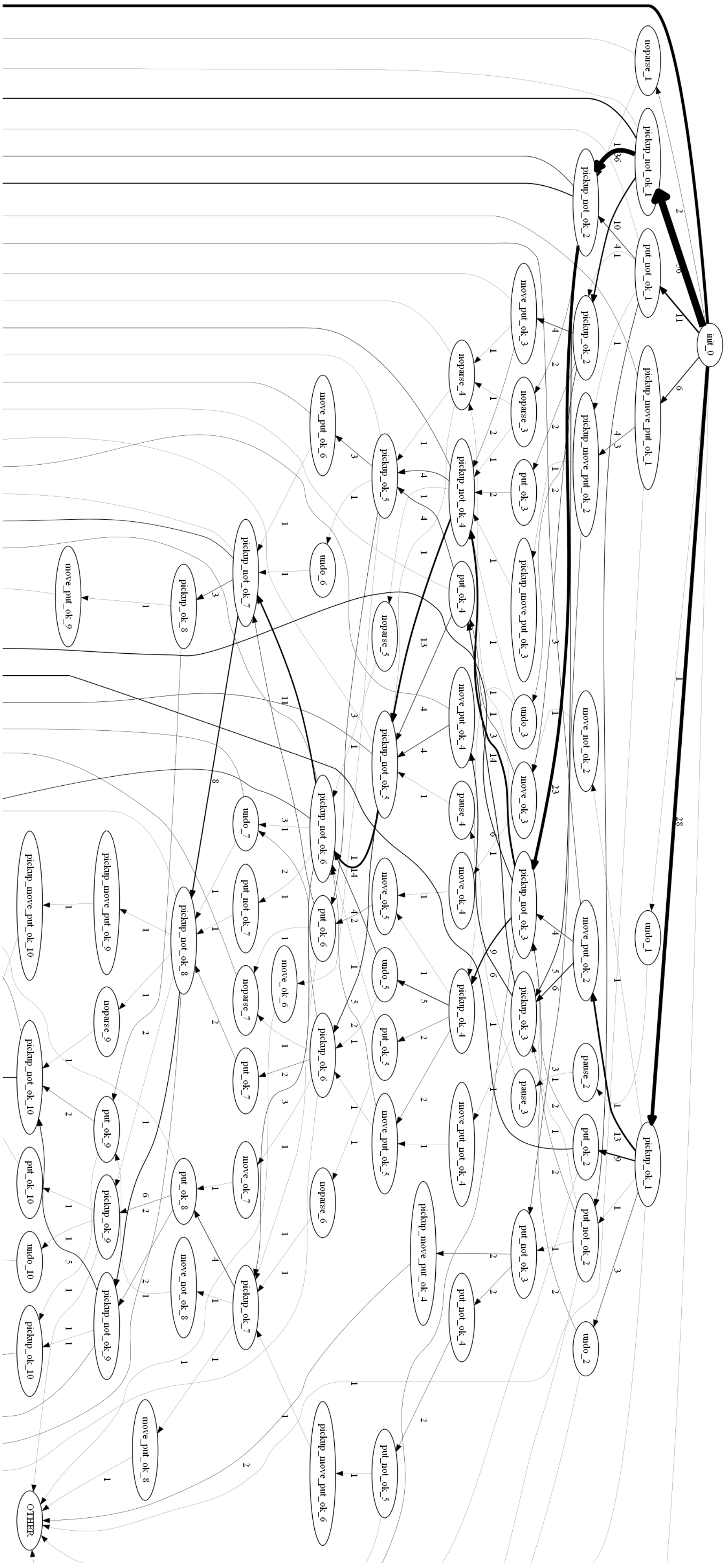


Figure 7.4: An excerpt from the full sequence graph from Error Phase.

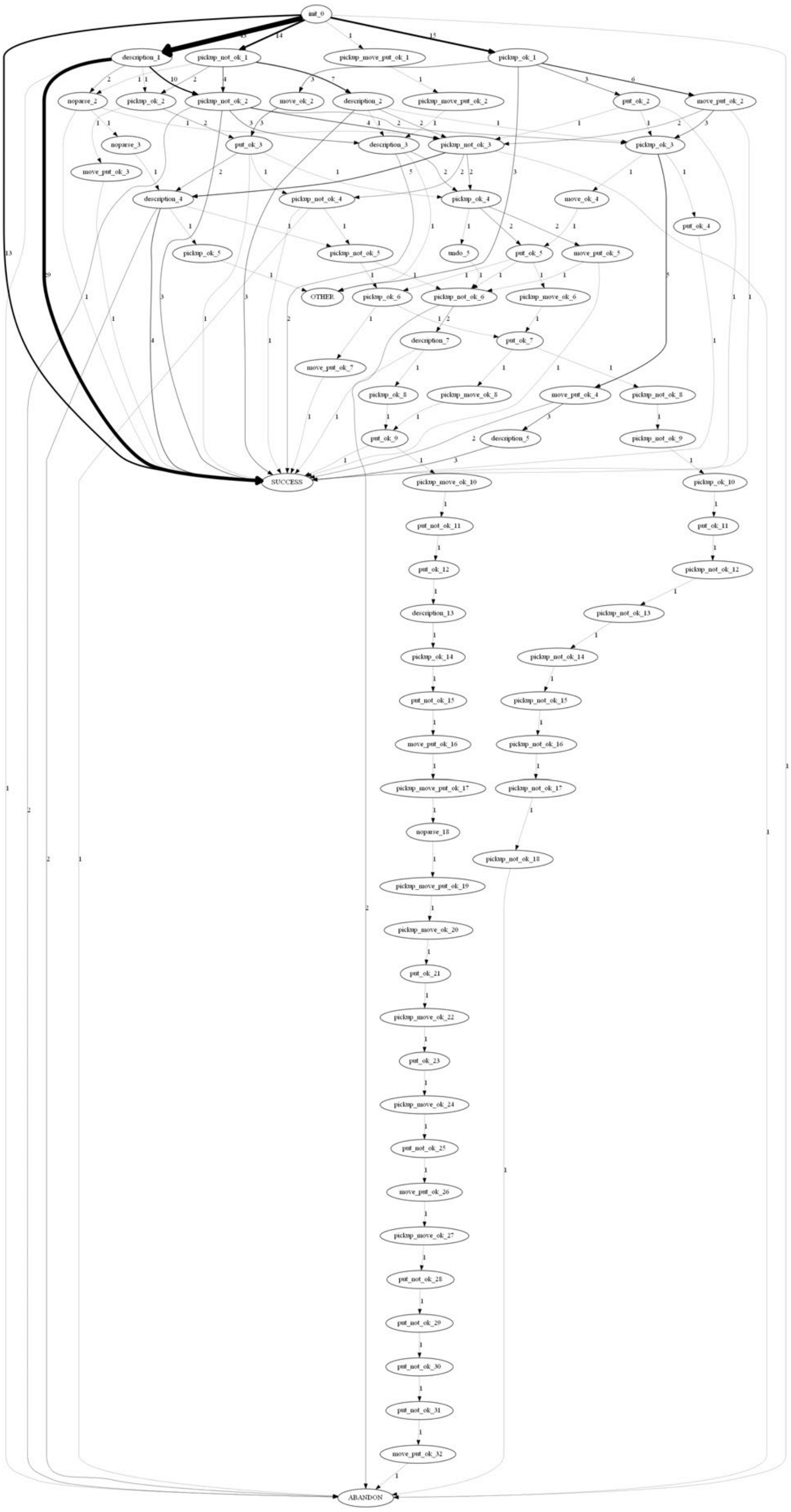


Figure 7.5: The sequence graph for all sequences from Description Phase.

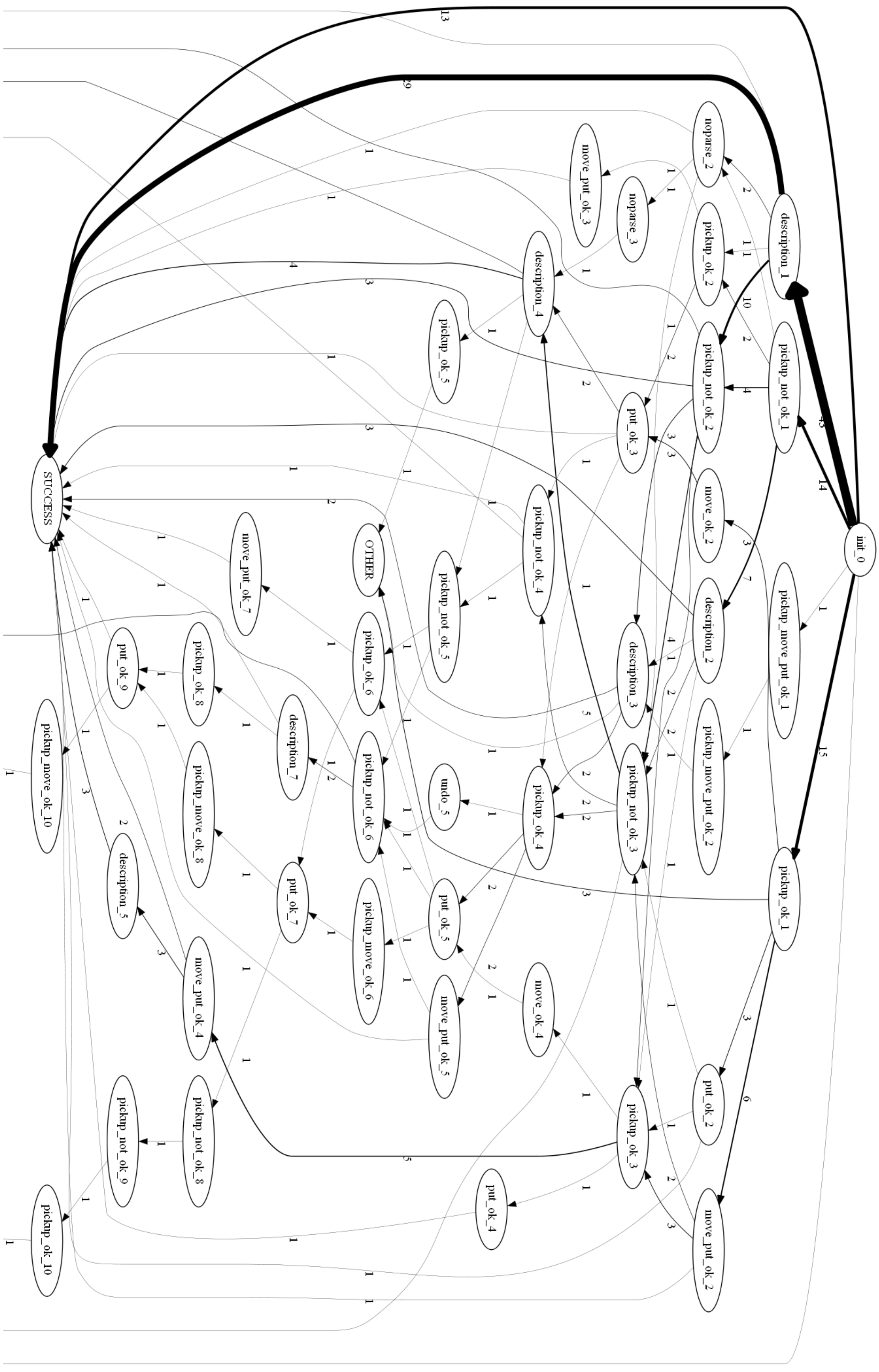


Figure 7.6: An excerpt from the full sequence graph from Description Phase.

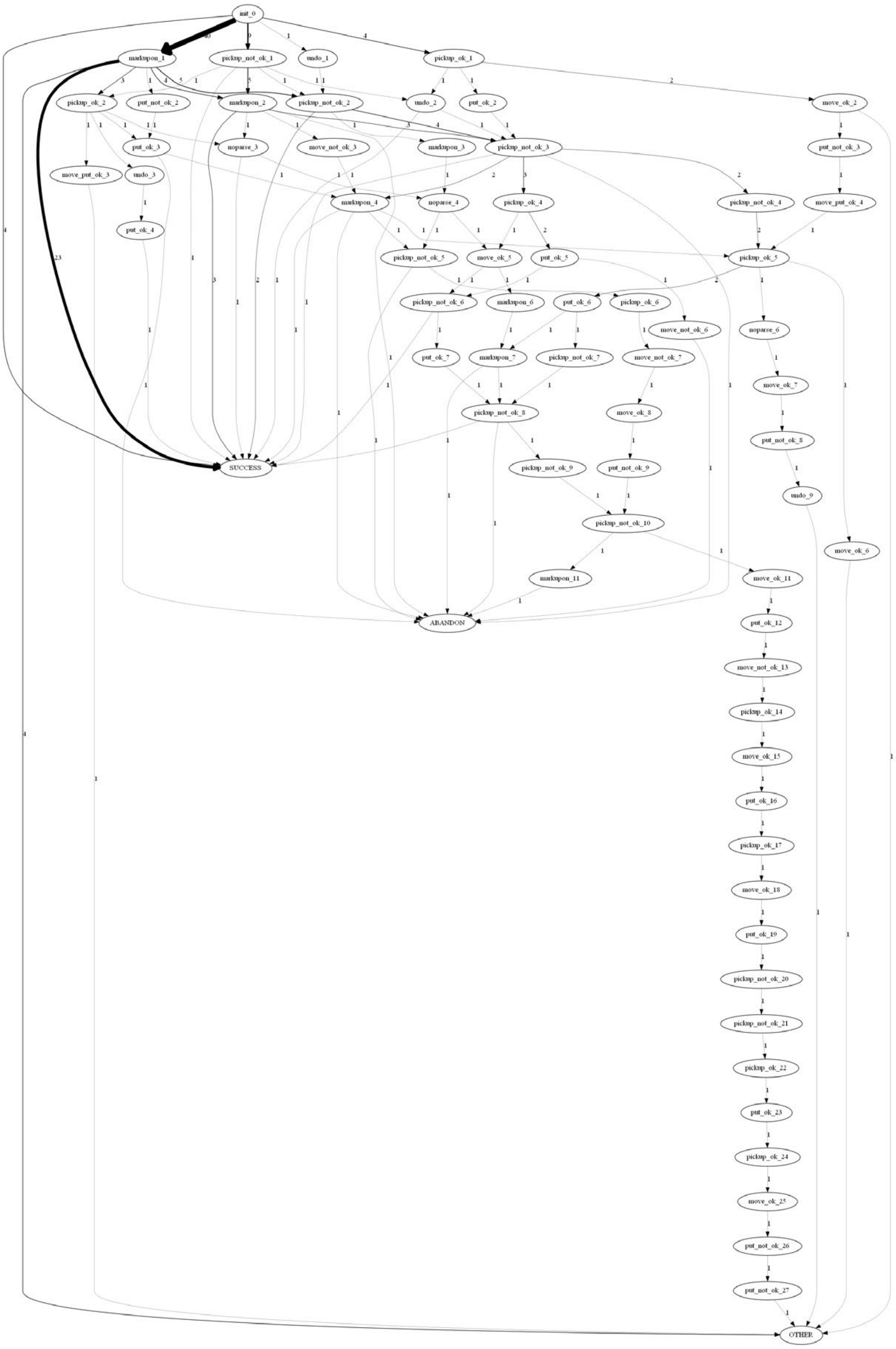


Figure 7.7: The sequence graph for all sequences from Markup Phase.

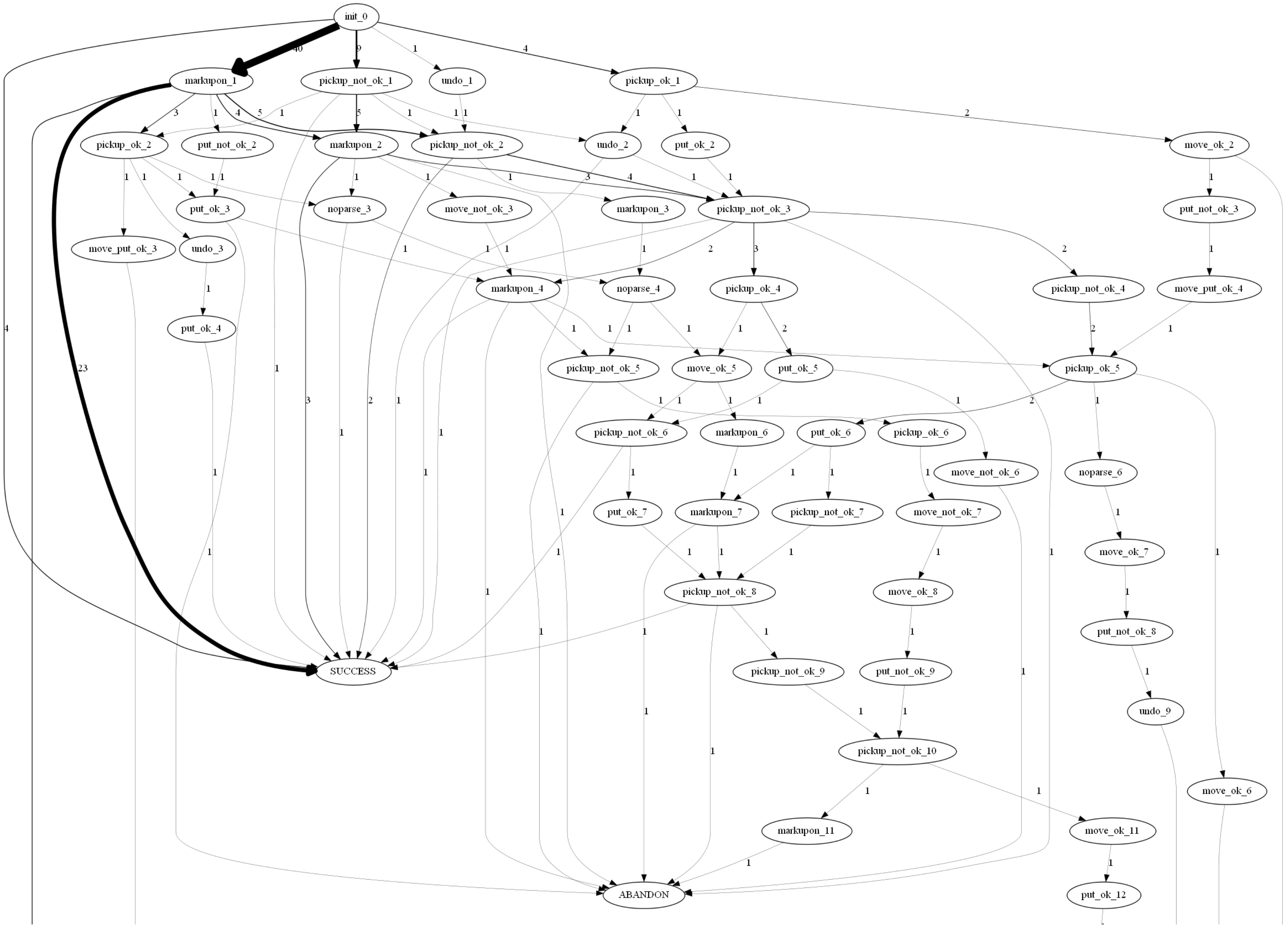


Figure 7.8: An excerpt from the full sequence graph from Markup Phase.

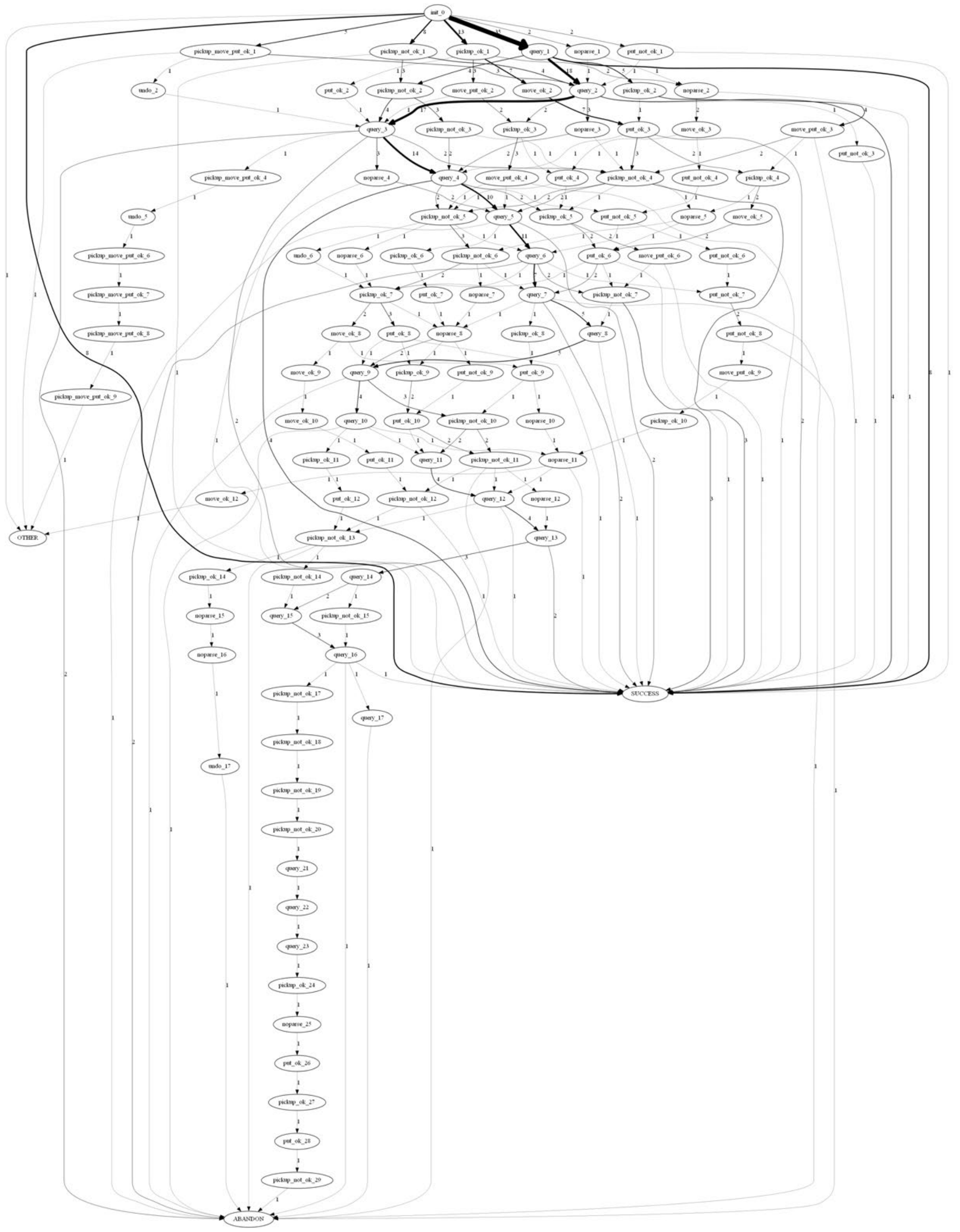


Figure 7.9: The sequence graph for all sequences from Querying Phase.

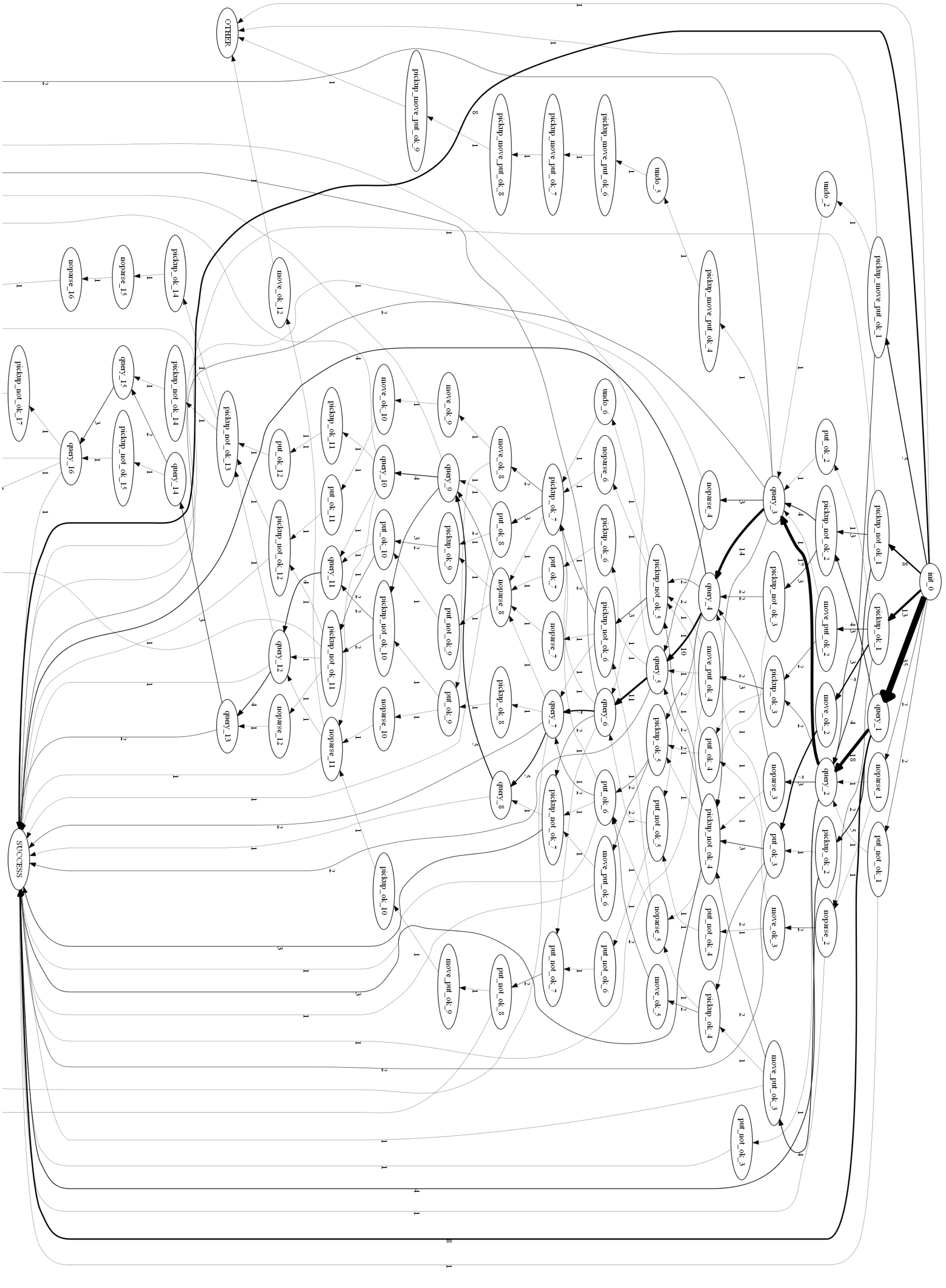


Figure 7.10: An excerpt from the full sequence graph from Querying Phase.

sequences.

7.5.3 Markov Graphs

In the graphs presented in Figure 7.11 to Figure 7.14 we present the events in the resolution sequences as **Markov chains**. A Markov chain consists of a set of states and a set of transitions between the states, where each transition has a transition probability that only is determined by the originating state only. In the graphs presented here, the states represent the types of actions participants could perform. The transition probabilities are calculated based on the observed sequences of actions (e.g. the transition probability between event *pickup_ok* and event *move_ok* is calculated as the number of times event *move_ok* followed event *pickup_ok* divided by the number of times event *pickup_ok* occured.). Edges are visually weighted based on their probability.

The graph for Error Phase is shown in Figure 7.11. Particularly interesting is the connection from the *pickup_not_ok*-node back to itself. It has a probability of 0.56 and is the arc with the highest probability exiting from this node. This indicates that participants often had to make multiple unsuccessful attempts to pick up an objects one after another. This suggests that participants followed a trial-and-error strategy where they repeatedly attempted to pick up the target object by guessing how the robot might perceive the object.

Figure 7.12 shows the graph for the Description Phase. Particularly interesting here is that there is a strong transition from the *init*-node to the *description*-node. This indicates that participants frequently requested a description as their first action after they encountered a problem (as represented by the *init*-node, which represents a pick-up action that failed due to a perception error. The strongest transition from the *description*-node leads to the *SUCCESS*-node. This indicates that participants were often able to resolve the problem after they requested a description.

Figure 7.13 shows the graph for the Markup Phase. The observations here are similar to the ones for the previous graph. There is a strong transition from the *init*-node to *markupon*-node, again indicating that participants tended to request information as their first action after encountering a problem. The strongest transition from the *markupon*-node leads to the *SUCCESS*-node, again indicating that participants were often able to resolve problems after they requested information.

Figure 7.14 shows the graph for the Querying Phase. As in the previous graphs, the transition from the the *init*-node to the *query*-node, which represents the information request in this phase, is strong. The strongest transition from the *query*-node however is a transition back to the *query*-node. This indicates that, rather than solving the the problem with one information request as in the Description Phase and the Markup Phase, participants had to ask multiple queries before they could resolve the problem.

The Markov graphs illustrate that information requests were an important part of the resolution attempts by the participants. In the following section we investigate individual sequences and the structures formed by unifying these sequences.

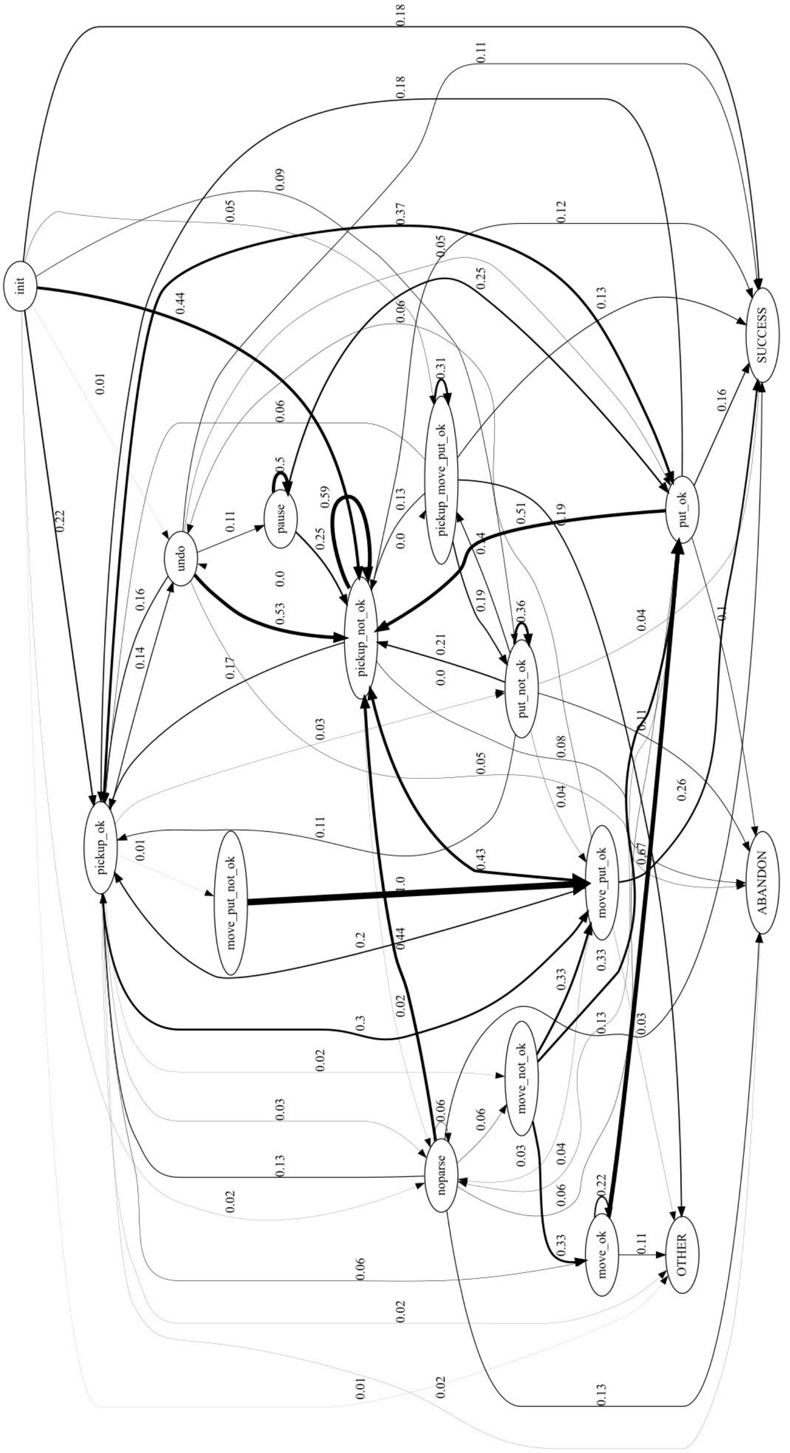


Figure 7.11: The Markov graph for the Error Phase.

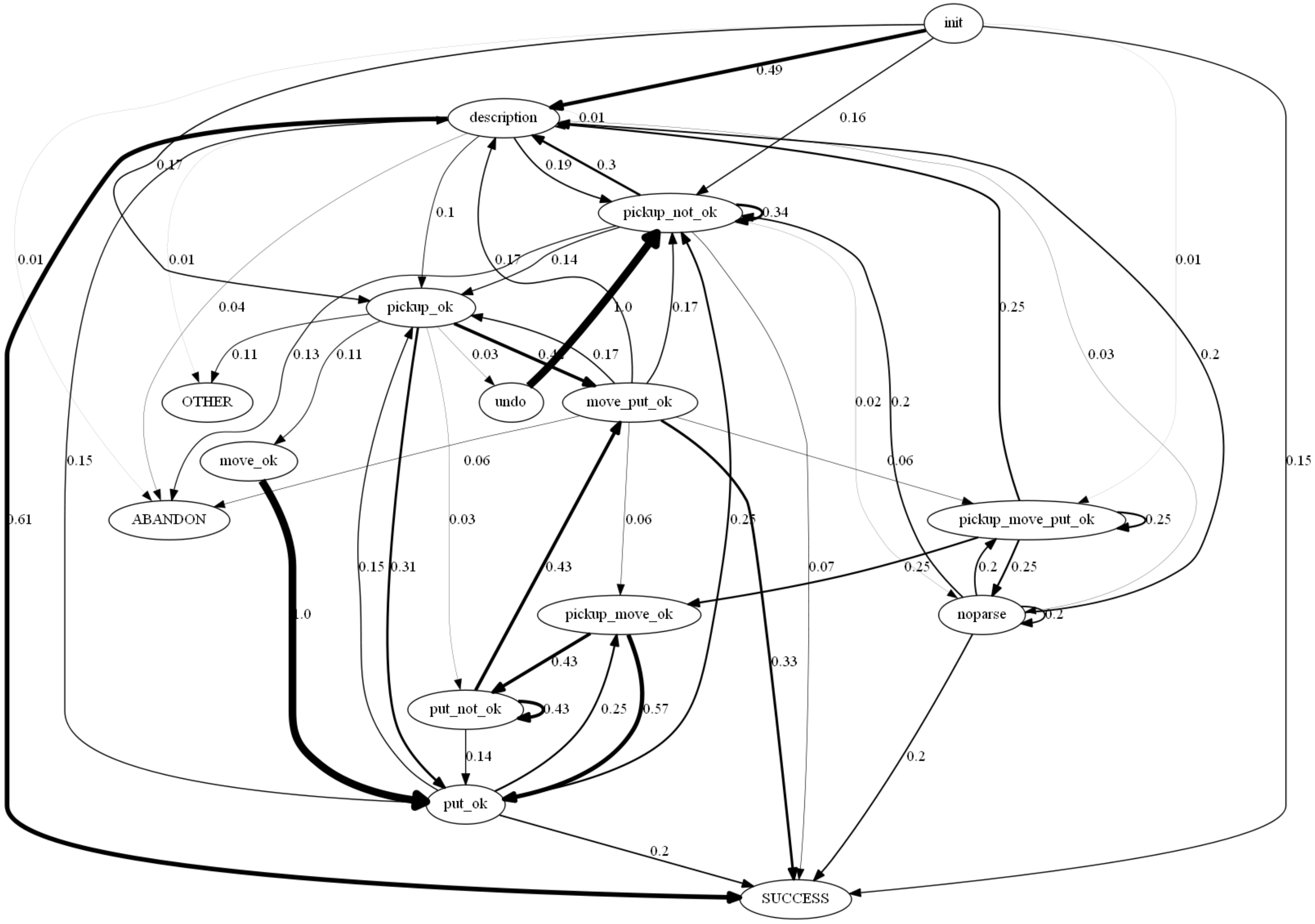


Figure 7.12: The Markov graph for the Description Phase.

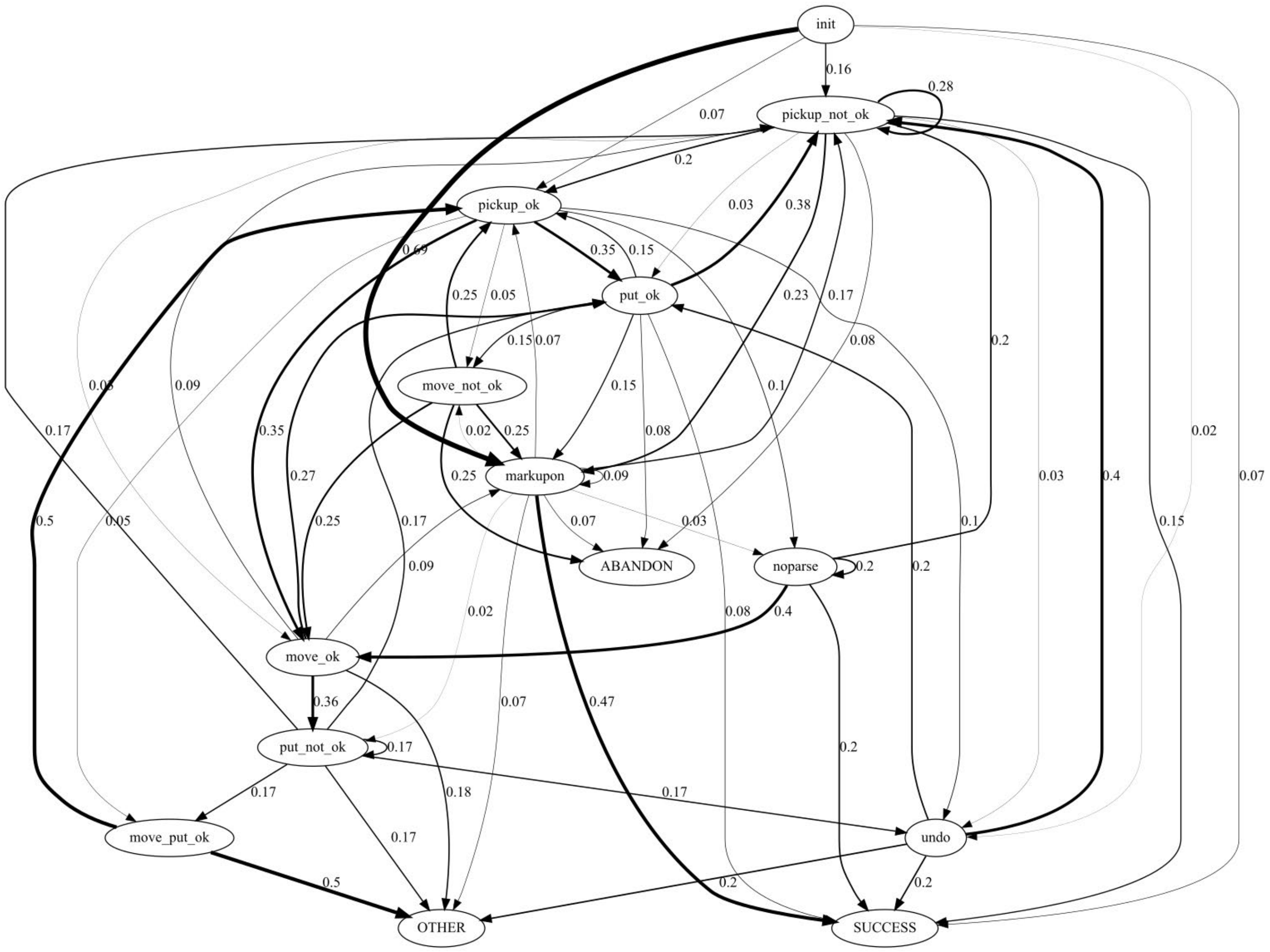


Figure 7.13: The Markov graph for the Markup Phase.

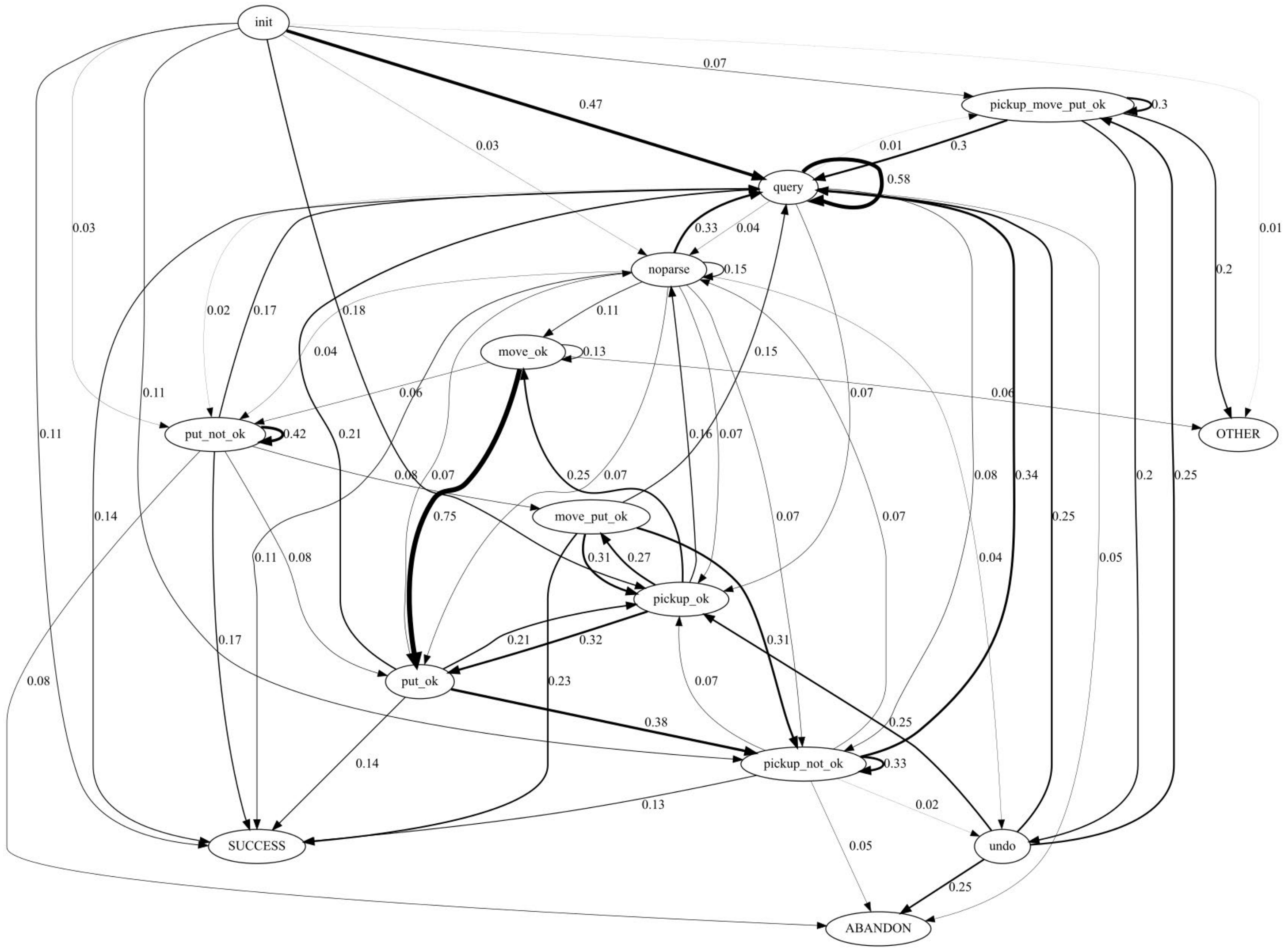


Figure 7.14: The Markov graph for the Querying Phase.

7.5.4 Most Frequent Sequences Graphs

In the first step we determined the dialogue action sequence for each resolution sequence and then counted how many resolution sequences were covered by each observed dialogue action sequence. Table 7.3 provides an overview of the 10 most frequently observed sequences. In Figure 7.15 to Figure 7.24 we present the five most frequent sequences for each phase as well as an example of the text of one resolution sequence from the data that was covered by the sequence (we chose to discuss only the five most frequent sequences to avoid discussing sequences that were only observed very few times).

As in the previous graphs, the *init* node represents the beginning of the sequence, where the participant unsuccessfully attempted to pick up an object that was affected by a perception error. The *SUCCESS* node represents the action in which the participant successfully picked up the object they meant to pick up in the initiating action. The *ABANDON* node represents the action in which the participant abandoned the scene, and the *OTHER* node represents cases where the participant finished the scene with an irregular outcome (as discussed in Section 5.3.1).

Overall we observe that the distributions are markedly different between the phases. In the Description Phase and the Markup Phase the highest ranked sequence makes up over 30% of the observed sequences. The two highest ranked sequences taken together make up over 40% of the observed data. In the Error Phase the two highest ranked sequences together make up only about 25%. In the Querying Phase the 3 highest ranked sequences make

	Error Phase			Description Phase			Markup Phase			Querying Phase		
Rank	**Prop.**	**Count**	**Figure**	**Prop.**	**Count**	**Figure**	**Prop.**	**Count**	**Figure**	**Prop.**	**Count**	**Figure**
1	17.97%	23	7.15	33.33%	29	7.19	39.66%	23	7.19	10.81%	8	7.20
2	7.03%	9	7.16	14.94%	13	7.19	6.90%	4	7.19	10.81%	8	7.21
3	4.69%	6	7.17	3.45%	3	7.19	6.90%	4	7.19	4.05%	3	7.22
4	4.69%	6	7.18	3.45%	3	7.19	3.45%	2	7.19	2.70%	2	7.23
5	3.13%	4	7.19	2.30%	2	7.19	3.45%	2	7.19	2.70%	2	7.24
6	2.34% -	3	-	2.30%	2	-	1.72%	1	-	2.70%	2	-
7	1.56% -	2	-	2.30%	2	-	1.72%	1	-	2.70%	2	-
8	1.56% -	2	-	2.30%	2	-	1.72%	1	-	1.35%	1	-
9	1.56% -	2	-	2.30%	2	-	1.72%	1	-	1.35%	1	-
10	1.56% -	2	-	1.15%	1	-	1.72%	1	-	1.35%	1	-

Table 7.3: The proportion of the most frequent sequences for each phase.

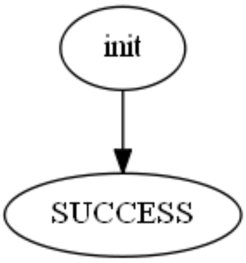
up about 25%. This suggests that the Description Phase and the Markup Phase had strategies that were generally well suited to resolve the problems and often used, while the Error Phase and the Querying Phase were not as straight-forward.

7.5.4.1 Individual Sequences

In this section we discuss, in detail, the most frequent sequences seen within each phase. We start with the Error Phase. The sequence presented in Figure 7.15 represents a structure observed in the Error Phase where the participants, after an initial unsuccessful reference, managed to produce a successful reference directly afterwards. This sequence also occurs in all of the other phases (presented in Figure 7.19, Figure 7.19 and Figure 7.21), but not as frequently. In the examples presented here, the participants achieved this by including a spatial direction attribute.

The sequences in Figure 7.16 and Figure 7.17 represent sequences where the participants (repeatedly) tried different referring expressions and eventually succeeded (or, as in the sequence in Figure 7.19 decided to abandon the scene).

The sequence in Figure 7.18 presents an exception. Here the participants successfully picked up an object that is not the one they originally intended, put it back down, and then picked up the intended object. In personal discussion with participants, some participants reported doing this in order to understand how the system would interpret a given expression. This case therefore represents a situation in which the participants used a pickup instruction to gain information about the robot's understanding of the scene and then produced a successful reference based on that.

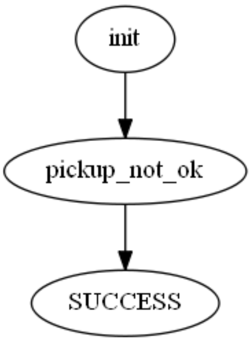


(a) The sequence.

U: pick up the red ball near the yellow ball
S: Sorry, I can't find any red balls that fit your description.
U: pick up the right most red ball
S: Okay, I will pick it up

(b) Error Phase, Rank 1.

Figure 7.15: An example of a resolution sequence.

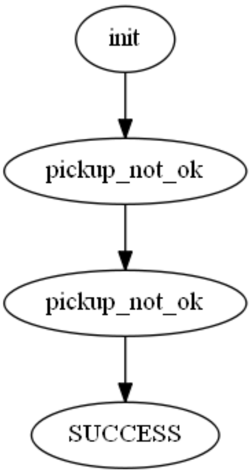


(a) The sequence.

U: pick the ball next the blue box
S: Sorry, I can't find any balls that fit your description.
U: pick the ball in front the blue box
S: Sorry, I can't find any balls that fit your description.
U: pick the left ball
S: Okay, I will pick it up

(b) A corresponding example sequence.

Figure 7.16: Error Phase, Rank 2.

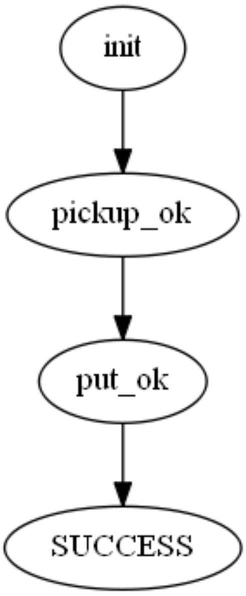


(a) The sequence.

U: pick up the red ball near the yellow ball
S: Sorry, I can't find any red balls that fit your description.
U: pick up the ball near the yellow ball
S: Sorry, I can't find any balls that fit your description.
U: pick up the red ball
S: Sorry, there is more than one red ball and I can't figure out which one you mean.
U: pick up the top red ball
S: Okay, I will pick it up

(b) A corresponding example sequence.

Figure 7.17: Error Phase, Rank 3.

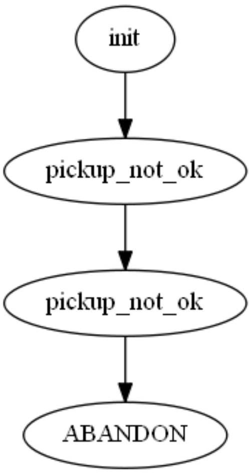


(a) The sequence.

U: pick the green box
S: Sorry, I don't see any green boxes.
U: pick the box
S: Okay, I will pick it up
U: put it down
S: Okay, I will put it down
U: pick the green object
S: Okay, I will pick it up

(b) A corresponding example sequence.

Figure 7.18: Error Phase, Rank 4.



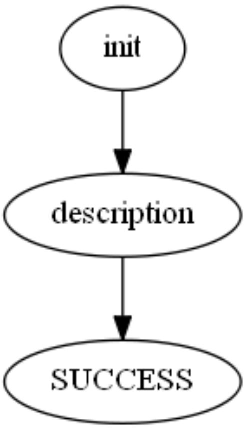
(a) The sequence.

U: pick up the red ball next to the green ball
S: Sorry, I can't find any red balls that fit your description.
U: pick up the upper most red ball
S: Sorry, I can't find any red balls that fit your description.
U: pick up the ball next to the green ball
S: Sorry, I can't find any balls that fit your description.
(User abandons the scene.)

(b) A corresponding example sequence.

Figure 7.19: Error Phase, Rank 5.

The sequences from the Description Phase and the Markup Phase mostly centre around information requests. Figure 7.19 and Figure 7.19 represent cases where the participants encountered a problem in the initial action, requested information (either a description or markup) and then solved the problem based on that information. Figure 7.19 and Figure 7.19 present cases where the participants unsuccessfully attempted to guess how to refer to the object, then requested information, and finally managed to pick up the intended object. Figure 7.19 presents a sequence in which the participants requested information, and afterwards made a unsuccessful attempt to pick up the intended object, but succeeded afterwards.

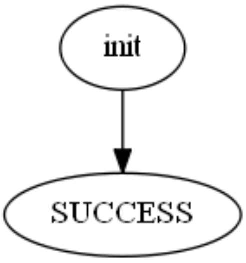


(c) The sequence.

U: pick up the blue ball
S: Sorry, I don't see any blue balls.
(User requests a description.)
S: There is a red box on a place named place 1 on the top right. There is a red ball on the left. There is a green ball on the bottom left. There is a place named place 2 on the right. There is a place named place 3 on the bottom right.
U: pick up the red ball
S: Okay, I will pick it up

(d) A corresponding example sequence.

Figure 7.19: Description Phase, Rank 1.

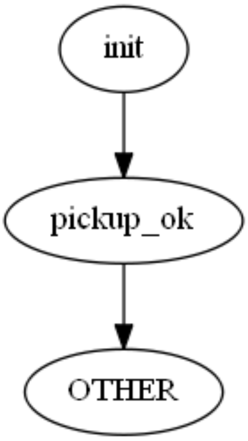


(e) The sequence.

U: pick up red ball to the left of a yellow box
S: Sorry, I can't find any red balls that fit your description.
U: Pick up ball in top right
S: Okay, I will pick it up

(f) A corresponding example sequence.

Figure 7.19: Description Phase, Rank 2.

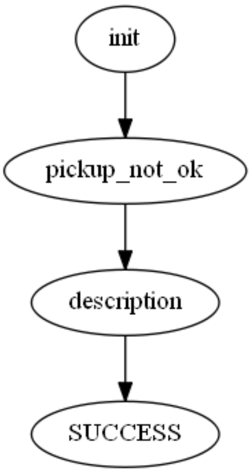


(g) The sequence.

U: pick up the red ball near the yellow ball
S: Sorry, I can't find any red balls that fit your description.
U: pick up the red ball in front of green box
S: Okay, I will pick it up
U: put it to place 1
S: Okay, I will put it there

(h) A corresponding example sequence.

Figure 7.19: Description Phase, Rank 3. The sequence ends with an irregular outcome when the participant places the object that is being held.

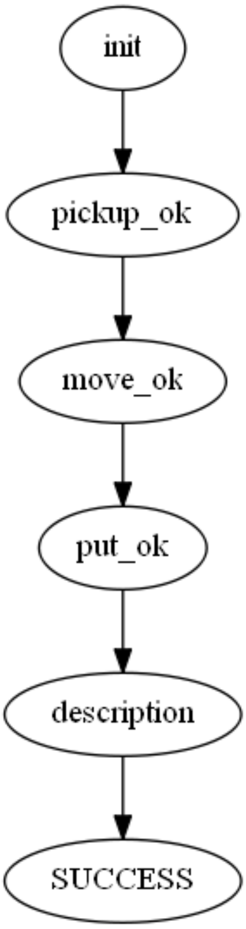


(i) The sequence.

U: pick up the green box
S: Sorry, I don't see any green boxes.
U: pick up the box on the left
S: Sorry, I can't find any boxes that fit your description.
(User requests a description.)
S: There is a blue box on a place named place 2 on the right. There is a red ball on a place named place 1 on the top right. There is a green ball on the bottom left. There is a place named place 3 on the bottom right.
U: pick up the green ball
S: Okay, I will pick it up

(j) A corresponding example sequence.

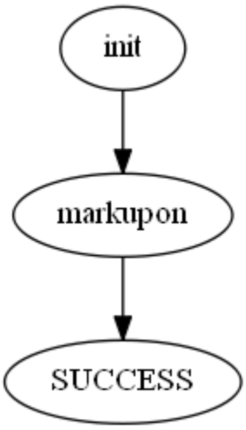
Figure 7.19: Description Phase, Rank 4.



U: pick up the green box
S: Sorry, I don't see any green boxes.
U: pick up the blue box
S: Okay, I will pick it up
U: move it to place 2
S: Okay, I will move it there
U: put it down
S: Okay, I will put it down
(User requests a description.)
S: There is a blue box on a place named place 2 on the right. There is a red ball on a place named place 1 on the top right. There is a green ball on the bottom left. There is a place named place 3 on the bottom right.
U: pick up the green ball
S: Okay, I will pick it up

(k) The sequence.

Figure 7.19: Description Phase, Rank 5.

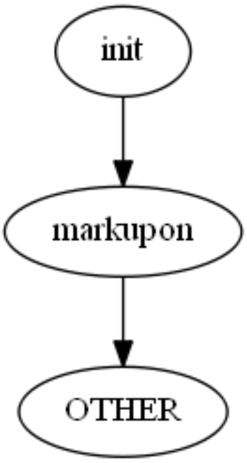


U: pick the blue ball
S: Sorry, I don't see any blue balls.
(User activates markup.)
U: pick up the red ball put it to the place 3
S: Okay, I will pick it up

(m) A corresponding example sequence.

(l) The sequence.

Figure 7.19: Markup Phase, Rank 1.

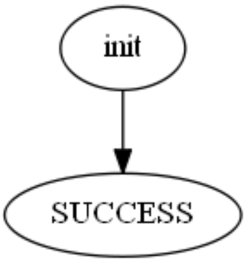


(n) The sequence.

U:	put the green box on place 3
S:	Sorry, I don't see any green boxes.
	(User activates markup.)
U:	put the green ball on place 3
S:	Okay, I will put it there

(o) A corresponding example sequence.

Figure 7.19: Markup Phase, Rank 2.

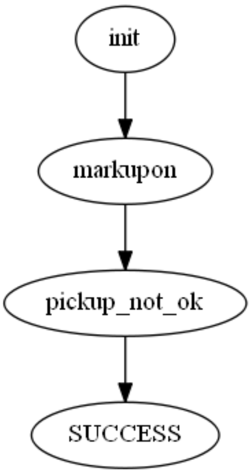


(p) The sequence.

U:	pick up the blue ball
S:	Sorry, I don't see any blue balls.
U:	pick up the ball on the left
S:	Okay, I will pick it up

(q) A corresponding example sequence.

Figure 7.19: Markup Phase, Rank 3.

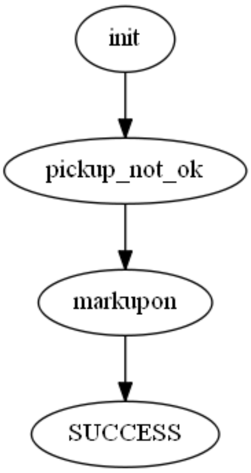


(r) The sequence.

U:	pick up the red ball beside the yellow box
S:	Sorry, I can't find any red balls that fit your description.
	(User activate markup.)
U:	pick up the red ball behind the blue box
S:	Sorry, I can't find any red balls that fit your description.
U:	pick up the red ball behind the red ball
S:	Okay, I will pick it up

(s) A corresponding example sequence.

Figure 7.19: Markup Phase, Rank 4.



(t) The sequence.

U:	pick up the blue ball
S:	Sorry, I don't see any blue balls.
U:	pick up the blue box
S:	Sorry, I don't see any blue boxes.
	(User activates markup.)
U:	pick up the red ball
S:	Okay, I will pick it up

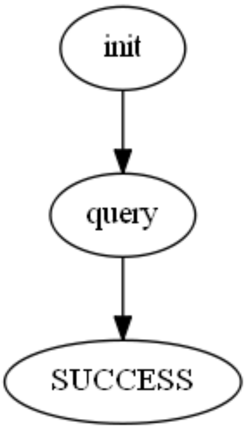
(u) A corresponding example sequence.

Figure 7.19: Markup Phase, Rank 5.

In the Querying Phase a similar approach to this structure where an information request precedes a successful conclusion of the resolution sequence can be observed. Figure 7.20, Figure 7.22 and Figure 7.23 present sequences in which the participants asked one, two or four queries before they picked up the intended object.

Figure 7.19 and Figure 7.19 present sequences with the **OTHER**-outcome, i.e. the participant created a configuration that was accepted by the system as a valid target scene, but was not actually the correct target scene. Apart from the fact that they result in the **OTHER**-outcome, they fit in structurally with the sequences with successful outcomes. From the perspective of a participant, the sequence in Figure 7.19 is not distinguishable from the from one in Figure 7.19, and the one in Figure 7.19 is equivalent to the one in Figure 7.19 because in both instances they (presumably) believed they were fulfilling the target conditions of the scene.

The sequences in Figure 7.19 and Figure 7.24 have not been discussed yet. They both represent sequences in which the participants, after the initial problem, left the originally targeted object alone and moved a different object to a target location, but then returned and successfully picked up the targeted object. In personal discussions, a few participants described that sometimes, when they could not pick up an object, they would first address other parts of the scene, and try to modify the context of the original object in order to allow it be identified with a simpler expression. In fact, it is plausible that the example for the sequence in Figure 7.24 is an example of this.

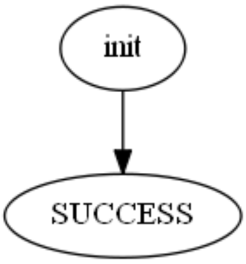


(a) The sequence.

U: pick up the green box
S: Sorry, I don't see any green boxes.
U: do you see a green ball
S: Yes, I see a green ball.
U: pick it up
S: Okay, I will pick it up

(b) A corresponding example sequence.

Figure 7.20: Querying Phase, Rank 1.

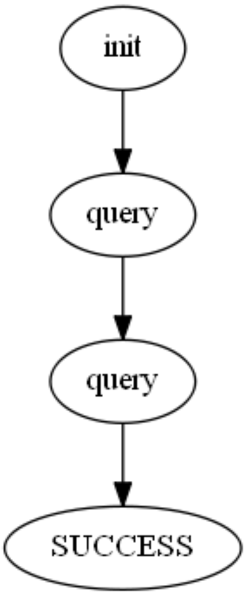


(a) The sequence.

U: place the blue ball on place 3
S: Sorry, I don't see any blue balls.
U: pick up the far left ball
S: Okay, I will pick it up

(b) A corresponding example sequence.

Figure 7.21: Querying Phase, Rank 2.

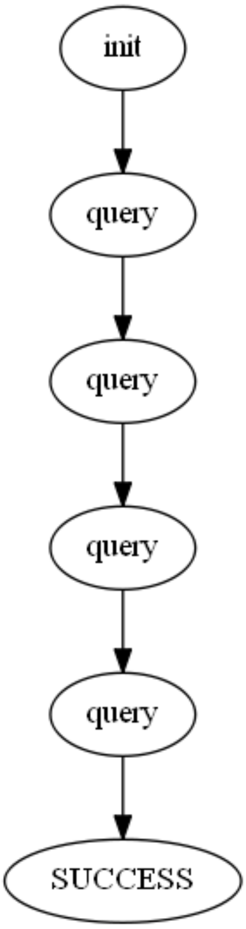


(a) The sequence.

U: pick up the red ball in front of the blue box
S: Sorry, I can't find any red balls that fit your description.
U: do u see a box
S: Yes, I see a box.
U: do u see a ball in front of the box
S: Yes, I see a ball that fits your description.
U: pick it up
S: Okay, I will pick it up

(b) A corresponding example sequence.

Figure 7.22: Querying Phase, Rank 3.

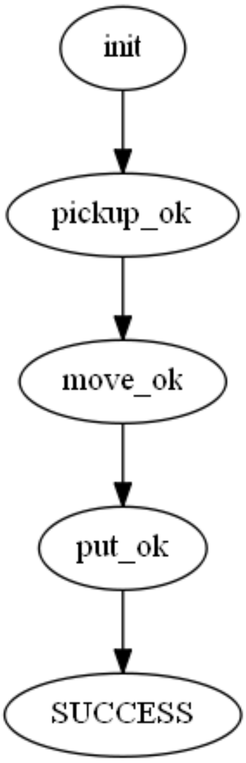


(a) The sequence.

U: pick up the red ball near the yellow ball
S: Sorry, I can't find any red balls that fit your description.
U: do you see two red balls
S: Yes, I see 2 red balls.
U: do you see a yellow ball
S: Sorry, I don't see any yellow balls.
U: do you see a yellox box
S: I see 2 boxes.
U: do you see a yellow box
S: Yes, I see a yellow box.
U: pick up the red ball near the yellow box
S: Okay, I will pick it up

(b) A corresponding example sequence.

Figure 7.23: Querying Phase, Rank 4.



(a) The sequence.

U: pick up the green box
S: Sorry, I don't see any green boxes.
U: pick up the red ball
S: Okay, I will pick it up
U: move it to place 1
S: Okay, I will move it there
U: put it down
S: Okay, I will put it down
U: pick up the most left object
S: Okay, I will pick it up

(b) A corresponding example sequence.

Figure 7.24: Querying Phase, Rank 5.

7.5.4.2 Composed Most Frequent Sequences Graphs

To put the individual most frequent graphs into context with each other, we created for each phase a graph that contains only the five most frequent sequences that were discussed in the previous section. As in the other graphs we presented, the edges are labelled with the number of times this particular edge occurred in an observed sequence. Again, the thickness of each edge is determined by the relative frequency of the edge.

The graph for the Error Phase is shown in Figure 7.25. It shows that participants often had to attempt (multiple) unsuccessful attempts before they resolved the problem.

The graph for the Description Phase in Figure 7.26 highlights the domi-

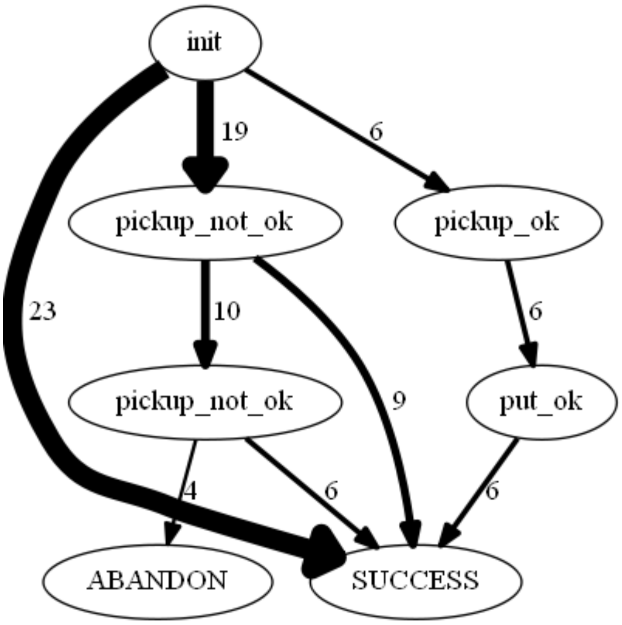


Figure 7.25: The sequence graph for the five most frequent sequences he in the Error Phase.

nance of the resolution sequence in which a description was requested, leading to a successful conclusion. The graph for the Markup Phase in Figure 7.27 shows a similar structure where the *markupon* node takes the place of the *description* node.

Similar to the graphs for the Description Phase and the Markup Phase, the graph for the Querying Phase in Figure 7.28 show strong arcs for branches that involve (multiple) queries.

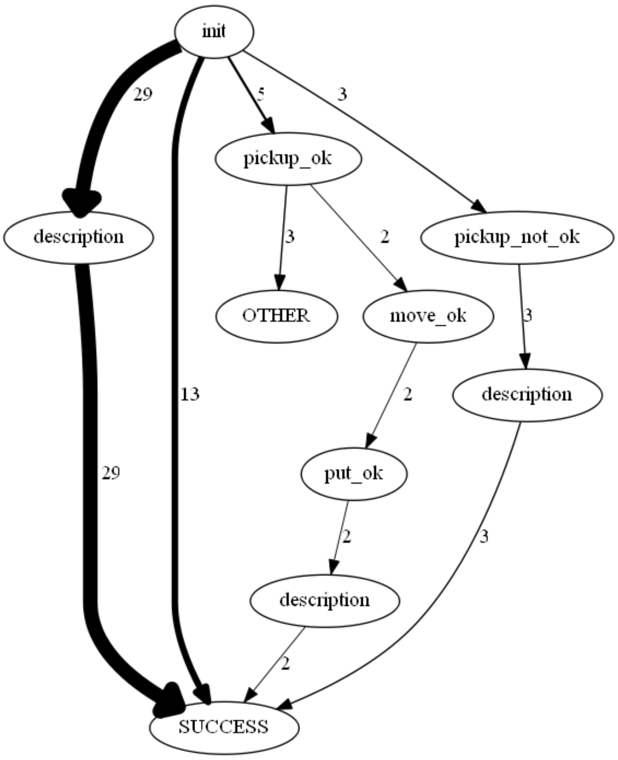


Figure 7.26: The sequence graph for the five most frequent sequences in the Description Phase.

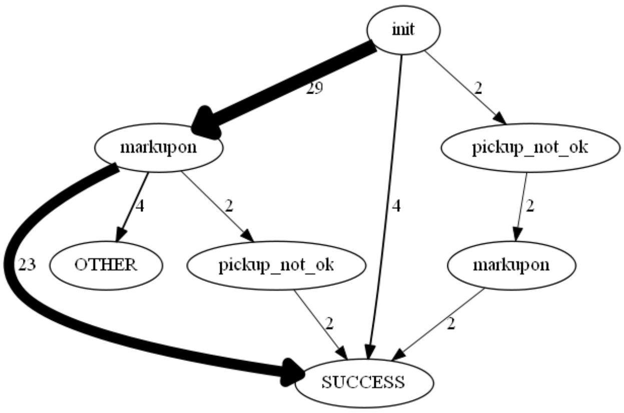


Figure 7.27: The sequence graph for the five most frequent sequences in the Markup Phase.

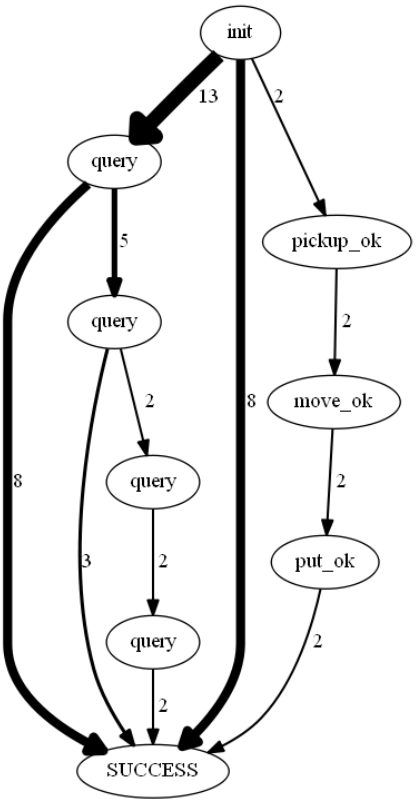


Figure 7.28: The sequence graph for the five most frequent sequences in the Querying Phase.

7.5.5 Summary

There is a considerable variation in how the participants reacted to perception based problems in the dialogue, and in how they resolved problems. By analysing the bigram graphs and the most frequent sequences, we found that there are structures that are distinct for each condition. One important aspect across all phases was the retrieval of information. The participants were not able to directly request information in the Error Phase, and therefore had to try to guess a successful way of referring to the intended object, if necessary by exhausting the different possibilities. An alternative option was used by a few participants who attempted to gain an understanding of the robot's understanding of the world by trying out different expressions to see which object the robot would resolve them to.

In the Description Phase and the Markup Phase, the participants could request information by the system through the description option and the markup option. Successful sequences therefore tend to involve direct information requests. In the Querying Phase, the participants were able to ask queries of the system. While in the two previous phases one information request always provided all the information the robot could provide at any given time, the information provided by queries depended on the content of the query. Consequently, the participants sometimes had to pose multiple queries in a row in order to be able to formulate a successful instruction.

Conclusions:

1. Information retrieval of different types depending on the information option available forms a central part of all successful strategies.

2. Information is either retrieved through explicit information requests or trial-and-error.

3. Participants tend to request information after they encounter a problem.

4. After they request information, participants are often able to directly solve the problem.

7.6 Summary

We began this chapter by defining the term *resolution sequence* and explaining how we extracted resolution sequences from the data. We then analysed the outcome and length of the resolution sequences and found that resolution dialogues were both shorter and more successful if the participants were able to request information from the robot. We then investigated the structure of resolution sequences at a dialogue act level. We found that in all phases participants performed actions to gather information in order to be able to successfully resolve the problem. Depending on the phase different strategies were used for this, and if explicit information request options were available, they were used. If they were not available, the participants tended to use a trial-and-error strategy.

In the following chapter we investigate resolution sequences from a content perspective by examining the expressions the participants used.

Chapter 8

Referring Strategies in Problem Resolution Sequences

In the previous chapter we introduced the concept of *resolution sequences*. Resolution sequences are the actions a participant performs (in co-operation with the robot) to resolve a problem that arose in the dialogue due to a perception error. We described how we identified and extracted resolution sequences in the Toy Block experiment. We then analysed the resolution sequences on a structural level. In this chapter we analyse them on a content based level by investigating the referring expressions the participants used.

We are particularly interested in identifying the differences between the expressions the participants used in the references that started the resolution sequences (i.e. the expressions that the robot could not resolve) and the expressions that terminated the resolution sequence (i.e. the successful final reference). We call the decisions the participants made about the choice of attributes **referring strategies**.

8.1 Research Questions

In this chapter the focus of the analysis is on how participants reformulated referring expressions in order to solve communication failures caused by divergent perceptual information between the robot and the participant. This analysis is structured around the following research questions:

Research Question 8.1: *What attributes did the participants include in their initial and final reference?* – This question refers to the participant's choice of whether or not to include a possible attribute into the description.

Research Question 8.2: *How did the participants modify their expressions between the initial and the final reference?* – To formulate a successful reference the participants had to formulate a new referring expression that is different from the initial one. With this question we investigate in what way the expressions differ, and analyse the strategies that may underlie these changes.

Research Question 8.3: *What effect did information requests have on how the participants modified the references?* – In the different phases of the experiment the participants were able to request information from the robot in different ways. With this question we investigate whether or not the different information sources had an impact on the choice of expressions by the participants and what the differences in the strategies they produced were.

8.2 RQ 8.1: Attribute Selection in the Resolution Sequences

What attributes did the participants include in their initial and final reference?

In this section we analyse the strategies participants used to resolve the problems in the dialogue that occur due to perception errors by comparing the attributes used in the initial references of resolution sequences to the attributes used to make the final successful reference.

8.2.1 Data

The investigation is based on the set of resolution sequences we presented in Section 7.2. A resolution sequence represents the sequence of actions a participant performed after they encountered a perception based problem in the dialogue by attempting to pick up an object that was affected by a perception error. Each sequence end when the participant was able to resolve the problem by successfully picking up the object they had originally intended to pick up when they or abandoned the scene. The term *initial reference* (as defined in our discussion of the resolution sequences in Section 7.2) denotes the referring expression used in the instruction that initiated the sequence (i.e. it was a referring expression the system could not (uniquely) resolve to a referent due to a perception error). The term *final reference* denotes the expression that was used in the instruction that successfully concluded the resolution sequence (i.e. it was an expression that was suitable to identify the object despite the perception error [1]. We determine which attributes participants used in their *initial references* and then compare the results to the attributes used in the *final references.* We distinguish between four types of attributes:

Type: The participant used an expression that described the specific type of the object (e.g. "ball' in "the green ball" or "box" in "the box"). If the participant used a expression that did not describe the specific type of the object (e.g. by describing the target an "object" or "thing", as in "the green thing" or "the red object"), we annotated the expression as not containing a type attribute.

Colour: The participant specified the colour of the object (e.g. "the green box" or "the red one"). Examples for expressions that did not contain a colour attribute are "the box" or "the ball on the left".

Landmark reference: The participant described the object in relation to another object (e.g. "the ball near the red box" or "the box between the two green balls"). This corresponds to the **relational attributes** discussed in Section 4.3.8.2.

Directional expression: The participant used a spatial direction to describe the object (e.g. "the ball on the left", "the box in the back").

[1] **(TODO: re-read)**

This corresponds to the **directional attributes** discussed in Section 4.3.8.1.

We selected all successful resolution sequences from the dataset and then determined automatically which attributes the initial and the final reference contained in each sequence. The results are presented in Table 8.1. For comparison, Table 8.2 contains an overview of the attributes that were used in pick-up instructions in general.

8.2.2 Summary

It appears that almost all initial references included the **type attribute**. Most of the final references also contained the *type* attribute. Only in the Error Phase did the participants tend to use not use specific type expressions more often than in the other phases. Removing the type attribute could be interpreted as an attempt to avoid an attribute the participants found to be unreliable. [1]

Most of the initial expressions included the **colour attribute**. In the final expressions, the colour attribute was used less often across all conditions. Again, this can be interpreted as an attempt to avoid an unreliable attribute. This hypothesis is supported by the observation that for the Error Phase and the Querying Phase the reduction in the number of colour attributes is particularly high. In these phases the participants were not given a complete description of the system's perception of the world, but had to elicit information step by step either through querying in the Querying Phase

[1] **(TODO: check discussion - type attributes, neutral types etc.)**

	Initial reference								Final reference							
	Type		Colour		LM		Dir		Type		Colour		LM		dir	
Phase	%	Count	%	Count	%	Count	%	Count	%	Count	%	Count	%	Count	%	Count
Error Phase	96.34	79	86.59	71	45.12	37	0	0	84.15	69	58.54	48	30.49	25	17.07	14
Description Phase	98.57	69	87.14	61	51.43	36	0	0	98.57	69	77.14	54	34.29	24	34.29	24
Markup Phase	100	40	92.5	37	45	18	0	0	95	38	95	38	37.5	15	17.5	7
Querying Phase	100	55	94.55	52	49.09	27	0	0	94.55	52	65.45	36	32.73	18	25.45	14

Table 8.1: Attributes that were included in initial and final references (LM = landmark reference, Dir = directional expression)

	All attributes	Type		Colour		LM		Direction		Any spatial attribute	
Phase	Count	%	Count	%	Count	%	Count	%	Count	%	Count
No Error Phase	420	100.00%	420	91.43%	384	6.67%	28	39.52%	166	46.19%	194
Error Phase	1447	84.24%	1219	65.65%	950	4.98%	72	33.66%	487	38.63%	559
Description Phase	1417	97.11%	1376	86.10%	1220	16.02%	227	35.07%	497	51.09%	724
Markup Phase	631	96.99%	612	76.23%	481	10.46%	66	26.78%	169	37.24%	235
All phases	3915	92.64%	3627	77.52%	3035	10.04%	393	33.69%	1319	43.73%	1712

Table 8.2: Attributes contained in pick-up instructions

or through trial-and-error in the Error Phase. Consequently, relative to the other phases in the experiment, the participants were less likely to have an understanding of how the system perceived an object, and therefore more likely to avoid attributes that, in their experience, the system had problems perceiving.

Landmark references were included in some of the initial references and also in some of the final references. Overall the number of expressions that included landmark references was smaller in the final expressions than in the initial expressions across all phases. This suggests that participants in some cases avoided landmarks after they found them to be unreliable.

The **directional expression** attribute is particularly interesting. In none of the observed cases was it included in an initial reference, but in some of the final references.[1] It is furthermore interesting that the number of directional descriptions was the highest in the Description Phase. In this phase, the participants were able to request a verbal description of the scene. As described in Section 4.3.5.2, the descriptions contained both landmark references and directional descriptions. One possible explanation for the higher number of directional descriptions would therefore be that the participants used the directional descriptions the system used and aligned to the descriptions given by the system.

Conclusions:

1. Overall, there is a difference between what attributes were used in the initial references and in the final references.

2. There are also differences that appear to be related to the information request option that was used.

3. There appears to be a tendency where participants substitute basic attributes with directional attributes.

8.3 RQ 8.2: Expression Modifications

How did the participants modify their expressions between the initial and the final reference?

In the first section we investigated at a high level which attributes the participants chose to include in their referring expressions. We found that the participants included other attributes in the final expressions than in the

[1] The fact that no directional attributes were included in the in the initial expressions may be a bit deceptive. We should keep in mind that resolution sequences were only initiated if a reference failed. This therefore probably reflects the fact that directional attributes were robust against perception errors.

initial expressions. We therefore set out to investigate how the expressions changed between the initial reference and the final reference in more detail, and in particular, whether the participants modified the values they used in the attributes. We compare the initial and the final reference and extract **modification events** that describe how the second expression differs from the first expression.

8.3.1 Data

We extracted the following modification events:

[**addColour**] The participant did not use a colour in the initial reference but did use a colour in the final reference. For example:

U: Pick up the ball.

U: Pick up the green ball.

[**addType**] The participant used a generic type expression in the first initial reference and then used a more specific type in the final reference. For example:

U: Pick up the green thing.

U: Pick up the green ball.

[**addLM**] The participant's initial reference did not contain a landmark reference, but the final reference did:

U: Pick up the ball.

U: Pick up the ball near the yellow box.

[**addDir**] The participant's initial reference did not contain a directional attribute, but the final reference did:

U: Pick up the ball.

U: Pick up the ball on the left.

[**dropColour**] The initial reference contained a colour attribute but the final reference did not.

U: Pick up the green ball.

U: Pick up the ball.

[**dropType**] The initial reference contained a type attribute with a specific type value, in the final reference it had a generic type value.

U: Pick up the green ball.

U: Pick up the green object.

[**dropLM**] The initial reference contained a landmark reference, but the final reference did not.

U: Pick up the green ball near the blue box.

U: Pick up the green ball.

[**dropDir**] The initial reference contained a directional attribute, but the final reference did not.

> **U:** Pick up the green ball in the back.
>
> **U:** Pick up the green ball.

[**changeColour**] Both the initial reference and the final reference contain a colour attribute but with different values.

> **U:** Pick up the green ball.
>
> **U:** Pick up the red ball.

[**changeType**] Both the initial reference and the final reference contain a type attribute but with the different values.

> **U:** Pick up the green box.
>
> **U:** Pick up the green ball.

[**changeLM**] The initial reference and the final reference contain a landmark reference, but the references were not identical.

> **U:** Pick up the green ball near the blue box.
>
> **U:** Pick up the green ball behind the green ball.

[**changeDir**] The initial reference and the final reference contain a direction attribute, but they are not identical.

U: Pick up the green ball in the back.

U: Pick up the green ball on the left.

Multiple events can occur between two references. For example, for the following two utterances

U: Pick up the green ball near the blue box.

U: Pick up the box on the left.

The following modification events would be extracted:

- [**dropColour**] because no colour was used to the describe the target object in the final expression while "green" was used in the initial expression.

- [**changeType**] because the term "ball" was used for the target object in the initial expression and "box" in the final expression.

- [**dropLM**] because in the initial expression the target object was described in relation to a landmark, but not in the final expression.

- [**addDir**] because in the final expression a directional expression was used but not in the initial expression.

The modification event for this sequence would therefore be denoted as [**addDir**] [**dropColour**] [**dropLM**] [**changeType**].

We compared the initial reference and final reference of each successful resolution sequence from the Error Phase, the Description Phase, the Markup

Phase and the Querying Phase and determined the modification events for each pair. In total we 30 distinct modification events were observed. The full set of modification events is listed in Table 8.3, along with each modification event's proportion in the total set of events and the number of times the modification event was observed.

8.3.2 Summary

Table 8.3 contains an overview of all events that were observed across all phases of the experiment, ordered by their frequency. The most frequent modification event is [**changeLM**]. It represents situations in which participants attempted to describe an object using a landmark in the initial reference. After the reference failed because the landmark was affected by a perception error, they used an alternative landmark to describe the object (e.g. "the box behind the green ball" instead of "the box in front the blue box"), chose a different description for the same landmark (e.g. "the green thing" instead of the "the green box"), or modified the landmark reference by including more or fewer objects into it (e.g. by "the ball near the yellow box" instead of "the ball between the yellow box and the blue box").

The second most frequent event is [**addDir**] [**dropLM**]. It represents an alternative solution to the same problem. The participants attempted to describe an object in relation to a landmark. However, after the reference failed they did not choose an alternative landmark, but instead abandoned the landmark based description and used a direction based description instead (e.g. "the ball on the bottom left" instead of "the ball near the yellow

Rank	**Event**	**Proportion**	**Count**
1	[changeLM]	22.52%	100
2	[addDir][dropLM]	15.09%	67
3	[changeType]	13.74%	61
4	[addDir][dropColour]	11.26%	50
5	[changeColour]	6.08%	27
6	[addDir][dropColour][dropType]	4.50%	20
7	[addColour][changeLM]	3.38%	15
8	[addDir][changeColour]	3.15%	14
9	[addDir][dropColour][changeType]	2.25%	10
10	[addDir][dropColour][dropLM]	2.25%	10
11	[addLM][changeColour]	2.25%	10
12	[addLM][dropColour]	2.03%	9
13	[dropType]	1.80%	8
14	[addDir]	1.35%	6
15	[addColour][addDir][dropLM]	0.90%	4
16	[addColour][addDir][dropType][dropLM]	0.90%	4
17	[addLM]	0.90%	4
18	[dropColour][changeType]	0.90%	4
19	[addColour][addDir][dropLM][dropNumber]	0.45%	2
20	[addDir][dropColour][dropDir][changeType]	0.45%	2
21	[addDir][dropColour][dropNumber][changeType]	0.45%	2
22	[addDir][dropLM][changeColour]	0.45%	2
23	[addLM][dropColour][dropType]	0.45%	2
24	[addType][changeColour]	0.45%	2
25	[dropColour]	0.45%	2
26	[dropColour][changeLM]	0.45%	2
27	[dropColour][dropLM][changeType]	0.45%	2
28	[addColour][addType][changeLM]	0.23%	1
29	[addDir][changeColour][changeType]	0.23%	1
30	[addDir][changeType]	0.23%	1

Table 8.3: The total set of actions observed across all conditions.

box").

[**changeType**] and [**changeColour**] are the third and fifth most frequently observed events. They represent instances where the participant changed the value of a single basic attribute between the initial and the final reference (e.g. instead of "the green box" they used the expression "the blue box" or "the green ball"). These modification events are remarkable because they represent a situation in which the participants abandoned a description which was valid from their understanding of the scene and adopted a description which was from their perspective was not necessarily valid from their perspective, but sufficient to accomplish their goal in the interaction. This can be interpreted as the participants aligning to the robot's understanding of the world, which they learnt either by requesting information about it (through descriptions, markup or queries) or by testing different descriptions through trial-and-error.

This observation is consistent with the results by Schneider & Luz (2011), who found that participants in a machine translation mediated dialogue scenario tended to adopt terms presented by the system, rather than attempting a repair dialogue, even if those terms were highly odd.

The fourth and the sixth most frequent events are [**addDir**] [**dropColour**] and [**addDir**] [**dropColour**] [**dropType**]. They represent cases where the participants removed a basic attribute (*colour*, or both *colour* and *type*) from their initial expression and instead added a directional attribute (e.g. they used the expression "the box on the right" instead of "the green box"). This approach is interesting, because it represents a substitution of information. The participants removed one piece of information (the basic attribute) that was potentially affected by perception errors, and instead added a different attribute (the spatial description) that was not affected by perception errors.

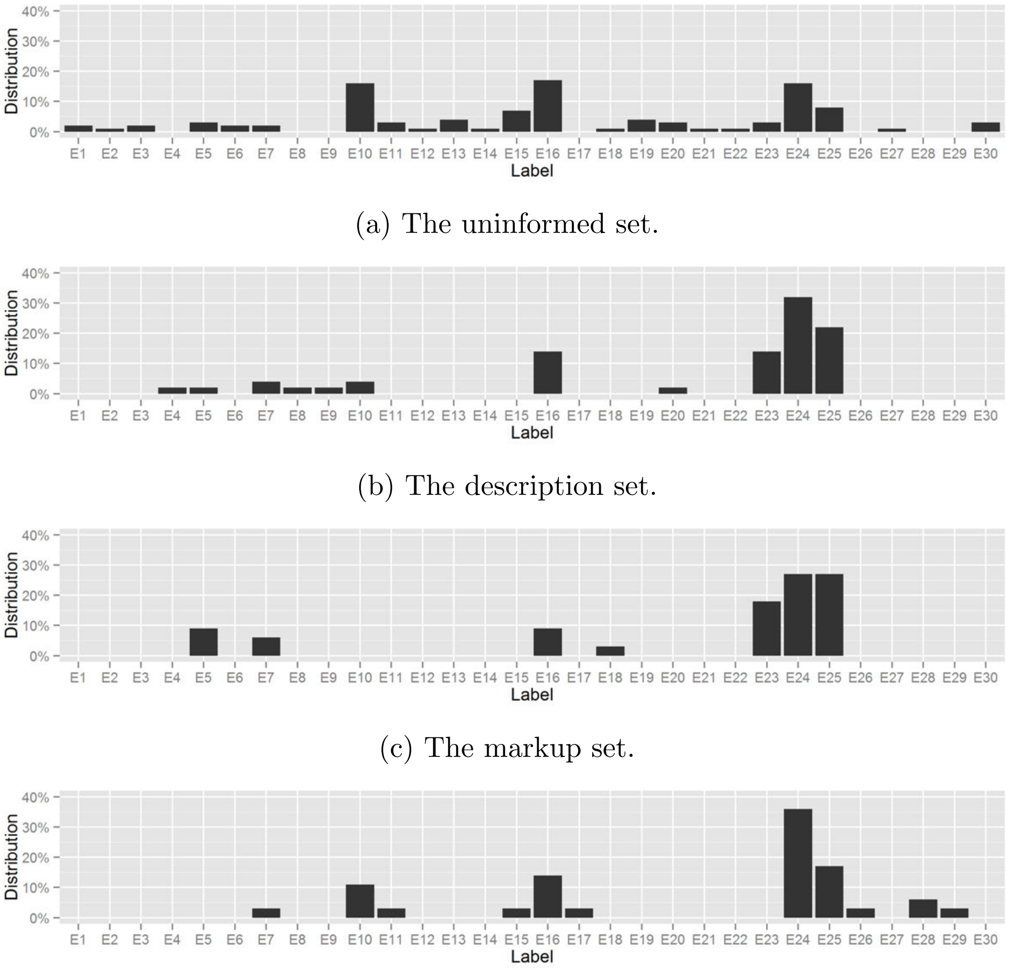
Overall we find that the participants changed the expressions in a number of ways by using different attributes and by changing the values of attributes. In some of the events multiple changes were made, e.g. one attribute was removed, while another was added. In some instances it appears that participants substituted attributes that were potentially affected by perception errors with attributes that were resistant to perception errors.

Conclusions:

1. A large range of different modification events was observed.
2. Some modifications suggest a strategy where participants substituted unreliable attributes for reliable ones, while other modifications suggest that the participants attempted to understand and align to the robot's model of the world.

8.4 RQ 8.3: The Effects of Information Requests

What effect did information requests have on how the participants modified the references?



(a) The uninformed set.

(b) The description set.

(c) The markup set.

(d) The querying set.

Figure 8.1: The distribution of the events.

In the previous section we investigated the differences between initial references and final references. We identified a number of modification events and considered what strategies might underlie these different events. In this section we investigate whether these different strategies are related to whether or not the participants were able to request information and to the type of information that was available. In the first step we are going to analyse the sets at a high level and attempt to describe and quantify the similarity between the sets. Based on this we are going to describe and motivate the differences and similarities between the sets.

8.4.1 Data

To achieve a clearer understanding of the effect of information request options, we split the events into separate **condition sets** depending on whether the participant requested information after the initial reference that triggered the problem had been made, and before the problem had been resolved. This resulted in four condition sets:

- The **uninformed** set contains all events from all the phases (including the Error Phase) in which no information requests or queries were used. This set includes 128 events.
- The **description** set contains all events from the Description Phase in which the participant requested a scene description. In total this set contains 50 events.
- The **markup** set contains all events from the Markup Phase in which

the participant requested markup. In total this set contains 33 events.

- The **query** set contains all events from the Querying Phase in which the participant posed a query. In total this set contains 36 events.

We present the modification events from the uninformed set in Table 8.4, for the description set in Table 8.5, for the markup set in Table 8.6 and for the query set in Table 8.7. Table 8.8 contains an overview of the number of modification events in each condition set and the number of distinct events. We also modelled each condition set as a probability distribution over the full set of observed events and calculated the entropy of each distribution. For easier comparison we normalized each entropy value by dividing it by the maximum entropy for the set of events.

The results show that the uninformed set is the largest and most evenly distributed, while the sets in the other conditions are smaller and more unevenly distributed.

This suggests that if the participants were able to request information about the robot's understanding of the scene, they tended to use fewer different strategies, while they used a wider range of strategies if no information was available. A reason for this observation could be that the information requests allowed the participants to employ particularly useful strategies that relied on the information they received from the robot. In the uninformed condition, the participants did not have this information and therefore used a wider range of less efficient strategies.

Rank	**Event**	**Proportion**	**Count**
1	[addDir][dropLM]	17.19%	22
2	[changeLM]	15.63%	20
3	[addDir][dropColour]	15.63%	20
4	[changeType]	7.81%	10
5	[addDir][dropColour][dropType]	7.03%	9
6	[addLM][changeColour]	3.91%	5
7	[addDir][dropColour][dropLM]	3.91%	5
8	[addColour][changeLM]	3.13%	4
9	[changeColour]	3.13%	4
10	[dropType]	3.13%	4
11	[addLM][dropColour]	3.13%	4
12	[addDir][dropColour][changeType]	3.13%	4
13	[addDir][changeColour]	2.34%	3
14	[addDir]	2.34%	3
15	[addColour][addDir][dropType][dropLM]	1.56%	2
16	[addColour][addDir][dropLM]	1.56%	2
17	[addLM][dropColour][dropType]	0.78%	1
18	[addDir][dropColour][dropDir][changeType]	0.78%	1
19	[addDir][dropColour][dropNumber][changeType]	0.78%	1
20	[dropColour][changeLM]	0.78%	1
21	[addLM]	0.78%	1
22	[addColour][addDir][dropLM][dropNumber]	0.78%	1
23	[addType][changeColour]	0.78%	1

Table 8.4: The modification events in the uninformed set.

Rank	Event	Proportion	Count
1	[changeLM]	32.00%	16
2	[changeType]	22.00%	11
3	[addDir][dropLM]	14.00%	7
4	[changeColour]	14.00%	7
5	[addDir][changeColour]	4.00%	2
6	[addDir][dropColour]	4.00%	2
7	[addColour][changeLM]	2.00%	1
8	[addDir][changeColour][changeType]	2.00%	1
9	[addColour][addType][changeLM]	2.00%	1
10	[addLM][dropColour]	2.00%	1
11	[addDir][changeType]	2.00%	1

Table 8.5: The modification events in the description set.

Rank	Event	Proportion	Count
1	[changeLM]	27.27%	9
2	[changeType]	27.27%	9
3	[changeColour]	18.18%	6
4	[addDir][dropLM]	9.09%	3
5	[addColour][changeLM]	9.09%	3
6	[addDir][changeColour]	6.06%	2
7	[addLM]	3.03%	1

Table 8.6: The modification events in the markup set.

Rank	**Event**	**Proportion**	**Count**
1	[changeLM]	36.11%	13
2	[changeType]	16.67%	6
3	[addDir][dropLM]	13.89%	5
4	[addDir][dropColour]	11.11%	4
5	[dropColour][changeType]	5.56%	2
6	[addDir][changeColour]	2.78%	1
7	[dropColour]	2.78%	1
8	[dropColour][dropLM][changeType]	2.78%	1
9	[addDir][dropLM][changeColour]	2.78%	1
10	[addDir][dropColour][dropType]	2.78%	1
11	[addDir][dropColour][changeType]	2.78%	1

Table 8.7: The modification events in the query set.

Set	**Number of instances**	**Number of distinct instances**	**Entropy**	**Entropy normalized**
Uninformed	128	23	2.64	0.77
Description	50	11	1.90	0.56
Markup	33	7	1.73	0.51
Query	36	11	1.94	0.57

Table 8.8: Information about the condition sets.

8.4.2 Differences Between the Condition Sets

In order to determine what the specific effects of the different sources of information were, we investigated the differences between the four condition sets. Table 8.9 shows for each event how often it was observed in each condition set, and its proportion of the total set of events in the condition set. We visualise the contents of this table in Figure 8.1. It contains a bar plot for each condition set that visualises the proportion of each event in the set. The bar plots show particularly clearly the differences in the distributions between the uninformed set and the other sets. In the uninformed set (Figure 8.1a) a large number of events are represented, but often only in very small numbers. In the other condition sets, fewer modification events were observed, but a small number of events are particularly frequent across all three sets.

In Figure 8.2 to Figure 8.7 we present *slope graphs*[1] that compare the ranks of the events in each set to the ranks of the events in other sets. Each side of the graph represents the events in one condition set ranked by their frequency. The lines connecting both axes indicate whether an event from the set represented by the axis on the left side was ranked higher or lower in the set represented by the right side. We primarily notice two things about the graphs. The graph comparing the description set and the markup set (Figure 8.5) shows that the events are mostly ranked similarly in both sets. The graphs that compare the uninformed set to the description set and, the markup set and the querying set show that the events are ranked noticeably

[1] As discussed by (Tufte, ND, 1986)

	Uninformed set		**Description set**		**Markup set**		**Querying set**	
Event set	**Proportion**	**Count**	**Proportion**	**Count**	**Proportion**	**Count**	**Proportion**	**Count**
[addColour][addDir][dropLM]	1.56%	2	0.00%	0	0.00%	0	0.00%	0
[addColour][addDir][dropLM][dropNumber]	0.78%	1	0.00%	0	0.00%	0	0.00%	0
[addColour][addDir][dropType][dropLM]	1.56%	2	0.00%	0	0.00%	0	0.00%	0
[addColour][addType][changeLM]	0.00%	0	2.00%	1	0.00%	0	0.00%	0
[addColour][changeLM]	3.13%	4	2.00%	1	9.09%	3	0.00%	0
[addDir]	2.34%	3	0.00%	0	0.00%	0	0.00%	0
[addDir][changeColour]	2.34%	3	4.00%	2	6.06%	2	2.78%	1
[addDir][changeColour][changeType]	0.00%	0	2.00%	1	0.00%	0	0.00%	0
[addDir][changeType]	0.00%	0	2.00%	1	0.00%	0	0.00%	0
[addDir][dropColour]	15.63%	20	4.00%	2	0.00%	0	11.11%	4
[addDir][dropColour][changeType]	3.13%	4	0.00%	0	0.00%	0	2.78%	1
[addDir][dropColour][dropDir][changeType]	0.78%	1	0.00%	0	0.00%	0	0.00%	0
[addDir][dropColour][dropLM]	3.91%	5	0.00%	0	0.00%	0	0.00%	0
[addDir][dropColour][dropNumber][changeType]	0.78%	1	0.00%	0	0.00%	0	0.00%	0
[addDir][dropColour][dropType]	7.03%	9	0.00%	0	0.00%	0	2.78%	1
[addDir][dropLM]	17.19%	22	14.00%	7	9.09%	3	13.89%	5
[addDir][dropLM][changeColour]	0.00%	0	0.00%	0	0.00%	0	2.78%	1
[addLM]	0.78%	1	0.00%	0	3.03%	1	0.00%	0
[addLM][changeColour]	3.91%	5	0.00%	0	0.00%	0	0.00%	0
[addLM][dropColour]	3.13%	4	2.00%	1	0.00%	0	0.00%	0
[addLM][dropColour][dropType]	0.78%	1	0.00%	0	0.00%	0	0.00%	0
[addType][changeColour]	0.78%	1	0.00%	0	0.00%	0	0.00%	0
[changeColour]	3.13%	4	14.00%	7	18.18%	6	0.00%	0
[changeLM]	15.63%	20	32.00%	16	27.27%	9	36.11%	13
[changeType]	7.81%	10	22.00%	11	27.27%	9	16.67%	6
[dropColour]	0.00%	0	0.00%	0	0.00%	0	2.78%	1
[dropColour][changeLM]	0.78%	1	0.00%	0	0.00%	0	0.00%	0
[dropColour][changeType]	0.00%	0	0.00%	0	0.00%	0	5.56%	2
[dropColour][dropLM][changeType]	0.00%	0	0.00%	0	0.00%	0	2.78%	1
[dropType]	3.13%	4	0.00%	0	0.00%	0	0.00%	0

Table 8.9: Distribution of events across all the conditions.

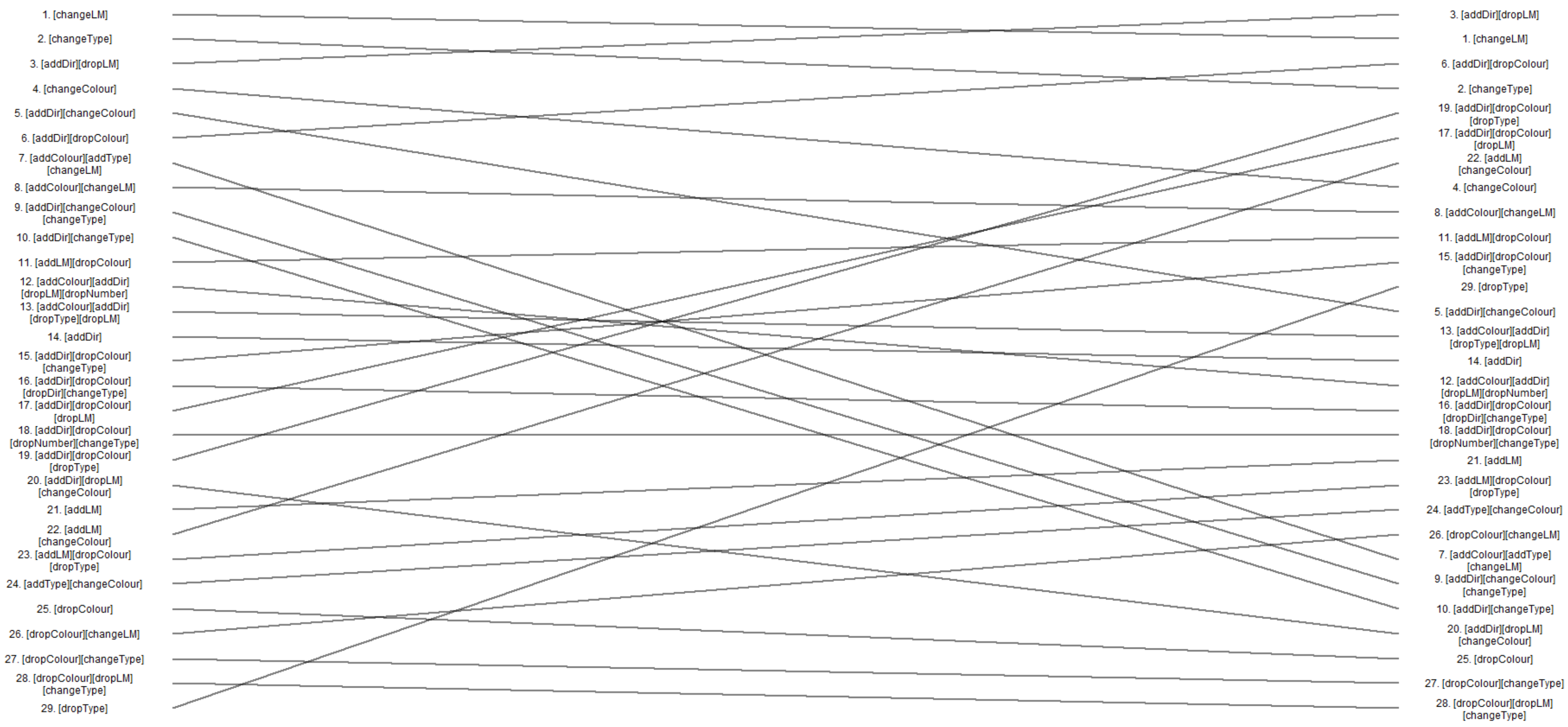

1. [changeLM]
2. [changeType]
3. [addDir][dropLM]
4. [changeColour]
5. [addDir][changeColour]
6. [addDir][dropColour]
7. [addColour][addType] [changeLM]
8. [addColour][changeLM]
9. [addDir][changeColour] [changeType]
10. [addDir][changeType]
11. [addLM][dropColour]
12. [addColour][addDir] [dropLM][dropNumber]
13. [addColour][addDir] [dropType][dropLM]
14. [addDir]
15. [addDir][dropColour] [changeType]
16. [addDir][dropColour] [dropDir][changeType]
17. [addDir][dropColour] [dropLM]
18. [addDir][dropColour] [dropNumber][changeType]
19. [addDir][dropColour] [dropType]
20. [addDir][dropLM] [changeColour]
21. [addLM]
22. [addLM] [changeColour]
23. [addLM][dropColour] [dropType]
24. [addType][changeColour]
25. [dropColour]
26. [dropColour][changeLM]
27. [dropColour][changeType]
28. [dropColour][dropLM] [changeType]
29. [dropType]
3. [addDir][dropLM]
1. [changeLM]
6. [addDir][dropColour]
2. [changeType]
19. [addDir][dropColour] [dropType]
17. [addDir][dropColour] [dropLM]
22. [addLM] [changeColour]
4. [changeColour]
8. [addColour][changeLM]
11. [addLM][dropColour]
15. [addDir][dropColour] [changeType]
29. [dropType]
5. [addDir][changeColour]
13. [addColour][addDir] [dropType][dropLM]
14. [addDir]
12. [addColour][addDir] [dropLM][dropNumber]
16. [addDir][dropColour] [dropDir][changeType]
18. [addDir][dropColour] [dropNumber][changeType]
21. [addLM]
23. [addLM][dropColour] [dropType]
24. [addType][changeColour]
26. [dropColour][changeLM]
7. [addColour][addType] [changeLM]
9. [addDir][changeColour] [changeType]
10. [addDir][changeType]
20. [addDir][dropLM] [changeColour]
25. [dropColour]
27. [dropColour][changeType]
28. [dropColour][dropLM] [changeType]

Figure 8.2: The description set and the uninformed set.

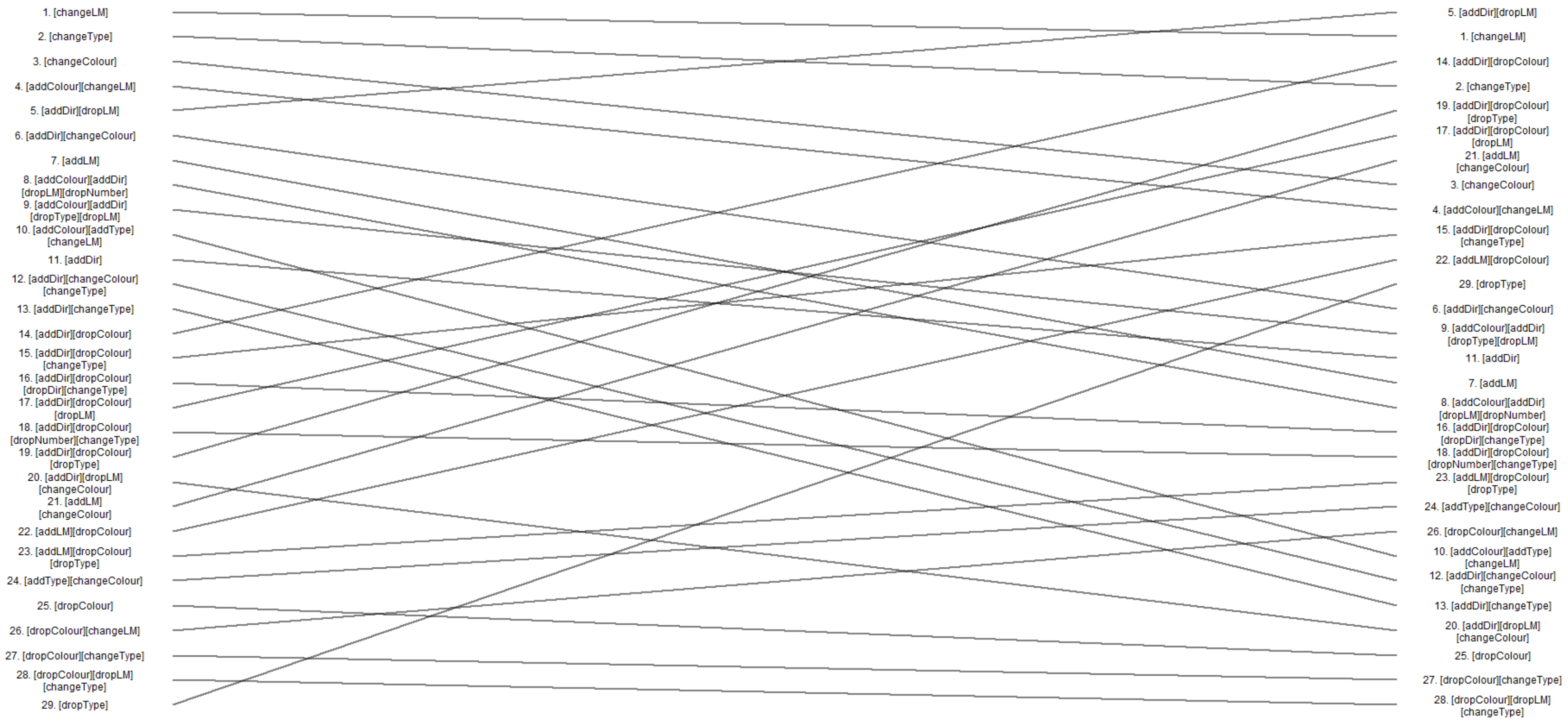

1. [changeLM]
2. [changeType]
3. [changeColour]
4. [addColour][changeLM]
5. [addDir][dropLM]
6. [addDir][changeColour]
7. [addLM]
8. [addColour][addDir] [dropLM][dropNumber]
9. [addColour][addDir] [dropType][dropLM]
10. [addColour][addType] [changeLM]
11. [addDir]
12. [addDir][changeColour] [changeType]
13. [addDir][changeType]
14. [addDir][dropColour]
15. [addDir][dropColour] [changeType]
16. [addDir][dropColour] [dropDir][changeType]
17. [addDir][dropColour] [dropLM]
18. [addDir][dropColour] [dropNumber][changeType]
19. [addDir][dropColour] [dropType]
20. [addDir][dropLM] [changeColour]
21. [addLM] [changeColour]
22. [addLM][dropColour]
23. [addLM][dropColour] [dropType]
24. [addType][changeColour]
25. [dropColour]
26. [dropColour][changeLM]
27. [dropColour][changeType]
28. [dropColour][dropLM] [changeType]
29. [dropType]
5. [addDir][dropLM]
1. [changeLM]
14. [addDir][dropColour]
2. [changeType]
19. [addDir][dropColour] [dropType]
17. [addDir][dropColour] [dropLM]
21. [addLM] [changeColour]
3. [changeColour]
4. [addColour][changeLM]
15. [addDir][dropColour] [changeType]
22. [addLM][dropColour]
29. [dropType]
6. [addDir][changeColour]
9. [addColour][addDir] [dropType][dropLM]
11. [addDir]
7. [addLM]
8. [addColour][addDir] [dropLM][dropNumber]
16. [addDir][dropColour] [dropDir][changeType]
18. [addDir][dropColour] [dropNumber][changeType]
23. [addLM][dropColour] [dropType]
24. [addType][changeColour]
26. [dropColour][changeLM]
10. [addColour][addType] [changeLM]
12. [addDir][changeColour] [changeType]
13. [addDir][changeType]
20. [addDir][dropLM] [changeColour]
25. [dropColour]
27. [dropColour][changeType]
28. [dropColour][dropLM] [changeType]

Figure 8.3: The markup set and the uninformed set.

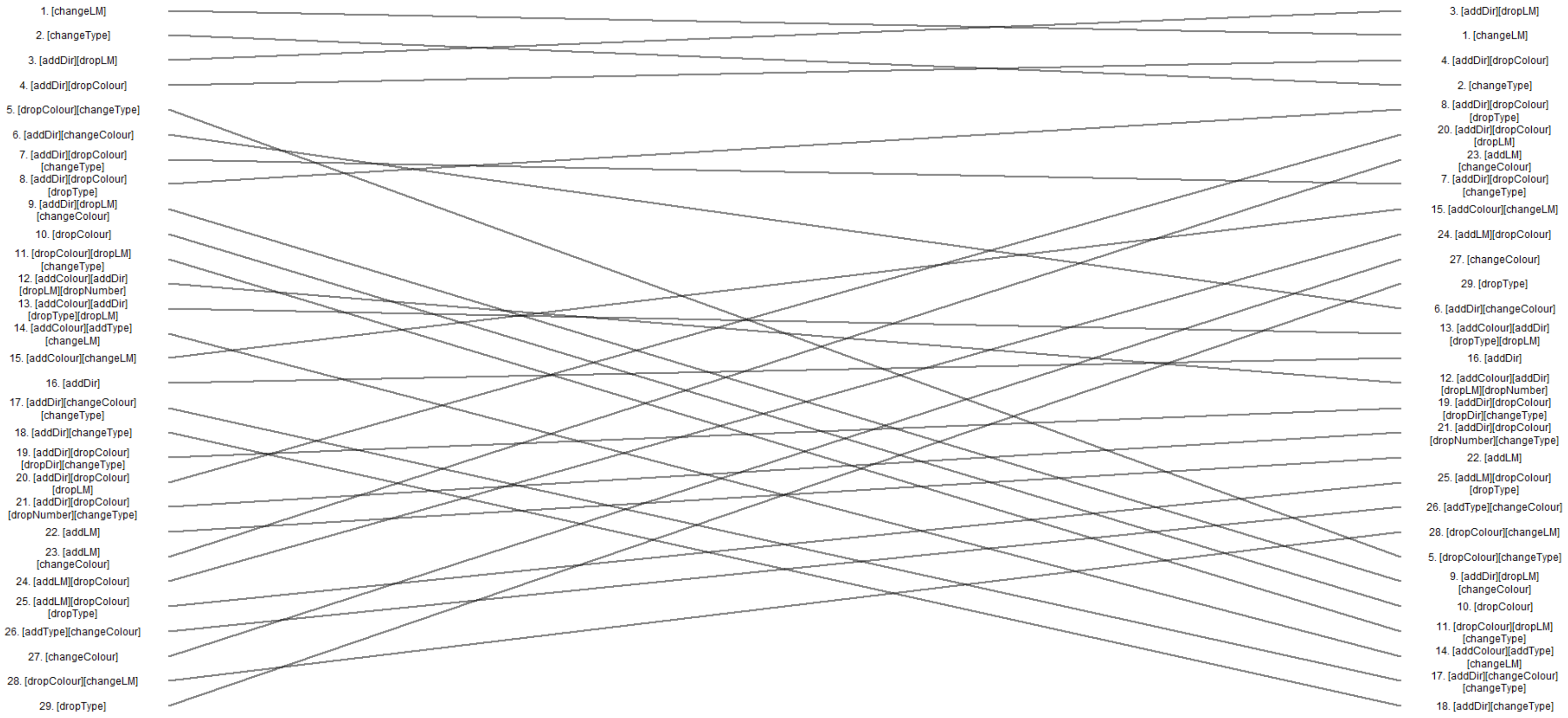

1. [changeLM]
2. [changeType]
3. [addDir][dropLM]
4. [addDir][dropColour]
5. [dropColour][changeType]
6. [addDir][changeColour]
7. [addDir][dropColour] [changeType]
8. [addDir][dropColour] [dropType]
9. [addDir][dropLM] [changeColour]
10. [dropColour]
11. [dropColour][dropLM] [changeType]
12. [addColour][addDir] [dropLM][dropNumber]
13. [addColour][addDir] [dropType][dropLM]
14. [addColour][addType] [changeLM]
15. [addColour][changeLM]
16. [addDir]
17. [addDir][changeColour] [changeType]
18. [addDir][changeType]
19. [addDir][dropColour] [dropDir][changeType]
20. [addDir][dropColour] [dropLM]
21. [addDir][dropColour] [dropNumber][changeType]
22. [addLM]
23. [addLM] [changeColour]
24. [addLM][dropColour]
25. [addLM][dropColour] [dropType]
26. [addType][changeColour]
27. [changeColour]
28. [dropColour][changeLM]
29. [dropType]
3. [addDir][dropLM]
1. [changeLM]
4. [addDir][dropColour]
2. [changeType]
8. [addDir][dropColour] [dropType]
20. [addDir][dropColour] [dropLM]
23. [addLM] [changeColour]
7. [addDir][dropColour] [changeType]
15. [addColour][changeLM]
24. [addLM][dropColour]
27. [changeColour]
29. [dropType]
6. [addDir][changeColour]
13. [addColour][addDir] [dropType][dropLM]
16. [addDir]
12. [addColour][addDir] [dropLM][dropNumber]
19. [addDir][dropColour] [dropDir][changeType]
21. [addDir][dropColour] [dropNumber][changeType]
22. [addLM]
25. [addLM][dropColour] [dropType]
26. [addType][changeColour]
28. [dropColour][changeLM]
5. [dropColour][changeType]
9. [addDir][dropLM] [changeColour]
10. [dropColour]
11. [dropColour][dropLM] [changeType]
14. [addColour][addType] [changeLM]
17. [addDir][changeColour] [changeType]
18. [addDir][changeType]

Figure 8.4: The querying set and the uninformed set.

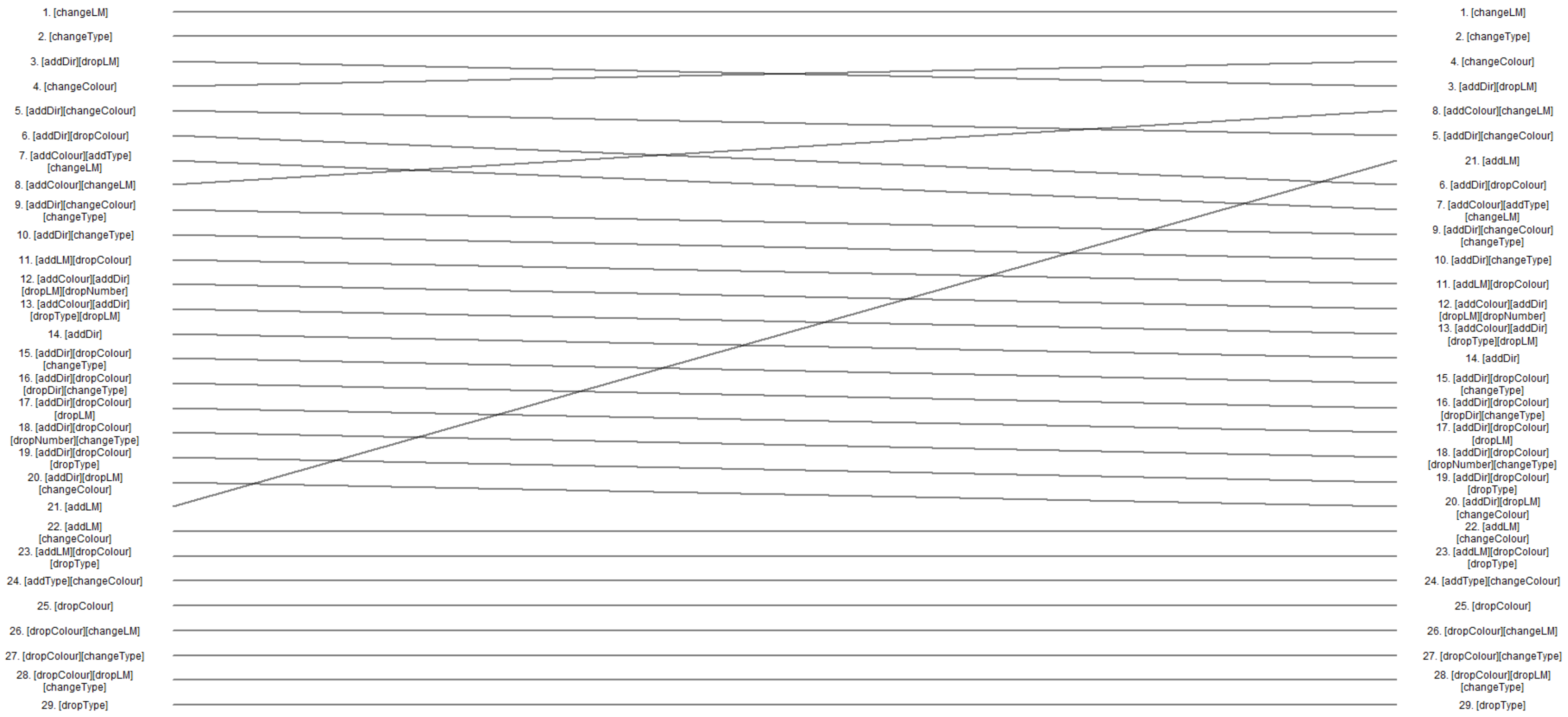

1. [changeLM]
2. [changeType]
3. [addDir][dropLM]
4. [changeColour]
5. [addDir][changeColour]
6. [addDir][dropColour]
7. [addColour][addType] [changeLM]
8. [addColour][changeLM]
9. [addDir][changeColour] [changeType]
10. [addDir][changeType]
11. [addLM][dropColour]
12. [addColour][addDir] [dropLM][dropNumber]
13. [addColour][addDir] [dropType][dropLM]
14. [addDir]
15. [addDir][dropColour] [changeType]
16. [addDir][dropColour] [dropDir][changeType]
17. [addDir][dropColour] [dropLM]
18. [addDir][dropColour] [dropNumber][changeType]
19. [addDir][dropColour] [dropType]
20. [addDir][dropLM] [changeColour]
21. [addLM]
22. [addLM] [changeColour]
23. [addLM][dropColour] [dropType]
24. [addType][changeColour]
25. [dropColour]
26. [dropColour][changeLM]
27. [dropColour][changeType]
28. [dropColour][dropLM] [changeType]
29. [dropType]
1. [changeLM]
2. [changeType]
4. [changeColour]
3. [addDir][dropLM]
8. [addColour][changeLM]
5. [addDir][changeColour]
21. [addLM]
6. [addDir][dropColour]
7. [addColour][addType] [changeLM]
9. [addDir][changeColour] [changeType]
10. [addDir][changeType]
11. [addLM][dropColour]
12. [addColour][addDir] [dropLM][dropNumber]
13. [addColour][addDir] [dropType][dropLM]
14. [addDir]
15. [addDir][dropColour] [changeType]
16. [addDir][dropColour] [dropDir][changeType]
17. [addDir][dropColour] [dropLM]
18. [addDir][dropColour] [dropNumber][changeType]
19. [addDir][dropColour] [dropType]
20. [addDir][dropLM] [changeColour]
22. [addLM] [changeColour]
23. [addLM][dropColour] [dropType]
24. [addType][changeColour]
25. [dropColour]
26. [dropColour][changeLM]
27. [dropColour][changeType]
28. [dropColour][dropLM] [changeType]
29. [dropType]

Figure 8.5: The description set and the markup set.

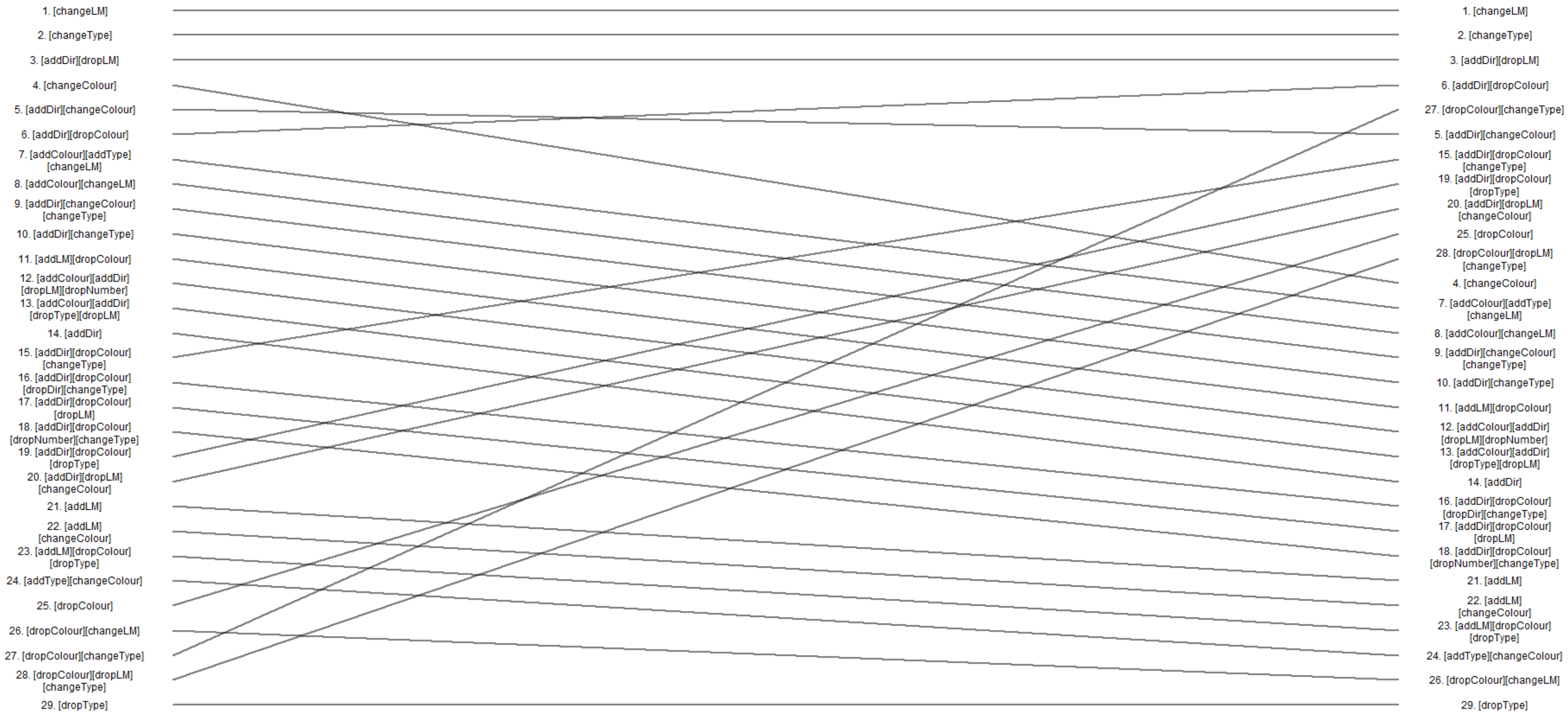

1. [changeLM]
2. [changeType]
3. [addDir][dropLM]
4. [changeColour]
5. [addDir][changeColour]
6. [addDir][dropColour]
7. [addColour][addType] [changeLM]
8. [addColour][changeLM]
9. [addDir][changeColour] [changeType]
10. [addDir][changeType]
11. [addLM][dropColour]
12. [addColour][addDir] [dropLM][dropNumber]
13. [addColour][addDir] [dropType][dropLM]
14. [addDir]
15. [addDir][dropColour] [changeType]
16. [addDir][dropColour] [dropDir][changeType]
17. [addDir][dropColour] [dropLM]
18. [addDir][dropColour] [dropNumber][changeType]
19. [addDir][dropColour] [dropType]
20. [addDir][dropLM] [changeColour]
21. [addLM]
22. [addLM] [changeColour]
23. [addLM][dropColour] [dropType]
24. [addType][changeColour]
25. [dropColour]
26. [dropColour][changeLM]
27. [dropColour][changeType]
28. [dropColour][dropLM] [changeType]
29. [dropType]
1. [changeLM]
2. [changeType]
3. [addDir][dropLM]
6. [addDir][dropColour]
27. [dropColour][changeType]
5. [addDir][changeColour]
15. [addDir][dropColour] [changeType]
19. [addDir][dropColour] [dropType]
20. [addDir][dropLM] [changeColour]
25. [dropColour]
28. [dropColour][dropLM] [changeType]
4. [changeColour]
7. [addColour][addType] [changeLM]
8. [addColour][changeLM]
9. [addDir][changeColour] [changeType]
10. [addDir][changeType]
11. [addLM][dropColour]
12. [addColour][addDir] [dropLM][dropNumber]
13. [addColour][addDir] [dropType][dropLM]
14. [addDir]
16. [addDir][dropColour] [dropDir][changeType]
17. [addDir][dropColour] [dropLM]
18. [addDir][dropColour] [dropNumber][changeType]
21. [addLM]
22. [addLM] [changeColour]
23. [addLM][dropColour] [dropType]
24. [addType][changeColour]
26. [dropColour][changeLM]
29. [dropType]

Figure 8.6: The description set and the querying set.

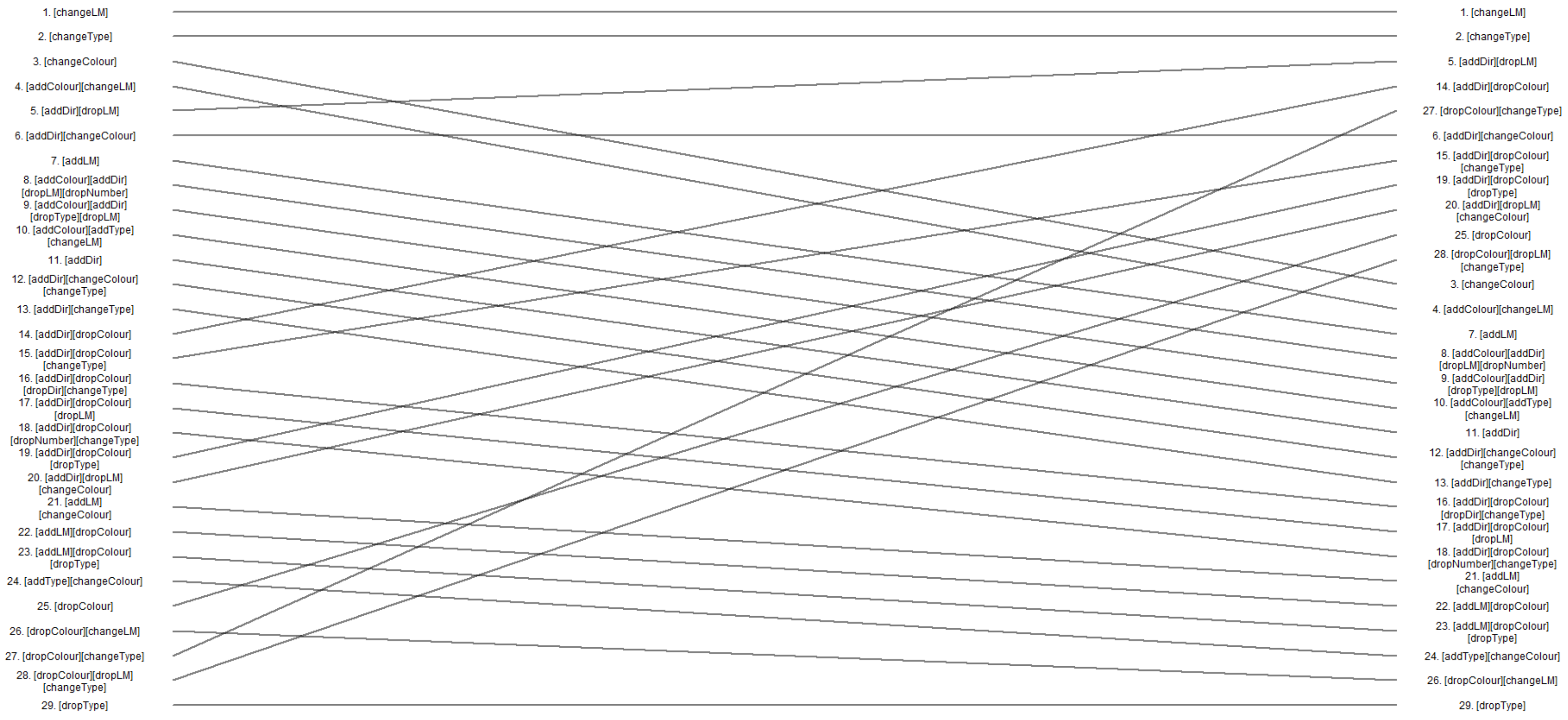

1. [changeLM]
2. [changeType]
3. [changeColour]
4. [addColour][changeLM]
5. [addDir][dropLM]
6. [addDir][changeColour]
7. [addLM]
8. [addColour][addDir] [dropLM][dropNumber]
9. [addColour][addDir] [dropType][dropLM]
10. [addColour][addType] [changeLM]
11. [addDir]
12. [addDir][changeColour] [changeType]
13. [addDir][changeType]
14. [addDir][dropColour]
15. [addDir][dropColour] [changeType]
16. [addDir][dropColour] [dropDir][changeType]
17. [addDir][dropColour] [dropLM]
18. [addDir][dropColour] [dropNumber][changeType]
19. [addDir][dropColour] [dropType]
20. [addDir][dropLM] [changeColour]
21. [addLM] [changeColour]
22. [addLM][dropColour]
23. [addLM][dropColour] [dropType]
24. [addType][changeColour]
25. [dropColour]
26. [dropColour][changeLM]
27. [dropColour][changeType]
28. [dropColour][dropLM] [changeType]
29. [dropType]
1. [changeLM]
2. [changeType]
5. [addDir][dropLM]
14. [addDir][dropColour]
27. [dropColour][changeType]
6. [addDir][changeColour]
15. [addDir][dropColour] [changeType]
19. [addDir][dropColour] [dropType]
20. [addDir][dropLM] [changeColour]
25. [dropColour]
28. [dropColour][dropLM] [changeType]
3. [changeColour]
4. [addColour][changeLM]
7. [addLM]
8. [addColour][addDir] [dropLM][dropNumber]
9. [addColour][addDir] [dropType][dropLM]
10. [addColour][addType] [changeLM]
11. [addDir]
12. [addDir][changeColour] [changeType]
13. [addDir][changeType]
16. [addDir][dropColour] [dropDir][changeType]
17. [addDir][dropColour] [dropLM]
18. [addDir][dropColour] [dropNumber][changeType]
21. [addLM] [changeColour]
22. [addLM][dropColour]
23. [addLM][dropColour] [dropType]
24. [addType][changeColour]
26. [dropColour][changeLM]
29. [dropType]

Figure 8.7: The markup set and the querying set.

differently. This observation is mostly based on visual impression, but it corresponds with our observations about the bar plots.

We attempt to quantify the similarity between the different condition sets by calculating the cosine similarity between them. To do this, we represent each condition set as a vector (where each element in the vector is filled with the proportion of an event in the set) and calculate the cosine between them. The results are presented in Table 8.10. Overall we notice that the markup and the description set have the highest similarity to each other while each has a larger distance to the uninformed set. The closest similarity between the uninformed set and any of the other sets is to the query set.

	Uninformed	**Description**	**Markup**	**Query**
Uninformed	-	0.747	0.632	0.815
Description	0.747	-	0.953	0.903
Markup	0.632	0.953	-	0.790
Query	0.815	0.903	0.790	-

Table 8.10: The cosine similarity between the different condition sets.

8.4.3 Strategies and the Different Condition Sets

In this section we investigate and discuss the differences between the condition sets. We base this discussion on the order of the most frequents events in the overall data set presented in Table 8.3. In Section 8.3.2 we discussed the most frequent events in the overall set and provided explanations for the strategies underlying the events.

The most frequent event was [**changeLM**]. It is also the most frequent event in the description set, the markup set and the query set. In the uninformed set it is the second most frequent event. The most frequent event in the query set is [**addDir**] [**dropLM**], which is the second most frequent event overall. It is only the third most frequent event in the query set and the description set and the fourth most frequent event in the markup set. This suggests that this event was particularly relevant when the participants could not request information from the robot. This is plausible because, as discussed earlier, it represents an action where the participants discarded an unreliable attribute and replaced it with a more reliable one. The third most frequent event overall is [**changeType**]. It is the second most frequent event in the description set, the markup set and the query set. It is only at Rank 4 in the uninformed data set. This suggests that this event is more likely to occur if the participants are able to request information from the robot, and less likely if they are not able to. This is consistent with our earlier observation that this event represents an action where the participants adopt the robot's model of the world even if it contradicts their own perception. It is plausible that participants are more willing to do so if they are able to gain direct information about the robot's understanding. In the description condition, they can do this by requesting a description from the robot. In the markup condition they do this through the markup provided by the system. In the querying condition they can use queries to indirectly form an understanding of the robot's understanding of the scene. In the uninformed condition however, they are not able to request information directly, and therefore the strategy becomes less attractive. A similar observation can be made for the [**changeColour**] event. It is the fifth most frequent event overall, the third most frequent event in the markup set and the fourth most frequent event in the description set. In both the uninformed set and the query set it is ranked much lower (in the uninformed set it ties for Rank 8 and in the query set it was not observed). The low rank in the uninformed set appears consistent with our earlier observation for the [**changeType**] event, namely that participants are less able and likely to adapt to the system's perception error if they are not able to request information about the robot's perception. The fourth most frequent event ([**addDir**] [**dropColour**]) is at Rank 3 (ties for Rank 2) in the uninformed set, at Rank 6 in the description set (ties for Rank 5) and at Rank 4 in the query set. It was not observed in the markup set. This event is similar to the event at Rank 2 ([**addDir**] [**dropLM**]) in that it represents an event in which a directional attribute was added to replace an attribute that could be affected by perception errors. The distribution of the ranks is also similar in the sense that its highest rank is in the uninformed set.

Overall we find that events are ranked differently between the conditions. The rankings in the Description Phase and the Markup Phase appear to be somewhat similar, and dissimilar from the other phases. This suggests that participants used similar strategies if they could request information through description or markup.

8.4.4 Summary

We find that there are differences between the sets of events observed in the different conditions. We furthermore find that the differences can be explained by the type of information that was available in each condition. If the participants were able to request information about the robot's understanding of the world, they tended to produce events in which they adapted their expressions to the robot's perception. Such events involve the change of the value of a basic attribute, such as [**changeType**] and [**changeColour**].

If the participants did not request information from the robot, they had no efficient way to adapt to the robot's understanding of the scene. As a consequence they were less likely to produce adapted expressions, and more likely to pursue approaches that were independent of the problematic parts of the robot's perception (the colour classification and the type classification). Instead they produced expressions that relied on spatial attributes that were robust against the robot's perception problems.

Conclusions:

1. If information request options were available, the participants used them to formulate expressions that were adjusted to the robot's understanding of the scene.

2. If no information was available, the participants tended to remove attributes that could be affected by perception errors. To ensure that the expressions were still distinguishing, they tended to include direction

based descriptions instead.

8.5 Summary

In this chapter we investigated the attribute selection in resolution sequences and the differences between the initial and the final, successful references. We found that, depending on the information request options used, the participants were more likely to use different reformulation strategies. If the participants were able to elicit information about the robot's understanding of the world they adapted their expressions to the robot's model of the world, even if it meant that they used expressions that were not appropriate for their actual perception. If they were not able to request information, they tended to use strategies in which they avoided attributes that could be affected by perception errors, and instead used attributes that robust against perception errors.

Chapter 9

Conclusion

9.1 Summary

In this thesis we investigated dialogues between a human user and a robot that is affected by perception errors. We investigated how participants resolved problems in the dialogue that were caused by the perception errors, and different methods of allowing the participants to request information about the robot's understanding of the world. In order to investigate this phenomenon, we performed the Toy Block experiment. In this experiment, a human user interacted with a simulated robot through a text based dialogue interface. The user instructed the robot to complete a series of object manipulation tasks.

In the first phase we performed a baseline version of the experiment where the robot was not affected by perception errors. In the second step we introduced errors into the robot's perception and analysed the effect of the errors on the participants' experience during the experiment and the participants' ability to solve the tasks. In the next three phases we introduced different options for the participants to request information about the robot's perception of the scene and investigated the effect these options had on the task success and the user experience. Information could be requested as verbal descriptions in the Description Phase and as visual markup in the Markup Phase. In the Querying Phase the participants could ask the system simple questions. Using these information request options, the participants were able to infer how the robot perceived the world, and they were able to determine where the robot's perception diverged from their own.

Overall we found that introducing errors into the robot's perception makes the task more difficult and increases the participants' frustration. On the other hand, after we introduced the information request options, the tasks became easier and the participants became more satisfied.

Based on this we conclude that in a situated human-robot dialogue, problems may arise if the robot is affected by perception errors. If we give the participants access to information about what the robot perceives, they are able to use it to compensate for the problems and they become less frustrated. This lower level of frustration may simply be due to the fact that they are more successful in the task. However, it may also be due to the fact that if participants are able to understand the robot's perception of the world, the errors the robot makes become more understandable.

We compared the success of the different information request options. We found that the visual markup option and the option to request information through dialogue from the Querying Phase were similarly effective and more effective than the description option. Also, even though participants posed queries more often on average than they used the markup option, there was almost no difference in the average completion times between the markup phase and the querying phase. This suggests that the markup option and the querying option are equivalent in terms of the potential to help the participants. It might be worth noting that the querying option only uses the spoken language modality in which the interaction already takes place, while the visual markup option uses visual output to present information. This does not present a problem in the application scenario investigated in this thesis since the user shares the robot's perspective through a video connection. In other scenarios however (e.g. a scenario where the user and the robot are present in the same space and no video connection is used or a scenario where the user has to visually attend to something else in parallel) it might be preferable to use a purely dialogue based option.

We investigated the strategies the participants used to resolve problems that were caused by perception errors. In the first step we analysed the actions the participants took between the action in which a problem arose and the action in which the problem was resolved. We found that retrieving and accumulating information was an important step of all successful strategies. Depending on the information available, the participants could either request a scene description, visual markup or ask questions about the robot's perception of the scene. If these options were not available, they could only use a trial-and-error method to test out the robot's understanding.

If no information request options were available, the participants used a trial-and-error strategy to resolve problems. Interestingly, we observed that some participants attempted to "query" the robot's understanding of the world in an indirect way. They asked the robot to pick up objects they did not actually need to move to complete the task to find out how the robot would interpret the instruction. This highlights the facts that the participants attempted to test out the robot's model of the world.

In the second step we analysed the referring expressions that were used in the problem resolutions. We compared the initial expressions that were used in the actions in which the problems arose to the expressions that were used in the actions in which the problems were successfully resolved. Overall we identified two major strategies that were also connected to the type of information that was available. If participants were able to gain explicit complete information about the robot's understanding of the scene, such as in the Description Phase and the Markup Phase, they tended to use the information about the robot's understanding to formulate expressions that were appropriate for the robot. The participants thereby aligned their model to the robot's model of the world.

If they were not able to request explicit complete information, such as in the Error Phase where no assistance was available, or in the Querying Phase, where every piece of information had to be individually requested through questions, they tended towards a different approach. We found that after a resolution problem occurred, the participants tended to avoid attributes that could be affected by errors and instead replaced them with attributes that were robust to errors.

9.2 Reflections

With this work we have shown that if a robot, that is engaged in a dialogue with a human user, is affected by perception errors, problems can occur in the dialogue. We have further shown that participants are, to a certain extent, able to resolve these problems. If participants are given an option to understand how the robot perceives the world, they are able to form a model of the robot's problems and compensate for them.

Overall, we believe that the results of the experiment are clear enough draw our conclusions. However, as we noted earlier, the experiment was perhaps lacking an element of pressure. This was reflected in the surprisingly low frustration ratings by the participants for the Error Phase. We nevertheless believe that the observed reactions and strategies are valuable. We suppose that participants might be more inclined to abandon difficult scenes if an element of time pressure is present, but we still believe that successful resolution strategies would be overall the same, whether participants are frustrated or not.

From this we would suggest for the design of future human-robot dialogue systems, that the systems should be designed with the possibility of errors in mind. While one option would be to modify the systems perceptual capabilities to accommodate divergences between the user's understanding of the scene and the robot's understanding, we suggest that another, perhaps in many situations simpler, approach would be to enable the user to understand the robot's perception, and thereby allow them to adjust to the robot's problems. In this sense we suggest to make the robot's model of the world understandable, similar to the way (Kruse *et al.*, 2010) suggest to make robot movement behaviour *legible* to humans.

There are however, some outstanding questions we were not able to answer in this work. For example, we were unfortunately not able to develop a proper model of the relationship between the sequences of actions in problem resolutions (discussed in Chapter 7) and our findings related to the choice of attributes in referring expressions in the same sequences (discussed in Chapter 8), which would have provided a more complete description of the events during problems resolutions.

While we investigated in this thesis how human users react to perception based problems in dialogue and how users can be enabled to resolve them, we unfortunately were not able to investigate the opposite direction, i.e. options to allow the robot to detect and resolve problems in the dialogue. We particularly believe that it would be interesting to investigate whether it is possible for the robot to infer what the perception error consists in, based on the type of the problem in the dialogue and its perception of the scene. It may be possible to address this in part with data generated in the experiments discussed in the thesis.

While the investigation in this thesis focused on problems arising from problems and mismatches in visual perception, we believe that the findings may be generalized to human-computer interaction in general. Human-computer interaction often involves different sources of background information or knowledge, such as databases or maps. A mismatch between what the background information contains and what a user presupposes as general knowledge, may lead to mismatches between the user's understanding and the system's understanding, similar to the way problems in perception may lead to mismatches. For example, map data for navigation may become outdated or may not contain points of interest that are relevant to a specific user's interest. Weather reports may diverge from the actual weather at a specific location. We therefore suggest to enable systems to explicitly account for such problems and to provide ways to discuss them with the user.

9.3 Future work

During the experiment we encountered a number of interesting problems that would be interesting topics of research, but that we had to exclude from the scope of this work. In particular we find the following topics could be of interest:

Alignment: It would interesting to investigate in how far the expressions used by the participants align with the expression used by the system, e.g. during the scene descriptions.

More realistic error models: The errors that were introduced into the robot's perception were manually designed in order to enable us to directly compare the different phases of the experiment to one another, and to clearly identify the points in the dialogues where they caused problems. It would be interesting to perform this experiment using an actual vision system (similar to the approach by Liu *et al.* (2012)). In particular it would also be important to test how well the effect of the different information request options scales with the extent of the perception errors.

More flexible queries: We found in the evaluation that the querying option was well accepted and effective, even though it only allowed a limited range of questions. One possible future direction would be to use a more flexible querying option that allows more complex questions and delivers partial descriptions of the scene.

More realistic application situations: We found during the evaluation that participant were less frustrated with the problems in the dialogue than we had anticipated (and than we would expect them be in a real-life robot interaction scenario). One possibility to address this issue would be to introduce an element of pressure into the tasks, for example by introducing a time limit, or a reward that is determined based on the performance of the participant.

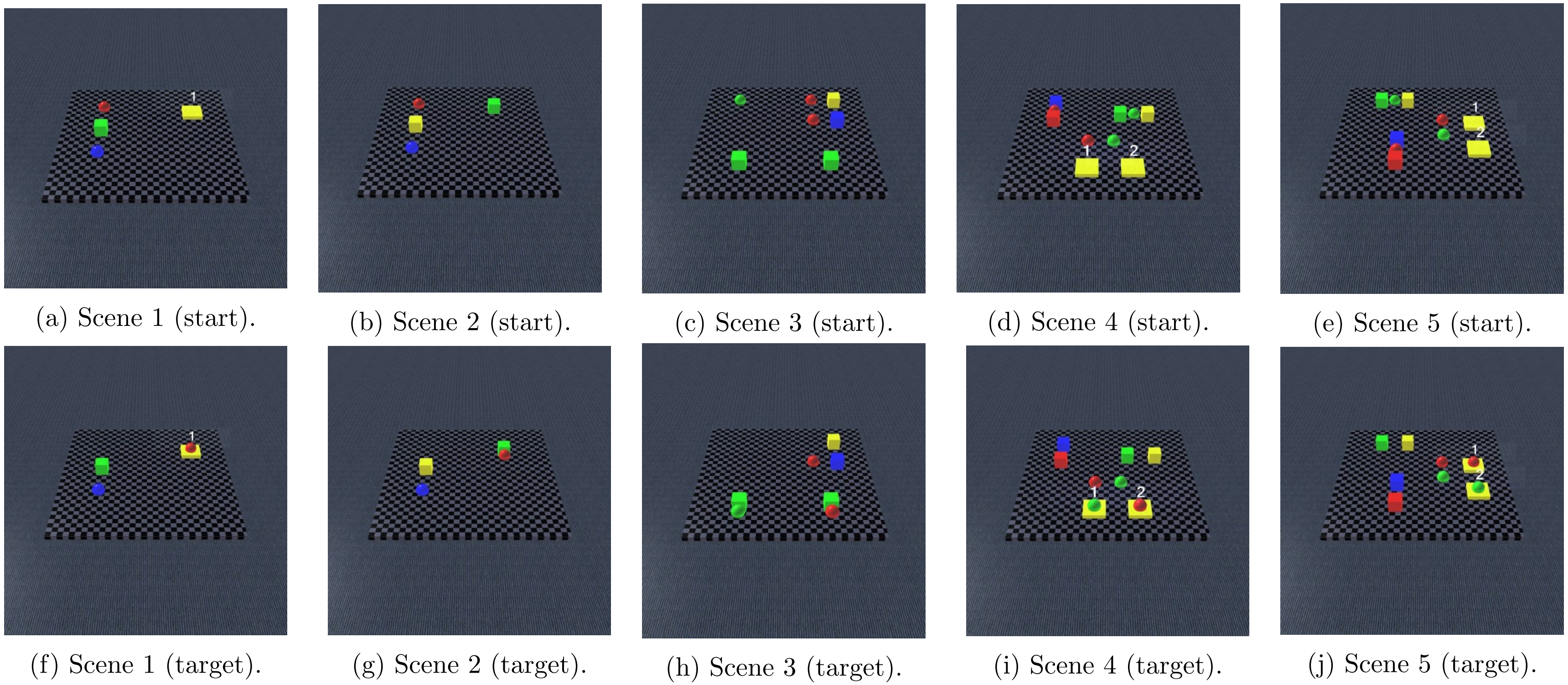
Robot side adapatation: Using the data from the experiment it would be interesting to investigate whether a machine learning system can learn to predict what the perception errors consist in based on the problem and the content of the scene.

Appendix A

Scenes and Errors

This appendix contains information related to the scenes that were used in the experiment and the errors that were introduced into the robot's perception.

- Figure A.1 to Figure A.4 contain images of the start scenes and the corresponding target scenes.
- Table A.1 contains descriptions of the errors that were introduced into the scenes during the Error Phase, the Description Phase, the Markup Phase and the Querying Phase.
- The referring expression templates that were used to capture references to objects that were affected by perception errors are presented in Figure A.5 and Figure A.6.



(a) Scene 1 (start).

(b) Scene 2 (start).

(c) Scene 3 (start).

(d) Scene 4 (start).

(e) Scene 5 (start).

(f) Scene 1 (target).

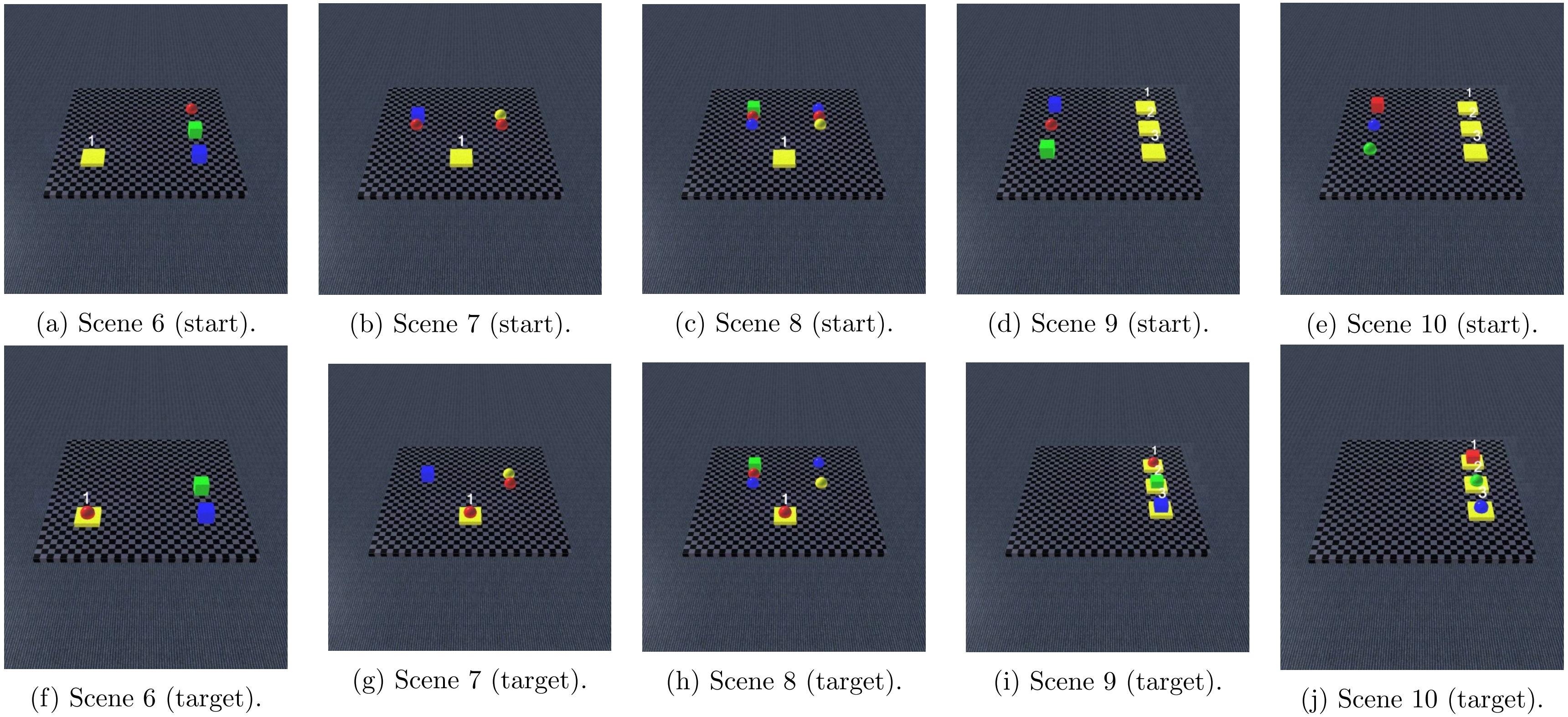
(g) Scene 2 (target).

(h) Scene 3 (target).

(i) Scene 4 (target).

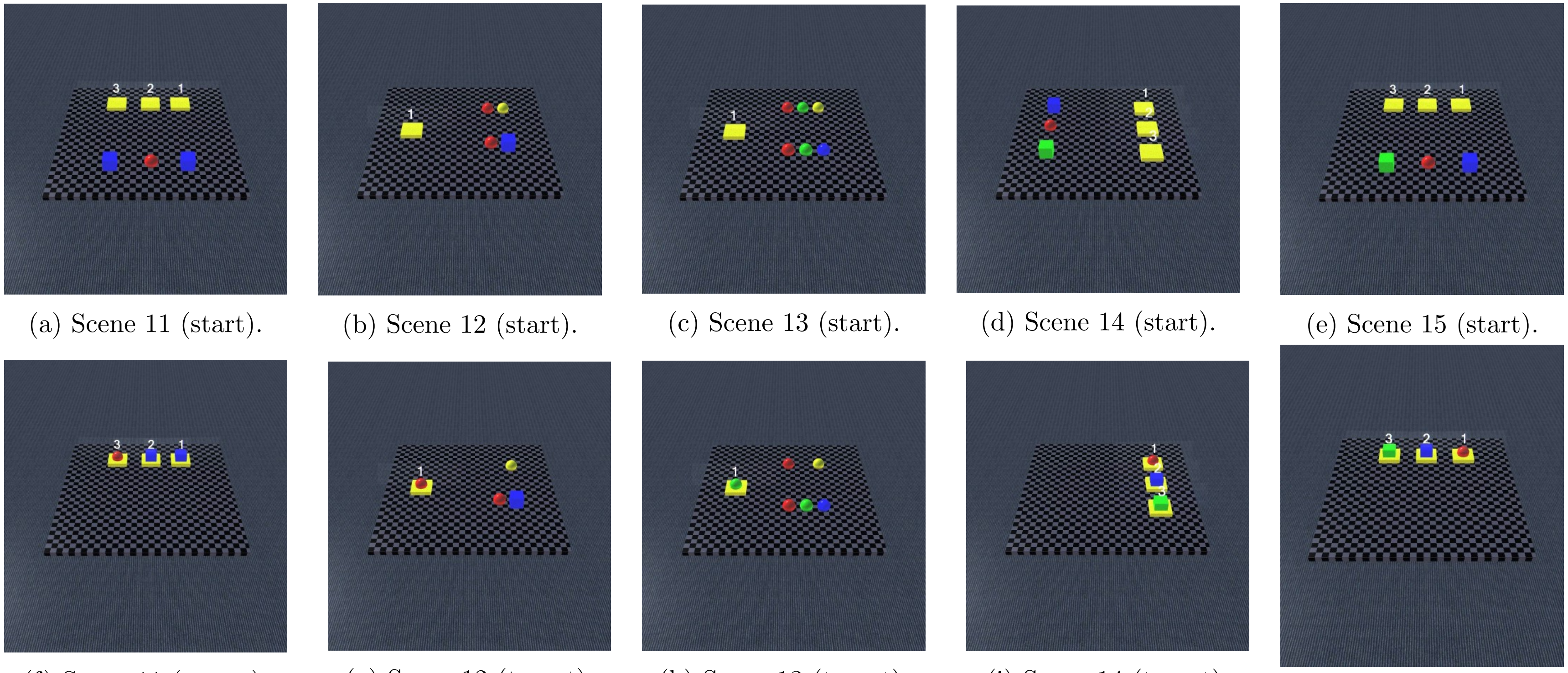
(j) Scene 5 (target).

Figure A.1: Start and target scenes for Scene 1 to Scene 5.



(a) Scene 6 (start).
(b) Scene 7 (start).
(c) Scene 8 (start).
(d) Scene 9 (start).
(e) Scene 10 (start).
(f) Scene 6 (target).
(g) Scene 7 (target).
(h) Scene 8 (target).
(i) Scene 9 (target).
(j) Scene 10 (target).

Figure A.2: Start and target scenes for Scene 6 to Scene 10.



(a) Scene 11 (start).

(b) Scene 12 (start).

(c) Scene 13 (start).

(d) Scene 14 (start).

(e) Scene 15 (start).

(f) Scene 11 (target).

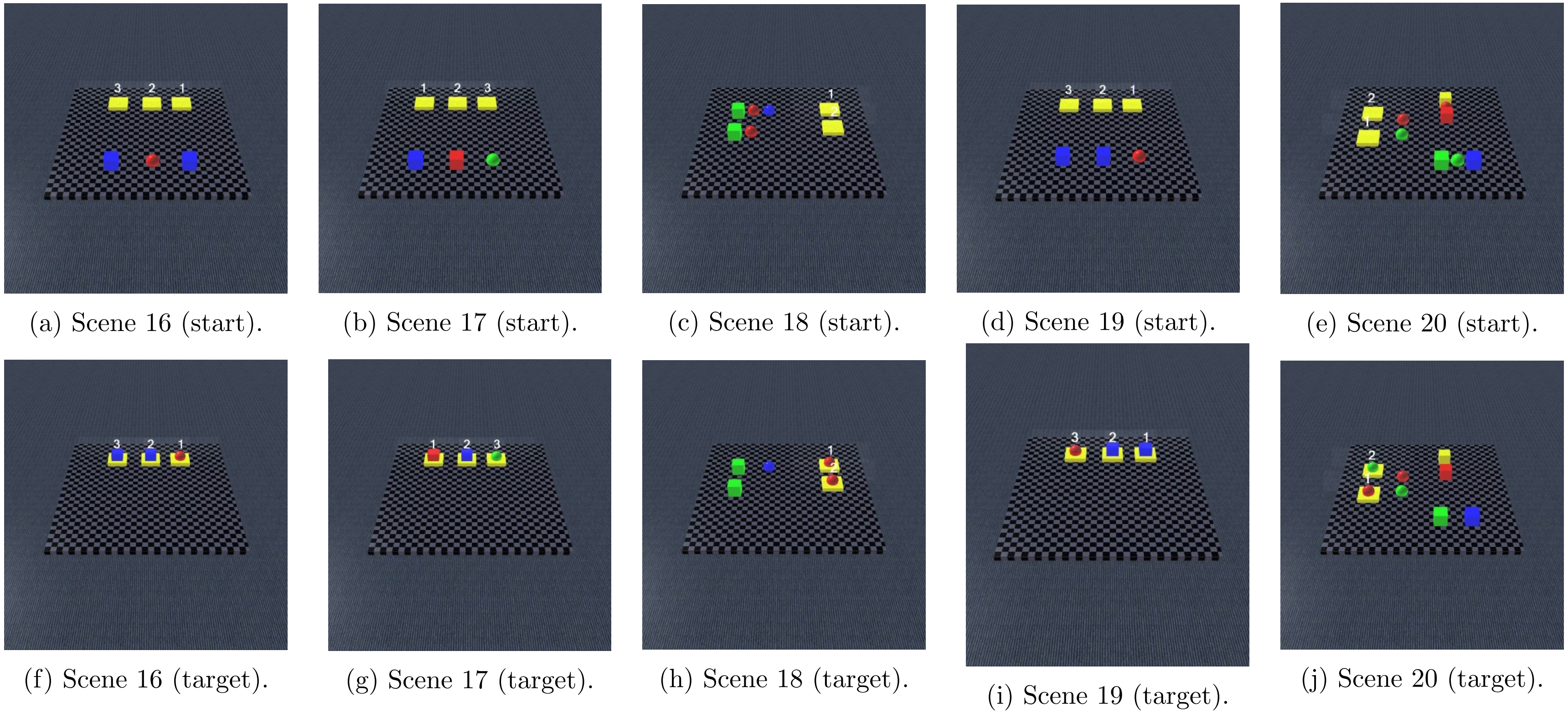
(g) Scene 12 (target).

(h) Scene 13 (target).

(i) Scene 14 (target).

(j) Scene 15 (target).

Figure A.3: Start and target scenes for Scene 10 to Scene 15.



(a) Scene 16 (start).

(b) Scene 17 (start).

(c) Scene 18 (start).

(d) Scene 19 (start).

(e) Scene 20 (start).

(f) Scene 16 (target).

(g) Scene 17 (target).

(h) Scene 18 (target).

(i) Scene 19 (target).

(j) Scene 20 (target).

Figure A.4: Start and target scenes for Scene 16 to Scene 20.

Scene	**Error**	**Error type**	**Error situation**
Scene 1	No error		
Scene 2	No error		
Scene 3	The yellow box next to the upper red ball is missing.	Missing object error	Landmark error 1
Scene 4	The yellow box next to the green ball is missing, but the green box remains.	Missing object error	Landmark error 2
Scene 5	The yellow box next to the green ball is missing, but the green box remains.	Missing object error	Landmark error 2.
Scene 6	A critical object is missing (i.e. this scene can not be completed successfully).	Missing object error	Critical object error
Scene 7	The blue box is perceived as a green box	Wrong colour error	Landmark error 1.
Scene 8	The yellow ball next to the red ball that has to be moved is missing, is perceived as green, but the blue ball next to it remains.	Wrong colour error	Landmark error 2
Scene 9	The blue box is perceived as green. It is therefore confusable with the actual green box in the scene.	Wrong colour error	Critical object error
Scene 10	The blue ball is perceived as a red ball.	Wrong colour error	Critical object error
Scene 11	The blue box on the right is perceived as a ball	Wrong type error	Critical object error
Scene 12	The yellow ball next to the red ball is perceived as a box	Wrong type error	Landmark error 1
Scene 13	The red ball next to the upper green ball is perceived as a box, but the yellow ball remains as a landmark.	Wrong type error	Landmark error 2
Scene 14	The green box is perceived as a green ball.	Wrong type error	Critical object error
Scene 15	The green box is perceived as a ball.	Wrong type error	Critical object error
Scene 16	The blue box on the right is perceived as a green box.	Wrong type error	Critical object error
Scene 17	No error		
Scene 18	No error		
Scene 19	No error		
Scene 20	No error		

Table A.1: The error conditions for each scene.

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & * \\ \text{colour} & \text{yellow} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball2} \end{bmatrix}$$

(a) Scene 3.

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & \text{box} \\ \text{colour} & \text{green} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball2} \end{bmatrix}$$

(b) Scene 4.

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & \text{box} \\ \text{colour} & \text{green} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball2} \end{bmatrix}$$

(c) Scene 5.

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & \text{ball} \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & * \\ \text{colour} & * \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball1} \end{bmatrix}$$

(d) Scene 6 (Expression 1).

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & \text{red} \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & * \\ \text{colour} & * \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball1} \end{bmatrix}$$

(e) Scene 6 (Expression 2).

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & * \\ \text{colour} & \text{blue} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball1} \end{bmatrix}$$

(f) Scene 7 (Expression 1).

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & \text{box} \\ \text{colour} & * \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball1} \end{bmatrix}$$

(g) Scene 7 (Expression 2).

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & * \\ \text{colour} & \text{yellow} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball1} \end{bmatrix}$$

(h) Scene 8.

Figure A.5: The referring expression templates for references that initiated clarification sequences (Part 1).

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & \text{thing} \\ \text{Colour} & \text{blue} \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & * \\ \text{colour} & * \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{box1} \end{bmatrix}$$

(a) Scene 9.

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & \text{ball} \\ \text{Colour} & \text{blue} \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & * \\ \text{colour} & * \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball1} \end{bmatrix}$$

(b) Scene 10.

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & * \\ \text{colour} & \text{yellow} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball2} \end{bmatrix}$$

(c) Scene 12 (Expression 1).

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & \text{ball} \\ \text{colour} & * \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball2} \end{bmatrix}$$

(d) Scene 12 (Expression 2).

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & * \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & \text{ball} \\ \text{colour} & \text{red} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{ball3} \end{bmatrix}$$

(e) Scene 13.

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & \text{green} \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & * \\ \text{colour} & * \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{box2} \end{bmatrix}$$

(f) Scene 14.

$$\begin{bmatrix} \text{expression} & \begin{bmatrix} \text{type} & * \\ \text{Colour} & \text{green} \\ \text{rel} & \begin{bmatrix} \text{reltype} & * \\ \text{relatum} & \begin{bmatrix} \text{type} & * \\ \text{colour} & * \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{target-id} & \text{box2} \end{bmatrix}$$

(g) Scene 15.

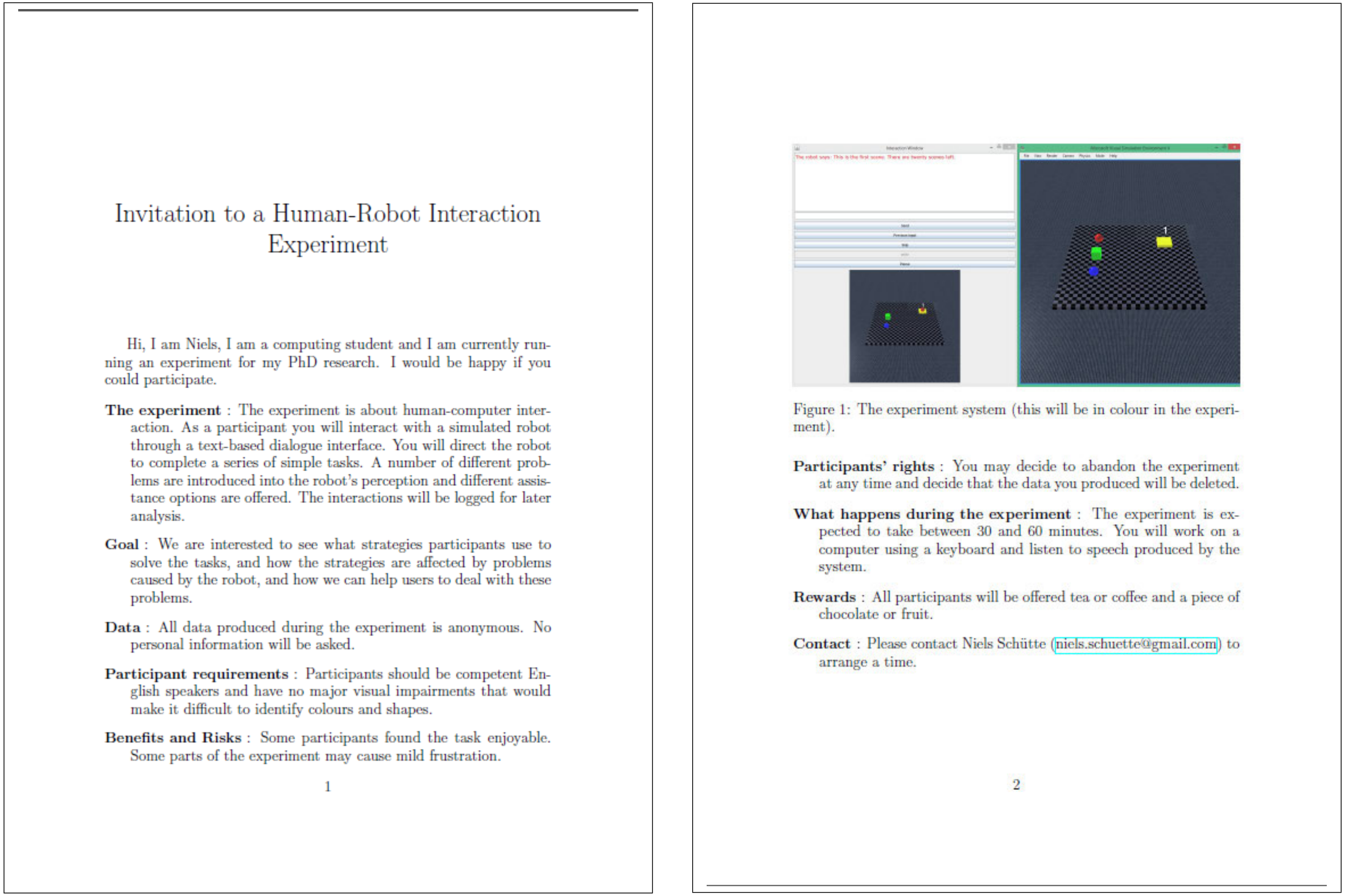
Figure A.6: The referring expression templates for references that initiated clarification sequences (Part 2).

Appendix B

Experiment materials

This appendix contains material that was handed out to the participants in the context of the experiment:

- The information sheet that was given to participants before they agreed to participate in the experiment (Figure B.1).
- The instruction sheets that were given to the participants before they began the experiment (Figure B.2 to Figure B.16).
- The questionnaires the participants were asked to complete after they had finished the experiment (Figure B.17 to Figure B.20).



Invitation to a Human-Robot Interaction Experiment

Hi, I am Niels, I am a computing student and I am currently running an experiment for my PhD research. I would be happy if you could participate.

The experiment : The experiment is about human-computer interaction. As a participant you will interact with a simulated robot through a text-based dialogue interface. You will direct the robot to complete a series of simple tasks. A number of different problems are introduced into the robot's perception and different assistance options are offered. The interactions will be logged for later analysis.

Goal : We are interested to see what strategies participants use to solve the tasks, and how the strategies are affected by problems caused by the robot, and how we can help users to deal with these problems.

Data : All data produced during the experiment is anonymous. No personal information will be asked.

Participant requirements : Participants should be competent English speakers and have no major visual impairments that would make it difficult to identify colours and shapes.

Benefits and Risks : Some participants found the task enjoyable. Some parts of the experiment may cause mild frustration.

1

(a) The front side of the information sheet.

The robot says: This is the first scene. There are twenty scenes left.

Figure 1: The experiment system (this will be in colour in the experiment).

Participants' rights : You may decide to abandon the experiment at any time and decide that the data you produced will be deleted.

What happens during the experiment : The experiment is expected to take between 30 and 60 minutes. You will work on a computer using a keyboard and listen to speech produced by the system.

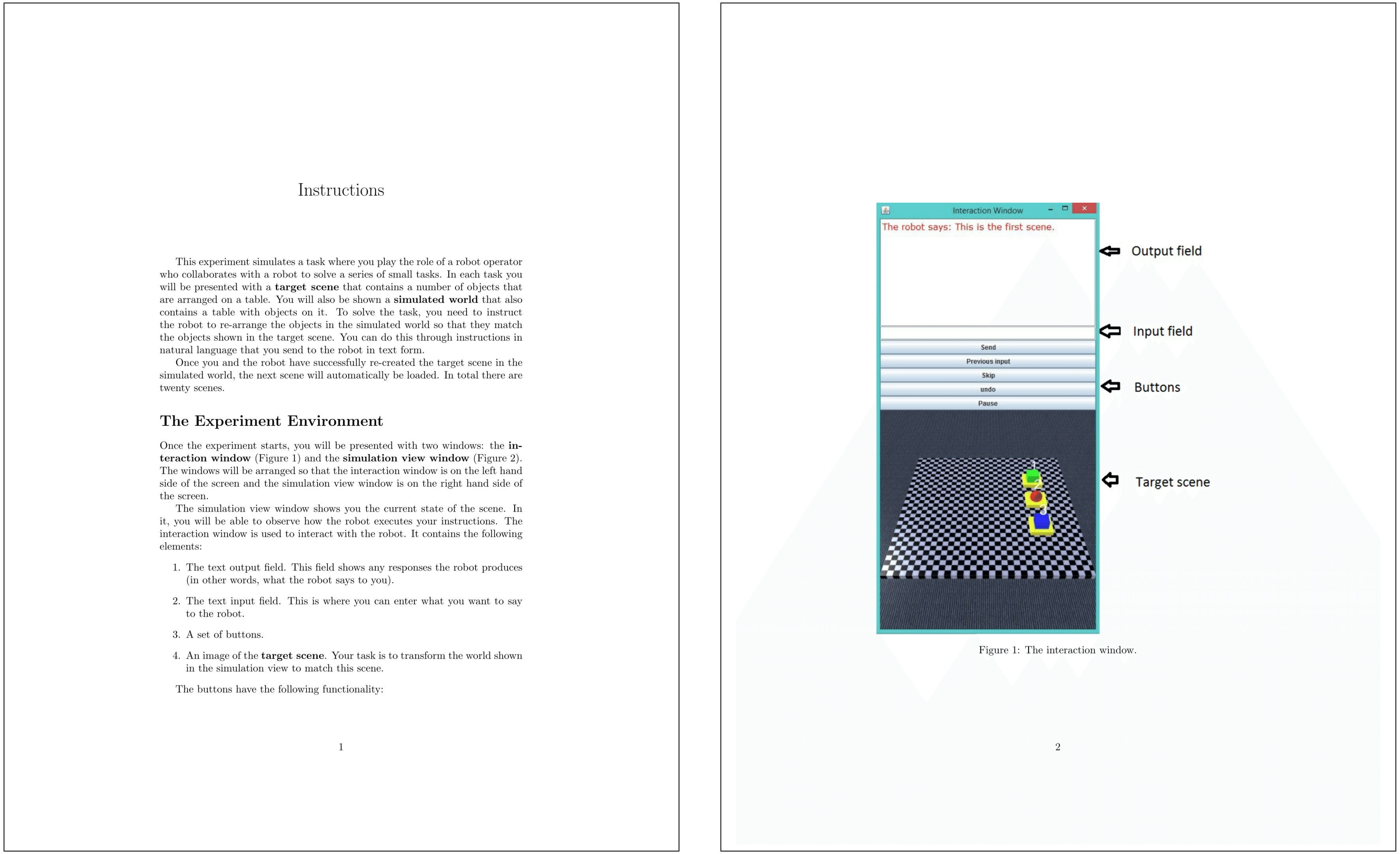
Rewards : All participants will be offered tea or coffee and a piece of chocolate or fruit.

Contact : Please contact Niels Schütte (niels.schuette@gmail.com) to arrange a time.

2

(b) The back side of the information sheet.

Figure B.1: The information sheet.



Instructions

This experiment simulates a task where you play the role of a robot operator who collaborates with a robot to solve a series of small tasks. In each task you will be presented with a **target scene** that contains a number of objects that are arranged on a table. You will also be shown a **simulated world** that also contains a table with objects on it. To solve the task, you need to instruct the robot to re-arrange the objects in the simulated world so that they match the objects shown in the target scene. You can do this through instructions in natural language that you send to the robot in text form.

Once you and the robot have successfully re-created the target scene in the simulated world, the next scene will automatically be loaded. In total there are twenty scenes.

The Experiment Environment

Once the experiment starts, you will be presented with two windows: the **interaction window** (Figure 1) and the **simulation view window** (Figure 2). The windows will be arranged so that the interaction window is on the left hand side of the screen and the simulation view window is on the right hand side of the screen.

The simulation view window shows you the current state of the scene. In it, you will be able to observe how the robot executes your instructions. The interaction window is used to interact with the robot. It contains the following elements:

1. The text output field. This field shows any responses the robot produces (in other words, what the robot says to you).
2. The text input field. This is where you can enter what you want to say to the robot.
3. A set of buttons.
4. An image of the **target scene**. Your task is to transform the world shown in the simulation view to match this scene.

The buttons have the following functionality:

1

(a) Page 1.

The robot says: This is the first scene.

Figure 1: The interaction window.

2

(b) Page 2.

Figure B.2: The instructions for the No Error Phase.

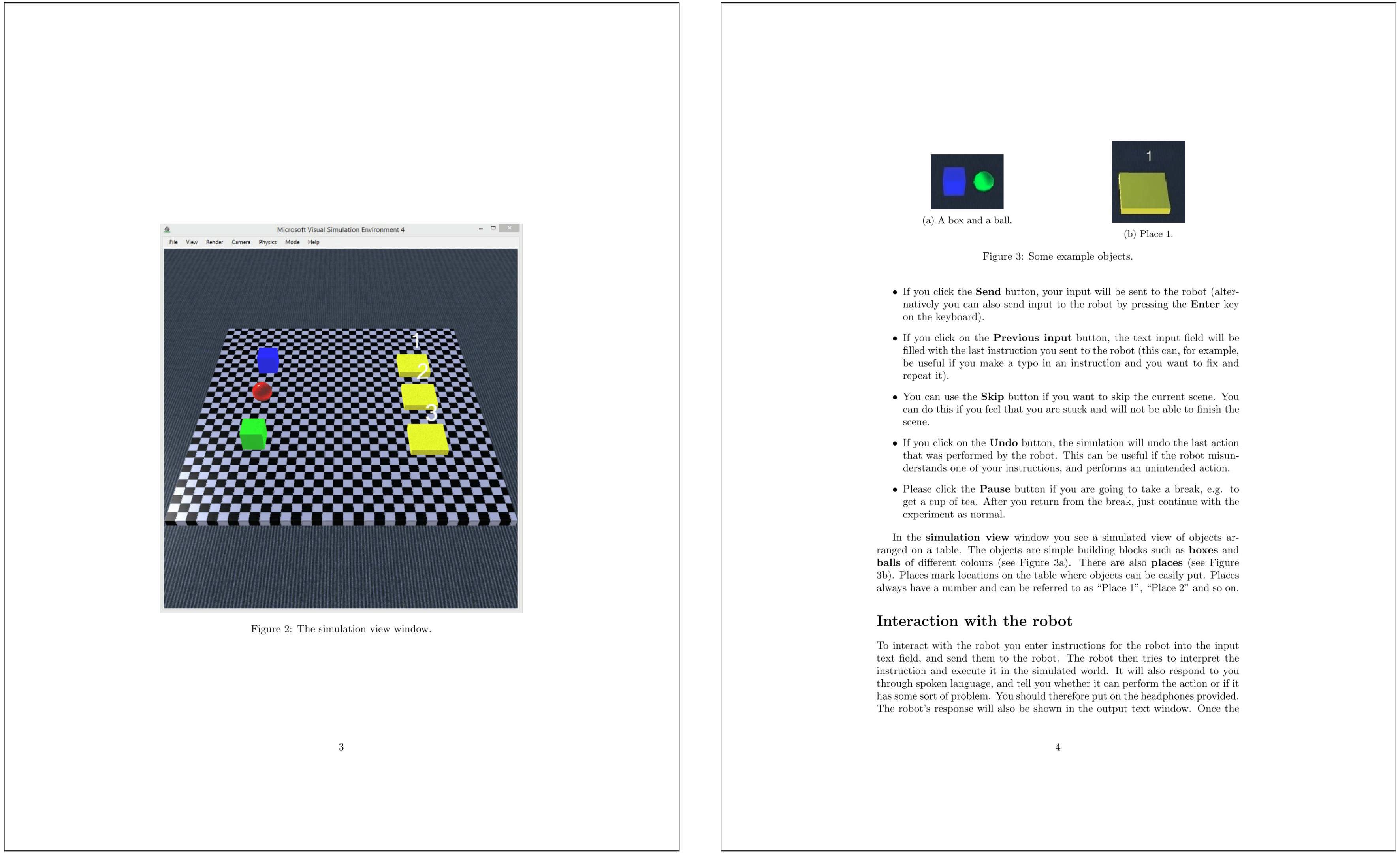


Figure 2: The simulation view window.

(a) Page 3.

(a) A box and a ball.

(b) Place 1.

Figure 3: Some example objects.

- If you click the **Send** button, your input will be sent to the robot (alternatively you can also send input to the robot by pressing the **Enter** key on the keyboard).
- If you click on the **Previous input** button, the text input field will be filled with the last instruction you sent to the robot (this can, for example, be useful if you make a typo in an instruction and you want to fix and repeat it).
- You can use the **Skip** button if you want to skip the current scene. You can do this if you feel that you are stuck and will not be able to finish the scene.
- If you click on the **Undo** button, the simulation will undo the last action that was performed by the robot. This can be useful if the robot misunderstands one of your instructions, and performs an unintended action.
- Please click the **Pause** button if you are going to take a break, e.g. to get a cup of tea. After you return from the break, just continue with the experiment as normal.

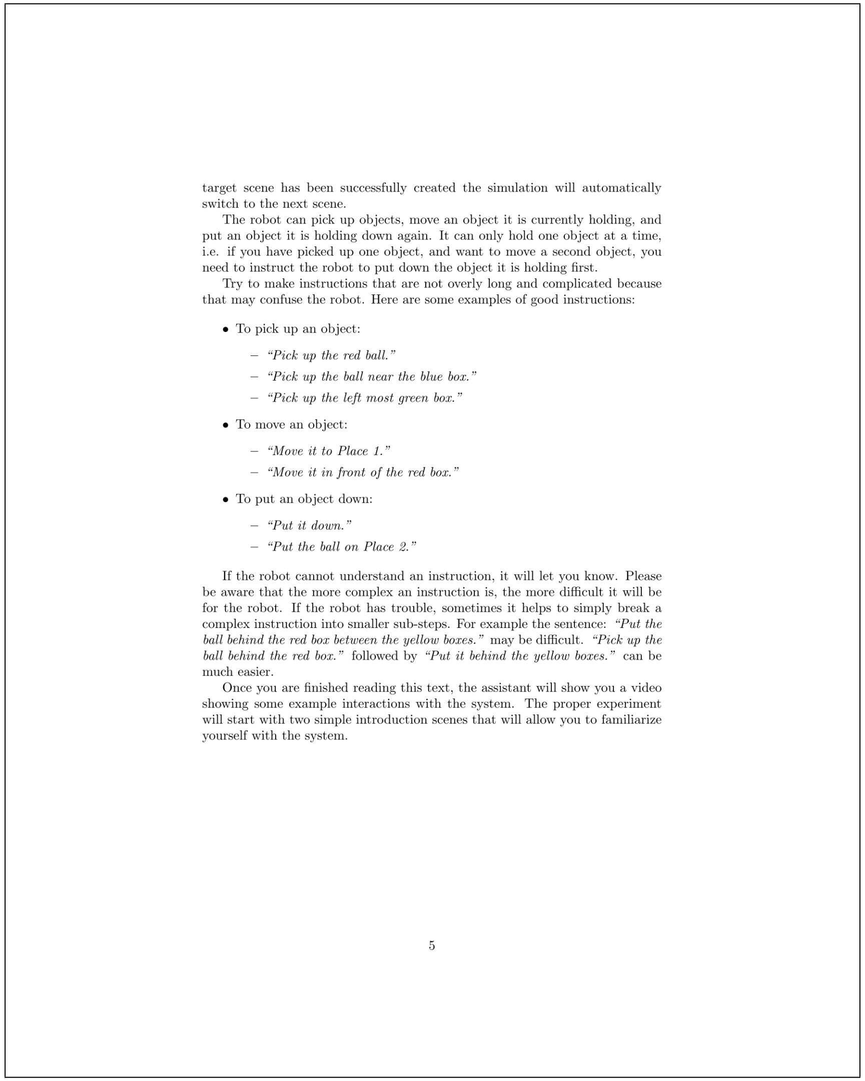
In the **simulation view** window you see a simulated view of objects arranged on a table. The objects are simple building blocks such as **boxes** and **balls** of different colours (see Figure 3a). There are also **places** (see Figure 3b). Places mark locations on the table where objects can be easily put. Places always have a number and can be referred to as "Place 1", "Place 2" and so on.

Interaction with the robot

To interact with the robot you enter instructions for the robot into the input text field, and send them to the robot. The robot then tries to interpret the instruction and execute it in the simulated world. It will also respond to you through spoken language, and tell you whether it can perform the action or if it has some sort of problem. You should therefore put on the headphones provided. The robot's response will also be shown in the output text window. Once the

(b) Page 4.

Figure B.3: The instructions for the No Error Phase (cont.).



target scene has been successfully created the simulation will automatically switch to the next scene.

The robot can pick up objects, move an object it is currently holding, and put an object it is holding down again. It can only hold one object at a time, i.e. if you have picked up one object, and want to move a second object, you need to instruct the robot to put down the object it is holding first.

Try to make instructions that are not overly long and complicated because that may confuse the robot. Here are some examples of good instructions:

- To pick up an object:
  - *"Pick up the red ball."*
  - *"Pick up the ball near the blue box."*
  - *"Pick up the left most green box."*
- To move an object:
  - *"Move it to Place 1."*
  - *"Move it in front of the red box."*
- To put an object down:
  - *"Put it down."*
  - *"Put the ball on Place 2."*

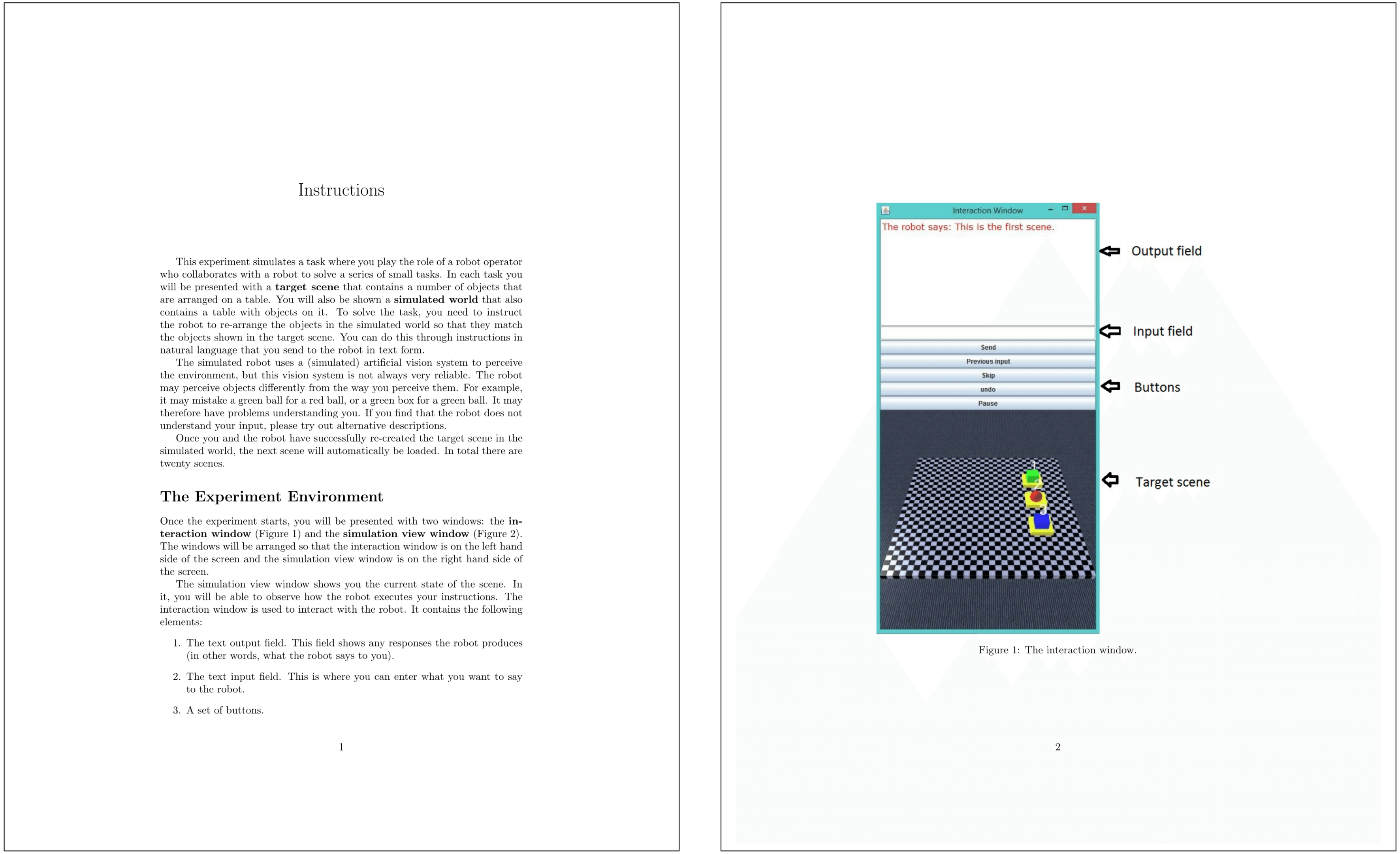
If the robot cannot understand an instruction, it will let you know. Please be aware that the more complex an instruction is, the more difficult it will be for the robot. If the robot has trouble, sometimes it helps to simply break a complex instruction into smaller sub-steps. For example the sentence: *"Put the ball behind the red box between the yellow boxes."* may be difficult. *"Pick up the ball behind the red box."* followed by *"Put it behind the yellow boxes."* can be much easier.

Once you are finished reading this text, the assistant will show you a video showing some example interactions with the system. The proper experiment will start with two simple introduction scenes that will allow you to familiarize yourself with the system.

5

(a) Page 5.

Figure B.4: The instructions for the No Error Phase (cont.).



Instructions

This experiment simulates a task where you play the role of a robot operator who collaborates with a robot to solve a series of small tasks. In each task you will be presented with a **target scene** that contains a number of objects that are arranged on a table. You will also be shown a **simulated world** that also contains a table with objects on it. To solve the task, you need to instruct the robot to re-arrange the objects in the simulated world so that they match the objects shown in the target scene. You can do this through instructions in natural language that you send to the robot in text form.

The simulated robot uses a (simulated) artificial vision system to perceive the environment, but this vision system is not always very reliable. The robot may perceive objects differently from the way you perceive them. For example, it may mistake a green ball for a red ball, or a green box for a green ball. It may therefore have problems understanding you. If you find that the robot does not understand your input, please try out alternative descriptions.

Once you and the robot have successfully re-created the target scene in the simulated world, the next scene will automatically be loaded. In total there are twenty scenes.

The Experiment Environment

Once the experiment starts, you will be presented with two windows: the **interaction window** (Figure 1) and the **simulation view window** (Figure 2). The windows will be arranged so that the interaction window is on the left hand side of the screen and the simulation view window is on the right hand side of the screen.

The simulation view window shows you the current state of the scene. In it, you will be able to observe how the robot executes your instructions. The interaction window is used to interact with the robot. It contains the following elements:

1. The text output field. This field shows any responses the robot produces (in other words, what the robot says to you).
2. The text input field. This is where you can enter what you want to say to the robot.
3. A set of buttons.

1

Interaction Window

The robot says: This is the first scene.

Figure 1: The interaction window.

2

(a) Page 1.

(b) Page 2.

Figure B.5: The instructions for the Error Phase.

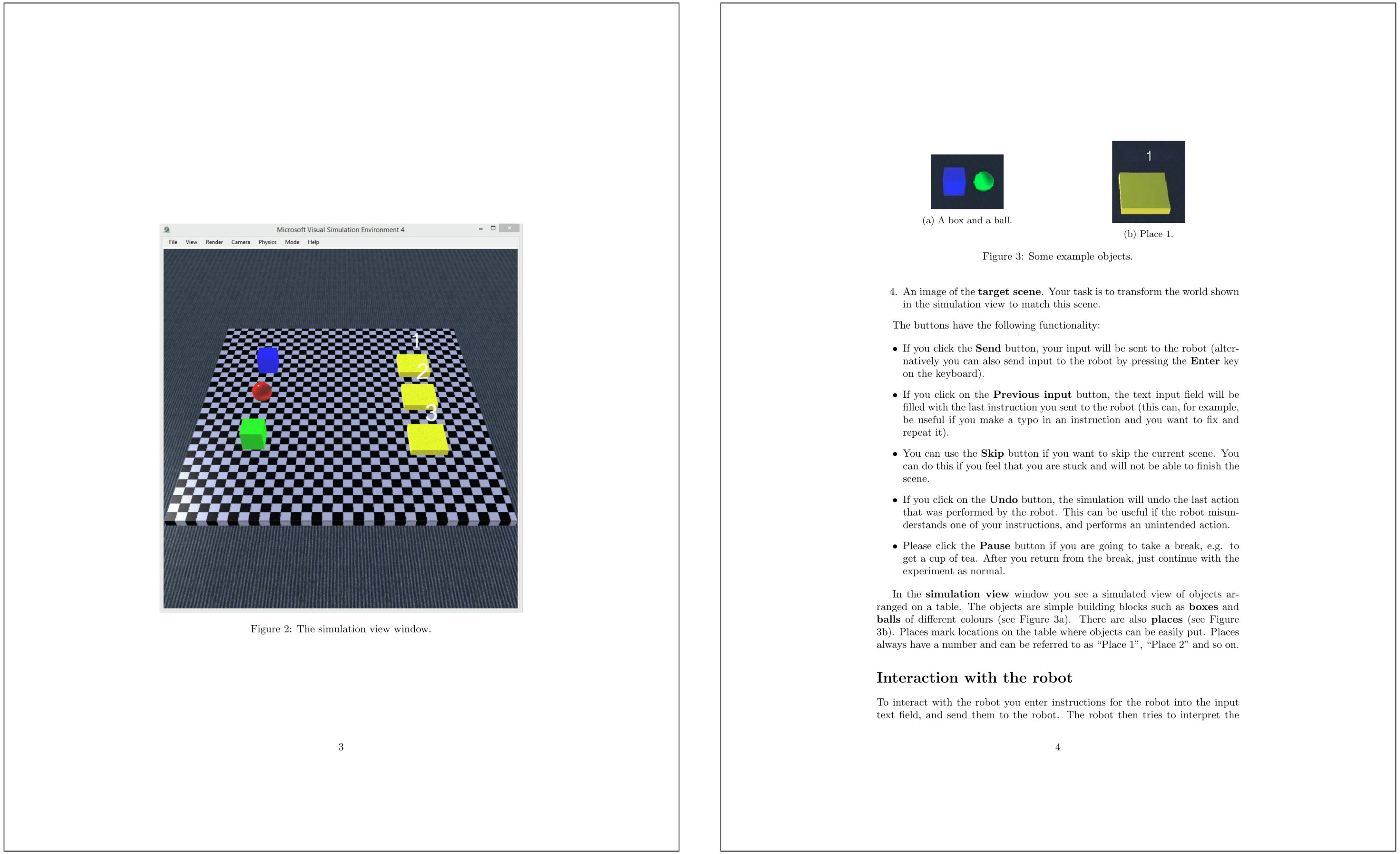


Figure 2: The simulation view window.

3

(a) A box and a ball.

(b) Place 1.

Figure 3: Some example objects.

4. An image of the **target scene**. Your task is to transform the world shown in the simulation view to match this scene.

The buttons have the following functionality:

- If you click the **Send** button, your input will be sent to the robot (alternatively you can also send input to the robot by pressing the **Enter** key on the keyboard).
- If you click on the **Previous input** button, the text input field will be filled with the last instruction you sent to the robot (this can, for example, be useful if you make a typo in an instruction and you want to fix and repeat it).
- You can use the **Skip** button if you want to skip the current scene. You can do this if you feel that you are stuck and will not be able to finish the scene.
- If you click on the **Undo** button, the simulation will undo the last action that was performed by the robot. This can be useful if the robot misunderstands one of your instructions, and performs an unintended action.
- Please click the **Pause** button if you are going to take a break, e.g. to get a cup of tea. After you return from the break, just continue with the experiment as normal.

In the **simulation view** window you see a simulated view of objects arranged on a table. The objects are simple building blocks such as **boxes** and **balls** of different colours (see Figure 3a). There are also **places** (see Figure 3b). Places mark locations on the table where objects can be easily put. Places always have a number and can be referred to as “Place 1”, “Place 2” and so on.

Interaction with the robot

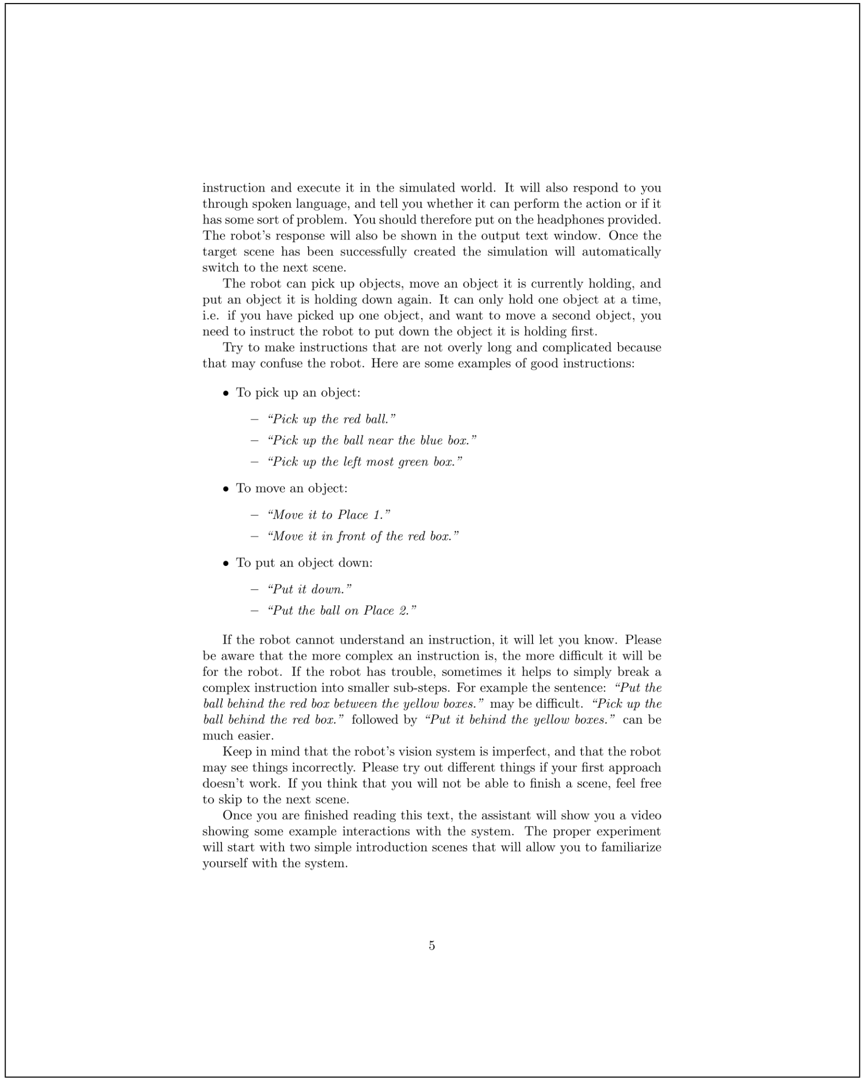
To interact with the robot you enter instructions for the robot into the input text field, and send them to the robot. The robot then tries to interpret the

4

(a) Page 3.

(b) Page 4.

Figure B.6: The instructions for the Error Phase (cont.).



instruction and execute it in the simulated world. It will also respond to you through spoken language, and tell you whether it can perform the action or if it has some sort of problem. You should therefore put on the headphones provided. The robot's response will also be shown in the output text window. Once the target scene has been successfully created the simulation will automatically switch to the next scene.

The robot can pick up objects, move an object it is currently holding, and put an object it is holding down again. It can only hold one object at a time, i.e. if you have picked up one object, and want to move a second object, you need to instruct the robot to put down the object it is holding first.

Try to make instructions that are not overly long and complicated because that may confuse the robot. Here are some examples of good instructions:

- To pick up an object:
  - *"Pick up the red ball."*
  - *"Pick up the ball near the blue box."*
  - *"Pick up the left most green box."*
- To move an object:
  - *"Move it to Place 1."*
  - *"Move it in front of the red box."*
- To put an object down:
  - *"Put it down."*
  - *"Put the ball on Place 2."*

If the robot cannot understand an instruction, it will let you know. Please be aware that the more complex an instruction is, the more difficult it will be for the robot. If the robot has trouble, sometimes it helps to simply break a complex instruction into smaller sub-steps. For example the sentence: *"Put the ball behind the red box between the yellow boxes."* may be difficult. *"Pick up the ball behind the red box."* followed by *"Put it behind the yellow boxes."* can be much easier.

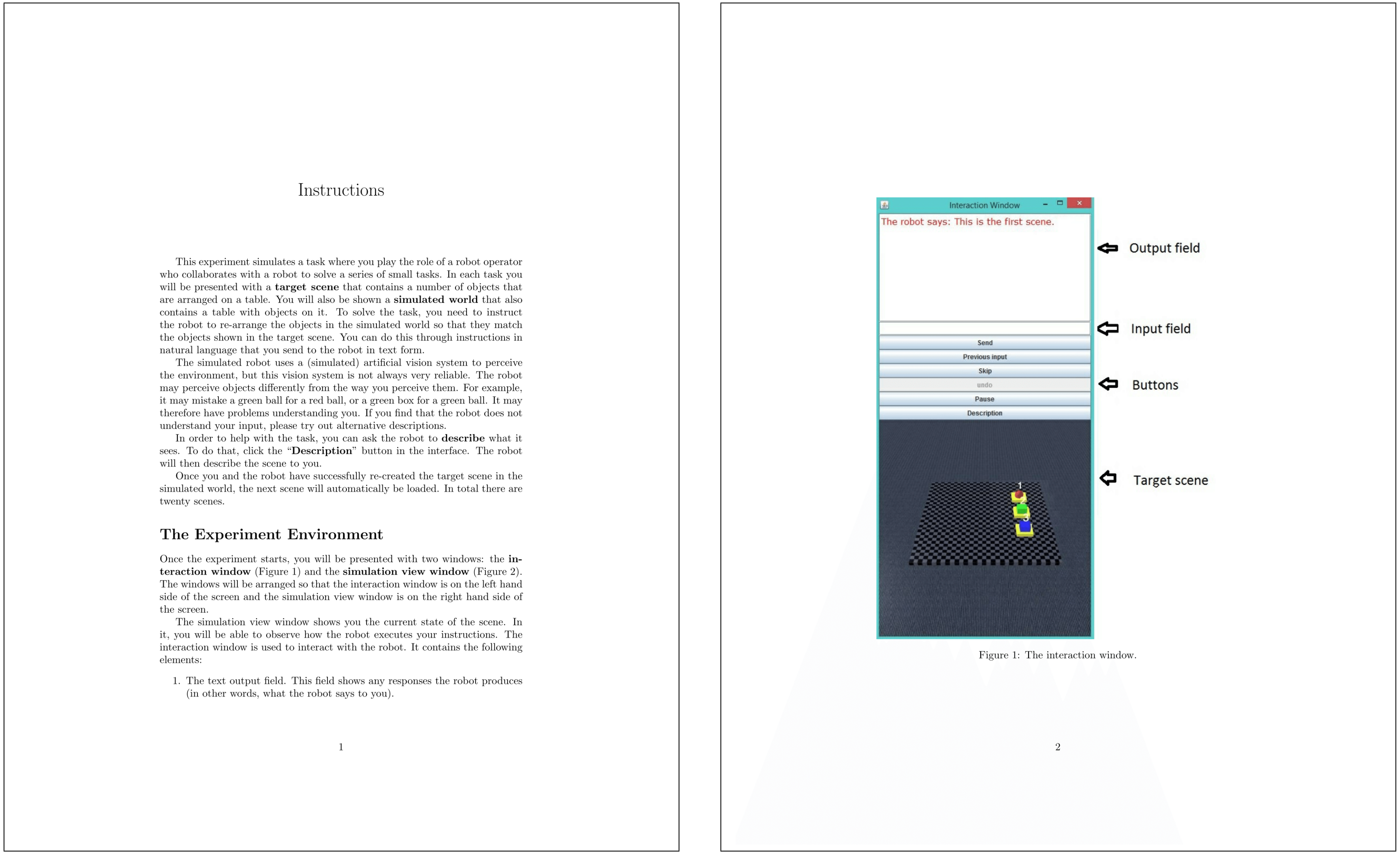
Keep in mind that the robot's vision system is imperfect, and that the robot may see things incorrectly. Please try out different things if your first approach doesn't work. If you think that you will not be able to finish a scene, feel free to skip to the next scene.

Once you are finished reading this text, the assistant will show you a video showing some example interactions with the system. The proper experiment will start with two simple introduction scenes that will allow you to familiarize yourself with the system.

5

(a) Page 5.

Figure B.7: The instructions for the Error Phase (cont.).



Instructions

This experiment simulates a task where you play the role of a robot operator who collaborates with a robot to solve a series of small tasks. In each task you will be presented with a **target scene** that contains a number of objects that are arranged on a table. You will also be shown a **simulated world** that also contains a table with objects on it. To solve the task, you need to instruct the robot to re-arrange the objects in the simulated world so that they match the objects shown in the target scene. You can do this through instructions in natural language that you send to the robot in text form.

The simulated robot uses a (simulated) artificial vision system to perceive the environment, but this vision system is not always very reliable. The robot may perceive objects differently from the way you perceive them. For example, it may mistake a green ball for a red ball, or a green box for a green ball. It may therefore have problems understanding you. If you find that the robot does not understand your input, please try out alternative descriptions.

In order to help with the task, you can ask the robot to **describe** what it sees. To do that, click the "**Description**" button in the interface. The robot will then describe the scene to you.

Once you and the robot have successfully re-created the target scene in the simulated world, the next scene will automatically be loaded. In total there are twenty scenes.

The Experiment Environment

Once the experiment starts, you will be presented with two windows: the **interaction window** (Figure 1) and the **simulation view window** (Figure 2). The windows will be arranged so that the interaction window is on the left hand side of the screen and the simulation view window is on the right hand side of the screen.

The simulation view window shows you the current state of the scene. In it, you will be able to observe how the robot executes your instructions. The interaction window is used to interact with the robot. It contains the following elements:

1. The text output field. This field shows any responses the robot produces (in other words, what the robot says to you).

1

(a) Page 1.

The robot says: This is the first scene.

Figure 1: The interaction window.

2

(b) Page 2.

Figure B.8: The instructions for the Description Phase.

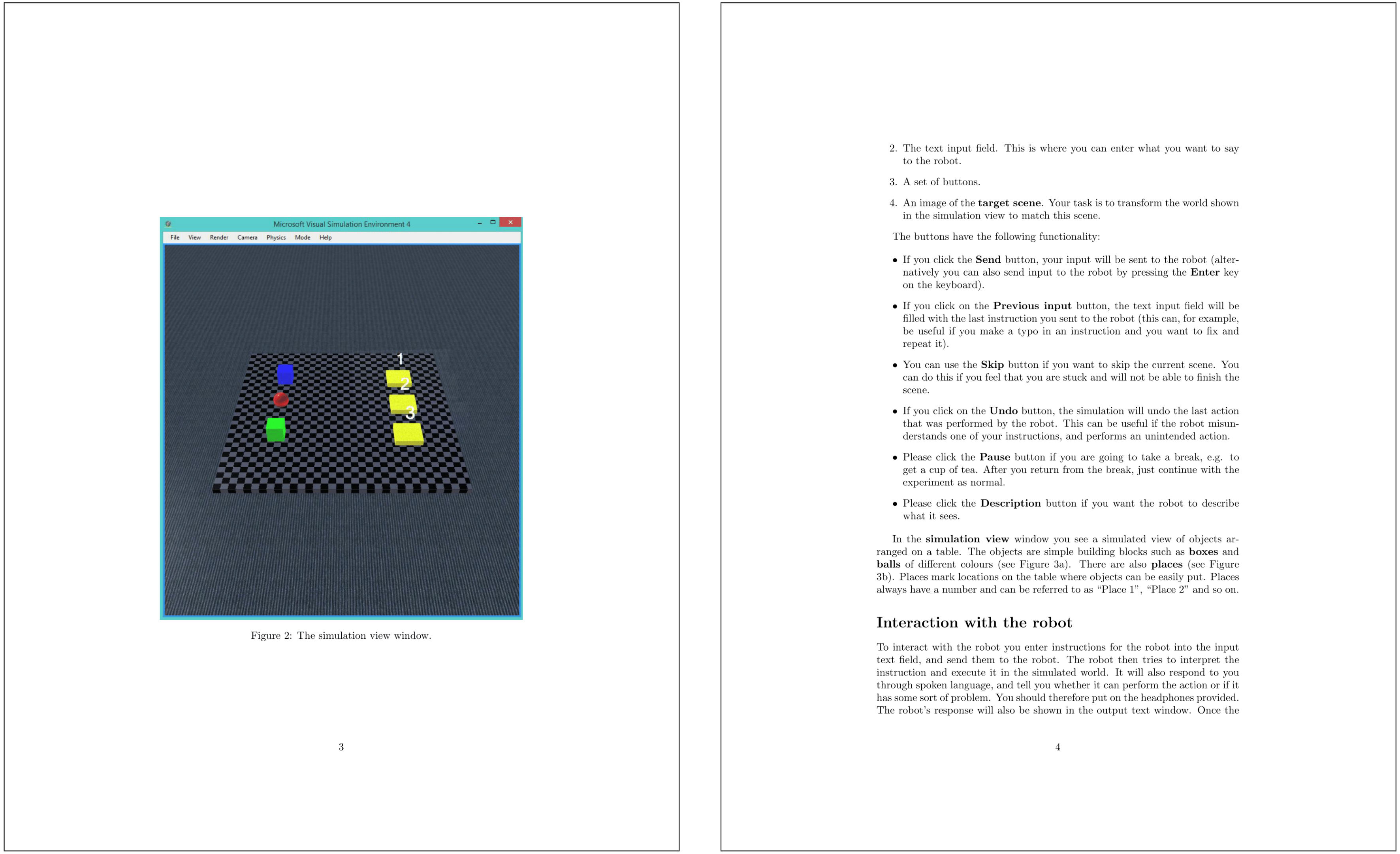


Figure 2: The simulation view window.

3

2. The text input field. This is where you can enter what you want to say to the robot.
3. A set of buttons.
4. An image of the **target scene**. Your task is to transform the world shown in the simulation view to match this scene.

The buttons have the following functionality:

- If you click the **Send** button, your input will be sent to the robot (alternatively you can also send input to the robot by pressing the **Enter** key on the keyboard).
- If you click on the **Previous input** button, the text input field will be filled with the last instruction you sent to the robot (this can, for example, be useful if you make a typo in an instruction and you want to fix and repeat it).
- You can use the **Skip** button if you want to skip the current scene. You can do this if you feel that you are stuck and will not be able to finish the scene.
- If you click on the **Undo** button, the simulation will undo the last action that was performed by the robot. This can be useful if the robot misunderstands one of your instructions, and performs an unintended action.
- Please click the **Pause** button if you are going to take a break, e.g. to get a cup of tea. After you return from the break, just continue with the experiment as normal.
- Please click the **Description** button if you want the robot to describe what it sees.

In the **simulation view** window you see a simulated view of objects arranged on a table. The objects are simple building blocks such as **boxes** and **balls** of different colours (see Figure 3a). There are also **places** (see Figure 3b). Places mark locations on the table where objects can be easily put. Places always have a number and can be referred to as “Place 1”, “Place 2” and so on.

Interaction with the robot

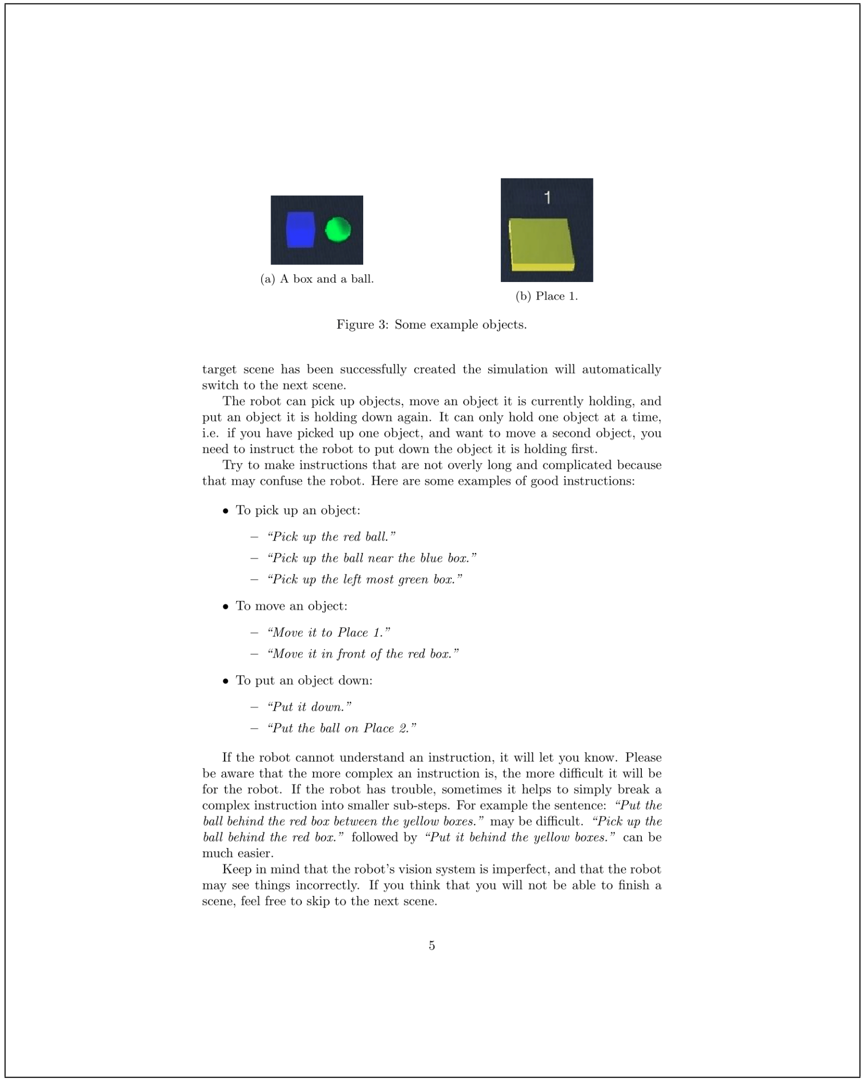
To interact with the robot you enter instructions for the robot into the input text field, and send them to the robot. The robot then tries to interpret the instruction and execute it in the simulated world. It will also respond to you through spoken language, and tell you whether it can perform the action or if it has some sort of problem. You should therefore put on the headphones provided. The robot’s response will also be shown in the output text window. Once the

4

(a) Page 3.

(b) Page 4.

Figure B.9: The instructions for the Description Phase (cont.).



(a) A box and a ball.

(b) Place 1.

Figure 3: Some example objects.

target scene has been successfully created the simulation will automatically switch to the next scene.

The robot can pick up objects, move an object it is currently holding, and put an object it is holding down again. It can only hold one object at a time, i.e. if you have picked up one object, and want to move a second object, you need to instruct the robot to put down the object it is holding first.

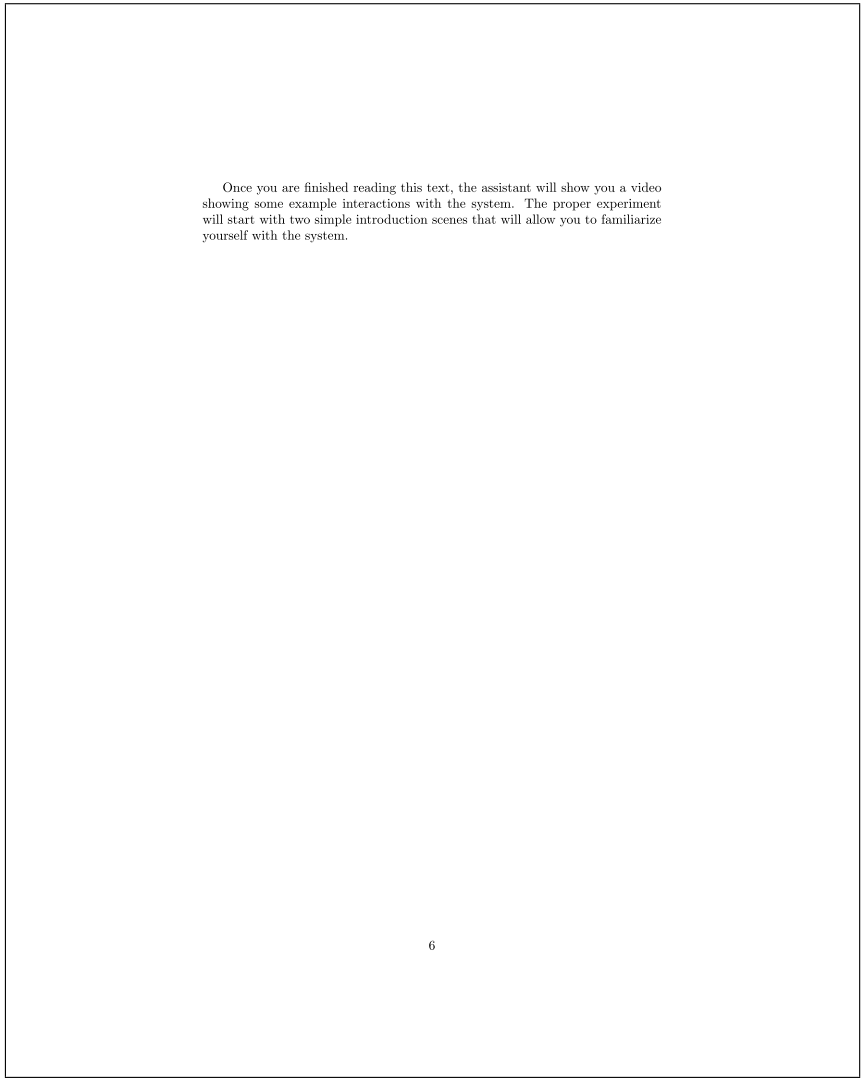
Try to make instructions that are not overly long and complicated because that may confuse the robot. Here are some examples of good instructions:

- To pick up an object:
  - *"Pick up the red ball."*
  - *"Pick up the ball near the blue box."*
  - *"Pick up the left most green box."*
- To move an object:
  - *"Move it to Place 1."*
  - *"Move it in front of the red box."*
- To put an object down:
  - *"Put it down."*
  - *"Put the ball on Place 2."*

If the robot cannot understand an instruction, it will let you know. Please be aware that the more complex an instruction is, the more difficult it will be for the robot. If the robot has trouble, sometimes it helps to simply break a complex instruction into smaller sub-steps. For example the sentence: *"Put the ball behind the red box between the yellow boxes."* may be difficult. *"Pick up the ball behind the red box."* followed by *"Put it behind the yellow boxes."* can be much easier.

Keep in mind that the robot's vision system is imperfect, and that the robot may see things incorrectly. If you think that you will not be able to finish a scene, feel free to skip to the next scene.

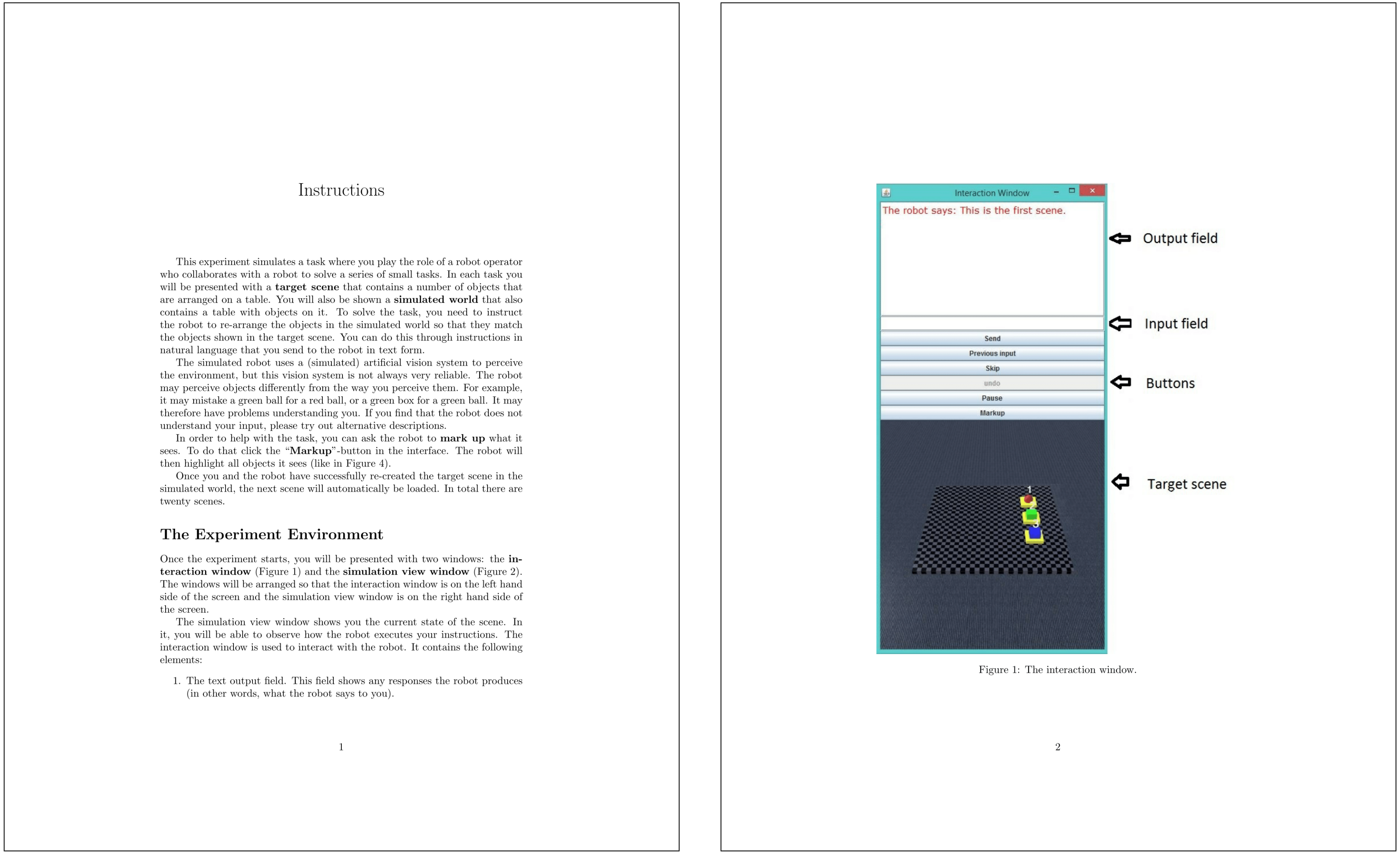
(a) Page 5.



Once you are finished reading this text, the assistant will show you a video showing some example interactions with the system. The proper experiment will start with two simple introduction scenes that will allow you to familiarize yourself with the system.

(b) Page 6.

Figure B.10: The instructions for the Description Phase (cont.).



Instructions

This experiment simulates a task where you play the role of a robot operator who collaborates with a robot to solve a series of small tasks. In each task you will be presented with a **target scene** that contains a number of objects that are arranged on a table. You will also be shown a **simulated world** that also contains a table with objects on it. To solve the task, you need to instruct the robot to re-arrange the objects in the simulated world so that they match the objects shown in the target scene. You can do this through instructions in natural language that you send to the robot in text form.

The simulated robot uses a (simulated) artificial vision system to perceive the environment, but this vision system is not always very reliable. The robot may perceive objects differently from the way you perceive them. For example, it may mistake a green ball for a red ball, or a green box for a green ball. It may therefore have problems understanding you. If you find that the robot does not understand your input, please try out alternative descriptions.

In order to help with the task, you can ask the robot to **mark up** what it sees. To do that click the "**Markup**"-button in the interface. The robot will then highlight all objects it sees (like in Figure 4).

Once you and the robot have successfully re-created the target scene in the simulated world, the next scene will automatically be loaded. In total there are twenty scenes.

The Experiment Environment

Once the experiment starts, you will be presented with two windows: the **interaction window** (Figure 1) and the **simulation view window** (Figure 2). The windows will be arranged so that the interaction window is on the left hand side of the screen and the simulation view window is on the right hand side of the screen.

The simulation view window shows you the current state of the scene. In it, you will be able to observe how the robot executes your instructions. The interaction window is used to interact with the robot. It contains the following elements:

1. The text output field. This field shows any responses the robot produces (in other words, what the robot says to you).

1

The robot says: This is the first scene.

Figure 1: The interaction window.

2

(a) Page 1.

(b) Page 2.

Figure B.11: The instructions for the Markup Phase.

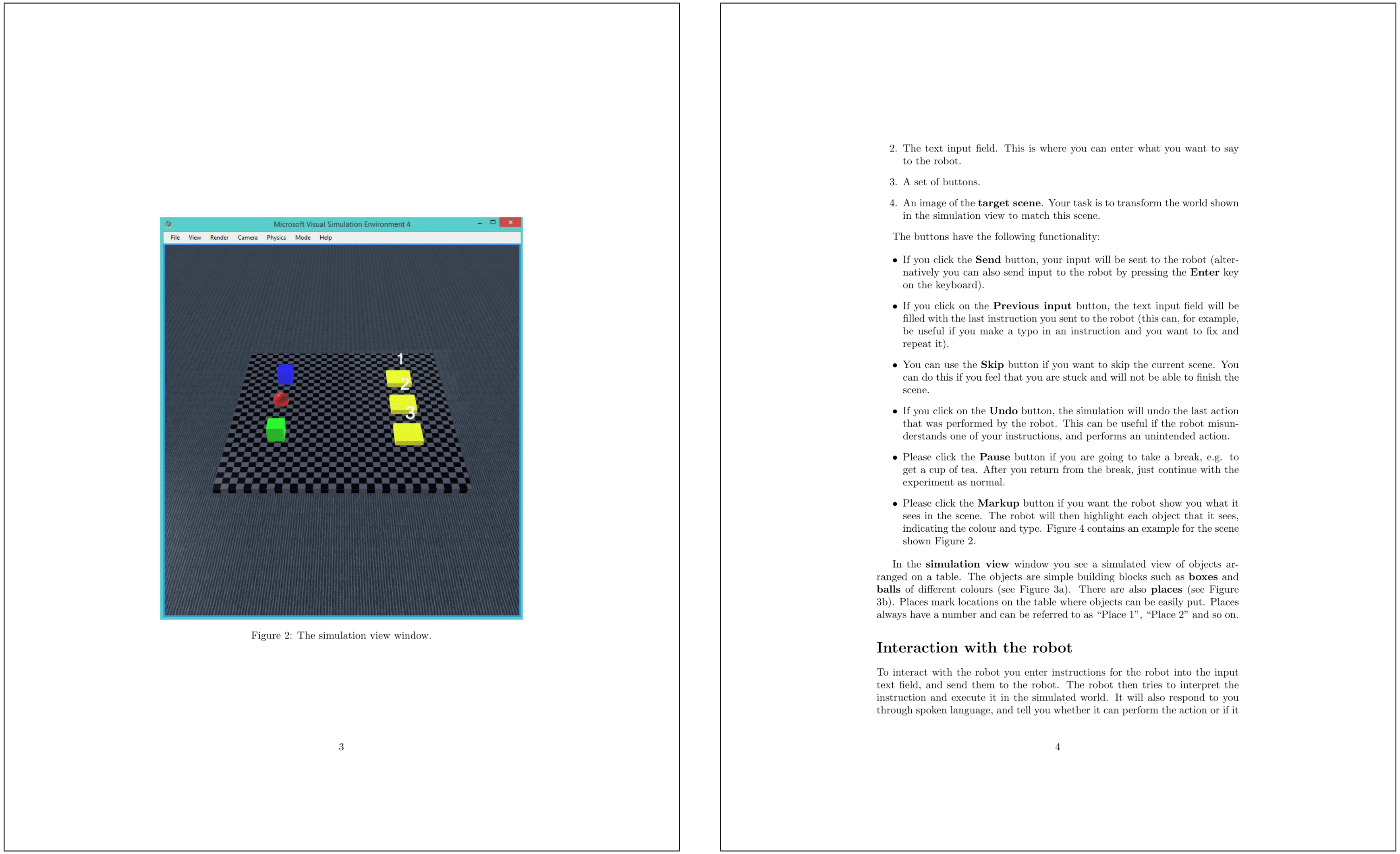

Microsoft Visual Simulation Environment 4
File
View
Render
Camera
Physics
Mode
Help
1
2
3

Figure 2: The simulation view window.

3

2. The text input field. This is where you can enter what you want to say to the robot.
3. A set of buttons.
4. An image of the **target scene**. Your task is to transform the world shown in the simulation view to match this scene.

The buttons have the following functionality:

- If you click the **Send** button, your input will be sent to the robot (alternatively you can also send input to the robot by pressing the **Enter** key on the keyboard).
- If you click on the **Previous input** button, the text input field will be filled with the last instruction you sent to the robot (this can, for example, be useful if you make a typo in an instruction and you want to fix and repeat it).
- You can use the **Skip** button if you want to skip the current scene. You can do this if you feel that you are stuck and will not be able to finish the scene.
- If you click on the **Undo** button, the simulation will undo the last action that was performed by the robot. This can be useful if the robot misunderstands one of your instructions, and performs an unintended action.
- Please click the **Pause** button if you are going to take a break, e.g. to get a cup of tea. After you return from the break, just continue with the experiment as normal.
- Please click the **Markup** button if you want the robot show you what it sees in the scene. The robot will then highlight each object that it sees, indicating the colour and type. Figure 4 contains an example for the scene shown Figure 2.

In the **simulation view** window you see a simulated view of objects arranged on a table. The objects are simple building blocks such as **boxes** and **balls** of different colours (see Figure 3a). There are also **places** (see Figure 3b). Places mark locations on the table where objects can be easily put. Places always have a number and can be referred to as “Place 1”, “Place 2” and so on.

Interaction with the robot

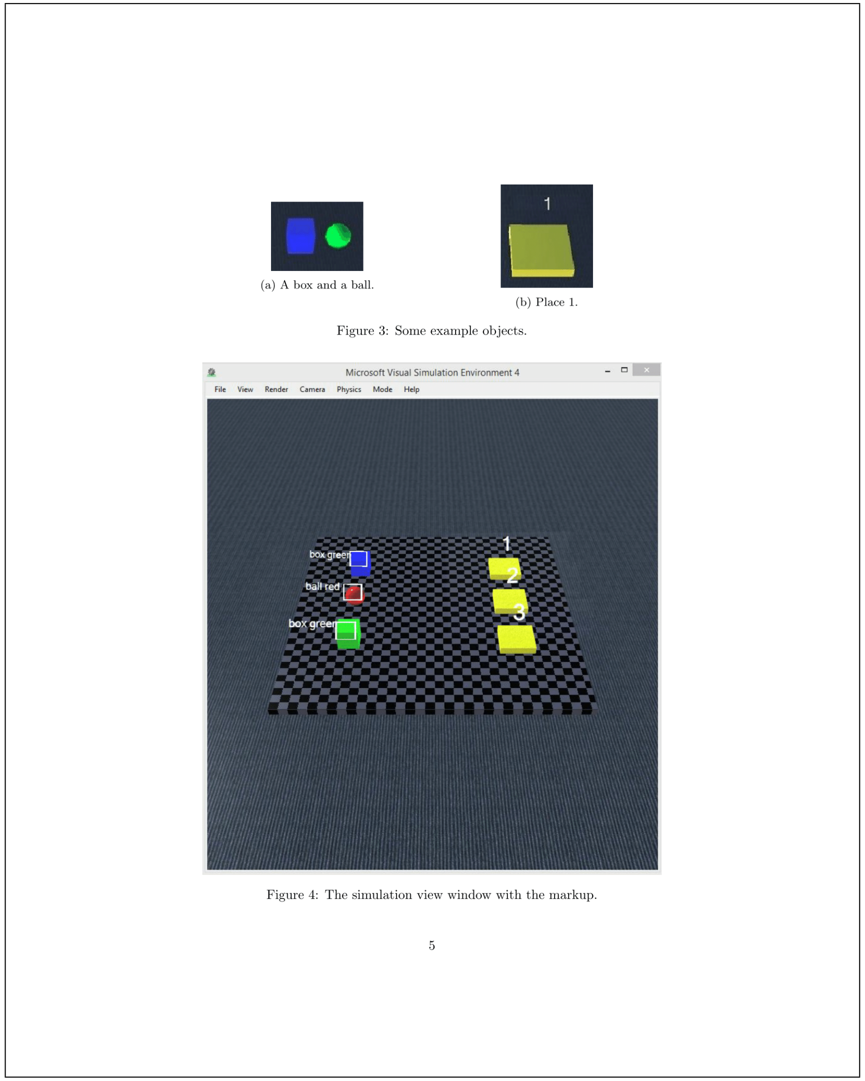
To interact with the robot you enter instructions for the robot into the input text field, and send them to the robot. The robot then tries to interpret the instruction and execute it in the simulated world. It will also respond to you through spoken language, and tell you whether it can perform the action or if it

4

(a) Page 3.

(b) Page 4.

Figure B.12: The instructions for the Markup Phase (cont.).



(a) A box and a ball.

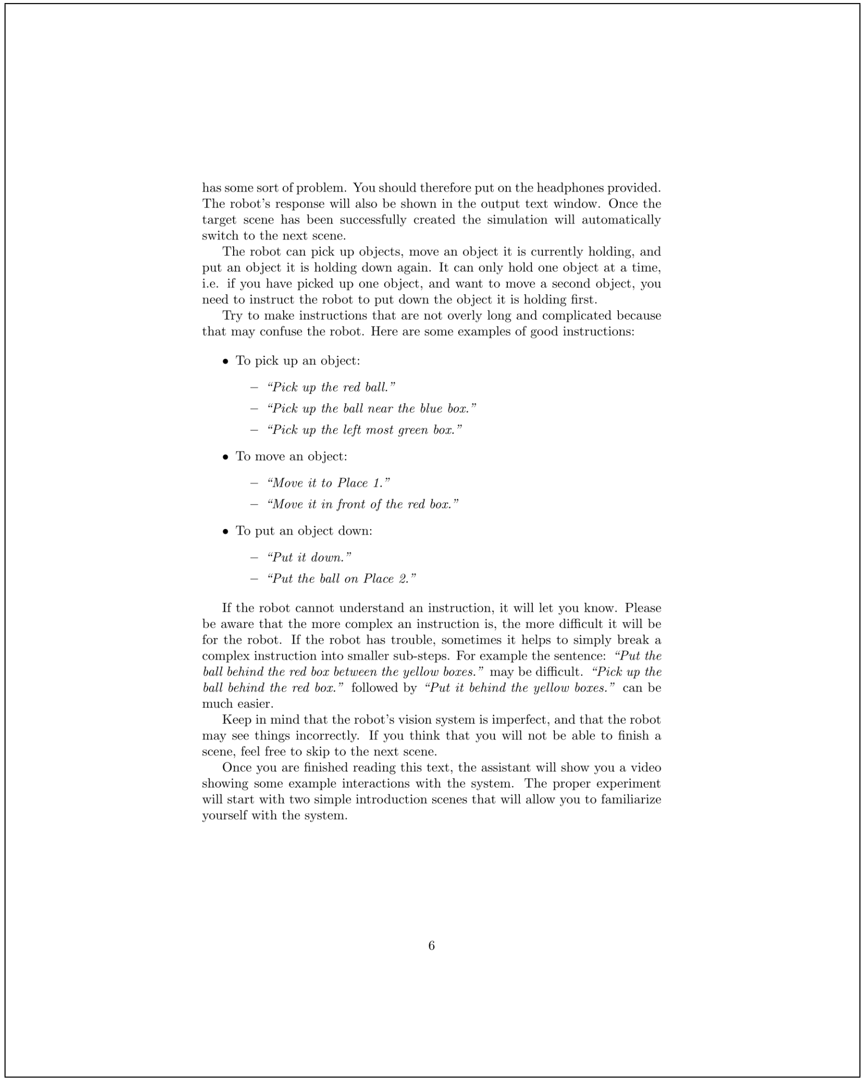
(b) Place 1.

Figure 3: Some example objects.

Figure 4: The simulation view window with the markup.

5

(a) Page 5.



has some sort of problem. You should therefore put on the headphones provided. The robot's response will also be shown in the output text window. Once the target scene has been successfully created the simulation will automatically switch to the next scene.

The robot can pick up objects, move an object it is currently holding, and put an object it is holding down again. It can only hold one object at a time, i.e. if you have picked up one object, and want to move a second object, you need to instruct the robot to put down the object it is holding first.

Try to make instructions that are not overly long and complicated because that may confuse the robot. Here are some examples of good instructions:

- To pick up an object:
  - *"Pick up the red ball."*
  - *"Pick up the ball near the blue box."*
  - *"Pick up the left most green box."*
- To move an object:
  - *"Move it to Place 1."*
  - *"Move it in front of the red box."*
- To put an object down:
  - *"Put it down."*
  - *"Put the ball on Place 2."*

If the robot cannot understand an instruction, it will let you know. Please be aware that the more complex an instruction is, the more difficult it will be for the robot. If the robot has trouble, sometimes it helps to simply break a complex instruction into smaller sub-steps. For example the sentence: *"Put the ball behind the red box between the yellow boxes."* may be difficult. *"Pick up the ball behind the red box."* followed by *"Put it behind the yellow boxes."* can be much easier.

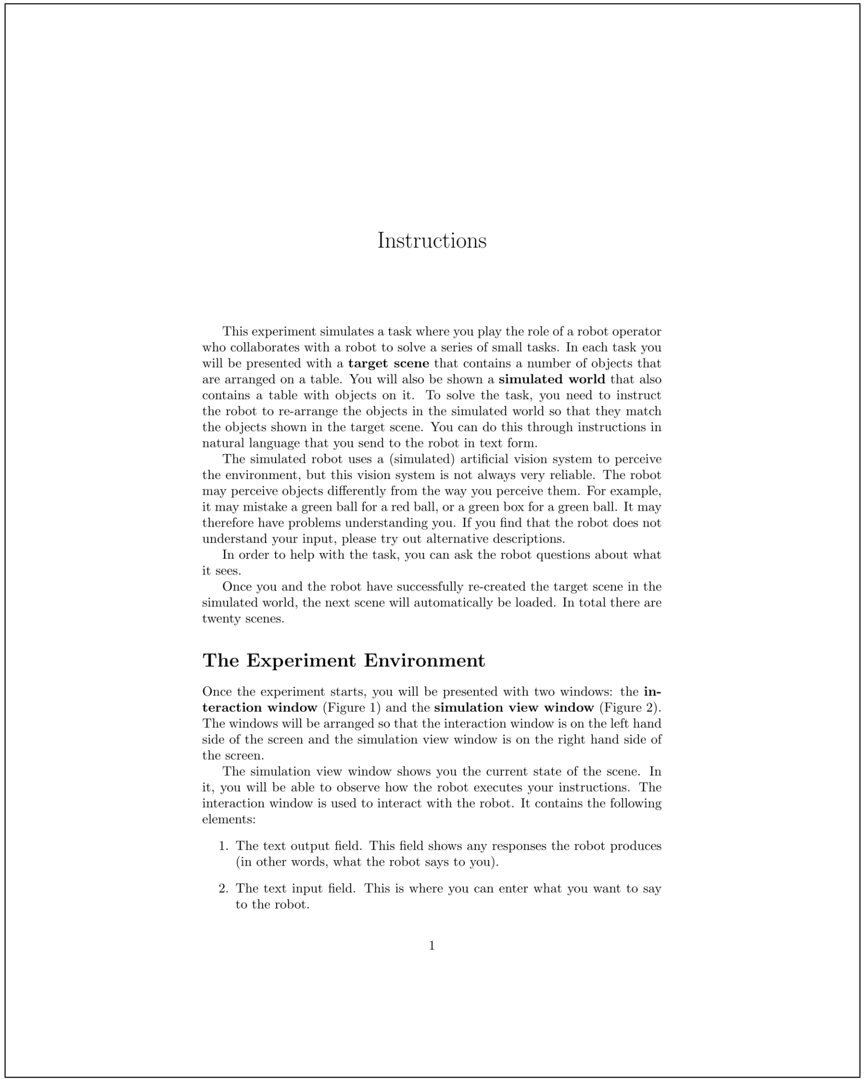
Keep in mind that the robot's vision system is imperfect, and that the robot may see things incorrectly. If you think that you will not be able to finish a scene, feel free to skip to the next scene.

Once you are finished reading this text, the assistant will show you a video showing some example interactions with the system. The proper experiment will start with two simple introduction scenes that will allow you to familiarize yourself with the system.

6

(b) Page 6.

Figure B.13: The instructions for the Markup Phase (cont.).



Instructions

This experiment simulates a task where you play the role of a robot operator who collaborates with a robot to solve a series of small tasks. In each task you will be presented with a **target scene** that contains a number of objects that are arranged on a table. You will also be shown a **simulated world** that also contains a table with objects on it. To solve the task, you need to instruct the robot to re-arrange the objects in the simulated world so that they match the objects shown in the target scene. You can do this through instructions in natural language that you send to the robot in text form.

The simulated robot uses a (simulated) artificial vision system to perceive the environment, but this vision system is not always very reliable. The robot may perceive objects differently from the way you perceive them. For example, it may mistake a green ball for a red ball, or a green box for a green ball. It may therefore have problems understanding you. If you find that the robot does not understand your input, please try out alternative descriptions.

In order to help with the task, you can ask the robot questions about what it sees.

Once you and the robot have successfully re-created the target scene in the simulated world, the next scene will automatically be loaded. In total there are twenty scenes.

The Experiment Environment

Once the experiment starts, you will be presented with two windows: the **interaction window** (Figure 1) and the **simulation view window** (Figure 2). The windows will be arranged so that the interaction window is on the left hand side of the screen and the simulation view window is on the right hand side of the screen.

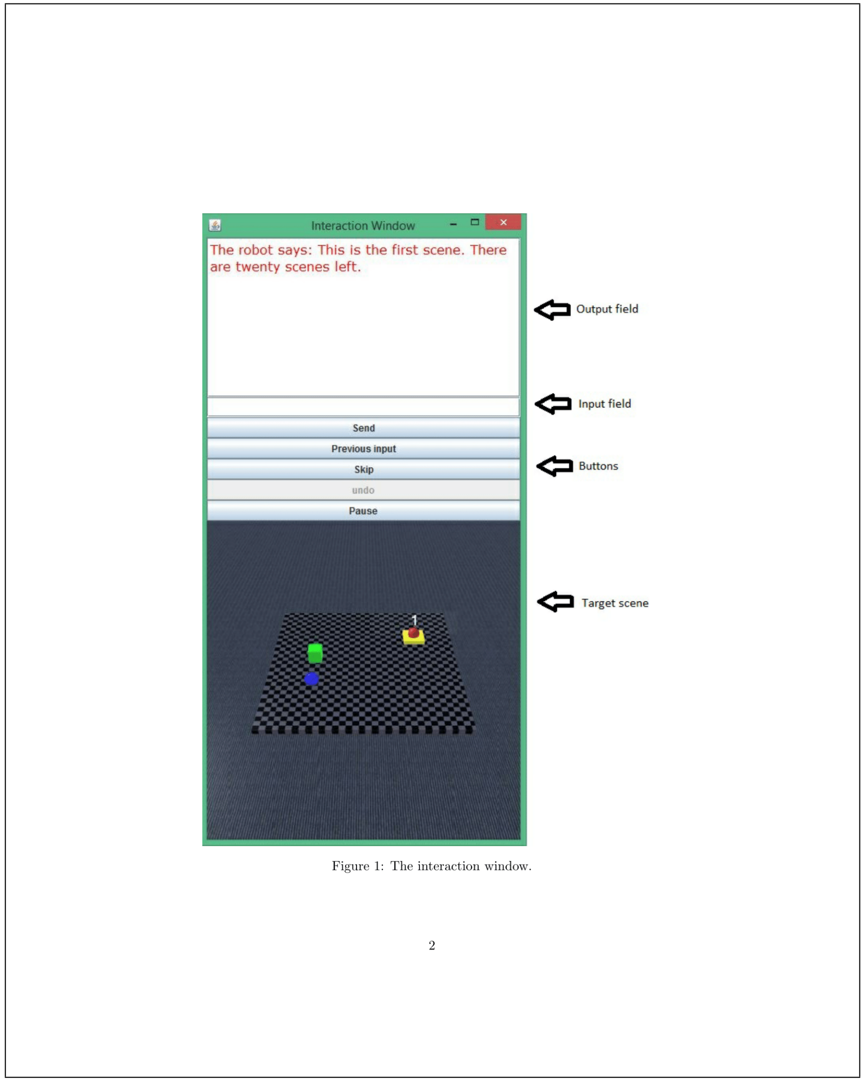
The simulation view window shows you the current state of the scene. In it, you will be able to observe how the robot executes your instructions. The interaction window is used to interact with the robot. It contains the following elements:

1. The text output field. This field shows any responses the robot produces (in other words, what the robot says to you).

2. The text input field. This is where you can enter what you want to say to the robot.

1

(a) Page 1.



The robot says: This is the first scene. There are twenty scenes left.

Figure 1: The interaction window.

2

(b) Page 2.

Figure B.14: The instructions for the Querying Phase.

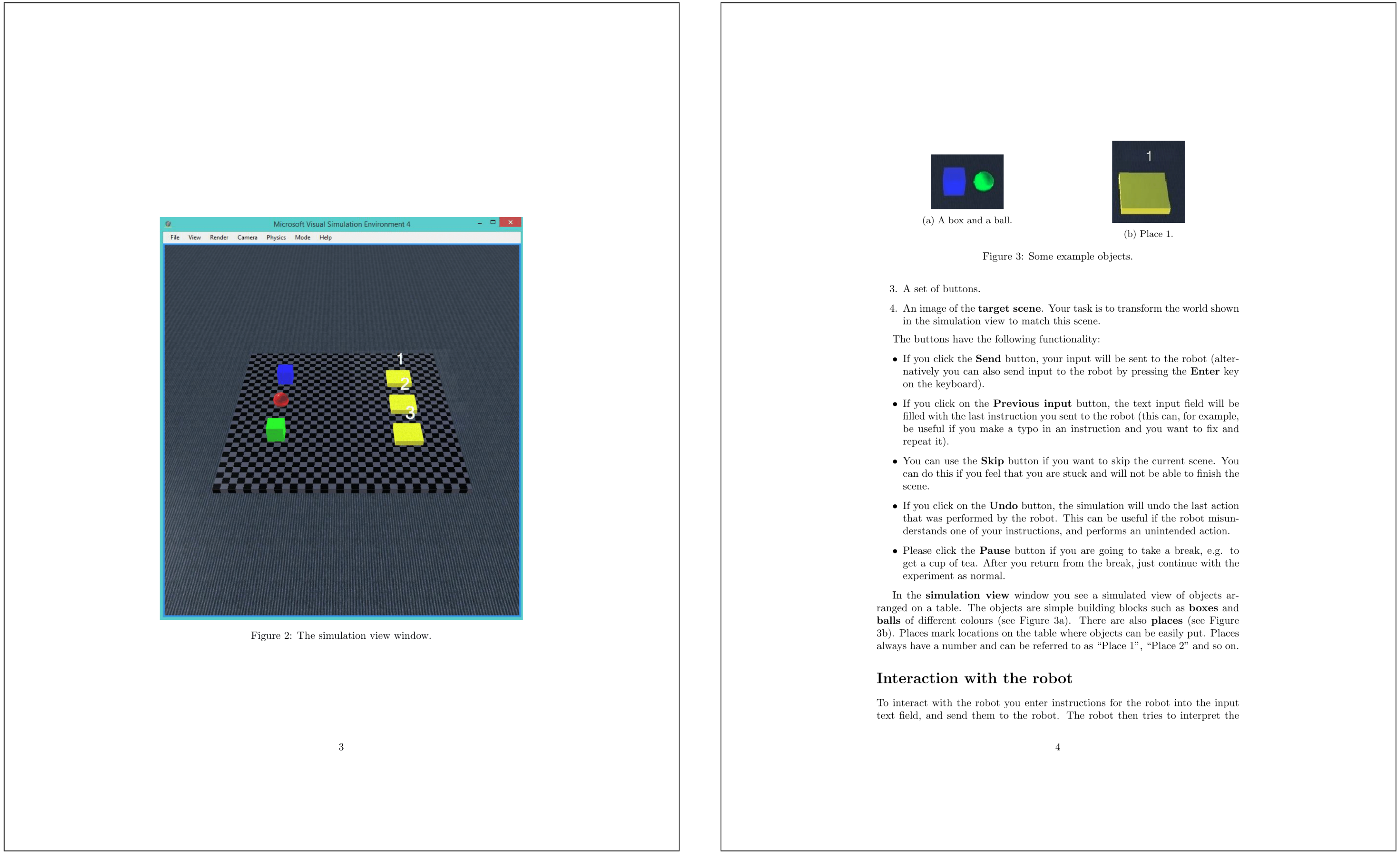


Figure 2: The simulation view window.

3

(a) A box and a ball.

(b) Place 1.

Figure 3: Some example objects.

3. A set of buttons.
4. An image of the **target scene**. Your task is to transform the world shown in the simulation view to match this scene.

The buttons have the following functionality:

- If you click the **Send** button, your input will be sent to the robot (alternatively you can also send input to the robot by pressing the **Enter** key on the keyboard).
- If you click on the **Previous input** button, the text input field will be filled with the last instruction you sent to the robot (this can, for example, be useful if you make a typo in an instruction and you want to fix and repeat it).
- You can use the **Skip** button if you want to skip the current scene. You can do this if you feel that you are stuck and will not be able to finish the scene.
- If you click on the **Undo** button, the simulation will undo the last action that was performed by the robot. This can be useful if the robot misunderstands one of your instructions, and performs an unintended action.
- Please click the **Pause** button if you are going to take a break, e.g. to get a cup of tea. After you return from the break, just continue with the experiment as normal.

In the **simulation view** window you see a simulated view of objects arranged on a table. The objects are simple building blocks such as **boxes** and **balls** of different colours (see Figure 3a). There are also **places** (see Figure 3b). Places mark locations on the table where objects can be easily put. Places always have a number and can be referred to as "Place 1", "Place 2" and so on.

Interaction with the robot

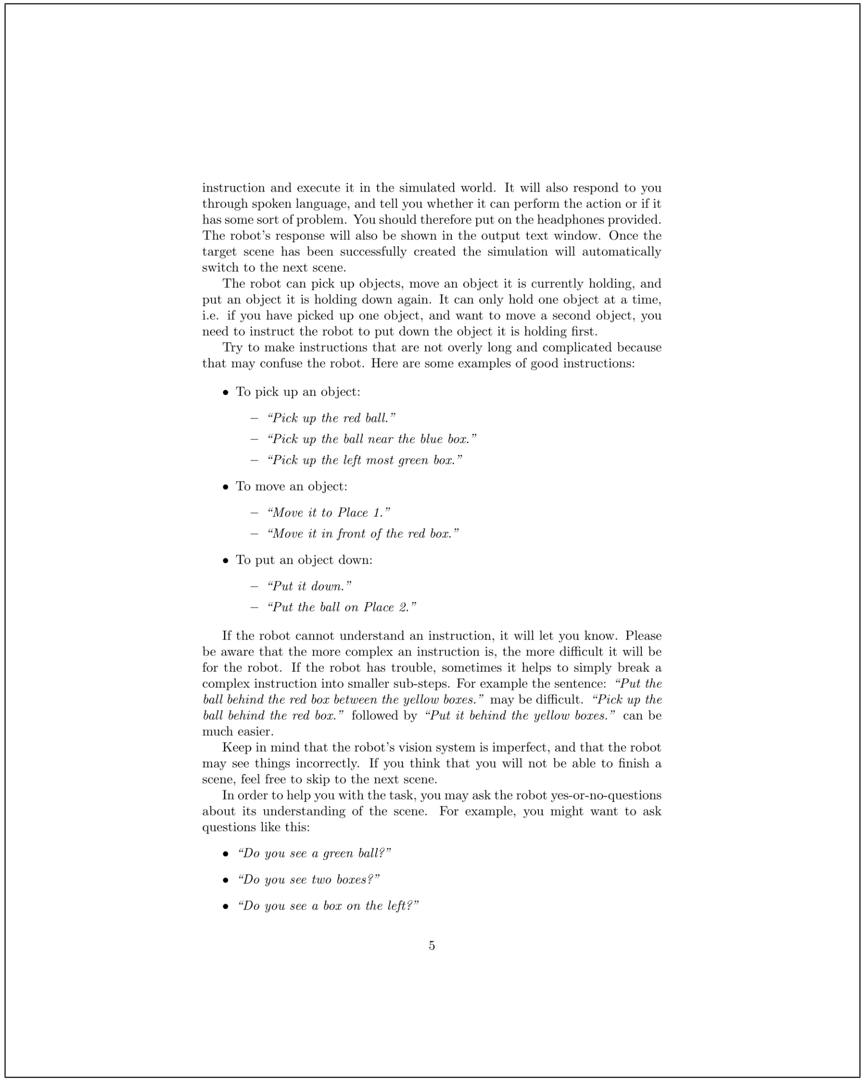
To interact with the robot you enter instructions for the robot into the input text field, and send them to the robot. The robot then tries to interpret the

4

(a) Page 3.

(b) Page 4.

Figure B.15: The instructions for the Querying Phase (cont.).



instruction and execute it in the simulated world. It will also respond to you through spoken language, and tell you whether it can perform the action or if it has some sort of problem. You should therefore put on the headphones provided. The robot's response will also be shown in the output text window. Once the target scene has been successfully created the simulation will automatically switch to the next scene.

The robot can pick up objects, move an object it is currently holding, and put an object it is holding down again. It can only hold one object at a time, i.e. if you have picked up one object, and want to move a second object, you need to instruct the robot to put down the object it is holding first.

Try to make instructions that are not overly long and complicated because that may confuse the robot. Here are some examples of good instructions:

- To pick up an object:
  - *"Pick up the red ball."*
  - *"Pick up the ball near the blue box."*
  - *"Pick up the left most green box."*
- To move an object:
  - *"Move it to Place 1."*
  - *"Move it in front of the red box."*
- To put an object down:
  - *"Put it down."*
  - *"Put the ball on Place 2."*

If the robot cannot understand an instruction, it will let you know. Please be aware that the more complex an instruction is, the more difficult it will be for the robot. If the robot has trouble, sometimes it helps to simply break a complex instruction into smaller sub-steps. For example the sentence: *"Put the ball behind the red box between the yellow boxes."* may be difficult. *"Pick up the ball behind the red box."* followed by *"Put it behind the yellow boxes."* can be much easier.

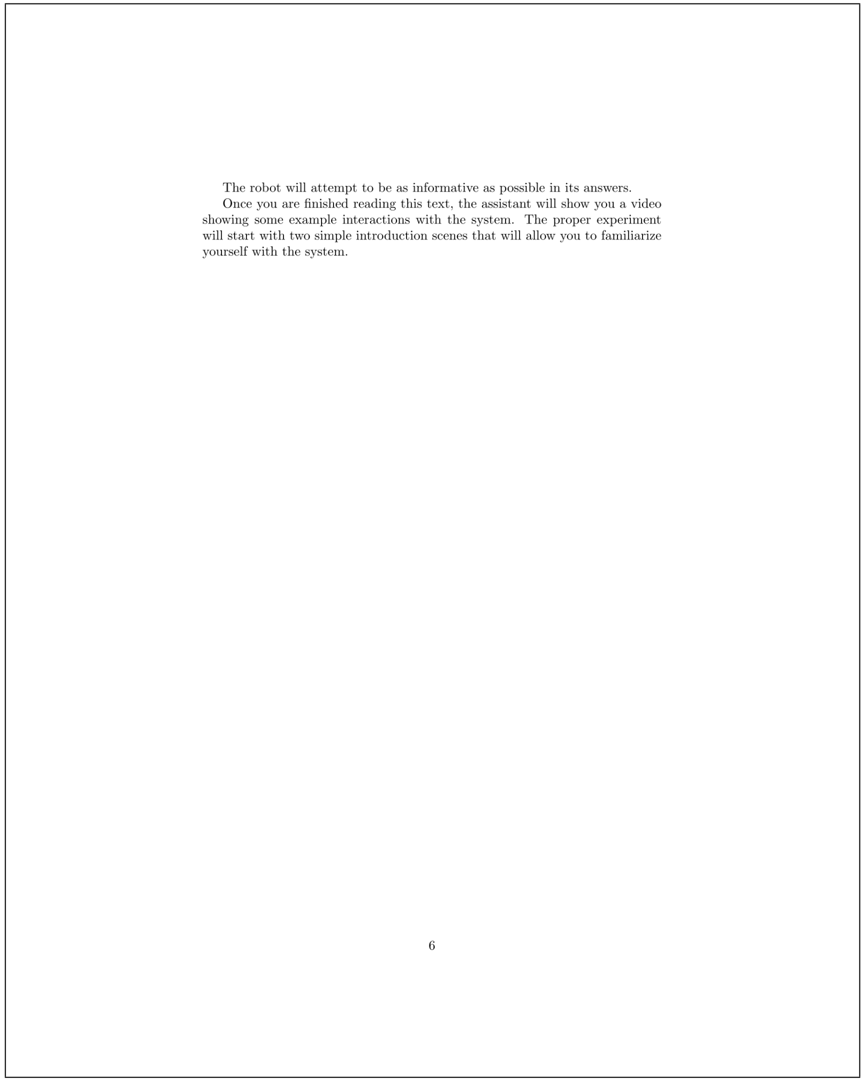
Keep in mind that the robot's vision system is imperfect, and that the robot may see things incorrectly. If you think that you will not be able to finish a scene, feel free to skip to the next scene.

In order to help you with the task, you may ask the robot yes-or-no-questions about its understanding of the scene. For example, you might want to ask questions like this:

- *"Do you see a green ball?"*
- *"Do you see two boxes?"*
- *"Do you see a box on the left?"*

5

(a) Page 5.



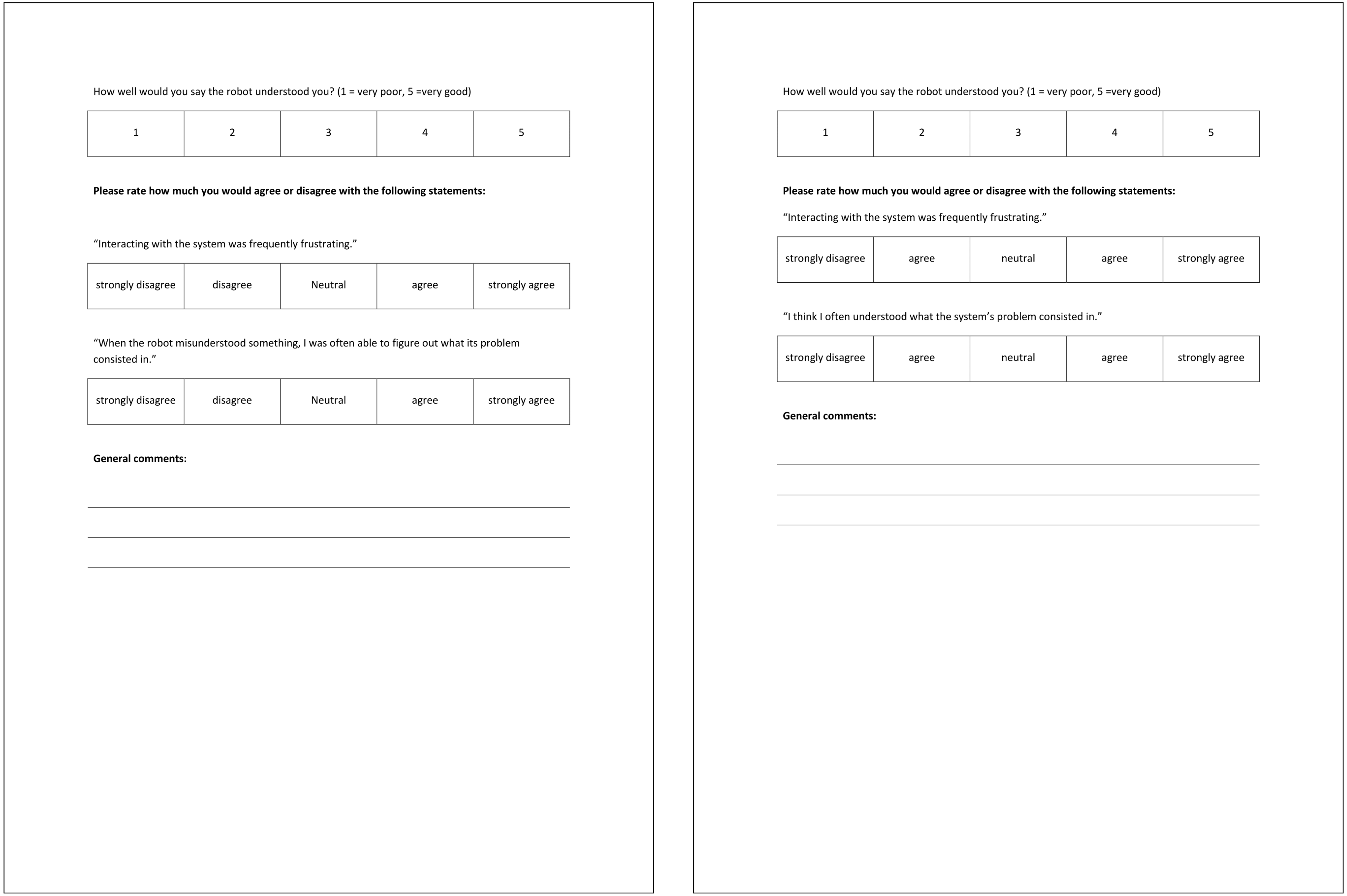
The robot will attempt to be as informative as possible in its answers.

Once you are finished reading this text, the assistant will show you a video showing some example interactions with the system. The proper experiment will start with two simple introduction scenes that will allow you to familiarize yourself with the system.

6

(b) Page 6.

Figure B.16: The instructions for the Querying Phase (cont.).



How well would you say the robot understood you? (1 = very poor, 5 =very good)

1	2	3	4	5

Please rate how much you would agree or disagree with the following statements:

“Interacting with the system was frequently frustrating.”

strongly disagree	disagree	Neutral	agree	strongly agree

“When the robot misunderstood something, I was often able to figure out what its problem consisted in.”

strongly disagree	disagree	Neutral	agree	strongly agree

General comments:

(a) The questionnaire for the No Error Phase.

How well would you say the robot understood you? (1 = very poor, 5 =very good)

1	2	3	4	5

Please rate how much you would agree or disagree with the following statements:

“Interacting with the system was frequently frustrating.”

strongly disagree	agree	neutral	agree	strongly agree

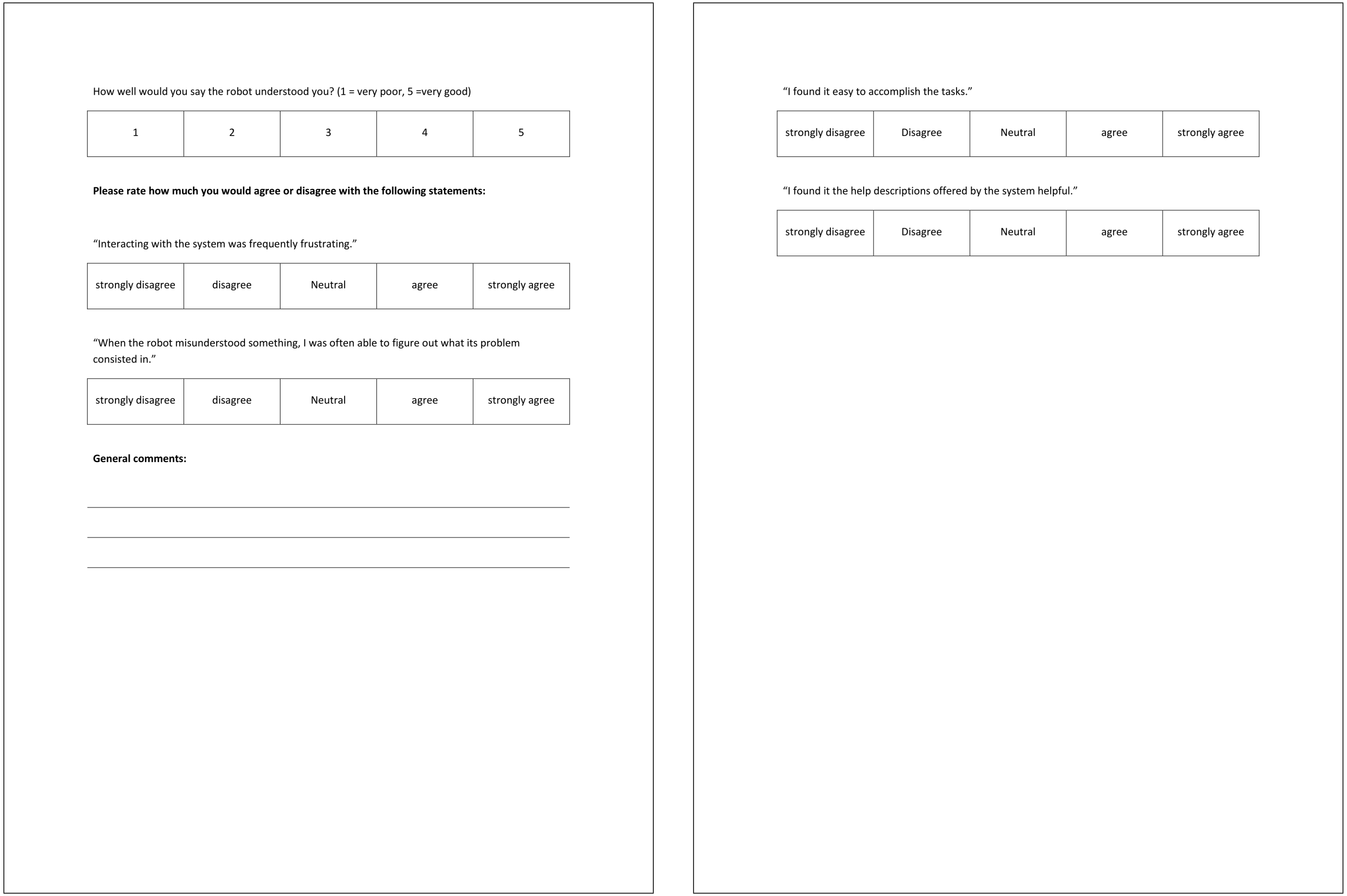
“I think I often understood what the system’s problem consisted in.”

strongly disagree	agree	neutral	agree	strongly agree

General comments:

(b) The questionnaire for the Error Phase.

Figure B.17: The questionnaires for the No Error Phase and the Error Phase.



How well would you say the robot understood you? (1 = very poor, 5 =very good)

1	2	3	4	5

Please rate how much you would agree or disagree with the following statements:

“Interacting with the system was frequently frustrating.”

strongly disagree	disagree	Neutral	agree	strongly agree

“When the robot misunderstood something, I was often able to figure out what its problem consisted in.”

strongly disagree	disagree	Neutral	agree	strongly agree

General comments:

(a) The questionnaire for the Description Phase (Page 1).

“I found it easy to accomplish the tasks.”

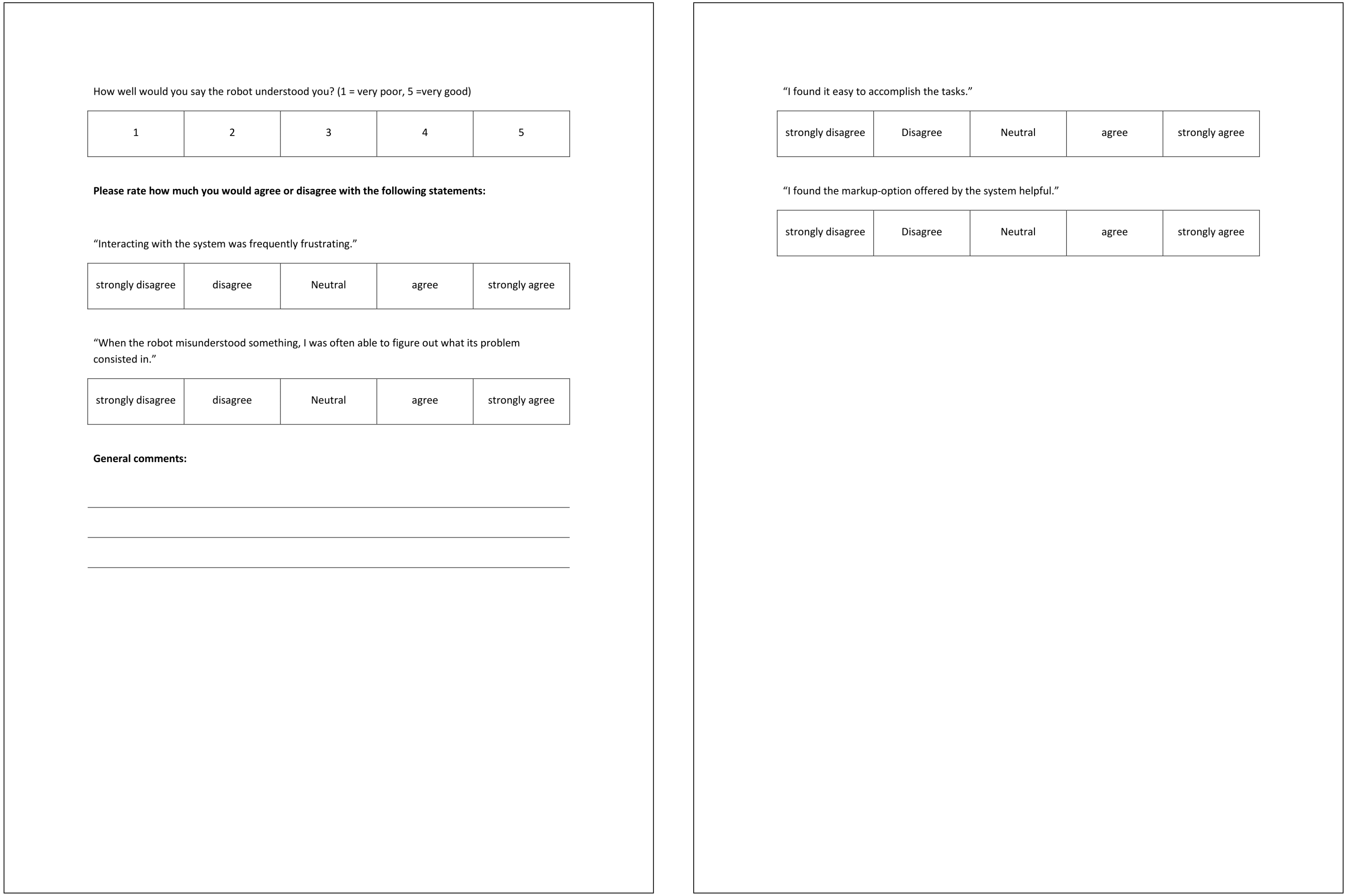
strongly disagree	Disagree	Neutral	agree	strongly agree

“I found it the help descriptions offered by the system helpful.”

strongly disagree	Disagree	Neutral	agree	strongly agree

(b) The questionnaire for the Description Phase (Page 2).

Figure B.18: The questionnaire for the Description Phase.



How well would you say the robot understood you? (1 = very poor, 5 =very good)

1	2	3	4	5

Please rate how much you would agree or disagree with the following statements:

“Interacting with the system was frequently frustrating.”

strongly disagree	disagree	Neutral	agree	strongly agree

“When the robot misunderstood something, I was often able to figure out what its problem consisted in.”

strongly disagree	disagree	Neutral	agree	strongly agree

General comments:

______

______

______

(a) The questionnaire for the Markup Phase (Page 1).

“I found it easy to accomplish the tasks.”

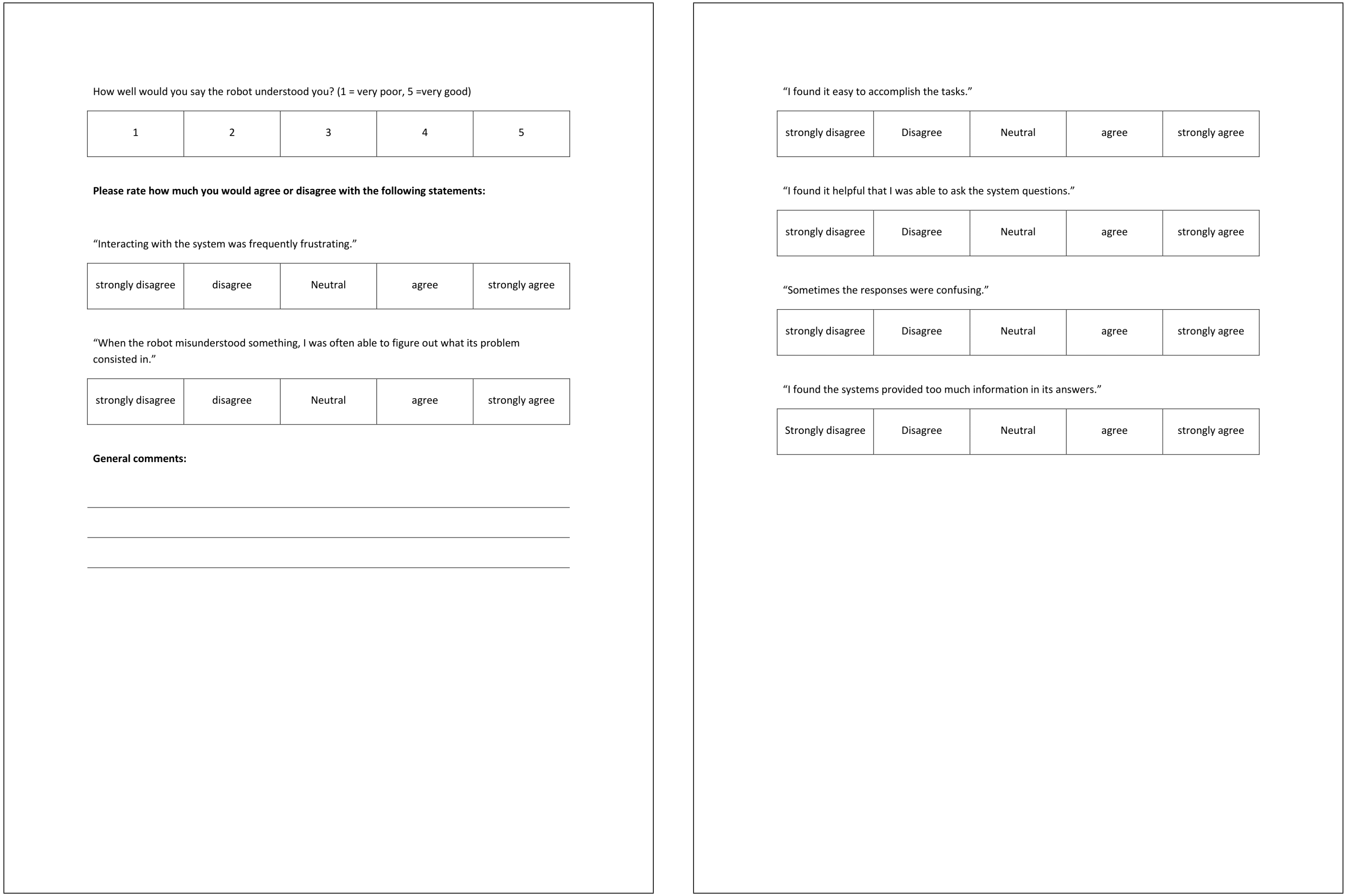
strongly disagree	Disagree	Neutral	agree	strongly agree

“I found the markup-option offered by the system helpful.”

strongly disagree	Disagree	Neutral	agree	strongly agree

(b) The questionnaire for the Markup Phase (Page 2).

Figure B.19: The questionnaire for the Markup Phase.



How well would you say the robot understood you? (1 = very poor, 5 =very good)

1	2	3	4	5

Please rate how much you would agree or disagree with the following statements:

“Interacting with the system was frequently frustrating.”

strongly disagree	disagree	Neutral	agree	strongly agree

“When the robot misunderstood something, I was often able to figure out what its problem consisted in.”

strongly disagree	disagree	Neutral	agree	strongly agree

General comments:

____

____

____

(a) The questionnaire for the Querying Phase (Page 1).

“I found it easy to accomplish the tasks.”

strongly disagree	Disagree	Neutral	agree	strongly agree

“I found it helpful that I was able to ask the system questions.”

strongly disagree	Disagree	Neutral	agree	strongly agree

“Sometimes the responses were confusing.”

strongly disagree	Disagree	Neutral	agree	strongly agree

“I found the systems provided too much information in its answers.”

Strongly disagree	Disagree	Neutral	agree	strongly agree

(b) The questionnaire for the Querying Phase (Page 2).

Figure B.20: The questionnaire for the Querying Phase.

References

Anderson, A., Bader, B., M., E., B., E., G.M., Doherty, Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H.S. & Weinert, R. (1992). The HCRC map task corpus. *Language and Speech*, **34**, 351–366. 61, 84, 95

Becker, T. (2006). Natural language generation with fully specified templates. In W. Wahlster, ed., *SmartKom: Foundations of Multimodal Dialogue Systems*, Cognitive Technologies, 401–410, Springer Berlin Heidelberg. 41

Brennan, S.E., Friedman, M.W. & Pollard, C.J. (1987). A centering approach to pronouns. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, 155–162, Association for Computational Linguistics. 37

Bunt, H. (1994). Context and dialogue control. *THINK Quarterly*, **3**, 19 – 31. 24

Byron, D.K. (2002). Resolving pronominal reference to abstract entities. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, 80 – 87, Philadelphia. 38

Byron, D.K. (2003). Understanding referring expressions in situated language some challenges for real-world agents. *Language Understanding and Agents for Real World Interaction*, 39. 17

Chernova, S., Orkin, J. & Breazeal, C. (2010). Crowdsourcing HRI through online multiplayer games. In *Dialog with Robots: AAAI fall symposium*. 17

Clark, H.H. (1996). *Using language*, vol. 1996. Cambridge University Press Cambridge. 45, 46

Clark, H.H. & Schaefer, E.F. (1989). Contributing to discourse. *Cognitive Science*, 259–294. 45

Cohen, P.R. & Perrault, C.R. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science*, **3**, 177–212. 24

Coradeschi, S. & Saffiotti, A. (2000). Anchoring symbols to sensor data: preliminary report. In *AAAI/IAAI*, 129–135. 39

Coradeschi, S. & Saffiotti, A. (2003). An introduction to the anchoring problem. *Robotics and Autonomous Systems*, **43**, 85–96. 39

Costello, F.J. & Kelleher, J.D. (2006). Spatial prepositions in context: The semantics of near in the presence of distractor objects. In *Proceedings of the Third ACL-SIGSEM Workshop on Prepositions*, 1–8, Association for Computational Linguistics. 55

Coulthard, M. & Brazil, D. (1981). Exchange Structure. In *Studies in discourse analysis*, 82–106, Routledge & Kegan Paul. 30

Dale, R. (1989). Cooking up referring expressions. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, 68–75, Association for Computational Linguistics. 42

Dale, R. & Haddock, N. (1991). Generating referring expressions involving relations. In *Proceedings of the fifth conference on European chapter of the Association for Computational Linguistics*, 161–166, Association for Computational Linguistics. 42

Dale, R. & Reiter, E. (1995). Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive science*, **19**, 233–263. 42

de Marneffe, M.C. & Manning, C.D. (2008). Stanford typed dependencies manual. `http://nlp.stanford.edu/software/dependencies_manual.pdf`, accessed: 2015-10-19. 104

Dzikovska, M., Steinhauser, N., Farrow, E., Moore, J. & Campbell, G. (2014). BEETLE II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. *International Journal of Artificial Intelligence in Education*, **24**, 284–332. 19

Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J. & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, **111**, 98–136. 51

Foster, M.E., White, M., Setzer, A. & Catizone, R. (2005). Multimodal generation in the COMIC dialogue system. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, 45–48, Association for Computational Linguistics. 41

Gabsdil, M. (2003). Clarification in spoken dialogue systems. In *Proceedings of the 2003 AAAI Spring Symposium. Workshop on Natural Language Generation in Spoken and Written Dialogue*, 28–35. 31

Gorniak, P. & Roy, D. (2004). Grounded semantic composition for visual scenes. *J. Artif. Intell. Res. (JAIR)*, **21**, 429–470. 39, 95

Grice, H.P. (1975). Logic and conversation. In Peter Cole & J.L. Morgan, eds., *Syntax and semantics*, vol. 3, Academic Press, New York. 41

Grosz, B.J. & Sidner, C.L. (1986). Attention, intentions, and the structure of discourse. *Computational linguistics*, **12**, 175–204. 30

Grosz, B.J., Joshi, A.K. & Weinstein, S. (1995). Centering: A framework for modelling the coherence of discourse. *Computational Linguistics*, **21**. 37

Hajičová, E., Kuboň, V. & Kuboň, P. (1992). Stock of shared knowledge: A tool for solving pronominal anaphora. In *Proceedings of the 14th conference on Computational linguistics-Volume 1*, 127–133, Association for Computational Linguistics. 37

Harnad, S. (1990). The symbol grounding problem. 39

Hawes, N., Klenk, M., Lockwood, K., Horn, G.S. & Kelleher, J.D. (2012). Towards a cognitive system that can recognize spatial regions based on context. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.* 17

Hirst, G., McRoy, S., Heeman, P., Edmonds, P. & Horton, D. (1994). Repairing conversational misunderstandings and non-understandings. *Speech communication*, **15**, 213–229. 48

Hoey, J., Poupart, P., Bertoldi, A.v., Craig, T., Boutilier, C. & Mihailidis, A. (2010). Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process. *Computer Vision and Image Understanding*, **114**, 503–519. 57

Horacek, H. (2005). Generating referential descriptions under conditions of uncertainty. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG)*, 58–67, Citeseer. 43

Itti, L. (2007). Visual salience. *Scholarpedia*, **2**, 3327, revision #72776. 40

Itti, L. & Koch, C. (2001). Computational modelling of visual attention. *Nature reviews neuroscience*, **2**, 194–203. 40

Jurafsky, D. & Martin, J.H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edn. 44

Keizer, S. (2001). A Bayesian approach to dialogue act classification. In *BI-DIALOG 2001: Proc. of the 5th Workshop on Formal Semantics and Pragmatics of Dialogue*, 210–218. 26

Kelleher, J. (2003). *A Perceptually Based Computational Framework for the Interpretation of Spatial Language*. Ph.D. thesis, Dublin City University. 53

Kelleher, J. & Costello, F. (2005). Cognitive representations of projective prepositions. In *Proceedings of the Second ACL-Sigsem Workshop of The Linguistic Dimensions of Prepositions and their Use in Computational Linguistic Formalisms and Applications*. 54

Kelleher, J. & van Genabith, J. (2004). Visual salience and reference resolution in simulated 3-d environments. *Artificial Intelligence Review*, **21**, 253–267. 40

Kelleher, J., Costello, F. & van Genabith, J. (2005). Dynamically structuring, updating and interrelating representations of visual and linguistic discourse context. *Artificial Intelligence*, **167**, 62–102. 38, 63, 94

Kelleher, J.D. (2006). Attention driven reference resolution in multimodal contexts. *Artificial Intelligence Review*, **25**, 21–35. 40, 53, 95

Kelleher, J.D. & Costello, F.J. (2009). Applying computational models of spatial prepositions to visually situated dialog. *Computational Linguistics*, **35**, 271–306. 19, 56

Kelleher, J.D. & Kruijff, G.J.M. (2005). A context-dependent algorithm for generating locative expressions in physically situated environments. *Proceedings of ENLG-05, Aberdeen, Scotland*. 42

Kelley, J.F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, **2**, 26–41. 91

Kim, Y.J. (2015). SPL (simple programming language). `http://www.helloapps.com/SPL/Overview_of_spl.html`. 94

Knoll, A., Hildenbrandt, B. & Zhang, J. (1997). Instructing cooperating assembly robots through situated dialogues in natural language. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 1, 888–894, IEEE. 63

Krahmer, E. & Van Deemter, K. (2012). Computational generation of referring expressions: A survey. *Computational Linguistics*, **38**, 173–218. 43

Kruijff, G.J.M., Kelleher, J.D., Berginc, G. & Leonardis, A. (2006a). Structural descriptions in human-assisted robot visual learning. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, 343–344, ACM. 53

Kruijff, G.J.M., Zender, H., Jensfelt, P. & Christensen, H.I. (2006b). Clarification dialogues in human-augmented mapping. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, 282–289, ACM. 58

Kruse, T., Kirsch, A., Sisbot, E.A. & Alami, R. (2010). Exploiting human cooperation in human-centered robot navigation. 192–197, IEEE. 296

Kuffner, J. & Latombe, J.C. (1999). Fast synthetic vision, memory, and learning models for virtual humans. 118–127, IEEE Comput. Soc. 53

Levelt, W.J. (1996). Perspective taking and ellipsis in spatial descriptions. *Language and space*, 77–107. 55, 56

Liu, C. & Chai, J.Y. (2015). Learning to mediate perceptual differences in situated human-robot dialogue. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, {USA.}*, 2288–2294. 53, 57

Liu, C., Fang, R. & Chai, J.Y. (2012). Towards mediating shared perceptual basis in situated dialogue. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 140–149, Association for Computational Linguistics. 51, 57, 298

Liu, C., Fang, R., She, L. & Chai, J.Y. (2013). Modeling collaborative referring for situated referential grounding. In *Proceedings of the SIGDIAL 2013 Conference*, 78–86. 57

Logan, G.D. & Sadler, D.D. (1996). A computational analysis of the apprehension of spatial relations. In P. Bloom, M. Peterson, M. Garrett & L. Nadel, eds., *Language and Space*, 493–529, MIT Press, Cambridge, MA. 55

Manning, C., Surdeanu, M., Bauer, J., Finekl, J., Bethard, S. & McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association of Computational Linguistics: System Demonstrations*, 55–60. 101

Mast, V. & Wolter, D. (2013). A probabilistic framework for object descriptions in indoor route instructions. In D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum, T. Tenbrink, J. Stell, A. Galton & Z. Wood, eds., *Spatial Information Theory*, vol. 8116, 185–204, Springer International Publishing, Cham. 58

Mast, V., Couto Vale, D., Falomir, Z. & Fazleh Elahi, M. (2014). Referential grounding for situated human-robot communication. In *Proceedings of SemDial (DialWatt)*, Edinburgh. 58

McTear, M.F. (2002). Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys (CSUR)*, **34**, 90–169. 16, 26

Melin, H.a., Sandell, A. & Ihse, M. (2001). CTT-bank: A speech controlled telephone banking system-an initial evaluation. *TMH-QPSR*, **1**, 1–27. 18

Mitchell, M. (2012). *Generating Reference to Visible Objects*. Ph.D. thesis, University of Aberdeen. 51

Noser, H., Renault, O., Thalmann, D. & Thalmann, N.M. (1995). Navigation for digital actors based on synthetic vision, memory, and learning. *Computers \& graphics*, **19**, 7–19. 53

Orkin, J. & Roy, D. (2007). The restaurant game: Learning social behavior and language from thousands of players online. *Journal of Game Development*, **3**, 39–60. 30

Orkin, J.D. (2013). *Collective artificial intelligence: simulated role-playing from crowdsourced data*. Ph.D. thesis, Massachusetts Institute of Technology. 92

Paek, T. (2003). Toward a taxonomy of communication errors. In *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*. 46

Petrick, R.P. & Foster, M.E. (2013). Planning for Social Interaction in a Robot Bartender Domain. In *ICAPS*. 17

Pickering, M.J. & Garrod, S. (2006). Alignment as the Basis for Successful Communication. *Research on Language and Computation*, **4**, 203–228. 5

Purver, M., Ginzburg, J. & Healey, P. (2003). On the means for clarification in dialogue. In J. van Kuppevelt & R.W. Smith, eds., *Current and New Directions in Discourse and Dialogue*, vol. 22 of *Text, Speech and Language Technology*, 235–255, Springer Netherlands. 31

Regier, T. & Carlson, L. (2001). Grounding spatial language in perception: An empirical and computational investigation. *Journal of experimental psychology: General*, **130**, 273 – 298. 55

Reiter, E. & Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press, New York, NY, USA. 21, 40, 41

Riek, L. (2012). Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. *Journal of Human-Robot Interaction*, 119–136. 92

Roy, D., Gorniak, P., Mukherjee, N. & Juster, J. (2002). A trainable spoken language understanding system for visual object selection. In *INTERSPEECH*, Citeseer. 39

Salmon-Alt, S. & Romary, L. (2009). Reference resolution within the framework of cognitive grammar. In *Proceedings of the Seventh International Colloquium on Cognitive Science (ICCS-01)*, 284–299, Donostia, Spain. 38

Schegloff, E.A. & Sacks, H. (1973). Opening up closings. *Semiotica*, **8**. 29

Schneider, A. & Luz, S. (2011). Speaker alignment in synthesised, machine translated communication. In *IWSLT*, 254–260. 270

Schröder, M. & Trouvain, J. (2003). The german text-to-speech synthesis system MARY: a tool for research, development and teaching. *International Journal of Speech Technology*, **6**, 365–377. 113

Schütte, N., Kelleher, J. & Mac Namee, B. (2010). Visual Salience and Reference Resolution in Situated Dialogues: A Corpus-based Evaluation. In *Proceedings of the AAAI Symposium on Dialog with Robots*, Arlington, Virginia, USA. 13, 94

Schütte, N., Kelleher, J.D. & Mac Namee, B. (2011). Automatic annotation of referring expression in situated dialogues. *International Journal Of Computational Linguistics And Applications*, **2**. 13

Schütte, N., Kelleher, J. & Mac Namee, B. (2012). A corpus based dialogue model for grounding in situated dialogue. In *Proceedings of the 1st Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Language, Motor Control and Vision (MLIS-2012).*, Montpellier, France. 61, 84

Schütte, N., Kelleher, J.D. & Mac Namee, B. (2014a). Clarification Dialogues for Perception-based Errors in Situated Human-Computer Dialogues. In *Proceedings of the 2014 Workshop on Multimodal, Multi-Party, Real-World Human-Robot Interaction*, MMRWHRI '14, 25–26, ACM Press. 8

Schütte, N., Kelleher, J.D. & Mac Namee, B. (2014b). The effect of sensor errors in situated human-computer dialogue. In *Proceedings of the The 3rd Annual Meeting Of The EPSRC Network On Vision & Language and The 1st Technical Meeting of the European Network on Integrating Vision and Language*, Dublin. 8

Schütte, N., Kelleher, J.D. & Mac Namee, B. (2014c). Perception Based Misunderstandings in Human-Computer Dialogues. In *Proceedings of SemDial (DialWatt)*, Edinburgh. 8

Schütte, N., Kelleher, J.D. & Mac Namee, B. (2015). Reformulation Strategies of Repeated References in the Context of Robot Perception Errors in Situated Dialogue. In *Workshop on Spatial Reasoning and Interaction for Real-World Robotics at the International Conference on Intelligent Robots and Systems 2015 (http://iros2015spatial-workshop.lsr.ei.tum.de)*. 8

Searle, J.R. (1975). A taxonomy of illocutionary acts. *Language, Mind, and Knowledge*, **7**, 344–369. 23

Seneff, S. & Polifroni, J. (2000). Dialogue management in the Mercury flight reservation system. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*, 11–16, Association for Computational Linguistics. 18

Sjöö, K. (2011). *Functional understanding of space: Representing spatial knowledge using concepts grounded in an agent's purpose*. Ph.D. thesis, KTH. 53

Strube, M. & Müller, C. (2003). A machine learning approach to pronoun resolution in spoken dialogue. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, 168–175, Association for Computational Linguistics. 38

Summers-Stay, D., Cassidy, T. & Voss, C.R. (2014). Joint Navigation in Commander/Robot Teams: Dialog & Task Performance When Vision is Bandwidth-Limited. *V&L Net 2014*, 9. 2, 53

Surendran, D. & Levow, G.A. (2006). Dialog act tagging with support vector machines and hidden markov models. In *Proceedings of Interspeech/ICSLP*. 26

Thórisson, K.R. (2002). Machine perception of real-time multimodal natural dialogue. *ADVANCES IN CONSCIOUSNESS RESEARCH*, **35**, 97–116. 17

Trask, R. & Stockwell, P. (2007). *Language and Linguistics: The Key Concepts*. Key Concepts Series, Routledge. 105

Traum, D.R. (2000). 20 questions on dialogue act taxonomies. *Journal of semantics*, **17**, 7–30. 25

Traum, D.R. & Larsson, S. (2003). The information state approach to dialogue management. In *Current and New Directions in Discourse and Dialogue*, vol. 22 of *Text, Speech and Language Technology*, 325–353, Springer Netherlands. 27

TUFTE, E. (ND). Slopegraphs for comparing gradients: Slopegraph theory and practice. `http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0003nk`, accessed: 2015-10-19. 278

TUFTE, E.R. (1986). *The Visual Display of Quantitative Information*. Graphics Press, Chesronhire, CT, USA. 278

VAN DER SLUIS, I.F. (2005). *Multimodal Reference, Studies in Automatic Generation of Multimodal Referring Expressions*. Ph.D. thesis. 42

WACHSMUTH, I. & CAO, Y. (1995). Interactive graphics design with situated agents. In *Graphics and Robotics*, 73–85, Springer. 40

WAHLSTER, W., ed. (2006). *SmartKom: Foundations of Multimodal Dialogue Systems*. Springer, Berlin. 41

WALKER, M.A., LITMAN, D.J., KAMM, C.A. & ABELLA, A. (1997). PARADISE: a framework for evaluating spoken dialogue agents. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, 271–280, Association for Computational Linguistics. 88

WINOGRAD, T. (1971). *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*. Ph.D. thesis. 52, 63

YOUNG, S., GASIC, M., THOMSON, B. & WILLIAMS, J.D. (2013). POMDP-Based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, **101**, 1160–1179. 28