GIVENNESS-HIERARCHY-INFORMED
DOCUMENT PLANNING

by

Kevin E. Spevak

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Robotics).

Golden, Colorado

Date ________________________

Signed: ________________________

Kevin E. Spevak

Signed: ________________________

Dr. Neil T. Dantam
Thesis Advisor

Golden, Colorado

Date ________________________

Signed: ________________________

Dr. Kevin L. Moore
Professor and Director of Mines Robotics Program
Interdisciplinary Programs

ABSTRACT

Robots that use natural language in collaborative tasks must refer to objects in their environment. Recent work has shown the utility of the linguistic theory of the *Givenness Hierarchy* (GH) in generating appropriate referring forms. But before *referring expression generation*, collaborative robots must determine the content and structure of a sequence of utterances, a task known as *document planning* in the natural language generation community. This problem presents additional challenges for robots in situated contexts, where described objects change both physically and in the minds of their interlocutors. In this work, we consider how robots can "think ahead" about the objects they must refer to and how to refer to them, sequencing object references to form a coherent, easy to follow chain. Specifically, we leverage GH to enable robots to plan their utterances in a way that keeps objects at a high *cognitive status*, which enables use of concise, anaphoric referring forms. We encode these linguistic insights as a *mixed integer program* within a planning context, formulating constraints to concisely and efficiently capture GH-theoretic cognitive properties. We demonstrate that this GH-informed planner generates sequences of utterances with high inter-sentential coherence, which we argue should enable substantially more efficient and natural human-robot dialogue.

This thesis is an extension of work originally presented at the International Conference on Intelligent Robots and Systems (IROS) 2022 [1]. Content from the original work is reused in this thesis with permission. See Appendix C for information on the original publication and copyright permissions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Set of grounded predicates . . . $\mathcal{P}$

Set of grounded actions . . . $\mathcal{A}$

Set of objects . . . $\mathcal{O}$

Set of predicates that are true in the initial state . . . $\mathcal{I}$

Set of predicates that are true in the goal state . . . $\mathcal{G}$

Boolean variable indicating whether grounded predicate p is true at time t . . . p_t

Boolean variable indicating whether grounded action a is taken at time t . . . a_t

Maximum number of time steps . . . T

Positive preconditions of grounded action a . . . $\text{pre}^+(a)$

Negative preconditions of grounded action a . . . $\text{pre}^-(a)$

Positive effects of grounded action a . . . $\text{eff}^+(a)$

Negative effects of grounded action a . . . $\text{eff}^-(a)$

Set of grounded actions with grounded predicate p as a positive effect . . . $\text{add}(p)$

Set of grounded actions with grounded predicate p as a negative effect . . . $\text{del}(p)$

Big O (asymptotic growth rate) . . . O

Boolean variable indicating whether object o is *in focus* at time step t . . . $I_{o,t}$

Boolean variable indicating whether object o is *activated* at time step t . . . $A_{o,t}$

Boolean variable indicating whether object o is *familiar* at time step t . . . $F_{o,t}$

Set of grounded actions for which object o is the *topic* . . . $\text{topic}(o)$

Set of grounded actions that refer to object o . . . $\text{ref}(o)$

LIST OF ABBREVIATIONS

Boolean Satisfiability . . . . SAT

Distributed Integrated Affect, Reflection, Cognition . . . . DIARC

Givenness Hierarchy . . . . GH

Human Robot Interaction . . . . HRI

Mixed Integer Programming . . . . MIP

Natural Language Generation . . . . NLG

Planning Domain Definition Language . . . . PDDL

Referring Expression Generation . . . . REG

Task Load Index . . . . TLX

ACKNOWLEDGMENTS

I would like to thank and acknowledge Neil Dantam, Tom Williams, and Zhao Han, who co-authored the original paper [1] presenting a large portion of this thesis. This work was made possible by their collaboration. Special thanks to my advisor Neil Dantam for guiding me in my graduate education and research, and helping me achieve my first publication and conference presentation. I would also like to thank my thesis committee, Neil Dantam, Tom Williams, and Alexandra Newman, for their time, effort, and feedback on my thesis and defense. Finally, I would like to thank the Office of Naval Research, whose funding through grant N00014-21-1-2418 helped make this work possible.

Dedicated to my parents, who fostered my curiosity from an early age,

and to my dearest Hailey, who supports me, inspires me,

and keeps going as I chase my dreams.

CHAPTER 1
INTRODUCTION

This chapter provides a high-level overview of this project. It begins with a discussion of our motivation. A deeper discussion of the background information relevant to our motivation can be found in Chapter 2. Next, we present the goals of our project and a summary of our approach, which is described in detail in chapter 3. We continue with a statement of the research questions we will attempt to answer with this project. Finally, we provide an outline of the remaining chapters in this document.

1.1 Motivation

Robots in domains ranging from collaborative manufacturing to intelligent tutoring will need to communicate with humans in order to coordinate, provide information, and fulfil social expectations. In collaborative manufacturing, for example, a robot may need to instruct a worker as to how to perform a complex task over several steps. In intelligent tutoring, a robot may need to instruct a child as to how to solve a mathematics problem. Using speech to communicate is an attractive choice, as it requires minimal training for human users, allows for hands-free operation. Robotic domains often involve reacting to the environment and handling varied and possibly unforeseen situations, requiring the speech-enabled robots to decide what to say in real time through Natural Language Generation (NLG).

In natural language there is substantial flexibility in the way information can be conveyed. A collaborative manufacturing robot, for example, may need to discuss multiple independent task steps with a human. This robot would need to decide how to order the information, and there may be multiple logically consistent ways to do so. The robot's agency as an embodied agent may lead to even more flexibility; if the robot can perform some of the steps on its own, it needs to choose which steps to discuss with the human. In the NLG community, this task of determining the overall content and structure of generated language is referred to as *document planning*. As the name suggests, however, traditional approaches to document planning are rooted in purely textual domains. But robots must grapple with a *situated context* involving both linguistic and non-linguistic elements. A variety of factors from non-verbal cues to the physical location conversation participants and objects impact language. In particular, speakers must take into account their interlocutors' cognitive context to determine the appropriate way to talk about things.

The linguistic theory of the *Givenness Hierarchy* (GH) [2] establishes a connection between how we refer to things and *cognitive status*. Cognitive status is a measure of how salient a referent is in our minds or in the (metaphorical) mind of the conversation. This theory has been widely accepted in the linguistics literature; at the time of this writing, the Gundel et al.'s seminal work has been referenced over three thousand times, including in numerous textbooks. Moreover, the predictions of the GH have been demonstrated in many different natural languages [3].

GH-based approaches have also seen significant attention in the robotics community. Williams et al. [4] demonstrated how this approach can be used for Natural Language Understanding: facilitating the understanding of a wide variety of referring forms (see also the chapter on this topic in the Oxford Handbook of Reference [5]). More recently, Pal et al. [6, 7] applied the GH to Natural Language Generation (NLG), demonstrating how cognitive status can be estimated across the course of a conversation and used to guide the selection of referring forms. In this work, we explore the application of GH theory in another stage of robot's language generation process. Specifically, we propose leveraging the GH for document planning in order to generate sequences of utterances that are sensitive to the cognitive status dynamics of a robot's human interlocutors. We argue that this approach is fundamentally well-suited for situated human-robot interaction (HRI) domains.

1.2 Goals and Solution

We intend to expand the use of GH theory for HRI applications. Prior work has shown the potential of GH-informed referring form selection. However, that approach is limited to choosing appropriate forms within otherwise static text. This work explores how GH theory can be leveraged at a higher level to make informed decisions about the overall text structure, what information to include in the text, and even the robot's non-linguistic action plans.

Towards this end, we propose GH-informed document planning. We present a proof-of-concept system that models the cognitive status of a human interlocutor and uses it to optimize document plans based on the cognitive status of referents. Our solution formulates a Mixed Integer Program (MIP) that encodes the cognitive status model and communicative task, and optimizes according to a GH-informed objective function. This approach enables generation of document plans with high inter-sentential coherence and facilitates effective use of anaphora over definite descriptions.

We evaluate our work on a hypothetical manipulation task in a human-robot collaborative tasking scenario, and show how sequences of instructions are generated by our approach in that domain. We then generate document plans for the robot's instructions to the human using both a classical planner (that does not model cognitive status) and our GH-informed planner, showing that our formulation enables the use of more concise referring forms.

We argue that such referring form usage should improve the inter-sentential coherence of the final generated language, and ultimately improve efficiency and usability in HRI applications. In particular, the highest cognitive statuses allow for the use of pronouns over definite noun phrases. There are numerous benefits to using these forms [8], as they make dialogue more efficient (and thus less costly to listen to) [9], more predictable (and thus cognitively easier to follow and more humanlike) [10], and conforming to Gricean conversational maxims of cooperative speech [11]. Additionally, the impact of these effects are magnified in situated contexts, in which the use of these forms facilitates—and is facilitated by—the use of deictic gesture to more effectively pick out objects at (and based on) varying distances [12]. We hypothesize that leveraging these benefits will improve task performance and user satisfaction, similar to the advantages gained through use of shorter object descriptions [13].

Another potential benefit of our approach is in reducing the cognitive workload (cf. [14]) of participants in human-robot collaborative tasks. It is well established that human performance is degraded when cognitive workload is too high [15]. Human-like, context-dependent referring forms have been shown to reduce workload [16], and high-cognitive-status referents are conducive to the use of such forms. Furthermore, high working memory load slows spoken-word recognition time [17]. This strain on language processing could be ameliorated by the shorter instructions generated by our method. Reduced cognitive workload, improved human performance on tasks, and increased user satisfaction are highly desirable traits for HRI applications. In addition to our technical approach, we present a human-subjects study design to evaluate these hypothesized benefits.

1.3 Research Questions

We have three questions regarding our proposed use of GH theory in document planning for human-robot communication.

Question 1. Does GH-informed document planning enable the use of more concise referring forms?

We are proposing the use of GH theory at the document planning level to impact the types of referring forms that can be used in the final text. The types of referring forms that are possible depends on the cognitive status of the referents at the time the reference is made. Thus, in order for our approach to be successful, it must "make a difference" in the cognitive status of referents. In other words, we want to verify that our optimal solution makes a significant improvement over an arbitrary satisficing solution.

Question 2. Is GH-informed document planning practical?

Robots may encounter unexpected and previously unseen situations. Thus an important feature is the ability to plan on-the-fly without relying on precomputed solutions. The computational demands of our approach are therefore an important factor to whether our approach could be utilized in practice. Modeling and optimizing for cognitive status increases the complexity of the document planning problem. We seek to determine how much this change impacts running time, and whether the difference could be a barrier to adoption of our approach.

Question 3. Can GH-informed document planning improve human understanding, task performance, and/or satisfaction in a human-robot collaborative interaction?

Whereas questions 1 and 2 concern our document planner in isolation, this question relates to its use in a real HRI scenario. Here we are interested in the impact GH-aware document planning will have on an actual interaction. This includes objective measures such as how well humans perform in cooperative tasks, as well as subjective measures such as humans' satisfaction with the robot. This question is an assessment of our approach's potential utility for practical HRI applications.

1.4 Document Outline

The remainder of this document is organized as follows. Chapter 2 presents background information and a review of related works. Chapter 3 gives an overview of our proposed GH-informed document planner. This includes a high-level description of our approach and system structure. Chapter 4 goes into the technical details that enable the application of GH Theory in document planning. Chapter 5 presents our technical evaluation of a proof-of-concept implementation, answering research questions 1 and 2. Chapter 6 presents the design of a human-subjects study intended to answer research question 3. Chapter 7 discusses ideas for the improvement and application of GH-informed document planning in future work. Finally, Chapter 8 provides a summary and discussion of this work.

CHAPTER 2

BACKGROUND AND RELATED WORK

The goal of this chapter is to provide background on the state of the art in areas related to this work. Towards this end, we review the literature on relevant topics in Linguistics, HRI, NLG, and Automated Planning.

2.1 The Givenness Hierarchy

Our approach is fundamentally grounded in the linguistic theory of the Givenness Hierarchy (GH) [2]. The GH is comprised of a hierarchically nested set of six cognitive statuses: $\{$in focus $\subseteq$ activated $\subseteq$ familiar $\subseteq$ uniquely identifiable $\subseteq$ referential $\subseteq$ type identifiable$\}$. These statuses apply to any *referent*, or thing which is referred to in language, but in this work we are primarily concerned with real-world objects in a robot's environment. The hierarchical structure of the GH implies that any referent that has a given cognitive status also has each lower status. The GH assigns one or more referring forms to each cognitive status level, and for a given from to be appropriate, the referent must have at least the corresponding status. For example, an object that is *in focus* can be referred to with the pronoun "it", but an object that is only *activated* cannot. The full GH and associated forms are shown in Figure 2.1

The GH theory posits that these six cognitive statuses are universal to human discourse, and has been validated across many disparate natural languages languages [3]. Furthermore, human use of referring forms demonstrates an implicit understanding of cognitive status from others' perspectives. For example, a speaker that uses the word "it" implicitly signals a belief that the object is *in focus* in the mind of the listener. This suggests that in order to accomplish natural and human-like speech, an NLG system needs to account for cognitive status.

2.2 The Givenness Hierarchy Coding Protocol

The GH maps cognitive status to referring form, but does not itself provide a means of determining cognitive status. Thus any application that utilizes GH theory requires a cognitive status model. One such model is the Givenness Hierarchy Coding Protocol [18] which was developed by the authors of the work [3] that initially proposed the GH. The protocol consists of a set of increasingly admissive criteria, corresponding to the six GH-theoretic cognitive statuses in high-to-low order. The highest status that a referent has, according to the protocol, is the first one whose criteria are satisfied.

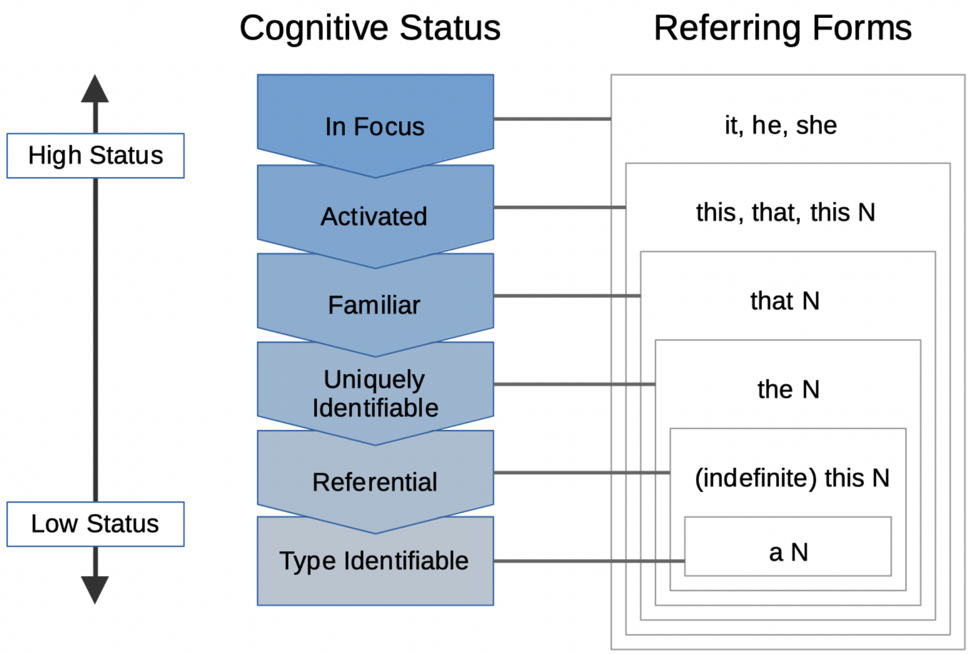

Figure 2.1 The Givenness Hierarchy

2.3 GH Applications in HRI

There is a growing body of work in using computational models of the GH for Natural Language Processing tasks in HRI applications [4, 5, 19, 20], which implement portions of the GH (although not attempting to directly model cognitive status). These works utilize the GH for the Natural Language Understanding task of *reference resolution*, which attempts to map references to specific referents. For example, if a human asks a robot to "give me that", reference resolution would determine what object "that" refers to. Pal et al. [6] built of these works to show that the GH can be used successfully for the NLG task of *referring form selection*. Referring form selection attempts to select the appropriate referring form for a given referent. Pal et al. used probabilistic *cognitive status filters* to predict the cognitive status of each referent, and used this along with distance and the presence of distractors to select referring form.

2.4 Document Planning

In this work we focus on using GH for a key NLG task not considered in prior work: document planning. NLG is often broken down into a series of steps known as a modular NLG pipeline [21]. Pipelines typically include modules for document planning, sentence planning, referring expression generation (REG), and linguistic realization [22]. These common modules are shown in Table 2.1.

Table 2.1 NLG Pipeline Modules

Step	Module	Purpose
1	Document Planning	Generate overarching sequence of sentiments
2	Sentence Planning	Decide how to communicate each sentiment
3	Referring Expression Generation	Construct expressions for each referent
4	Linguistic Realization	Form grammatically valid sentences

At the highest level is the document planner, which decides on an overarching sequence of sentiments to communicate a larger point or achieve a larger goal [23]. This involves content selection, or determining what needs to be said, and also text structuring, or deciding how to order the chosen content. The resulting document plan is then refined to text in the remainder of the pipeline. Sentence planning converts the chosen content into sentence outlines. These outlines contain references, and for each referent, REG constructs a referring expression by selecting appropriate descriptive properties. Finally, Linguistic realization organizes everything into grammatically correct text.

Referring forms include anaphora such as the pronouns "it" and "that," as well as noun phrases. Thus Pal et al.'s GH-informed referring form selection [6] could be applied between steps 2 and 3 in the above example pipeline. After sentence planning, referring forms would be selected for each referent, and REG would be applied for any required noun phrases. In this work, we move the application of GH Theory to earlier in the pipeline with a GH-informed document planner.

2.5 Constraint-Based Planning

From a technical perspective, our approach is grounded in constraint-based planning techniques. Planning is an established technique for NLG. We briefly review the constraint-based planning formulation and refer the reader to [22, 24, 25] for more background.

Classical approaches for constraint-based planning encode a scenario as a logical formula, then use a constraint solver to find a satisfying variable assignment for that formula [25, p69]. In the classical paradigm [26], actions are treated as tuples of preconditions and effects. Variables in the formula represent the state and the action to take for a fixed number of steps. The following constraints are encoded in the formula:

- the start state holds at the initial step;
- the goal condition holds at the final step;
- preconditions and effects of selected actions hold;
- only one action is taken at a time (a.k.a *operator exclusion*).

- state does not change unless modified by an action (a.k.a. *frame axioms*);

The planner progressively increases step count until the formula is satisfiable. The true action variables in the satisfying assignment encode the action to take for each step of the plan.

SAT solvers [27, 28], have been a popular choice for the underlying solver, and SAT-based planners have performed well in planning competitions [29]. Other types of solvers have been used as well, including solvers for Satisfiability Modulo Theories, an extension of SAT [30].

There is a long history of planning based approaches to NLG [22]. For example, Appelt [31] proposed the KAMP system for REG by reasoning a hearer's belief state. Koller and Stone [32] leveraged the formalism of Lexicalised Tree Adjoining Grammar [33] for REG by specifying semantic preconditions and effects.

2.6 PDDL

Planning Domain Definition Language (PDDL) [34] is a formal language used to represent planning problems. PDDL has seen widespread adoption, and is supported as an input to many popular planners. PDDL divides planning tasks into two concepts: the *domain* and the *problem*. The PDDL domain specifies predicates (relations between objects) and possible actions (sets of preconditions and effects, parameterized by objects). The domain can be seen as a representation of a "type" of problem. The PDDL problem belongs to a particular PDDL domain, and specifies the objects, initial state, and goal condition. This can be seen as a "problem instance." In this document, we will use the term *domain* to refer both to a type of robot task (i.e. generate instructions for a human) and the PDDL encoding of that task.

PDDL uses *parenthesized prefix notation*; expressions have the form (op arg1 arg2 . . .) where op is an operator followed by any number of arguments. Object parameter names for predicates and actions begin with the character "?". For example, the expression (holding ?o) represents a predicate named "holding" with a single object parameter "?o". Semantically, this could represent whether the robot is holding a particular object. For further background on PDDL, we refer the reader to [34]. In this work, we will also use action expressions in PDDL-like syntax. For example, we use the expression (pick-up ?o) to mean an action named "pick-up" with a single object parameter named "?o". These expressions do not appear within PDDL definitions, but they are convenient for concisely representing actions defined in PDDL.

PDDL is a *lifted* or first-order representation of planning problems. *Grounding* is the process of converting to a propositional representation by expanding each possible set of object parameters for actions and predicates. In the example above, (pick-up ?o) is a lifted action in PDDL-like syntax. For a problem instance with two objects, block-A and block-B, the grounded representation would contain the actions (pick-up block-A) and (pick-up block-B). If the effects of the lifted action (pick-up ?o) contain the lifted predicate (holding ?o), then the effects of grounded action (pick-up block-A) contain the grounded predicate (holding block-A). A grounded PDDL problem can be encoded with the constraints given in Section 2.5. Thus, constraint-based planning approaches that accept PDDL as an input generally include grounding as a pre-processing step. For simplicity, we will use the sets of grounded actions and predicates when specifying formulations in this work.

2.7 Planning as Mixed Integer Programming

The relationship between SAT and integer programs is long-established [35], and integer programming has been used in efficient planners [36]. Mixed integer programming (MIP) is a generalization of integer programming that allows for real-valued variables. Thus, MIP can be used for planning, and also supports real values that cannot be represented in SAT. Furthermore, MIP finds optimal solutions while SAT can only find satisfying ones. In this work, we use MIP-based planning to optimize based on cognitive status. We need an optimizing solver in order to find plans that maximally benefit from our GH insights. MIP's capabilities in real-valued optimization enable the modeling of relevant real-world values such as distance, as well as probabilistic models of cognitive status such as Pal et al.'s Cognitive Status Filter [6]. Strictly speaking, our formulation is pure integer programming because we use a deterministic model of cognitive status, and we have only discrete-valued variables. However, our implementation uses the commercial Gurobi Optimizer [37] which can handle MIP, in order to make it extensible for future work involving real values. While MIP is at least as hard as SAT [35], MIP formulations critically let us leverage highly engineered solution techniques [37–39].

The reduction of planning to MIP is well known in the art. Here, we provide a formulation for the reader's reference. In Chapter 4, we augment this formulation for our GH-informed planner. Both formulations assume a planning problem instance specified according to Definition 1

Definition 1 (Classical Planning) *A planning instance is the tuple* $(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{I}, \mathcal{G}, T)$*, where:*

- $\mathcal{P}$ *is the set of grounded predicates*
- $\mathcal{A}$ *is the set of grounded actions*

  $\mathrm{pre}^-(a) \subseteq \mathcal{P}$ *is the set of negative preconditions of* $a \in A$

  $\mathrm{pre}^+(a) \subseteq \mathcal{P}$ *is the set of positive preconditions of* $a \in A$

  $\mathrm{eff}^-(a) \subseteq \mathcal{P}$ *is the set of negative effects of* $a \in A$

  $\mathrm{eff}^+(a) \subseteq \mathcal{P}$ *is the set of positive effects of* $a \in A$

  $\mathrm{del}(p) \subseteq \mathcal{A}$ *is the set of actions* $a \in A$ *such that* $p \in \mathrm{eff}^-(a)$

$\text{add}(p) \subseteq \mathcal{A}$ *is the set of actions* $a \in A$ *such that* $p \in \text{eff}^+(a)$

- $\mathcal{O}$ *is the set of objects*
- $\mathcal{I} \subseteq \mathcal{P}$ *is the set of true predicates in the initial state*
- $\mathcal{G} \subseteq \mathcal{P}$ *is the set of true predicates in the goal state*
- T *is the maximum number of steps in a plan*

Classical constraint-based approaches do not involve the set of objects outside of the grounding step. Since $\mathcal{P}$ and $\mathcal{A}$ in Definition 1 are already grounded, Formulation 1 does not reference $\mathcal{O}$. However, our GH-informed approach references objects directly, so $\mathcal{O}$ is needed in Formulation 2, given in Chapter 4.

Formulation 1 (Classical Planning as MIP) *Given a planning instance* $(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{I}, \mathcal{G}, T)$

Decision Variables

$$p_t \in \{0,1\}, \quad \forall p \in \mathcal{P},\ t \in [0,T] \qquad \text{(1 iff predicate } p \text{ is true at time step } t) \tag{2.1}$$
$$a_t \in \{0,1\}, \quad \forall a \in \mathcal{A},\ t \in [1,T] \qquad \text{(1 iff action } a \text{ is taken at time step } t) \tag{2.2}$$

Objective: *minimize steps to achieve goal state*

$$\min \sum_{a \in \mathcal{A}, t \in [1,T]} a_t \tag{2.3}$$

Subject to:

$$p_0 = 1, \quad \forall p \in \mathcal{I} \tag{2.4}$$
$$p_0 = 0, \quad \forall p \notin \mathcal{I} \tag{2.5}$$
$$p_T = 1, \quad \forall p \in \mathcal{G} \tag{2.6}$$
$$a_t \leq p_{t-1}, \quad \forall a \in \mathcal{A}, p \in \text{pre}^+(a), t \in [1,T] \tag{2.7}$$
$$a_t \leq 1 - p_{t-1}, \quad \forall a \in \mathcal{A}, p \in \text{pre}^-(a), t \in [1,T] \tag{2.8}$$
$$a_t \leq p_t, \quad \forall a \in \mathcal{A}, p \in \text{eff}^+(a), t \in [1,T] \tag{2.9}$$
$$a_t \leq 1 - p_t, \quad \forall a \in \mathcal{A}, p \in \text{eff}^-(a), t \in [1,T] \tag{2.10}$$
$$\sum_{a \in A} a_t \leq 1, \quad \forall t \in [1,T] \tag{2.11}$$
$$p_t \leq p_{t-1} + \sum_{a \in \text{add}(p)} a_t, \quad \forall p \in \mathcal{P}, t \in [1,T] \tag{2.12}$$
$$p_t \geq p_{t-1} - \sum_{a \in \text{del}(p)} a_t, \quad \forall p \in \mathcal{P}, t \in [1,T] \tag{2.13}$$

Variables (2.1) encode the state space of the planning problem by tracking the value of each grounded predicate at each time step. Variables (2.2) encode the plan by tracking which action is taken at each time step. Conditions (2.4) and (2.5) encode the initial state, (2.6) encodes the goal condition, (2.7) through (2.10) enforce consistency of action preconditions and effects, (2.11) encodes operator exclusion, and (2.12) and (2.13) are the frame axioms. The objective function (2.3) counts the actions taken which ensures that the optimal solution contains a minimum number of steps.

The optimal solution to MIP Formulation 1 can be converted to a plan by selecting each action variable a_t which is assigned a value of 1, and ordering by the time step t.

CHAPTER 3
APPROACH

This chapter introduces a novel document planning approach developed in this work. We begin by defining the problem at hand, which we call *situated document planning*. Then, we describe our solution, a GH-informed document planner.

3.1 Situated Document Planning

We define the NLG task of *situated document planning* as a special case of document planning applied to HRI contexts. For a robot that sharing a physical environment with a human, situated document planning is the task of generating a document plan that references physical objects in the shared environment.

In this work, we will focus on a particular situated document planning domain based on *instruction giving*. In this task, the robot's communicative goal is to describe an embodied task to a human. Specifically, the human is capable of performing a set of actions on the objects in the environment, and the robot must generate a sequence of utterances instructing the human on which actions to take. This situated document planning problem is represented as a constraint-based planning problem as follows:

- Objects in the planning problem correspond to the physical objects that the human can interact with.
- Predicates represent the state of the world based on relationships between objects.
- Actions represent utterance skeletons that instruct the interlocutor to manipulate physical objects.
- The initial state corresponds to no instructions having been given
- The goal condition is that each step in the manipulation task has been described.

For simplicity, we will assume there is a one-to-one correspondence between these utterance skeletons and the actions the human is capable of performing. In other words, for each manipulation action m from the human's perspective, there is a communicative action a that represents the robot telling the human to perform m. This assumption is purely for clarity of our example; the domain is easily generalizable to include more types of utterances that the robot can use. As an example, using PDDL-like syntax, a plan step (pick-up block-A) would correspond to an utterance by the robot instructing the human to pick up the item designated as block-A. Planning actions take objects as parameters, and each parameter assignment (block-A the example above) in the plan is equivalent to the selection of a referent in the instruction-giving task. Table 3.1 summarizes this representation. The bracketed ellipsis [...] in the utterance skeleton example represents a placeholder where an object reference can be inserted. This placeholder corresponds to the object parameter "?o" in the planning representation.

Table 3.1 Classical Approach to Situated Document Planning

Instruction Giving	**Example**	**Planning Representation**	**Example**
physical object	block A	object	block-A
utterance skeleton	"Pick up [...]."	action	(pick-up ?o)
utterance	"Pick up that block"	grounded action	(pick-up block-A)

Table reused with modifications from [1]. © 2022 IEEE.

While not performed in this work, these forms could easily be translated into the types of utterance representations typically used in cognitive robotic architectures like DIARC [40]. Specifically, each plan step, when translated into a predicate p (e.g. pick-up(block-A)) can be assumed to be part of an utterance of form $Inst(r, h, p)$, i.e. an Instruction from the speaker (robot r) to the hearer h instructing them to perform action p. However, this could be formulated in other ways, based on pragmatic inference, to achieve various communicative goals such as politeness [41].

3.2 GH-informed Document Planning

We propose a document planner that models and optimizes for the GH-theoretic cognitive status of referents. Our key insight is that document planning approaches that fail to account for cognitive status may exhibit decreased inter-sentential coherence. For instance, these approaches may introduce more referents than is strictly needed (or repeatedly re-introduce referents that are no longer activated), requiring full definite descriptions rather than shorter anaphoric phrases. In contrast, an approach that aims to use and continue referring to task-relevant entities that are already in focus or activated would lead to greater inter-sentential coherence, shorter and easier-to-follow dialogues, and perhaps even fundamentally simpler plans overall. We also believe this approach is particularly well-suited for the unique demands of situated document planning. Cognitive status is highly dynamic in situated contexts, as it is affected not only by language but by the location and physical actions of both the human and robot.

Figure 3.1 shows the structure of our proposed document planner. The input to the system is a problem definition which specifies the situated document planning task, as well as a model for determining the cognitive status of objects in the environment. We reproduce this model within planner state so that it can inform content determination and text structuring. The system's output is a document plan which can be utilized in a traditional NLG pipeline.

Graphic reused from presentation accompanying [1]. © 2022 IEEE.

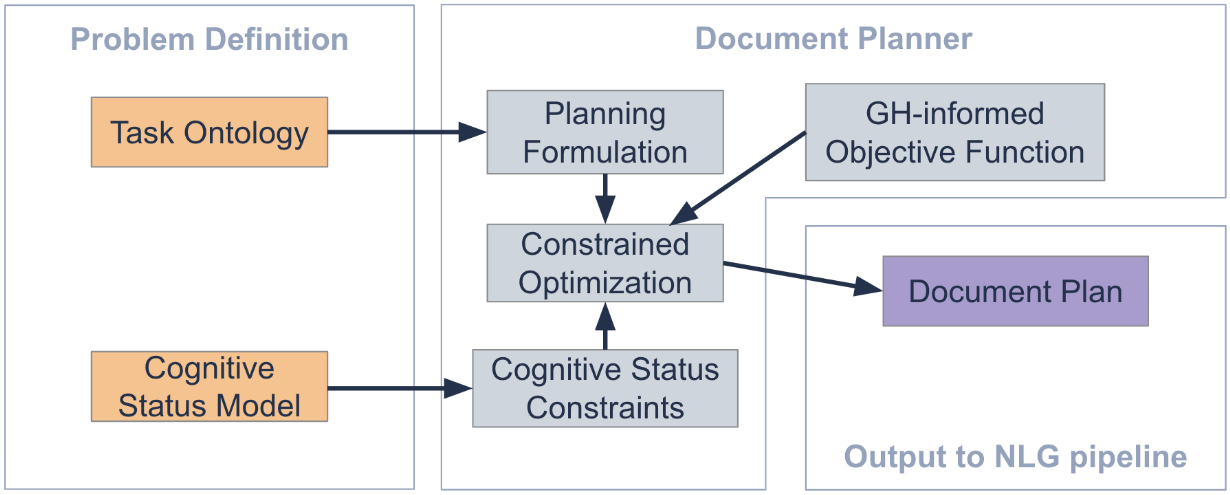

Figure 3.1 Structure of GH-Informed Document Planner

3.3 Task Ontology

The task ontology encodes a situated document planning task within a formal language. This allows our document planner to be domain-independent. In other words, a single generic document planner can be re-used for different robots and scenarios. Changing the objects in the environment, the capabilities of the robot, or the communicative goal requires only a change in the planner's input. In our implementation, we use PDDL as the task ontology language. PDDL has the advantage of being widely adopted within the planning community. Moreover, it is sufficiently expressive to model a realistic situated document planning task, as demonstrated by the example in Chapter 5.

3.4 Cognitive Status Model

There are multiple existing models for cognitive status, including the Givenness Hierarchy Coding Protocol [18] and Pal et al.'s Cognitive Status Filter [6]. We show this choice as an input to our system to highlight how it could be changed to accommodate a different model. However, there is no established formal language that we know of to represent this input. Thus changing the cognitive status model would require manually updating the cognitive status constraints used by the planner. In our implementation, we use the Givenness Hierarchy Coding Protocol as our model. It consists of simple deterministic criteria, which makes our proof-of-concept more straightforward. Other options that could be explored in future work are discussed in Chapter 7.

3.5 Planning Formulation

We use the Formulation 1, given in Section 2.7, to encode the task ontology. This converts the PDDL input into a MIP problem with constraints that ensure our solution satisfies the communicative goal. Solving the MIP at this point would give a valid, GH-uninformed document plan. However, we will first augment the problem to model and optimize for cognitive status.

3.6 Cognitive Status Constraints

We add MIP variables to track the cognitive status for each object in the environment. Then, we add MIP constraints that enforce the criteria of our cognitive status model. These constraints do not impact the solution on their own, as they merely track cognitive status in the newly defined variables. However, the new variables make cognitive status available to the objective function. Unlike the task ontology which is first represented as a planning problem in PDDL, we do not represent cognitive status at the planning level. Existing planning languages impact the expressivity and capability of planning approaches [42] and do not easily represent the behavior of cognitive status. Instead, we incorporate cognitive status directly into the MIP formulation. As we will show in Chapter 5, our MIP formulation succeeds in our goal of generating document plans with fewer and higher-cognitive-status referents.

3.7 GH-informed Objective Function

Using the cognitive status variables, we can now create an objective function that is informed by the GH. In this proof of concept, we use this to maximize the cognitive status of referents at the time step that they are referenced. This straightforward objective function demonstrates the capability to impact possible referring forms, but much more sophisticated objective functions could be developed, as discussed in Chapter 7.

3.8 Constrained Optimization

The planning formulation, cognitive status constraints, and GH-informed objective function together form the final MIP encoding. This MIP problem is then solved with the Gurobi Optimizer. Utilizing this commercial solver allows us to take advantage of a highly efficient constrained optimization process.

3.9 Document Plan

The solution to the MIP problem is converted directly to a document plan. The values of action variables determine the plan as in Section 2.7. Since each action represents an utterance skeleton, the full plan can be refined into a sequence of utterances.

CHAPTER 4

PROBLEM FORMULATION

This chapter presents the technical details of our approach to Situated Document Planning defined in Sections 3.6 and 3.7. We provide a novel encoding of GH-theoretic cognitive status as MIP and a sample GH-informed objective function. These are used to augment the MIP formulation of classical planning given in Section 2.5 to form our GH-informed document planner.

4.1 Cognitive Status Constraints

Our GH-informed document planner models the GH-theoretic cognitive status of each object in the scene at each planning time step. We accomplish this by introducing MIP variables which track the cognitive statuses, and adding constraints which enforce the criteria in our cognitive status model.

Our formulation uses a simplified version of the criteria specified in the GH coding protocol [18], including only what is applicable in the instruction-giving task described in Section 3.1. The exact criteria we use are shown in Table 4.1.

Table 4.1 Cognitive Status Criteria

Cognitive Status	**Condition**
In Focus	Topic of previous utterance
Activated	Referenced in previous two utterances
Familiar	Referenced in any previous utterance
Uniquely identifiable	Always

Table reused with modifications from [1]. © 2022 IEEE.

Note that "topic" in Table 4.1 encompasses the linguistically distinct terms *subject*, *syntactic topic*, and *syntactic focus*. We leave the specification of an utterance's topic to the planning domain (see Section 5.2) so the planning formulation is independent of these distinctions. In the example situated document planning problem given in Chapter 5, we use the special parameter name `?topic` in the PDDL domain to specify a speech action's topic. In contrast, "reference" means that the object in question is referenced in any way. Thus all object parameters of a planning action are referenced by that action.

Definition 2 (Topic and Reference) *Given a situated document planning instance* $(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{I}, \mathcal{G}, T)$*:*

- $\text{topic}(o) \subseteq \mathcal{A}$ *is the set of actions* $a \in A$ *with topic* $o \in \mathcal{O}$
- $\text{ref}(o) \subseteq \mathcal{A}$ *is the set of actions* $a \in A$ *that reference* $o \in \mathcal{O}$

Definition 2 defines two sets of actions used in our MIP formulation, which can be derived from the PDDL task ontology.

In this work, we assume that every object is at least *uniquely identifiable*, which forms the implicit default if none of the other criteria are met. This assumption holds true in the situated context of our sample problem: a workspace with a collection of distinct objects. The presented formulation can easily be extended to include the omitted statuses for applications where they are applicable.

We encode the cognitive status model directly as MIP rather than including it at the planning level due to practical considerations regarding representation. The key challenge lies in grounding actions that affect cognitive status. Grounding is the process of converting from a first order representation (e.g. PDDL, with actions and predicates that are functions over objects) to a propositional logic (e.g. SAT, with only symbolic variables). The size of the grounded representation significantly impacts running time [24, 27, 30]. A naïve representation of cognitive status in planning results in a large number of constraints in the grounded representation. The number of grounded actions is exponential in action arity: a single action requires $T|\mathcal{O}|^n$ grounded variables where T is the number of time steps, $\mathcal{O}$ is the set of objects, and n is the action's arity. Typical planning problems have low-arity actions that only affect a small number of objects. However, GH-theoretic cognitive status is different: the cognitive status of an object may change at each time step even if it is not referred to at that time step. For example, if object A is *in focus*, then an utterance referencing only object B is used, the cognitive status of A changes to *activated*. Thus, accounting for cognitive status requires each grounded action to update a number of state variables proportional to the number of objects, resulting in an additional $O(T|\mathcal{O}|^{|\mathcal{O}|})$ constraints per action.

We address the challenge of concisely modeling cognitive status updates with a formulation of MIP constraints that capture GH-theoretic structure. Our formulation introduces only a $O(T|\mathcal{O}|)$ new constraints, instead of the exponential number of constraints required for a naïve encoding.

Our MIP encoding of cognitive status introduces the variables $I_{o,t}$, $A_{o,t}$, and $F_{o,t}$, which appear in lines (4.1) to (4.3) of Formulation 2. These variables respectively represent whether object o is in focus, activated, and familiar at time step t. Table 4.2 shows the values that these variables should hold in order to be consistent with a given value of cognitive status. Recall that cognitive status are hierarchically nested, and thus an object that is in focus is also activated, and so on.

In order to enforce our cognitive status model, we introduce constraints 4.5 through 4.9 of Formulation 2. These constraints ensure that values assigned to the cognitive status variables are consistent with the criteria in Table 4.1. Constraint 4.10 encodes the initial condition for the cognitive status variables: at the start of the interaction, each object is at the default cognitive status (uniquely identifiable).

Table 4.2 Cognitive Status Variable Values

Status Values	*In Focus*	*Activated*	*Familiar*	*Uniquely Identifiable*
$I_{o,t}$	1	0	0	0
$A_{o,t}$	1	1	0	0
$F_{o,t}$	1	1	1	0

4.2 GH-informed Objective Function

The new cognitive status variables allow us to optimize our document planning according to the cognitive status of referenced objects. We provide an example objective function to demonstrate how our approach can be used enable the use of the linguistic forms associated with higher cognitive statuses—i.e., closer to *in focus* in the hierarchy. Therefore, we introduce a cost for each object reference depending on its cognitive status at the time of the reference. Higher statuses have lower costs, encouraging document plans that support the use of concise referring forms. The costs we used in our implementation are shown in Table 4.3. The objective function (4.4) of Formulation 2 captures these costs.

Table 4.3 Cognitive Status Costs

Status	*In Focus*	*Activated*	*Familiar*	*Uniquely Identifiable*
Cost	1	2	4	8

As this implementation is meant to serve as a proof-of-concept, we did not investigate the impact of different sets of costs. In our evaluation presented in Chapter 5, we found that our approach had the intended effect on document planning even with these naïvely chosen costs. Ideas for future work related to the objective function are discussed in Chapter 7.

4.3 MIP Formulation of GH-informed Document Planning

Our complete MIP formulation of GH-informed document planning is given in Formulation 2. This is an extension of the classical planning given in Formulation 1. Variables and constraints without a line number in Formulation 2 are unchanged from the classical formulation.

The new variables on lines (4.1) through (4.3) and constraints (4.5) through (4.9) model cognitive status as discussed in discussed in Section 4.1. The objective function (4.4) replaces the objective function used in Formulation 1, and encodes the reference costs discussed in Section 4.2. Aside from the replaced objective function, nothing from Formulation 1 is removed in Formulation 2.

Formulation 2 *Given a situated document planning instance* $(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mathcal{I}, \mathcal{G}, T)$

Decision Variables

$$p_t \in \{0,1\}, \quad \forall p \in \mathcal{P},\ t \in [0,T] \qquad \textit{(1 iff predicate p is true at time step t)}$$

$$a_t \in \{0,1\}, \quad \forall a \in \mathcal{A},\ t \in [1,T] \qquad \textit{(1 iff action a is taken at time step t)}$$

$$I_{o,t} \in \{0,1\}, \quad \forall o \in \mathcal{O},\ t \in [0,T] \qquad \textit{(1 iff object o is in focus at time step t)} \tag{4.1}$$

$$A_{o,t} \in \{0,1\}, \quad \forall o \in \mathcal{O},\ t \in [0,T] \qquad \textit{(1 iff object o is activated at time step t)} \tag{4.2}$$

$$F_{o,t} \in \{0,1\}, \quad \forall o \in \mathcal{O},\ t \in [0,T] \qquad \textit{(1 iff object o is familiar at time step t)} \tag{4.3}$$

Objective: *minimize reference costs*

$$\min \sum_{t \in [1,T]} \sum_{o \in O} \sum_{a \in \mathrm{ref}(o)} a_t(8 - 4F_{o,t} - 2A_{o,t} - I_{o,t}) \tag{4.4}$$

Subject to:

$$p_0 = 1, \quad \forall p \in \mathcal{I}$$

$$p_0 = 0, \quad \forall p \notin \mathcal{I}$$

$$p_T = 1, \quad \forall p \in \mathcal{G}$$

$$a_t \leq p_{t-1}, \quad \forall a \in \mathcal{A}, p \in \mathrm{pre}^+(a), t \in [1,T]$$

$$a_t \leq 1 - p_{t-1}, \quad \forall a \in \mathcal{A}, p \in \mathrm{pre}^-(a), t \in [1,T]$$

$$a_t \leq p_t, \quad \forall a \in \mathcal{A}, p \in \mathrm{eff}^+(a), t \in [1,T]$$

$$a_t \leq 1 - p_t, \quad \forall a \in \mathcal{A}, p \in \mathrm{eff}^-(a), t \in [1,T]$$

$$\sum_{a \in A} a_t \leq 1, \quad \forall t \in [1,T]$$

$$p_t \leq p_{t-1} + \sum_{a \in \mathrm{add}(p)} a_t, \quad \forall p \in \mathcal{P}, t \in [1,T]$$

$$p_t \geq p_{t-1} - \sum_{a \in \mathrm{del}(p)} a_t, \quad \forall p \in \mathcal{P}, t \in [1,T]$$

$$I_{o,t} = \sum_{\mathrm{topic}(o)} a_t, \quad \forall o \in \mathcal{O},\ t \in [1,T] \tag{4.5}$$

$$A_{o,t} \leq \sum_{\mathrm{ref}(o)} a_t + a_{t-1}, \quad \forall o \in \mathcal{O},\ t \in [1,T] \tag{4.6}$$

$$A_{o,t} \geq \frac{1}{2} \sum_{\mathrm{ref}(o)} a_t + a_{t-1}, \quad \forall o \in \mathcal{O},\ [1,T] \tag{4.7}$$

$$F_{o,t} \leq F_{o,t-1} + \sum_{\mathrm{ref}(o)} a_t, \quad \forall o \in \mathcal{O},\ t \in [1,T] \tag{4.8}$$

$$F_{o,t} \geq \frac{1}{2} \left(F_{o,t-1} + \sum_{\mathrm{ref}(o)} a_t \right), \quad \forall o \in \mathcal{O},\ t \in [1,T] \tag{4.9}$$

$$I_{o,0}, A_{o,0}, F_{o,0} = 0, \quad \forall o \in \mathcal{O} \tag{4.10}$$

Formulation 2 uses symbols defined in both Definition 1 (Chapter 2) and Definition 2 (Chapter 4). These definitions are collected here in Table 4.4 for the reader's reference.

Table 4.4 Symbols Used in MIP Formulation

Symbol	Definition
$\mathcal{P}$	The set of grounded predicates
$\mathcal{A}$	The set of grounded actions
$\mathcal{O}$	The set of objects
$\mathcal{I}$	The set of true predicates in the initial state
$\mathcal{G}$	The set of true predicates in the goal state
T	The maximum number of steps in a plan
$\text{pre}^-(a)$	The set of negative preconditions of $a \in A$
$\text{pre}^+(a)$	The set of positive preconditions of $a \in A$
$\text{eff}^-(a)$	The set of negative effects of $a \in A$
$\text{eff}^+(a)$	The set of positive effects of $a \in A$
$\text{del}(p)$	The set of actions $a \in A$ such that $p \in \text{eff}^-(a)$
$\text{add}(p)$	The set of actions $a \in A$ such that $p \in \text{eff}^+(a)$
$\text{topic}(o)$	The set of actions $a \in A$ with topic $o \in \mathcal{O}$
$\text{ref}(o)$	The set of actions $a \in A$ that reference $o \in \mathcal{O}$

CHAPTER 5

PLANNER EVALUATION

This chapter discusses our evaluation of the approach and formulation presented in Chapters 3 and 4. The goals of this evaluation are to: (1) verify that the system satisfies the document planning requirements and correctly models cognitive status; (2) quantify the impact of modeling cognitive status in a sample situated document planning problem; (3) assess the impact of modeling cognitive status on planner performance.

5.1 Experimental Setup

We evaluate our approach on a sample problem to identify the impact of GH-theoretic cognitive status. We solved the sample problem using a proof-of-concept document planner implemented in Python with the Gurobi Python Interface. We compared our GH-theoretic optimizing planner to a classical planner with no cognitive status model. The classical planner is equivalent to our system with the Cognitive Status Constraints and GH-Informed Objective Function removed (see Chapter 3) and using Formulation 1 rather than Formulation 2 for the MIP encoding. We compare the resulting plans, and show that GH-theoretic optimization does produce plans that use higher cognitive status referents.

```
(define (domain gadgets)
  (:predicates (screwdriver ?o) (wrench ?o) (part ?o) (box ?o) (out ?o) (in ?o ?b)
          (attached ?o1 ?o2) (gripper ?o) (wired ?o) (screwable ?o) (boltable ?o))
  (:action take-out
     :parameters (?topic ?b)
     :precondition (and (in ?topic ?b) (box ?b))
     :effect (and (out ?topic) (not (in ?topic ?b))))
  (:action screw-in
     :parameters (?topic ?p ?s)
     :precondition (and (out ?topic) (out ?p) (out ?s) (part ?topic)
                        (part ?p) (screwdriver ?s) (screwable ?topic))
     :effect (and (attached ?topic ?p)))
  (:action bolt-in
     :parameters (?topic ?p ?w)
     :precondition (and (out ?topic) (out ?p) (out ?w) (part ?topic)
                   (part ?p) (wrench ?w) (boltable ?topic))
     :effect (and (attached ?topic ?p)))
  (:action wire
     :parameters (?topic ?g)
     :precondition (and (out ?topic) (out ?g) (part ?topic) (gripper ?g))
     :effect (wired ?topic)))
```

Figure 5.1 Gadgets domain in PDDL

```
(define (problem assemble)
  (:domain gadgets)
  (:objects toolbox partbox multitool allen phillips
            motor axle gear board chip led pliers)
  (:init (box toolbox) (box partbox) (part motor) (part axle) (gripper multitool)
         (part gear) (part led) (part board) (part chip) (screwdriver multitool)
         (gripper pliers) (wrench allen) (wrench multitool) (screwdriver phillips)
         (screwable axle) (screwable chip) (boltable gear) (out motor) (out axle)
         (boltable led) (out allen) (out phillips) (out board) (out pliers)
         (out gear) (in chip partbox) (in led partbox) (in multitool toolbox))
  (:goal (and (attached chip board) (attached axle motor)
              (attached gear axle) (attached led board) (wired board))))
```

Figure 5.2 Assemble problem in PDDL

5.2 Sample Problem

We evaluate the two planners on a sample situated document planning domain *gadgets* which involves generating instructions to assemble gadgets from parts. We use the sample problem *assemble* within the gadgets domain. The PDDL specification of this domain and problem are shown in Figure 5.1 and Figure 5.2. This domain is similar to the tower construction task used in Robotics [43], and has potential utility in key HRI domains like collaborative manufacturing. However, this gadgets problem is not meant to directly represent a realistic scenario, but rather serve as an example to compare the plans generated by a classical planner and our optimizing planner. In the *gadgets* domain, there are parts, tools, and boxes. Items in boxes must be taken out before they can be used. There are three types of tools: screwdrivers, wrenches, and grippers. Certain parts can be attached to another part using a screwdriver or using a wrench, and parts can be wired using a gripping tool.

Table 5.1 Classical Encoding Plan

Planner Output	**Possible Surface Realization**
`(take-out led partbox)`	*“Take* ***the LED*** *out of* ***the box of parts****”* U[8] U[8]
`(take-out chip partbox)`	*“Take* ***the chip*** *out of* ***that****”* U[8] A[2]
`(screw-in axle motor phillips)`	*“Screw* ***the axle*** *into* ***the motor*** *with* ***the phillips screwdriver****”* U[8] U[8] U[8]
`(bolt-in gear axle allen)`	*“Bolt* ***the gear*** *onto* ***it*** *with* ***the allen wrench****”* U[8] I[1] U[8]
`(screw-in chip board phillips)`	*“Screw* ***that chip*** *into* ***the breadboard*** *with* ***that****”* F[4] U[8] A[2]
`(wire board pliers)`	*“Wire* ***that*** *with* ***the pliers****”* A[2] U[8]
`(bolt-in led board allen)`	*“Bolt* ***that LED*** *onto* ***it*** *with* ***that allen wrench****”* F[4] I[1] F[4]

Table 5.2 GH-aware Encoding Plan

Planner Output	**Possible Surface Realization**
`(take-out multitool toolbox)`	“*Take* ***the multi-tool*** *out of* ***the toolbox***” U[8] U[8]
`(screw-in axle motor multitool)`	“*Screw* ***the axle*** *into* ***the motor*** *with* ***it***” U[8] U[8] I[1]
`(bolt-in gear axle multitool)`	“*Bolt* ***the gear*** *onto* ***it*** *with* ***this tool***” U[8] I[1] A[2]
`(wire board multitool)`	“*Wire* ***the breadboard*** *with* ***that***” U[8] 2[A]
`(take-out chip partbox)`	“*Take* ***the chip*** *out of* ***the box of parts***” U[8] U[8]
`(screw-in chip board multitool)`	“*Screw* ***it*** *into* ***this board*** *with* ***that***” I[1] A[2] A[2]
`(take-out led partbox)`	“*Take* ***the LED*** *out of* ***this box***” U[8] A[2]
`(bolt-in led board multitool)`	“*Bolt* ***it*** *onto* ***this board*** *with* ***that***” I[1] A[2] A[2]

5.3 Results

For the sample problem discussed in Section 5.2, the classical encoding generated the plan in Table 5.1, and our GH-aware encoding generated the plan in Table 5.2. The left column in each table gives the document planner output in PDDL-like syntax. Although this work focuses on the document planning level, we include possible surface realizations for each plan step in the right column of each table. These natural language examples were not generated by our system; they are included to facilitate understanding and comparison of the two plans. The references in each example utterance are shown in bold. The referring form of these references are informed by cognitive status as the most prominent feature [7]. Below each reference, the cognitive status of the referent is given using the following abbreviations: I for *in focus*, A for *activated*, F for *Familiar* and U for *uniquely identifiable*. The objective function cost for the references is given in brackets.

The GH-aware plan makes several changes that result in an increased use of high-cognitive-status referents, as summarized below:

- *Object reuse:* The classical plan uses three separate tools, while the optimized plan uses only the multi-tool. This avoids the low-cognitive-status references required in switching tools.

- *Planning ahead for object reuse:* In order to access the multi-tool, the new planner must first take it out of the box. This shows how the GH-aware planning approach goes beyond greedily reusing objects.

- *Accepting longer plans:* The new planner gives a plan that is longer than the classical plan, due to the step of taking the multi-tool out. This demonstrates that our system is able to handle a trade-off between brevity and complexity. Note that the GH-aware plans are not always longer, and that this trade-off can be tuned with the objective function.

- *Separating sub-tasks:* The classical plan switches back and forth between working on the motor gadget and the chip gadget. The new planner completes work on one before starting the other to keep objects at maximum cognitive status.

Figure 5.3 shows the total references used in each plan by cognitive status. The GH-informed plan made more references at the two highest cognitive statuses, *in focus* and *activated*, and fewer at the lower statuses. This indicates that our approach could enable the concise referring forms associated with high cognitive statuses. All objects start at *uniquely identifiable*, and cognitive status only changes as the result of object references. Furthermore, there is a minimum number of unique objects that must be referenced in order to produce a valid set of instructions. This results in a large number of references at *uniquely identifiable* in both plans. However, the GH-informed planner did manage to reduce the number of unique objects referenced by one. Further improvements in this area could be achieved through diectic gesture, which is can influence cognitive status of objects non-verbally. The total costs of the classical and optimized plans according to our objective function are 100 and 90 respectively.

Table 5.3 Running Times

Classical encoding	**GH-aware Encoding**
0.59 seconds	14.12 seconds

Table reused from [1]. © 2022 IEEE.

We compare running times of our proof of concept implementation for the classical and GH-aware planners in Table 5.3. The GH-aware encoding does increase running time over the classical encoding. There are a number of potential causes for the difference, including the quadratic objective function of the GH-aware encoding and the existence of many of plans that satisfy the goal conditions. The GH-aware encoding must find not only a shortest, satisficing plan but one that optimizes the objective function (4.4). In our sample problem, the shortest, satisficing plan takes 7 steps while the optimal plan takes 8 steps. The optimal plan takes an extra step in order to avoid the high costs of introducing new referents. Note that this difference in plan length is a byproduct of our objective function choice. If minimal length plans are also desired, this could be accomplished by changing the objective function as discussed in Section 7.10. There are also several potential refinements to our implementation that would improve running time, as

Graphic reused from [1]. © 2022 IEEE.

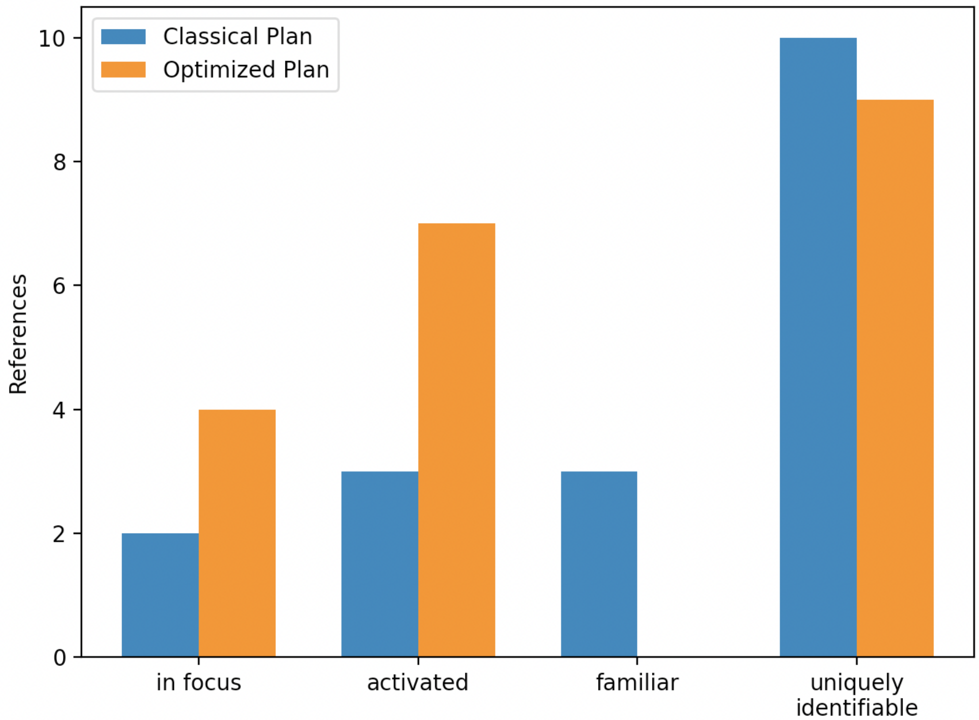

Figure 5.3 References made by cognitive status

discussed in Section 7.2. However, we believe that computational demands will not likely be a barrier to the practical use of GH-informed document planning. Our implementation is a highly unrefined proof of concept, and commercial planners can be expected to greatly outperform it.

5.4 Discussion

The plans generated by our GH-informed planner differ substantially in structure from those generated with a classical approach. These differences result in more references to high-cognitive-status objects while still accomplishing the same communicative goal. Since cognitive status is closely tied to referring form choice [7], plans generated with our approach will result in shorter language after text realization. The language may sound more natural as well, since humans tend to avoid forming complex referring expressions when possible [44]. We believe our results demonstrate the utility of the modeling cognitive status at the document planning level. There has been recent interest in computational models of cognitive status and GH-based referring form selection has been demonstrated with good results [7]. However, previous work has focused on tasks that fall under text realization. Our results are a proof of concept of how document planning can benefit from these works as well.

CHAPTER 6

HUMAN-SUBJECTS STUDY DESIGN

This chapter presents the design of a human-subjects study investigating the practical application of GH-informed document planning. Whereas the planner evaluation presented in Chapter 5 is concerned with the GH-informed document planner's performance in isolation, this human-subjects study intends to evaluate the planner's impact in real-world HRI scenario. Results from this study will be an indication of potential of GH-informed document planning for practical HRI applications.

6.1 Experimental Setup

Study subjects will participate in a human-robot collaborative task similar to the hypothetical *gadgets* problem given in Chapter 5. This allows us to investigate whether the observed differences in the document plans correspond to changes in objective and subjective measures of the interaction. Subjects will sit at a table with an assortment of objects and a speech-capable robot. They will be instructed to follow the instructions given by the robot. Robots will then deliver the instructions without otherwise interacting with the scene. In order to evaluate a variety of situations, each subject will perform three tasks of varying difficulty and intuitiveness. The experiment will use three conditions, with the robot's instructions varying between conditions. In order to account for differences in the tasks, as well as familiarity and attention factors associated with the order of the conditions, we use a 3-way Graeco-Latin counterbalanced within-subjects design. The Graeco-Latin square for this setup is shown in Table 6.1. Each cell in the table gives a task-condition combination, with the tasks represented by the letters { A, B, C } and the conditions represented by the numbers { 1, 2, 3 }. Participants will be divided into three groups, with each group receiving the pairings from a single row in the table. After each task, subjects will be given a survey regarding their perception of the interaction.

Table 6.1 Graeco-Latin Square for Tasks and Conditions

A1	B3	C2
B2	C1	A3
C3	A2	B1

6.2 Domain

We formalize the robot's task of generating instructions for this experiment as a Situated Document Planning domain called *assembly*. It is similar to the *gadgets* domain used in our evaluation in Chapter 5. However, we made minor changes for practical considerations of our experimental setup. For example, we used more general actions such as `attach` rather than specific methods such as `screw-in` in order to enable testing a wider variety of manipulation tasks. The domain contains four types of instructions that the robot can give to a human: putting an object into or taking an object out of a container, attaching two objects together, and fastening two objects together using a tool. The PDDL specification of the *assembly* domain is given in Appendix A.

6.3 Problems

We designed three experiment tasks within the *assembly* domain. The PDDL specifications of the problems are given in A. The first problem, *fan*, involves the assembly of an electric fan using an Elenco Snap Circuits Junior kit. The kit allows for many configurations, so subjects need to understand and follow the robot's instructions in order to correctly assemble it. However, the assembly itself is quite easy as this kit is designed as a children's toy. The task is expected to be moderately intuitive to the subjects, as it uses widely familiar objects such as batteries, but also requires actions that subjects likely have no experience with (i.e. connecting proprietary Snap Circuits Junior parts). The second task, *supplies*, involves preparing an assortment of school supplies, such as placing pencils inside a case. This task is both easy and intuitive, as it involves everyday objects and actions which participants have likely taken many times. The last task, *booktand* involves assembling a book stand from 8020 extruded aluminum parts. It is moderately difficult as it requires the use of multiple tools. It is also quite unintuitive as the parts are generic and used in a wide variety of constructions, so participants will likely not be able to predict how they are intended to fit together.

6.4 Conditions

The three conditions in the experiment are: classical document planning, GH-informed document planning with deterministic cognitive status model, and GH-informed document planning with probabilistic status model. The first condition uses the classical planner from Chapter 5 (without a model of cognitive status), and the latter two use the GH-informed planner. The two GH-informed approaches will use the same document planner, but differ in the cognitive status model used for referring form selection after the plans are generated.

6.5 Document Plans

We present the document plans to be used in the proposed study in Appendix B. For each experiment task, we provide document plans from the classical approach and our GH-informed approach. These document plans will be refined to realized text using established methods.

6.6 Dependent Measures

We will collect two objective and three subjective measures. The first objective measure is the total time spent to complete (or attempting to complete) the assembly task. The second objective measure is the accuracy, or the fraction of the task steps that were completed correctly. The first two subjective measures are perceived naturality and clarity of the instructions. Participants will be collected from Likert scales of "Natural-sounding" to "Unnatural-sounding" and "Clear" to "Unclear", respectively. The last subjective measure is the NASA Task Load Index (TLX) [45], which measures subjective workload. In particular, we are interested in the metal effort scale of TLX.

6.7 Hypotheses and Research Questions

We hypothesize that the GH-informed conditions will be more natural and clear than the classical condition. The increased use of low-cognitive-status referents allows for a greater variety of referring forms, and more anaphora in the instructions. These qualities are characteristic of human speech. Moreover, we noted in our original evaluation that the GH-informed document plans seemed to order instructions in a more intuitive way. We further hypothesize that GH-informed conditions will require less mental effort. If the instructions are more natural, interpreting them should be easier, lowering the cognitive load of listening to the robot. Between the two GH-informed conditions, we expect the probabilistic model to slightly outperform the deterministic model, similar to the difference between Pal et al.'s Cognitive Status Filter and Finite State Machine models [6]. Finally we have the following research question: how does the use of GH-informed document planning impact accuracy in the tasks? There seem to be opposing factors that could impact accuracy. If the GH-informed conditions have clearer instructions and more intuitive instructions, one would expect accuracy to increase. On the other hand, increased use of anaphora also makes speech more ambiguous, potentially leading to misinterpretations. The inherent difficulty of the task could also make a difference; perhaps ambiguous language is sufficient for easier tasks but insufficient for harder tasks.

CHAPTER 7
FUTURE WORK

In this work, we introduced a novel Document Planning approach, implemented a proof-of-concept, and conducted preliminary analysis on its performance. While we are optimistic on the potential of GH-informed document planning, and there are still many opportunities for further development.

7.1 Human-Subjects Study

Our first line of future work is to conduct the Human-Subjects Study discussed in Chapter 6. The results of this study should serve as a strong indicator of the potential of GH-informed document planning for practical use in HRI applications. If our hypotheses are supported by the data, the study will also motivate the exploration of other ideas for future work discussed in this chapter.

7.2 Performance

There are numerous refinements that could improve the performance of our implementation. Typed PDDL would greatly reduce the size of the grounded problem by avoiding grounding actions that don't "make sense". For example, in the sample *gadgets* problem, our current implementation grounds (and thus creates a MIP variable and constraints for) actions such as "screw the pliers into the toolbox with the motor". Object types would allow us to restrict the objects which can be selected for each action parameter, and reduce the size of the encoded problem. The specification of the *assembly* domain for the proposed human study, given in Appendix A, uses typed PDDL. However, updating our GH-informed document planner to take advantage of this typed representation is still in progress. Parallel actions [24] would also improve performance significantly, be reducing the length of the optimal plan. Parallel actions allow multiple actions to be taken in the same time step, as long as they do not interfere with one another. The parallel plans can then be linearized efficiently as a post-processing step. Further heuristics to prune satisficing but nonoptimal solutions may also yield improvements.

7.3 Cognitive Status Model

We used the Givenness Hierarchy Coding Protocol as our cognitive status model, which is simple and deterministic. However, the literature suggests probabilistic models may be more accurate [7]. Thus, modifying the cognitive status model used for GH-informed document planning could improve its performance in practice.

7.4 Modeling Additional Factors

For this work, we considered only cognitive status; but how speakers choose to refer to objects depends on a myriad of other situational values that could also be modeled. For example, referring form selection also depends on the physical distance between speaker and referent, as well as the presence of distractors [7]. Much of this information could be derived from planner state in our current implementation. Our use of MIP as the underlying constraint solver creates the flexibility for tracking attributes such as the real-valued position of objects as well. Our document planner's model of cognitive status could be improved by incorporating these features.

7.5 Referential Assumptions

The use of the multi-tool in the sample problem in Chapter 5 suggests another potential line of investigation. The planner uses the tool so consistently that it remains *activated* from the first time it is referenced to the end of the plan. In such a situation, a human instructor would likely stop referring to it altogether, assuming that the instruction follower will continue using the same tool if unspecified. This leads to the notion of leveraging referential assumptions at the document planning level. In a similar vein to how ambiguous anaphora are appropriate when the referent is at a high cognitive status, we could attempt to model when a certain idea is so given that it needs not even be stated.

7.6 Nonverbal Communication

Nonverbal communication is prevalent in human conversation, and it can influence cognitive status as well. In particular, diectic gesture (i.e. pointing to something that you are talking about) can completely change an object's cognitive status. Gestures can occur at any time, and thus hapen "between" time steps in our current formulation. This could be addressed by adding non-verbal communicative actions to the situated document planning domain. This would enable a robot to generate document plans that include gestures, which could lead to robots with more effective communication abilities.

7.7 Non-Communicative Robot Actions

Similar to the idea of adding gesture capabilities, it would be possible to include completely non-communicative robot actions to the document planning problem. This could be useful in domains such as collaborative manipulation tasks, where a pool of subtasks can be divided between the robot and the human. In such a scenario, the robot's decisions on physical actions impact the document planning problem (and thus cognitive status considerations) by changing what the robot needs to communicate. Even more generally, the document planning problem could be subsumed as a subtask within overall robot planning, allowing the robot to be cognizant of how all of its actions impact the quality of the language it will need to generate.

7.8 Iterative Planning

In real life, things do not always go as planned. For example, the robot might receive real-time feedback on its communication, or perceive that a human has made a mistake in following instructions. Iterative planning could enable our GH-informed document planner to recover from unexpected situations.

7.9 Larger Groups

This work assumed a single human listener. However, a real robot may need to model multiple listeners who come and go. A single, global model of cognitive status is not sufficient when listeners have different cognitive contexts. Furthermore, the robot would need a means to select referring forms when a referent is at a different cognitive status in the minds of different listeners (e.g. a new person joins a conversation). These considerations could motivate an extension to our approach for a larger, dynamically changing human audience.

7.10 Objective Function

Our proof-of-concept implementation used a simple objective function that penalizes references to higher cognitive statuses. There is a lot of room to improve on this objective function. Only considering cognitive status, the objective function could be improved based on the linguistic literature on, e.g., cost-of-comprehension or other psycholinguistically relevant criteria [8]. Furthermore, other factors could be added to the objective function, such as number of steps, expected time to completion, expected difficulty, and monetary cost. Perhaps multi-objective optimization could be used to intelligently balance other concerns with cognitive status related benefits.

CHAPTER 8
CONCLUSION

In this work, we propose modeling GH-theoretic cognitive status within document planning for situated human-robot interaction. We present a proof of concept by encoding the GH coding criteria in a MIP-based planner and demonstrate the solution to a sample instruction-giving problem. Our MIP encoding captures the structure of GH-theoretic cognitive status to form a concise set of constraints. Our resulting plans indicate the utility of our approach and motivate further investigation in GH-aware, situated document planning.

We also present the design of a human-subjects study on the generated language. Carrying out this experiment will provide valuable insight into how GH-informed document planning could be used in practical applications. There are many promising areas for future work, including generalizing the planner to non-speech actions such as gesture and developing more informed objective functions. This work could serve as a framework for human-robot collaboration systems that exhibit complex decision-making behavior that maximizes human understanding of natural language prompts and reduces human cognitive load.

REFERENCES

[1] Kevin Spevak, Zhao Han, Tom Williams, and Neil T Dantam. Givenness hierarchy informed optimal document planning for situated human-robot interaction. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[2] Jeanette K Gundel, Nancy Hedberg, and Ron Zacharski. Cognitive status and the form of referring expressions in discourse. *Language*, pages 274–307, 1993.

[3] Jeanette K Gundel, Mamadou Bassene, Bryan Gordon, Linda Humnick, and Amel Khalfaoui. Testing predictions of the givenness hierarchy framework: A crosslinguistic investigation. *Journal of Pragmatics*, 42(7):1770–1785, 2010.

[4] Tom Williams, Saurav Acharya, Stephanie Schreitter, and Matthias Scheutz. Situated open world reference resolution for human-robot dialogue. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 311–318. IEEE, 2016.

[5] Tom Williams. A givenness hierarchy theoretic approach. *The Oxford handbook of reference*, page 457, 2019.

[6] Poulomi Pal, Lixiao Zhu, Andrea Golden-Lasher, Akshay Swaminathan, and Tom Williams. Givenness hierarchy theoretic cognitive status filtering. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2020.

[7] Poulomi Pal, Grace Clark, and Tom Williams. Givenness hierarchy theoretic referential choice in situated contexts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43, 2021.

[8] Jennifer E Arnold and Sandra A Zerkle. Why do people produce pronouns? pragmatic selection vs. rational models. *Language, Cognition and Neuroscience*, 34(9):1152–1175, 2019.

[9] Harry Tily and Steven Piantadosi. Refer efficiently: Use less informative expressions for more predictable meanings. In *Proceedings of the workshop on the production of referring expressions: Bridging the gap between computational and empirical approaches to reference*. Citeseer, 2009.

[10] E Williams and JE Arnold. Priming discourse structure guides pronoun comprehension. In *Poster, CUNY conference on human sentence processing, University of Colorado*, 2019.

[11] Herbert P Grice. Logic and conversation. In *Speech acts*, pages 41–58. Brill, 1975.

[12] Stephen C Levinson. Deixis. In *The handbook of pragmatics*, pages 97–121. Blackwell, 2004.

[13] Christopher D Wallbridge, Alex Smith, Manuel Giuliani, Chris Melhuish, Tony Belpaeme, and Séverin Lemaignan. The effectiveness of dynamically processed incremental descriptions in human robot interaction. *ACM Transactions on Human-Robot Interaction (THRI)*, 11(1):1–24, 2021.

[14] Zhao Han, Adam Norton, Eric McCann, Lisa Baraniecki, Will Ober, Dave Shane, Anna Skinner, and Holly A Yanco. Investigation of multiple resource theory design principles on robot teleoperation and workload management. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3858–3864. IEEE, 2021.

[15] Bin Xie and Gavriel Salvendy. Review and reappraisal of modelling and predicting mental workload in single-and multi-task environments. *Work & stress*, 14(1):74–99, 2000.

[16] Ellen Campana, Michael K Tanenhaus, James F Allen, and R Remington. Natural discourse reference generation reduces cognitive load in spoken systems. *Natural Language Engineering*, 17(3):311–329, 2011.

[17] Britt Hadar, Joshua E Skrzypek, Arthur Wingfield, and Boaz M Ben-David. Working memory load affects processing time in spoken word recognition: Evidence from eye-movements. *Frontiers in neuroscience*, 10:221, 2016.

[18] Jeanette K Gundel, Nancy Hedberg, Ron Zacharski, Ann Mulkern, Tonya Custis, Bonnie Swierzbin, Amel Khalfoui, Linda Humnick, Bryan Gordon, Mamadou Bassene, et al. Coding protocol for statuses on the givenness hierarchy. *Unpublished manuscript (1993/2006). http://www. sfu. ca/hedberg/Coding_ for_Cognitive_Status. pdf*, 2006.

[19] Andrew Kehler. Cognitive status and form of reference in multimodal human-computer interaction. In *AAAI/IAAI*, pages 685–690, 2000.

[20] Joyce Y Chai, Pengyu Hong, and Michelle X Zhou. A probabilistic approach to reference resolution in multimodal user interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 70–77, 2004.

[21] Ehud Reiter. Pipelines and size constraints. *Computational Linguistics*, 26(2):251–259, 2000.

[22] Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170, 2018.

[23] David D McDonald. Issues in the choice of a source for natural language generation. *Computational Linguistics*, 19(1):191–197, 1993.

[24] Jussi Rintanen. Engineering efficient planners with SAT. In *European Conference on Artificial Intelligence (ECAI)*, pages 684–689, 2012.

[25] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[26] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

[27] Henry A. Kautz and Bart Selman. Blackbox: A new approach to the application of theorem proving to problem solving. In *AIPS98 Workshop on Planning as Combinatorial Search*, pages 58–60, 1998.

[28] Jussi Rintanen. Madagascar: Scalable planning with SAT. In *8th International Planning Competition (IPC-2014)*, pages 66–70, 2014.

[29] Mauro Vallati, Lukas Chrpa, Marek Grześ, Thomas Leo McCluskey, Mark Roberts, Scott Sanner, et al. The 2014 international planning competition: Progress and trends. *Ai Magazine*, 36(3):90–98, 2015.

[30] Neil T. Dantam, Zachary K. Kingston, Swarat Chaudhuri, and Lydia E. Kavraki. An incremental constraint-based framework for task and motion planning. *The International Journal of Robotics Research*, 37(10):1134–1151, 2018.

[31] Douglas Appelt. *Planning english sentences*. Cambridge University Press, 1985.

[32] Alexander Koller and Matthew Stone. Sentence generation as a planning problem. 2007.

[33] Aravind K Joshi and Yves Schabes. Tree-adjoining grammars. In *Handbook of formal languages*, pages 69–123. Springer, 1997.

[34] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. *PDDL – the planning domain definition language*. AIPS-98 Planning Competition Committee, 1998.

[35] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[36] Menkes Van Den Briel, Thomas Vossen, and Subbarao Kambhampati. Reviving integer programming approaches for ai planning: A branch-and-cut framework. In *ICAPS*, pages 310–319, 2005.

[37] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2022. URL http://www.gurobi.com.

[38] Nikolaj Bjørner, Anh-Dung Phan, and Lars Fleckenstein. νz-an optimizing SMT solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 15, pages 194–199, 2015.

[39] Laurent Perron. Operations research and constraint programming at Google. In *International Conference on Principles and Practice of Constraint Programming*, pages 2–2. Springer, 2011.

[40] Matthias Scheutz, Thomas Williams, Evan Krause, Bradley Oosterveld, Vasanth Sarathy, and Tyler Frasca. An overview of the distributed integrated cognition affect and reflection DIARC architecture. In *Cog. Arch.* 2019.

[41] Tom Williams, Gordon Briggs, Bradley Oosterveld, and Matthias Scheutz. Going beyond command-based instructions: Extending robotic natural language interaction capabilities. In *Proc. AAAI*, 2015.

[42] Jussi Rintanen. Impact of modeling languages on the theory and practice in planning research. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[43] Malte F Jung, Dominic DiFranzo, Solace Shen, Brett Stoll, Houston Claure, and Austin Lawrence. Robot-assisted tower construction—a method to study the impact of a robot's allocation behavior on interpersonal dynamics and collaboration in groups. *ACM Transactions on Human-Robot Interaction (THRI)*, 10(1):1–23, 2020.

[44] Donna Byron, Alexander Koller, Jon Oberlander, Laura Stoia, and Kristina Striegnitz. Generating instructions in virtual environments (give): A challenge and an evaluation testbed for nlg. 2007.

[45] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.

APPENDIX A

PDDL DEFINITIONS OF TASKS FOR HUMAN SUBJECTS STUDY

This appendix contains the PDDL representation of the tasks designed for the human-subjects study outlined in Chapter 6. The three tasks fall under the common domain *assembly*, given in Section A.1. Each task is encoded as a separate PDDL problem, given in Sections A.2-A.4. Unlike the domain and problem used for the evaluation in Chapter 5, these definitions are given in typed PDDL. Typing reduces the size of the grounded problem, thereby improving performance.

A.1 Assembly Domain

```
(define (domain assembly)
  (:requirements :strips)
  (:requirements :typing)

  (:types container - object
          fastener - object
          tool - object)

  (:predicates (in ?x - object ?c - container)
               (out ?x - object)
               (connected ?x - object ?y - object)
               (attachable ?x - object ?y - object)
               (fastenable ?x - object ?y - object ?t - tool))

  (:action put-in
           :parameters (?topic - object ?c - container)
           :precondition (and (out ?topic) (out ?c))
           :effect (and (not (out ?topic))
                        (in ?topic ?c)))

  (:action take-out
           :parameters (?topic - object ?c - container)
           :precondition (and (out ?c) (in ?topic ?c))
           :effect (and (out ?topic)
                        (not (in ?topic ?c))))

  (:action attach
           :parameters (?topic - object ?o - object)
           :precondition (and (out ?topic) (out ?o)
                              (not (connected ?topic ?o))
                              (attachable ?topic ?o))
           :effect (and (connected ?topic ?o)))

  (:action fasten
           :parameters (?topic - object ?o - object ?t - tool)
           :precondition (and (out ?topic) (out ?o) (out ?t)
                              (not (connected ?topic ?o))
                              (fastenable ?topic ?o ?t))
           :effect (and (connected ?topic ?o))))
```

A.2 Fan Problem

```
(define (problem fan)
  (:domain assembly)
  (:objects propeller - object
            motor - object
            lamp - object
            switch - object
            battery - object
            battery-case - container
            battery-box - container
            propeller-case - container)
  (:init (out motor)
         (out lamp)
         (out switch)
         (out battery-box)
         (out battery-case)
         (out propeller-case)
         (in battery battery-box)
         (in propeller propeller-case)
         (attachable propeller motor)
         (attachable motor lamp)
         (attachable lamp switch)
         (attachable switch battery-case))
  (:goal (and (connected propeller motor)
              (connected motor lamp)
              (connected lamp switch)
              (connected switch battery-case)
              (in battery battery-case))))
```

A.3 Supplies Problem

```
(define (problem school-supplies)
  (:domain assembly)
  (:objects pencil - object
            pencil-grip - object
            eraser - object
            pen - object
            pencil-case - container
            calculator - container
            batteries - object
            battery-box - container
            sticker - object)
  (:init (out pencil) (out pencil-grip)
         (out eraser) (out pen)
         (out pencil-case)
         (out calculator)
         (out battery-box)
         (out sticker)
         (in batteries battery-box)
         (attachable eraser pencil)
         (attachable pencil-grip pencil)
         (attachable sticker calculator))
  (:goal (and (connected eraser pencil)
              (connected pencil-grip pencil)
              (in pencil pencil-case)
              (in pen pencil-case)
              (in batteries calculator)
              (connected sticker calculator))))
```

A.4 Bookstand Problem

```
(define (problem bookstand)
  (:domain assembly)
  (:objects base - object
            post - object
            support - object
            foot - object
            pivot - object
            arm - object
            lip - object
            grip - object
            plate - object
            sm-hex - tool
            med-hex - tool
            lg-hex - tool
            m5-hex - tool
            multi-hex - tool
            toolbox - container)
  (:init (out base)
         (out post)
         (out support)
         (out foot)
         (out pivot)
         (out arm)
         (out lip)
         (out grip)
         (out plate)
         (out sm-hex)
         (out med-hex)
         (out lg-hex)
         (out m5-hex)
         (out toolbox)
         (in multi-hex toolbox)
         (fastenable post base lg-hex)
         (fastenable post base multi-hex)
         (fastenable support post med-hex)
         (fastenable support post multi-hex)
         (fastenable foot support lg-hex)
         (fastenable foot support multi-hex)
         (fastenable pivot post med-hex)
         (fastenable pivot post multi-hex)
         (fastenable arm pivot lg-hex)
         (fastenable arm pivot multi-hex)
         (fastenable lip arm m5-hex)
         (fastenable lip arm multi-hex)
         (fastenable plate lip sm-hex)
         (fastenable plate lip multi-hex)
         (attachable grip lip))
  (:goal (and (connected post base)
              (connected support post)
              (connected foot support)
              (connected pivot post)
              (connected arm pivot)
              (connected lip arm)
              (connected plate lip)
              (connected grip lip))))
```

APPENDIX B

DOCUMENT PLANS FOR HUMAN SUBJECTS STUDY

This appendix contains document plans for the three tasks in the human subjects study design presented in Chapter 6.

B.1 Classical Planner

This section provides document plans generated by a classical (GH-unaware) document planner, to be used as a baseline in the study.

Fan Problem

```
(take-out battery battery-box)
(attach motor lamp)
(put-in battery battery-case)
(attach lamp switch)
(take-out propeller propeller-case)
(attach switch battery-case)
(attach propeller motor)
```

Supplies Problem

```
(attach pencil-grip pencil)
(take-out batteries battery-box)
(put-in batteries calculator)
(attach eraser pencil)
(put-in pencil pencil-case)
(put-in pen pencil-case)
(attach sticker calculator)
```

Bookstand Problem

```
(fasten lip arm m5-hex)
(fasten arm pivot lg-hex)
(fasten pivot post med-hex)
(fasten post base lg-hex)
(fasten foot support lg-hex)
(fasten support post med-hex)
(fasten plate lip sm-hex)
(attach grip lip)
```

B.2 GH-informed Planner

This section provides document plans generated by our GH-informed document planner.

Fan Problem

```
(take-out battery battery-box)
(put-in battery battery-case)
(attach switch battery-case)
(attach lamp switch)
```

```
(attach motor lamp)
(take-out propeller propeller-case)
(attach propeller motor)
```

Supplies Problem

```
(attach sticker calculator)
(take-out batteries battery-box)
(put-in batteries calculator)
(attach eraser pencil)
(attach pencil-grip pencil)
(put-in pen pencil-case)
(put-in pencil pencil-case)
```

Bookstand Problem

```
(take-out multi-hex toolbox)
(fasten post base multi-hex)
(fasten support post multi-hex)
(fasten foot support multi-hex)
(fasten pivot post multi-hex)
(fasten arm pivot multi-hex)
(fasten lip arm multi-hex)
(fasten plate lip multi-hex)
(attach grip lip)
```

APPENDIX C
COPYRIGHT PERMISSIONS

This thesis is an extension of a work that has previously been published:

> Kevin Spevak, Zhao Han, Tom Williams, and Neil T. Dantam. "Givenness Hierarchy Informed Optimal Document Planning for Situated Human-Robot Interaction." In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6109-6115.
>

Content from the original publication has be reused for this thesis, with permission from the coauthors and according to the IEEE reuse policies. Reused content includes Figures, Tables, and segments of text.

C.1 Coauthor Permission

This thesis was produced by Kevin Spevak, one of the authors of the original publication. The three other coauthors all gave permission to reuse content for this thesis. Email correspondences between Kevin Spevak and the coauthors, which indicate permission, are provided on ProQuest as supplemental files under the names han.eml, williams.eml, and dantam.eml.

C.2 IEEE Reuse Policy

The IEEE policy does not require formal permission to reuse copyrighted content for a thesis or dissertation, provided that certain requirements are met. Two of the requirements are applicable for this thesis.

- As this thesis uses text excerpts from the publication, it must provide the full citation, followed by the IEEE copyright line. This requirement is fulfilled by the citation at the beginning of this appendix.
- As this thesis uses graphics and tables from the publication, it must include the IEEE copyright line within each reprinted graphic and table. Each graphic and table that is reused in this thesis (with or without modification) complies with this requirement.