KNOWLEDGE-BASED SPATIAL REASONING FOR AUTOMATED SCENE GENERATION FROM TEXT DESCRIPTIONS

BY

DANIEL ALLEN TAPPAN, B.A., M.S.E.

A dissertation submitted to the Graduate School

in partial fulfillment of the requirements

for the degree

Doctor of Philosophy

Major Subject: Computer Science

New Mexico State University

Las Cruces, New Mexico

May 2004

Copyright 2004 by Daniel Allen Tappan

"Knowledge-Based Spatial Reasoning for Automated Scene Generation from Text Descriptions," a dissertation prepared by Daniel Allen Tappan in partial fulfillment of the requirements for the degree, Doctor of Philosophy, has been approved and accepted by the following:

Linda Lacey
Dean of the Graduate School

Roger T. Hartley
Chair of the Examining Committee

Date

Committee in charge:

Dr. Roger T. Hartley, Chair

Dr. Kerry J. Alt

Dr. Peter W. Foltz

Dr. Son C. Tran

ABSTRACT

KNOWLEDGE-BASED SPATIAL REASONING FOR

AUTOMATED SCENE GENERATION

FROM TEXT DESCRIPTIONS

BY

DANIEL ALLEN TAPPAN, B.A., M.S.E

Doctor of Philosophy

New Mexico State University

Las Cruces, New Mexico, May 2004

Dr. Roger T. Hartley, Chair

Understanding text is a trivial task for literate humans. For computers, however, it is extremely difficult due to (among other reasons) a lack of knowledge about language and the world, as well as an intelligent reasoning mechanism to process such resources. As a result, computational approaches to text understanding generally lack common sense and suffer from poor performance. This work presents a system that addresses a set of critical cognitive, linguistic, and computational issues. On the cognitive level, it considers the role of mental models in the mind and how humans internally abstract and conceptualize spatial characteristics of the external world. On the linguistic level, it considers the roles of underspecification, vagueness, uncertainty, context, and frame of reference in how humans communicate about space. On the computational level, it implements a constraint-based, declarative knowledge representation for qualitative spatial reasoning over the dimensions, positions, and orientations of representative objects (primarily animals and plants) in a simulated microworld of a zoo environment. This system extracts into a semantic network the explicit information in rudimentary text descriptions of static, spatial scenes, integrates it with implicit, background information from an object-oriented, commonsense knowledge base, reasons over the combined representation, and renders a set of corresponding graphical interpretations. From these depictions, it extracts new information that iteratively feeds back into the original description to augment the understanding further. As part of a Monte Carlo simulation, the architecture supports a multidimensional test-and-evaluation framework to investigate a variety of related issues that apply to many applications in artificial intelligence.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

"Picture yourself on a boat in a river" [76, 23]. This famous line from the Beatles' song *Strawberry Fields Forever*, like most descriptions, leads the reader to form an abstract picture of the scene in his or her mind. This mental representation plays an essential role in how humans understand and manipulate the meaning of a description. For humans, the task seems trivial. But what if a person did not know the simple facts about what a boat and river are or what it means to be "on" and "in" one, respectively? For computers, this is exactly the predicament. This lack of knowledge and the ability to reason over it intelligently greatly hinders their performance in almost all areas of processing human language.

This project investigates several key issues in computational text understanding and demonstrates a unified approach toward solving them. Its primary goal is to translate text descriptions into corresponding graphical renderings for simple scenarios that could be found in a typical zoo. Its secondary goal is to infer simple, unstated information from the underlying representation of the renderings as humans might; e.g., the boat is in the middle of the river, on its surface, and facing north. A common framework exists for addressing each goal as well as for analyzing various contributing factors.

1.1 Problem Statement

Language processing, whether by computers or humans, is a very complex task that must resolve countless issues. Of interest in this project are five tightly intertwined issues that are believed to cause the greatest difficulty in computational text understanding: underspecification, vagueness, uncertainty, context, and frame of reference. Humans acquire the skills and knowledge to resolve these issues from a lifetime of experience interacting with and communicating about the world [82, 85]. Computers obviously lack such preparation and consequently perform poorly on language.

1.1.1 Underspecification

A description explicitly states very little information. Almost all the content must be derived implicitly by "reading between the lines" and reasoning over various types of knowledge from other sources. Two broad categories of knowledge are considered necessary for this project.

Knowledge of language is critical to decoding the grammatical framework. For example, *the big tree is in front of the dog* differs significantly in meaning from *the dog is in front of the big tree*, although the structures are nearly identical. Even parsing the words requires an understanding of how they should be interpreted:

[*the* [*big*]PROPERTY *tree*]OBJECT [*is*]IGNORE [*in front of*]RELATION [*the dog*]OBJECT

Knowledge of the world is critical to understanding what words describe. For example, the dog has an accepted front to its body, so the tree in front of it implies that it is facing the tree. On the other hand, the tree has no accepted front—it is uniformly the same all around—so the dog in front of it implies that the dog is located between it and the viewer of the description. There is also no implication of its direction.

1.1.2 Vagueness

Adjectives of size are not as straightforward as they may appear [87]. For example, *big* generally implies greater size than *small* does. This relationship indeed holds between objects of the same type (e.g., a *big duck* is greater in size than a *small duck*) but not necessarily between those of different types (e.g., a *big duck* is not greater in size than a *small giraffe* but is to a *small mouse*). Furthermore, the application of size often differs between objects. For example, a *big tree* is great in height, but a *big lake* is great in length and width. Finally, how great is a *big duck* in specific numerical terms of height, width, and length? The proper size of a rendered object depends on the correct interpretation of these adjectives.

1.1.3 Uncertainty

Exact positions and orientations cannot be determined from the imprecise details in non-technical descriptions. For example, *the dog is in the north, is facing south, and is far from the cat*. First, this statement legitimately places the dog anywhere within a large, vaguely defined upper area of the world. Second, as the lower area is equally vague, the orientation of the dog can differ considerably and still be interpreted as correct. Finally, the distance from the cat is subjective and depends somewhat on the size of the cat and what the writer might believe it perceives. These three issues in combination allow for an infinite number of plausible interpretations, although some are more favored than others [47, 118].

1.1.4 Context

The meaning of most words depends on their usage, or as Firth [39] succinctly expresses, "[y]ou shall know a word by the company it keeps." For example, the preposition *in* primarily means "contained or enclosed by" [92]. This definition holds true for both *the boat is in the lake* and *the hippo is in the lake*, but the terms of enclosure differ significantly enough (i.e., *on* versus *under* the surface) to make a noticeable difference in the interpretation and consequently the graphical rendering of each.

1.1.5 Frame of Reference

A description of the world requires the writer to commit explicitly or implicitly to a particular vantage point (position and orientation) so the reader can correspondingly orient himself or herself to reconstruct it mentally. A computational solution must orient itself as well. A simpler form of explicit commitment such as *the dog is in front of the cat as seen from the fountain* states that the viewer is located at the fountain and is facing both the dog and cat. The position of the dog depends on the position and orientation of the cat, but neither depends on the passive viewer. A more difficult form such as *the dog is in front of the tree as seen from the fountain*, states the same vantage point for the viewer. However, the position of the dog depends on the position of the tree and the position and orientation of the viewer because the tree has no true "front" of its own. An implicit commitment introduces these complexities as well, but it is even more troublesome because the description does not state the vantage point of the viewer; e.g., *the dog is in front of the tree*.

1.2 Research Value

From a theoretical perspective, text understanding is a highly interdisciplinary task that draws upon the research areas of natural language processing, machine translation, artificial intelligence, computer science, linguistics, psychology, cognitive science, and others. Any advances to it propagate to a considerable amount of related work.

From a practical perspective, text understanding has the potential to improve human-computer interaction. Humans communicate most comfortably in natural language; whereas computers demand a very unfriendly, arcane mechanism of control. Text understanding helps bridge this gap. It also supports a more effective way of indexing and retrieving documents by considering their meaning instead of their surface text. The spatial component of this project lays the groundwork for advanced search queries and question-and-answer interfaces for databases of pictures and geographical features, especially for geographical information systems [43, 35]. Finally, there is the obvious value to graphical modeling and rapid prototyping of visual scenes.

1.3 Overview of Solution

This project provides a flexible framework to investigate underspecification, vagueness, uncertainty, context, and frame of reference in computational text understanding. Each issue is addressed with respect to how it affects the interpretation and graphical rendering of the dimensions, position, and orientation of objects that are consistent with the zoo theme.

This project adheres to the philosophy of so-called *weak artificial intelligence* [106, 83]: it is primarily designed and defended as a computational system for solving a problem. Although the solution is based on research and limited observation of how humans solve similar problems, no claim is made that this solution emulates these cognitive processes. The following functional overview outlines the six main processing stages:

- Stage 1 converts the input description from English text into a simple semantic network to represent its explicit details.
- Stage 2 interprets the semantic network using a complex knowledge base to infer implicit details that augment the explicit details.
- Stage 3 spatially reasons over the combined details to produce valid dimensions, positions, and orientations for each object. From this solution, it infers new details and adds them back into the semantic network from Stage 1.
- Stage 4 collects multiple, independent solutions by repeating Stage 3 in a simulation.
- Stage 5 analyzes the set of solutions to extract a common set of interpretations.
- Stage 6 renders the common interpretations graphically.

This project is implemented in Java 1.4.1 with Blackdown Java 3D 1.3 and JavaCC 2.0 running on Red Hat Linux 8.0 The source code consists of 7 packages, 64 classes, and roughly 8,000 statements over 33,000 lines. All tests were executed on a 450MHz single-processor machine with 384MB RAM.

1.4 Overview of Discussion

This project is a complete, self-contained system for end-to-end processing. As such, it covers a broad range of topics across the stated areas of interest. The organization of this dissertation reflects the issues in Section 1.1 and the stages of processing in Section 1.3. Specifically, Chapter 2 reviews related systems and establishes a foundation of the cognitive, linguistic, and computational issues of interest. Chapter 3 covers the structure and content of the text descriptions to processes. Chapter 4 discusses the semantic network that represents the explicit information in a description. Chapter 5 complements this discussion with extensive coverage of the knowledge base that represents the implicit, commonsense, background knowledge that is not present in a description but is essential for its processing. This chapter also addresses constraints, which declaratively define all aspects of knowledge in this project. Chapter 6 steps through the process of combining the explicit and implicit representations to reason over the spatial constraints. Chapter 7 explains how this process contributes to a simulation that supports a test-and-evaluation framework. Chapter 8 presents the graphical rendering engine that converts the internal results into a collection of three-dimensional, virtual worlds for interactive inspection. Chapter 9 presents and discusses a collection of representative results, both positive and negative, and also considers future work. Chapter 10 summarizes what was accomplished and what was learned from this project. Appendix A lists the complete, annotated grammar of the knowledge representation language that was created. Appendix B lists a representative subset of the knowledge base. Finally, Appendix C shows a sample vignette, which states a description and defines the parameters that configure all components of this project.

2 BACKGROUND

Research in artificial intelligence naturally draws from many sources due to its interdisciplinary nature. In particular, this project considers three general areas that are believed to contribute most to representing and reasoning over spatial descriptions [56]:

- The cognitive foundation considers how the human mind conceptualizes the external world into its own internal representation that it uses for spatial reasoning.
- The linguistic foundation considers how humans use natural language to communicate about the external world.
- The computational foundation considers how a computer can satisfactorily emulate the essential aspects of these cognitive and linguistic processes.

2.1 Related Systems

Despite the theoretical and practical value of research in generating pictures from text, there is a surprising paucity of related systems [29, 120, 110, 126, 118, 56]: only five in the 33-year history of the area reasonably overlap with the goals and underlying issues that this project addresses. Moreover, collectively they have produced no more than a handful of publications. Most contemporary research (for which there is considerably more activity) addresses the converse process of generating text from pictures. Although many of the same issues arise in both directions, this dissertation does not address them because their focus is different.

2.1.1 SHRDLU

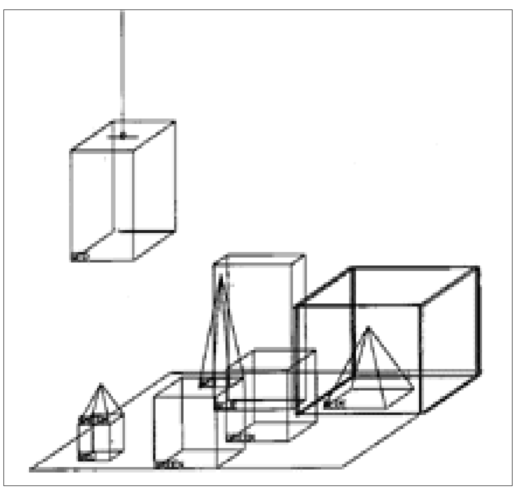
The first text-to-image system, SHRDLU,[1] appeared to great acclaim in 1971 [131]. It was a revolutionary advance in artificial intelligence and fostered claims that it could actually "understand text" [132]. On the surface, it basically extended Weizenbaum's [129] remarkably popular (but baseless[2]) Eliza dialog system from 1966, which (disturbingly) convinced more than a few people that a computer could conduct psychotherapy. SHRDLU, however, legitimately focused on a tabletop world of colored blocks, cones, and balls. It also maintained and manipulated complex (for 1971) internal representations of its world and was able to reason and communicate effectively over it in response to natural-language input for the user.

---

[1]The name SHRDLU refers to a keyboard arrangement that was common to the period. Analogously, if the same naming scheme were followed today, it would be called QWERTY.

[2]It was actually intended as a hoax! The entire program consists of 256 lines of BASIC code available at `http://hps.elte.hu/~kampis/Eliza/ELIZA.BAS`.

SHRDLU focused on grammar, semantics, and deduction. It defined a basic vocabulary of objects and properties, a general semantics for interpreting them, and an environment to which they applied. As Figure 2.1 illustrates, this work combined four simple concepts in natural language processing to great effect: a small vocabulary in a knowledge base of expected behaviors and interpretations, limited context for coreference resolution (e.g., "it" refers to the previously mentioned "red ball"), a true question-and-answer framework with remarkable linguistic freedom, and the capability to compose objects (e.g., *a steeple is a small triangle on top of a tall rectangle*).



```
> Pick up a big red block.
  Ok.

> Grasp the pyramid.
  I don't understand which pyramid
  you mean.

> What did you do that?
  Because you asked me to.

> What does the box contain?
  The blue pyramid and the blue block.

> What is the pyramid supported by?
  The box.

> How many blocks are not in the box?
  Four of them.
```

Figure 2.1: SHRDLU

Despite its groundbreaking implementation, SHRDLU did not actually understand text as its developer, Terry Winograd, originally believed [132]. The underlying mechanisms operate well over their restricted, literal domain, but they are too rigid for most aspects of language processing [114, 117, 6]. In fact, Winograd eventually considered the work a dead end and distanced himself from it. Nevertheless, other researchers have revived the fundamental idea for different applications over the years, so it was hardly a failure [16].

2.1.2 Natural Language Image Generation System

The next system to appear is the optimistically named Natural Language Image Generation System (NALIG) in 1984 [1, 2, 56]. It deserves honorable mention because its areas of interest overlap considerably with those of this project:

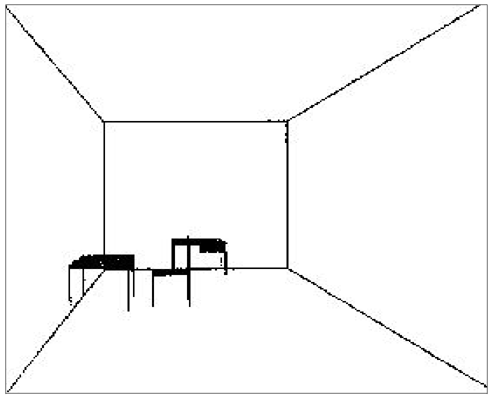
- Semantic processing for computational text understanding of natural language input.
- Rudimentary taxonomic knowledge representation of objects.
- Constraints and consistency checking between objects and relations.
- Treatment of "fuzziness"; i.e., vagueness and uncertainty.
- Commonsense, qualitative spatial reasoning over static scenes.
- Graphical rendering.
- Potential simulation capabilities.

The preliminary work for NALIG introduced these issues, but no follow-up publications ensued. In contrast to SHRDLU, there is no published explanation for abandoning the work.

2.1.3 Words Into Pictures

Words Into Pictures (WIP) followed in 1994 [94, 118, 123]. It is another ambitious system that inexplicably disappeared from the publication record after its introduction. Its researchers, however, remain prominent in the field and continue to publish significant work on the issues that WIP was supposed to address (see [93]). Several overlap with this project:

- Investigation of major cognitive aspects of natural language, especially of preferences in spatial prepositions.
- Qualitative and quantitative, probabilistic models for objects in a conceptual representation.
- Capture of inherently fuzzy meaning in spatial language.
- Introduction of fuzzy, potential fields, which are the basis of constraints in this project.
- Ambiguity in frame of reference.
- Generation of static, indoor room arrangements as Figure 2.2 shows.



A chair is in front of the left desk.

Figure 2.2: Words Into Pictures

2.1.4 CarSim

CarSim, which appeared in 2001, is an on-going project that generates a short, animated depiction for French reports of automobile accidents such as Figure 2.3 [29, 120, 107].[3] It is the only text-to-image system that uses non-contrived input. As such, the reported correctness of its interpretations is quite low at 10–17%. Unrestricted input in systems for natural language processing generally introduces many problems that overshadow their focus. In this case, the incomplete, inconsistent, ambiguous, sloppy, and sometimes incoherent form of accident reports substantially degrades performance [124]. CarSim focuses on the static and dynamic components of an accident scene; e.g., a tree and a

[3]The current version of this work now processes English reports from the public website of the U.S. National Transportation Safety Board, which is a suggestion I made to the principal investigator Pierre Nugues at the ACL2001 conference in Toulouse, France.

car, respectively. It performs linguistic analysis on the input and processes the intermediate XML representation through several stages of planning for position, trajectory, accident dynamics, and temporal aspects. The output feeds into a virtual scene generator.

I was driving on a crossroads with a slow speed, approximately 40 km/h. Vehicle B arrived from my left, ignored the priority from the right and collided with my vehicle. On the first impact, my rear fender on the left side was hit and because of the slippery road, I lost control of my vehicle and hit the metallic protection of a tree, hence a second frontal collision. [sic]

Figure 2.3: CarSim

2.1.5 WordsEye

WordsEye, which also appeared in 2001, seems on the surface to be the Holy Grail of text-to-image systems. Its broad domain of input and rich graphical output, as Figure 2.4 exemplifies, clearly outperform all related work, including this project. It is a large system that defines roughly 1,300 objects, 2,300 verbs, and 2,000 3D models (with 10,000 more planned) [116]! It also relies heavily on major, external components like Church's part of speech tagger [13], Collins' parser [20], and the WordNet lexical-semantic database [38].

Originally under the auspices of AT&T Research Labs, its developers recently spun it off as their own commercial venture, Semantic Light, LLC.

John uses the crossbow. He rides the horse by the store The store is under the large willow. The small stegosaurus is in front of the horse. The dinosaur faces John. A gigantic teacup is in front of the store. The dinosaur is in front of the horse. The gigantic mushroom is in the teacup. The castle is to the right of the store.

Figure 2.4: WordsEye

Despite its magnificent capabilities, WordsEye does not solve all problems in computational text understanding. In fact, its stated objective is actually to facilitate and expedite the generation of static, three-dimensional graphics for general-purpose applications. Natural language is an ideal medium for such work because humans manipulate it so readily and expressively. As a result, WordsEye reflects effort in linguistic processing, knowledge representation, and reasoning, but it does not do so with the intent of introspectively investigating these areas as this project does [65, 117]. In other words, this foundation is the means toward the goal but not the research area *per se*. Furthermore, in contrast to this project, WordsEye focuses primarily on the contextually appropriate depiction of poses for entities and the kinematics of implied actions

[23]; e.g., a man throwing a ball. It does so by heavily articulating its models, which this project does not support at all.

2.2 Cognitive Foundation

The cognitive foundation of this project superficially considers how humans mentally view and manipulate the external world that descriptions depict. This project, like most related work, does not focus on the psychology and cognition of text understanding, and it does not claim to rely on or advance any of this foundation [44, 83]. Therefore, this section presents only an overview. Johnson-Laird [63], Lakoff [74], Talmy [121], Langacker [75], Glasgow and Papadias [44], and Mark [83] together provide a more comprehensive review.

The cognitive aspects of this project are based on the notion that the human mind abstracts the real world into internal representations, which it manipulates as a surrogate for spatial reasoning (and many other purposes) [105, 7, 67]. While the existence of some innate mechanism to this effect is not in question, its form and function are highly contested. In fact, it fuels one of the fiercest battles in cognitive science [87]. As a result of widely differing views, the same general idea appears under many names; e.g., mental models, mental maps, mental imagery, cognitive maps, and others [15, 26]. This project stays out of the fray by separating its limited cognitive interests into two areas that reflect its computational goals with respect to its approach of weak artificial intelligence. The first resides at the physical or implementation level of the mind and corresponds to hardware in the computational sense. The brain is hardwired to perform many tasks in spatial reasoning, just as it is for language, vision, and other actions that are critical for survival [95, 121, 83]. Thus, it is not surprising that humans universally share some common set of cognitive abilities and likewise exhibit common weaknesses and deficiencies. This property is advantageous to a computational approach because certain solutions should apply uniformly across all people, languages, cultures, etc. [27]. Like universal grammar [95], however, which would serve the same ideal purpose for natural language processing, a comprehensive theory remains elusive [58, 35].

This project subscribes to the view of [43, 93, 118] that the purpose of a mental model is to simulate a behavior and thus predict and plan for its contingencies. For example, envisage a description of furniture in a room, then rearrange the pieces mentally [56, 67]. Humans do this relatively easily on multiple, possibly hierarchical levels [43, 71, 11, 35, 56]. The process accounts for fundamental constraints on the objects such as their location and orientation, but it also factors in preferences to cull the many possible mental configurations of the scene into favored ones that somehow contribute to the goal [15]. This project acknowledges both aspects as important in spatial reasoning over its domain of interpretation [4].

The second area of cognitive interest resides at the abstract or symbolic level of the mind and corresponds to software in the computational sense. The ways in which humans perceive the world and conceptualize space vary widely by language, culture, education, gender, personality, and so on, and they are incomplete, inconsistent, and often erroneous [121, 35, 8, 111, 96, 85, 26]. For example, peculiar for English speakers, some cultures use compass directions for tabletop objects like plates and utensils in place of relations like *in front of* [83]. Likewise, many people are under the impression that the east coast of the United States is a due north-south boundary [35]. Nevertheless, this confusing mix of world views does not appear to impair the spatial abilities of humans.

The richest source of research in spatial conceptualization comes from geographic information systems, which has a long history of trying to represent the real world on various forms of maps and, more recently, in computers [15]. Research by [35, 11, 32, 31] provides a comprehensive review of important considerations, and [82, 84] augment it well with a breakout of at least 18 different types of space.

2.3 Linguistic Foundation

The linguistic foundation of this project considers how humans use natural language to communicate about the external world. It focuses on the five issues from the problem statement in Section 1.1. However, because they are not disjoint and independent as presented, this discussion of their background does not have such a straightforward organization. In reality, these issues overlap and interact in ways that are still poorly understood (e.g., Sowa's "knowledge soup" [114]). As an implementation of weak artificial intelligence, this project does not attempt to advance any of this foundation. Rather, it simply uses selected aspects of a wide range of linguistic theories, principles, observations, philosophies, and so on to solve individual computational problems. In combination, they comprise no unified, linguistic approach.

Almost all computational work in the area of spatial language builds upon the foundation of Talmy [121], Herskovits [58], and Langacker [75]. Talmy's seminal work on how language structures space [121] forms an integral part of his larger work that defines the field of cognitive semantics [122]. Herskovits' thorough investigation of language and spatial cognition [58] focuses on the many and varied roles of prepositions [57]. Langacker [75] defines a comprehensive cognitive grammar that attempts to tie these and other aspects of visual language into a unified theory. Bierwisch and Lang [9] present a popular conceptual representation of physical objects. Hernández [56] provides an extensive literature review of these authors and other linguistically motivated research.

Language is always interpreted in some form of context [60, 51], which serves many complex, intricate, and subtle roles in resolving a wide range of issues, including those of interest in this project. Research into context covers a broad range of linguistic areas; e.g., lexical semantics, pragmatics, semiotics (a combination of syntax, semantics, and pragmatics), and many others [51, 52, 60, 4, 62]. To identify, represent, and emulate essential aspects of context, this project considers at a high level what it is, why it is important, and how it operates. Context in general serves as a filler, corrector, or focal mechanism to mitigate inherent deficiencies from underspecification, vagueness, and uncertainty [60, 125]. It other words, it fills in gaps between the explicit and implicit interpretations of a description. Hausser [51] presents this view as a five-level hierarchy of nine types of context that distinguishes between sincere and ironic use, literal and metaphoric use, and precise and vague use. As Chapter 3 will discuss, this project addresses only literal and vague use.

The appropriate semantic and pragmatic interpretation of linguistic components first requires the recognition that a context is present, then an identification of its effects [51, 52, 4]. Recognition is relatively straightforward in this project because contexts can exist only in the form of two objects that are bound with a spatial relation, which is roughly equivalent to two nouns joined by a preposition. Identification is far more tricky, however. Every combination of objects and prepositions has the potential to impart its own contextual skew on their combined interpretation [84]. Research into these effects often considers how specific, contextual meanings differ from ideal, generic meanings that serve as a baseline definition [58, 56]. Such work generally considers functional dependencies of behavior and interaction to establish how salience contributes to certain interpretations and the preference of one over another [59, 90, 105, 125].

Very little linguistic background focuses exclusively on underspecification, vagueness, and uncertainty. One reason is that these issues fall squarely into the realm of context and usually find themselves addressed as part of that work [114, 26, 125, 70]. Another is that, from a purely linguistic perspective, solutions to them are not particularly relevant. Cognitive science and computer science need to address them for practical reasons, but linguistics can generally assume that its subjects are fully functional humans whose theoretical ability to use language is enhanced rather than undermined by them [125, 52, 51]. In other words, underspecification, uncertainty, and vagueness are issues that linguistics acknowledges as inherent difficulties in language processing, but it does not necessarily work toward a theory to solve them. Hence, for the purposes of this project, they are primarily computational problems.

Frame of reference, which Section 1.1.5 introduced, does have a strong linguistic background to complement its cognitive and computational backgrounds [15]. The notion of a non-fixed location of the viewer in a scene varies in name (e.g., observer location [57], point of view [56, 121], vantage point [93]), but the same principles apply. In general, the speaker's intent or purpose plays the greatest role in the choice of frame of reference [57, 123, 93]. As Section 2.2 discussed, the way different languages and cultures conceptualize the world varies. English is relatively straightforward[4]—but by no means easy to process—with only three frames of reference [56, 85, 63, 15]:

- *Intrinsic*: an object-centered perspective in which a natural, inherent face of an object defines the region canonically understood to be its front;

[4]Cora, a language in Mexico, partitions its world into at least 137 forms [8]!

e.g., a person, a dog, and a car can support such an interpretation; whereas a tree and a lake cannot. A definite front implies a back, left, and right as well.

- *Extrinsic*: an environment-centered perspective in which a transient location defines a contextual vantage point; e.g., the "front" region of a car may actually be behind it if it is moving backwards. As this project forbids motion, this frame of reference does not apply.
- *Deictic*: a viewer-centered perspective in which the viewer serves as a reference object. For example, in *the dog is in front of the tree*, the position of the dog is between the tree and the implicit position of viewer. A second form situates the viewer explicitly; e.g., *the dog is in front of the tree as seen from the car*.

2.4 Computational Foundation

The computational foundation of this project considers how a computer can satisfactorily emulate the essential aspects of the cognitive and linguistic issues of interest with respect to its goals. As the areas of artificial intelligence, knowledge representation, spatial reasoning, and natural language processing are far too large and interdependent to cover in any detail, this section reviews only the computational background that directly contributes to or influences the solutions in this project. Davis [26], Hernández [56], Hernández and Mukerjee [54], Rashid and Shariff [108], Srihari [118], Oliver and Tsujii [93], and Davis, Schrobe, and Szolovits [27] together provide a well-rounded review of a wider range of related areas. Section 5.3 provides additional details as well.

For the purposes of this project, computation in artificial intelligence focuses on the tightly intertwined issues of what to represent, how to represent it, and how to reason over it [27]. Contrary to what its name implies, however, most work in artificial intelligence places relatively little emphasis on the cognitive aspects of the areas that Section 2.2 outlined [44]. This project takes a similar position that the cognitive foundation lays an important foundation and therefore must be considered and acknowledged, but its mechanisms and theories do not necessarily translate into a computational solution [26]. This project diverges, however, from the claim of Davis, Schrobe, and Szolovits [27] that ". . . one significant part of the representation endeavor—capturing and representing the richness of the natural world—is receiving insufficient attention" and Guarino [48] that the artificial intelligence community is ". . . more interested in the nature of reasoning rather than in the nature of the real world." As Section 2.3 outlined and this dissertation continually addresses, linguistic issues reflect subtle yet important distinctions in the real world and in how humans perceive and manipulate it. An understanding of these details is essential to emulate their behaviors. Chapters 3, 4, and 5 will discuss them further.

This discussion partitions the computational foundation into three interdependent levels according to Russell and Norvig [106]. The *knowledge or epistemological level* addresses what a system knows about the world; the *logical level* addresses the form of the representation of this knowledge; and the *implementation level* address how a system internally manipulates this representation for reasoning.

On the knowledge or epistemological level, decisions about what to encode in a spatial reasoning system are difficult because there is nothing resembling a comprehensive theory of space [124, 84, 90]. This situation is exacerbated on three fronts [106]:

- *Theoretical ignorance*: the knowledge to represent is not completely understood. Without deeper understanding, however, a high-quality computational model is unlikely to reflect many important nuances of behavior.
- *Practical ignorance*: no representation is ever complete because the conditions of its use are unique and infinitely complex. Humans are flexible enough to mitigate this problem, but computers are not.

- *Laziness*: the knowledge to identify, consider, encode, test, and evaluate has immense breadth and depth. Even if complete details were available and understood, the amount to encode is intractable.

Although a comprehensive theory of space is lacking, general agreement within the research community is that there are relatively few epistemological problems with representing space [26, 19]. As Section 2.2 mentioned, the greatest body of research in this area comes from geographic information systems [15, 83]. While the majority of this work applies to large-scale space and issues like naive geography and physics, which are beyond the scope of this project, the foundation of cognitive, linguistic, and computational issues it lays is invaluable [35, 2, 109]. Of particular interest here is knowledge about the behavior of spatial interactions between objects, which fall primarily into the categories of *topological* and *metrical* relations. Topological relations define relatively simple interactions of boundary, contact, and separation between objects [85, 90, 55, 111, 17, 128, 57, 54,90, 121]; e.g., *in*, *on*, *at*, *adjoining*, *overlapping*. Metrical relations define imprecise, fuzzy interactions of distance and angles between objects [55, 85, 88, 90, 59, 121]; e.g., *near*, *far*, *in front of*, *next to*. In general, topological relations *define* spatial configurations, and metrical relations subsequently *refine* them [35]. From the cognitive standpoint of Section 2.2, cognitive maps are mostly metrical and rarely topological [35]. In terms of spatial content, however, topological relations are considered more important [108, 56]. They are also more flexible because they are not affected by translation, rotation, or scaling [33, 93]. This project employs both topological and metric relations, which Section 5.5.1.3 and Section 5.5.2.3 discuss.

Topological relations are the most studied and consequently have the strongest theoretical and practical foundation [15, 90, 17]. Region connection calculus, the seminal work of Cohn et al. [19], uses just two primitives to define hundreds of possible topological configurations of two objects [102]: a function to define the convex hull of a region, and a predicate to test for the connection of two such regions. Hernández [56] and Cohn [18] exhaustively list these compositions and several others. The similar 9-intersection model of Egenhofer [33] overlaps and extends this set further [108]. It is based on the original interval calculus of Allen [3] for time, which has been adapted for space by others as well [100, 26, 28, 98, 17]. It defines 13 interval relations of spatial topology; e.g., *before*, *meets*, *contains*, *contained-by*, *after*.

The logical and implementation levels blend in this review. In theory, the logical formalism is independent of its implementation, but, in practice, it is difficult to decouple them in a computational system [106]. This discussion addresses both levels from the mixed perspective of *quantitative* and *qualitative* representation and reasoning [55]. Hernández and Mukerjee [54] and Mukerjee [90] provide an extensive review. A quantitative approach is purely numerical, generally in terms of hard, absolute, unambiguous coordinates, angles, and distances. Representation of space is traditionally quantitative because it meshes well with the implementation requirements for computation [54, 109]. However, this approach exhibits several disadvantages with respect to this project [56, 109, 72, 114]:

- *Complexity*: many precise details are required to define objects and their interrelations. A representation must commit to a level of detail or granularity, but one size does not fit all possible cases.
- *Partial and uncertain information*: details must be completely and unambiguously defined, but the inherent underspecification, vagueness, and uncertainty in descriptions never provide them.
- *Cognitive inadequacy*: humans do not process the world quantitatively and thus have difficulty defining it in precise, artificial terms. Therefore, two simultaneous representations must be available or derivable for interactive manipulation between a human and computer.

- *Transformational impedance*: additional loss of information and introduction of errors results as users interact with a system because the different representations require iterative, bidirectional translation.
- *Falsifying effects*: the spatial world and its representations rarely—if ever—align perfectly, so approximations and concessions are often necessary to force-fit the former into the latter. Corruption and loss of information results, which exacerbates the previous issues.

A qualitative approach is soft and fuzzy and typically defines its knowledge in terms of rules, heuristics, guidelines, and so on for generalized degrees of truth. Their interpretation depends heavily on the context of their use [51]. They exhibit several advantageous properties with respect to this project [56, 55, 54, 90, 134, 37, 106, 26, 84, 111, 114]:

- *Flexible level of detail*: only the details necessary to solve a problem are needed, and they are defined at the most appropriate granularities.
- *Relative generalities*: details are defined in relative terms of themselves as comparisons, constraints, intervals, assertions, axioms, and so on, which are independent of scale.
- *Support of vagueness*: ranges in details can be represented in terms of loosely defined gradations.

- *Support of underspecification*: partial details contribute to a solution.
- *Compositionality*: partial details can interact for higher precision.
- *Structural similarity*: a qualitative representation can closely reflect the commonsense structure of the real world without unnecessary contortions, commitments, concessions, and so on.
- *Situated nature*: qualitative definitions map well to particular goals. The level of detail can be higher for specialized contexts and lower for generalized ones.

In practice, there is no clear boundary between quantitative and qualitative representations [90]. Each serves its owns purpose, and any reasonably broad system needs to rely on properties of both regardless of whether it acknowledges the form of representation. Section 5.1 and Section 5.3 will discuss the aspects that relate to this project.

3 SPATIAL DESCRIPTIONS

The form of the input to this system is tailored to showcase how nouns, adjectives, and prepositions encode spatial knowledge in a description. As in most related systems, the input here uses contrived scenarios such as Figure 3.1 instead of passages of authentic text from existing sources [29]. This approach generally reflects a concession that real-world text is unsuitable for or beyond the capabilities of a system [124]. In this case, descriptions that are both consistent with a zoo theme and limited to the scope of investigation simply do not exist in any usable quantity. This situation is actually advantageous here because contrived examples facilitate formal, structured experiments, which are the foundation of the analysis component that Chapter 7 will discuss.

The scene contains a tree, a zebra named Zeus, and a giraffe.
Zeus is in front of the giraffe.
Zeus is at the fringe of the tree.
The giraffe is in front of the tree.
The tree is in front and left of the giraffe.
The tree is small.
The giraffe is big.

Figure 3.1: Sample Description[1]

Parsing English text to extract and decipher its grammatical structures is a complex task that is not the focus of this project (see [13, 20]). Nevertheless, as

[1]See Figure 8.1 for the corresponding graphical rendering.

Section 1.1.1 illustrated, some capability of this type must be provided. To simplify its development, the format of a description is rigidly defined according to the grammar in Figure 3.2. Additional, non-linguistic configuration comes from the vignette file that Section 8.1.1 and Chapter 9 will discuss.

```
<SCENE>       := 'The scene contains' <CONCEPTS> '.' <DESCRIPTOR>*

<CONCEPTS>    := ['a'|'an'] <CONCEPT> ['named' <INSTANCE>]
                 [',' ['and'] <CONCEPTS>]

<DESCRIPTOR>  := <IDENTIFIER> 'is' <DESCRIPTION> '.'

<DESCRIPTION> := (<ADJECTIVE> | (<PREPOSITION> <IDENTIFIER>))
                 [',' ['and'] <DESCRIPTION>]

<IDENTIFIER>  := 'the' <CONCEPT> | <INSTANCE>

<INSTANCE>    := single-word alphanumeric string

<CONCEPT>     := see Section 5.5.1.1

<ADJECTIVE>   := see Section 5.5.1.2

<PREPOSITION> := see Section 5.5.1.3
```

Figure 3.2: Description Grammar

3.1 Spatial Nouns

The nouns in a description specify the entities that play a role. For a zoo theme, unsurprisingly, they are limited in scope. Akin to a virtual Noah's Ark, this project employs a strategy of selecting its 41 animals and 11 plants from a broad cross-section of the living world. This collection represents diverse exemplars that exhibit the real-world and linguistic properties and spatial behaviors of interest here. The eclectic mix showcases animals and plants that vary structurally in combinations of height, width, and depth, as well as behaviorally in terms of being predominantly land-, water-, and/or air-dwelling. Other miscellaneous objects are included as well for experimentation; e.g., structures, enclosures, vehicles, bodies of water. See Section 5.5.1.1 for a complete list.

The choice of a zoo theme is not fanciful or arbitrary. Aside from the obvious visual appeal and entertainment value, a zoo and its contents also exhibit several excellent characteristics from the perspective of this project [27, 111, 127]. First, its scale is linguistically appropriate [84, 35, 83, 50]. As objects scale down in size, the interpretation of their description becomes more literal, and the uncertainty and tolerance to error decrease [90]. For example, in a tabletop environment containing a glass and a dinner plate, the possible spatial configurations are very limited [35]. A range of plausible solutions may allow a freedom of only plus or minus a few centimeters in placing these objects with respect to each other. The particular solutions within this small range may hardly differ enough to consider significant, and those immediately outside the range may be completely incorrect. While such spatial reasoning undoubtedly has its uses (e.g., robotics), it does not satisfy the goals of this project. As objects scale up in size, the opposite effect occurs [90]. For example, in a geographic environment containing a town and a mountain, the possible spatial configurations are extremely loose. A range of plausible solutions may allow a freedom of many kilometers in placing these objects with respect to each other. The particular solutions within this huge range may differ too much to consider significant, and those outside the range may be only marginally incorrect.

The choice of a zoo theme allows the world to be fixed at 100 meters square in this project. This scale falls nicely between the minimum and maximum scales as described and provides ample space for large and small objects to interact on near and far levels. Another advantageous characteristic of a zoo is that the size and shape of animals is dictated by nature. As later sections will discuss, a range of plausible sizes from minimum to maximum must be assigned for the dimensions of each object. The range for an adult male animal (the default) is relatively constant in nature. Contrast this to non-natural entities such as cars and buildings, the size and shape of which are—for all practical purposes—unrestricted. Finally, animal size ranges are conveniently documented in many nature resources,[2] which assist greatly in compiling the knowledge and validating the results of this project.

[2]See `www.enature.com`, `www.wikipedia.org`, and `link.bubl.ac.uk/ISC7842`.

3.2 Spatial Adjectives

The spatial adjectives in a description affect the size of the noun/object they modify. All objects have default dimensions of height, width, and depth as part of their range of plausible sizes that the previous section described. In the absence of any spatial adjective modifying a noun, the defaults are used; otherwise, the adjective is interpreted with respect to the noun so that it corresponds to the contextually appropriate size in the contextually appropriate dimensions. For example, a *big tree* is great in height; whereas a *big lake* is great in width and depth.[3] The qualifier prefix *very* is also permitted to extend the minimum and maximum range of sizes as contextually appropriate. Section 5.5.1.2 describes the spatial adjectives that are supported. The set is quite small because the number of ways each can be interpreted in context is large. The adjective *big*, for instance, has six possible combinations of height, width, and depth.

3.3 Spatial Prepositions

The spatial prepositions in a description specify how two nouns/objects interact in terms of their position and/or orientation. All objects maintain a

---

[3]Case in point: notice how *depth* in the context of this discussion on dimensions corresponds to *length*, but in the context of a *lake*, the reader probably thinks of the customary interpretation of how far down the bottom is.

three-dimensional position in the world and a 360-degree orientation that indicates where they are facing. Prepositions contextually limit the range for these values. For example, *the dog is in front of the cat* specifies that the dog must located somewhere in front of the cat, and the position of the dog depends on both the position and the orientation of the cat. Similarly, *the giraffe is near the lake* specifies that the giraffe must be located near the lake, but the position of the giraffe depends only on the position of the lake and not on its orientation. The interpretation of what it means to be *in front of* or *near* something depends heavily on the context of the objects.

Section 5.5.1.3 describes the spatial prepositions that are supported. All are of the binary form <NOUN$_1$> <PREPOSITION> <NOUN$_2$>, which is consistent with all English spatial prepositions of interest in this project except *between* [24, 25, 58, 56]. The qualifier prefix *directly* is also permitted on some prepositions to tighten the interpretation.

3.4 Restrictions and Limitations

A description consists of spatial nouns, adjectives, and prepositions. One major grammatical category, verbs, is conspicuously missing. Except for the copular *is*, no verbs are supported. This limitation is intentional as it cleanly eliminates the complexities associated with verbal constructions, all of which are beyond the scope of this project. It also forces scenes to be static, which precludes any adverse issues of time or movement [106, 26, 77, 44, 87].

Only existential quantification of objects is permitted; e.g., *there is a dog*. Universal quantification (e.g., *all the dogs*) is arguably within the capabilities of this solution, but it is beyond the scope of investigation. For similar reasons, no plural forms of nouns are allowed; e.g., *the dogs*. It is possible, however, to have multiple instances of nouns provided that each is specified separately; e.g., *the dog named Spot and the dog named Fido*. No numerical descriptions are allowed; e.g., *the dog is four meters from the cat*. Although many usages of negation are possible in this solution, it is not permitted; e.g., *the dog is not big* or *the cat is not near the giraffe*. No vertical relations like *above* or *below* are supported, except *on* in certain cases. Finally, objects are not decomposable or articulated, so it is not possible to refer to or manipulate any subcomponent; e.g., *the dog's head is facing left.* Similarly, they are not physics-based, so unrealistic interpretations are possible; e.g., *the elephant is in the raft*, but the raft continues to float.

4 REPRESENTATION OF EXPLICIT KNOWLEDGE

The previous three chapters discussed and demonstrated in many ways how spatial descriptions contain remarkably little explicit content and how most of the details for interpretation are actually supplied by the implicit background knowledge of the reader [61, 113, 114]. This chapter and the next respectively address how this project represents such explicit and implicit knowledge.

4.1 Description Transformation

The structure and content of how spatial descriptions must be expressed in English was laid out in the previous chapter. This sentence-based representation is convenient for humans to read, but it is not well-suited for computers to process. The use of articles and conjunctions, punctuation, capitalization, and certain whitespace makes the text easier to read, but they are extraneous because they contribute no additional information. Reformatting or removing them reduces the text to its core content. For example, Figure 4.1 is equivalent in meaning to its base form in Figure 3.1.

4.2 Semantic Network Representation

Descriptions in this form are (cryptically) concise, but they are still based on the computationally unappealing formalism of characters and words. A graph-based abstraction known as a *semantic network*, in contrast, can represent the same information through an intuitive, visual formalism that is easy to process computationally [114, 87]. It is a collection of one or more disjoint directed graphs consisting of nodes and arcs as Figure 4.2 illustrates.

The rectangular nodes are *objects*, the rounded nodes are *attributes*, and the arcs are *relations*. Objects correspond directly to spatial nouns in Section 3.1, attributes to spatial adjectives in Section 3.2, and relations to spatial prepositions in Section 3.3. In certain cases, correspondences are realigned to simplify later processing. For example, the phrase *dog facing*[1] *north* translates to the attribute expression `dog→[is]→facing-north` because *north* is not an object and cannot be the destination of a relation arc.

At this stage in processing a description, the representation as a semantic network is static: it serves merely as a data structure that holds information without manipulating it [51, 27, 44]. Section 5.5.2, Section 5.8.2.2, and Section 6.2 and will discuss the dynamic characteristics of a semantic network when it is augmented with inferences.

[1]In strict grammatical terms, *facing* is actually part of a verb phrase. Due to the similarity between both grammatical roles, however, it can be easily reclassified here as a preposition to circumvent the lack of support for verbs.

```
tree zebra/zeus giraffe
zeus in-front-of giraffe
zeus at-fringe-of tree
giraffe in-front-of tree
tree in-front-left-of giraffe
tree small
giraffe big
```

Figure 4.1: Reduced Description

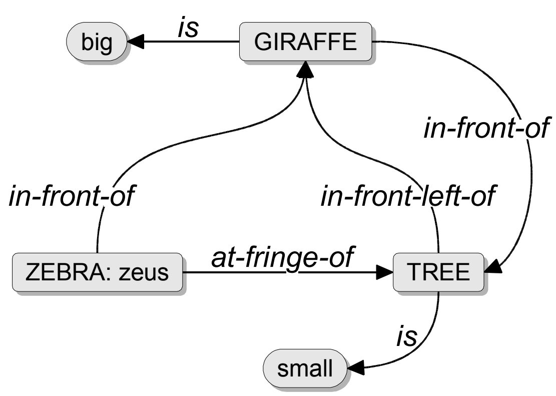


Figure 4.2: Semantic Network

5 REPRESENTATION OF IMPLICIT KNOWLEDGE

The vast majority of knowledge that a person brings to bear in understanding a description is so commonplace and obvious that humans take it for granted and are not even consciously aware of its contribution [114, 106]. This so-called "commonsense" knowledge defines unseen behaviors, relationships, constraints, and so on in the world [121, 35]. Without this implicit understanding of what should and should not occur in particular contexts, as well as what is more likely or favored, humans would not be able to reason so powerfully over spatial descriptions [15]. Such is the plight of many computational approaches to text understanding, which focus on the grammar and vocabulary of the text and ignore the deeper significance of meaning [114].

Implicit spatial knowledge in this project provides a limited commonsense contextual framework for filling in major gaps in the explicit knowledge of a description [78, 121]. It addresses the five stated lexical-semantic issues as follows:

- *Underspecification*: lack of explicit detail is addressed by providing formal definitions for many commonsense, spatial aspects of objects.

- *Vagueness*: inconsistent application of size properties to objects is addressed by formally binding supported English adjectives to a commonsense interpretation of their effect.
- *Uncertainty*: impreciseness in specifying the position and orientation of objects is addressed by formally constraining plausible interpretations to a range of possibilities and favoring certain ones.
- *Context*: effects on meaning from interactions of objects are addressed by categorizing similar behavior and formally assigning it as part of their definitions.
- *Frame of reference*: the role of the viewer and objects in a description is addressed by applying variations of definitions according to context.

5.1 Form of Representation

Among many other distinctions, spatial knowledge representations are traditionally classified as either *quantitative* or *qualitative* [35, 55, 54, 90. 56]. A quantitative representation defines its contents primarily with absolute, numerical values and equations [90]. For example, the attribute *big* could mean *1.5 meters high by 1.3 wide by 1.6 deep*. Similarly, the relation *in front of* could mean *the z position of object B is 2.0 meters greater than the z position of object A with respect to the azimuth of object A*. While this form is useful for many purposes like robotics and tabletop object placement, it is generally incompatible with the vague, uncertain, contextually dependent nature of natural-language descriptions that are the focus of this project [72, 109].

A qualitative representation defines its contents primarily with relative, non-numerical values and generalized degrees of truth over loose constraints [90, 51]. For example, the attribute *big* could mean *roughly 20% above the norm for the appropriate dimensions of an object in context*. Similarly, the relation *in front of* could mean *object B is somewhere within a wedge-shaped region centered on the line of sight of object A, with the middle most likely*. The parallel to a description in natural language is clear.

A qualitative representation does not preclude the use of numerical values for many purposes for which they are more appropriate [37]. Also, deep at the implementation level of any computational reasoning system, all knowledge is eventually processed numerically, of course. A major advantage of a qualitative representation is that it can abstract this degree of detail away from the definitions and defer it to lower levels where it interferes less with the conceptual definitions above it [56, 26, 27, 54]. Where numerical values are unavoidable or indeed desirable, such as in defining the realistic dimensions of animals, they are defined in this project as probabilistic intervals; e.g., *the height of an adult* *male giraffe is normally distributed between 3.9 and 5.5 meters and must not exceed 3.5 and 6.0 meters, respectively.*

5.2 Structural Overview

The previous chapter addressed the representation of explicit knowledge with a semantic network. It was a short and relatively simple discussion because the semantic network serves in this capacity as a static data structure only. In other words, it does not perform any processing. This chapter addresses the representation of implicit world knowledge with a knowledge base. In stark contrast, it is a long and very complex discussion because this dynamic structure is responsible for the majority of the reasoning in this project [77]. Although the reasoning engine in the next chapter is credited in name with this task, the "intelligent" processing is actually realized here by setting up the framework to be used by the "dumb" reasoner.

The knowledge base is complex because the issues it addresses are complex. Unfortunately, its tightly intertwined components cannot be easily broken into independent discussions. To mitigate the difficulty in reading the remainder of this chapter, the organization of its sections is outlined here. They are primarily oriented toward the issues the knowledge base solves instead of toward the knowledge representation language, but there is some crossover. For the complete grammar, see Appendix A.

The knowledge base is a parallel structure to the semantic network in the sense that it operates on the same framework of objects, properties, and relations toward the same goal of supporting reasoning over the dimensions, position, and orientation of entities in a description. Its two main purposes are to represent the implicit semantics and the implicit pragmatics of these components and goals. Semantic representation in Section 5.5 focuses on the default definitions of objects, properties, and relations that are most commonly used in general contexts [26, 27]. Pragmatic representation in Section 5.6 focuses on extending or overriding the default definitions for use in specialized contexts [130]. Each of these representations plays two mutually contributing roles. The first is to define the details necessary to generate a solution from the semantic network of a description. The second is to define how to find new knowledge about the description based on this solution and to add it back into the semantic network. Finally, Section 5.7 focuses on the structural details of constraints, which are the primary means to define and derive numerical values and form the basis of the reasoning processes in this project. This discussion serves as a bridge into the next chapter on the reasoning engine that operates on them.

5.3 Computational Foundation

The knowledge base in this project is defined as *an ontology of concepts that are organized in an object-oriented structure of declarative definitions*. Each of the four components of this definition plays an integral role in the behavior of the knowledge base and will be addressed separately.

5.3.1 Ontological Representation

Definitions vary widely on what an ontology is and is not according to different philosophical and practical views [47, 36, 106, 114, 26, 4, 52, 17, 8, 78, 48]. What is agreed, however, is that "[i]n the field of natural language processing (NLP), there is now a consensus that all NLP systems that seek to represent and manipulate meanings of texts need an ontology . . . ." [78]. The following features of a knowledge representation (adapted primarily from [27]) summarize the points that are considered most important in an ontology for this project [114, 106, 47, 130, 17, 11]:

- *It is a surrogate*. A computational model has no access to the real world and instead must operate on some abstraction of it. An abstraction can never be complete or exact. The ontology in this project reduces the knowledge to fit the stated requirements and goals.

- *It is a set of ontological commitments*. An abstraction is arbitrary and requires its developer to decide and formally define what is important. This definition maps to some form of the conceptualization of the world that Section 2.2 discussed. Thus, the organization of an ontology is usually "situated" in a role it must play [78, 106]; e.g., lexical-semantic issues of space in a zoo.
- *It is a fragmentary theory of intelligent reasoning*. An abstraction requires a definition of the behaviors of its components, not just of their existence, because a reasoning system must operate on them. The ontology in this project defines what its components are and how to interpret them.
- *It is a medium for efficient computation*. An abstraction must be amenable to processing, not just interesting or theoretically elegant. The graph-based formalism and inheritance mechanisms of an ontology directly support this requirement.
- *It is a medium of human expression*. An abstraction should retain the features of the surrogate in a form that humans can understand. The

formalism of definitions in this ontology balance computational and human needs. It is not optimal or ideal for either, but it is understandable and usable by both.

Above all else, this project conforms to the view that an ontology is a taxonomy of definitions [106, 4, 87]. Each of its nodes defines information about a single entry in the knowledge base; e.g., a prototypical `DOG`.[1] The nodes are organized hierarchically from most general at the top of the tree (e.g., `THING`) to most specific at the bottom (e.g., `GREAT-WHITE-SHARK`). Every node except the root one has at least one parent node, which, by nature of the hierarchy, is inherently more general than its itself. An entry is thus defined by the information its node contains, as well as by the information its parent node or nodes contain. This behavior, known as *inheritance*, holds recursively, so each parent is likewise described by its own parents, and so on up the tree.

An analogous example is a taxonomy of the animal kingdom, a grossly simplified branch of which Figure 5.1 shows. A `DOG`, for example, is a `CANINE`, which is a `MAMMAL`, which is an `ANIMAL`, which is a `THING`, and so on. Thus by looking up the entry for `DOG`, the hierarchy directly supplies information about `DOG` and indirectly about its ancestors `CANINE`, `MAMMAL`, `ANIMAL`, and `THING`. If the

[1] Entries in the knowledge base are denoted here by monotype script. Concepts are capitalized.

definition of ANIMAL indicates that they must be mortal, for instance, then all its descendants, including DOG, are by default mortal without having to define them individually in each node. Inheritance plays a powerful role in addressing underspecification in this project, and the next section and Section 5.5.1.1.1 will discuss it in more detail.

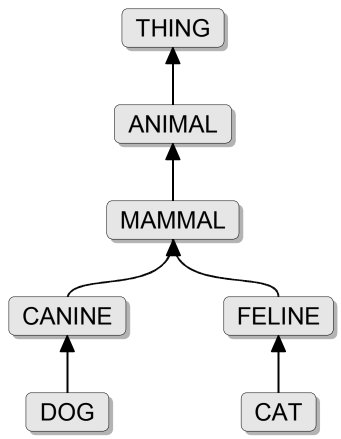


Figure 5.1: Simplistic Taxonomy

There is no single, theoretically "correct" hierarchy for anything that optimally serves all purposes all the time [79, 114, 111]. Life scientists, for example, organize the entries in the taxonomy of the animal kingdom according to the physiological and morphological characteristics they believe are most representative in showing how these animals are related. Many other organizations are possible and might even be more appropriate for other uses; e.g., genetic similarity. Appropriate for this project is an organization that addresses the spatial behaviors of entries. As such, even though it represents many of the same animals as the taxonomy of the animal kingdom, the two differ significantly in structure.

5.3.2 Conceptual Templates

The "entries" in the knowledge base, as the previous section loosely called them, are not instances of what they represent in the real world. For example, DOG does not refer to any particular dog like Rover, Fido, or Spot. Rather, it specifies a *conceptual template* or, for short, the *concept*, of a prototypical class of dog, from which such instances can be built [27, 87, 114, 83]. A concept defines—directly and through inheritance—all the relevant and appropriate information for this project that would be shared by any dog. Concepts are analogous to simple *frames*, which encapsulate related knowledge and structure it with respect to other encapsulations [26, 114, 4, 130, 27]; e.g., a bedroom frame decomposes into (at least) a bed, which links with and/or decomposes into a box spring, a mattress, blankets, etc. The primary distinction in this project is that its concepts do not form a deeply tangled subsumption and decomposition hierarchy of such interconnections, which is common in other knowledge-based work [79, 115]. Its separation of explicit and implicit knowledge representations generally contributes to fewer, cleaner, more straightforward interconnections. Also, its scale does not consider compositional or articulated objects; e.g., a giraffe has a head and body, and the head can move in certain ways independent of the body.

5.3.3 Object Orientation

A concept in the knowledge base is analogous to a class in an object-oriented programming language in several ways [114, 11, 112, 130]:

- Concepts encapsulate related information.
- The same structural relationships hold between superconcept, concept, and subconcept as between superclass, class, and subclass.
- The contents of any superconcept can be inherited or overridden.
- Multiple inheritance allows a concept can have more than one superconcept.
- The process of creating an instance from a concept is instantiation.
- Any number of unique instances can be created from a concept.

5.3.4 Declarative Paradigm

Although the *structure* of the concepts in the knowledge base exhibits great similarity with object-oriented programming languages, the *contents* of the concepts rely on a substantially different formalism. Most object-oriented languages follow an imperative paradigm that precisely specifies the exact procedures to execute in order to produce the desired result. In contrast, the definitions in concepts in this project follow a declarative paradigm that specifies only the necessary and sufficient conditions that a correct solution must satisfy, but not the steps to produce it [101]. As a result, both the knowledge base (see Appendix B) and the grammar that defines it (see Appendix A) are compact, straightforward, and relatively easy to define and understand [106].

At some point in the processing of this knowledge, of course, procedural execution is unavoidable because Java is not a declarative programming language. A declarative representation does not eliminate or even mitigate this need. Rather, it cleanly separates the declarative knowledge details from the procedural implementation details, both of which must still be defined somewhere [4, 26]. The former is here in the knowledge base, and the latter is in the spatial reasoning engine, which the next chapter will discuss. The beauty of this architecture is that it separates *what to do* from *how to do it* and allows the designer of the knowledge base to focus on the issues of interest instead of on low-level programming details [54, 27, 56].

5.4 Mapping of Semantic Network to Knowledge Base

The semantic network and the knowledge base complement each other. Recall that the semantic network represents the explicit details of a description, and the knowledge base represents the implicit details. The mechanism of connection is straightforward: every object node in the semantic network has a link to its corresponding concept node in the knowledge base, and so does every attribute node and relation arc originating from it [43]. For example, Figure 5.2 depicts this mapping for the description *the big dog named Rambo is in front of the cat, and the cat is near small dog named Angel*. A similar two-level approach is used in the Mikrokosmos project as a text meaning representation to link instances of concepts in text to their framework of interpretation in an ontology [113].

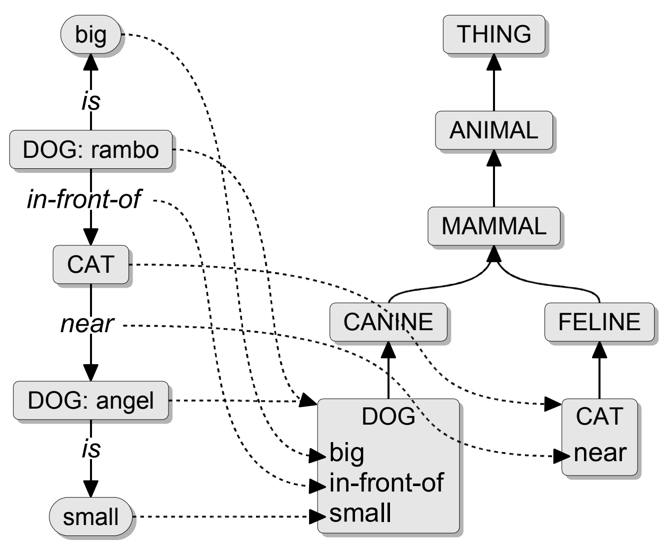


Figure 5.2: Mapping of Semantic Network to Knowledge Base

The interpretation of an object node and its attributes and relationships is defined entirely by the contents of the concept node to which they point in the knowledge base.[2] For example, Rambo and Angel are unique instances of a DOG, but both link to the same concept. Their interpretations will differ, however, because each supplies different attributes and relationships.

5.5 Semantic Representation for Intrinsic Context

The appropriate interpretation of an object node in the semantic network depends on its links both to the knowledge base and from any other object nodes. This mapping establishes the context, which is considered in this project as either *intrinsic* or *extrinsic* [69]. In an intrinsic context, an object node has a prototypical (or "default") interpretation that is independent of any other object nodes in the semantic network. In the most basic example, *there is a dog*, not enough information is available to posit any interpretation beyond its simple, inert presence in the scene regardless of what else is described. In the extension to this example, *there is a big dog*, the interpretation of the size of the dog now plays a role as well, but again, no other external influences apply. The default information for intrinsic contexts is defined in a concept node by a semantic representation. This project takes the position that *semantics* refers to meaning

[2] As all the links that originate from an object node point to the same concept node, there is actually no need to depict them with multiple arrows. The attribute and relationship links are shown for clarity only.

that is independent of context [62, 52, 106, 51]. Section 5.6 will discuss its complement: pragmatic representation for extrinsic context.

5.5.1 Role in Solution Generation

The knowledge base plays two complementary roles evenly across both semantic and pragmatic representation. The first role, which this section discusses, addresses the implicit knowledge that is needed to produce a plausible interpretation from a description. The second role, which Section 5.5.2 will discuss, addresses the knowledge that is essential to handle the converse task—to infer from an interpretation additional spatial information that was not explicitly stated in the original description.

As the semantic network and knowledge base are both defined in terms of objects, attributes, and relations, these three components translate into a natural organization for discussing the details of the knowledge base. Furthermore, the declarative representation of the knowledge base lends itself to an organization that addresses the role of each component with respect to the stated issues and goals of this project. As such, the syntactic details of the knowledge representation language that defines each component are omitted, as well as the procedural details of how it is processed. See Appendix A for the grammar and Appendix B for the knowledge base. Chapter 6 will discuss the procedural details.

5.5.1.1 Objects

Objects in the semantic network correspond to nouns in a description and map to concepts in the knowledge base. As Chapter 3 discussed, these concepts are related to the zoo theme and organized in an ontology. As a suitable ontology of the animal kingdom already exists, this project does not attempt to reinvent the wheel by defining a radically new ontology of the same concepts for its own specialized purposes. A more effective strategy is to augment[3] this existing structure to accommodate new knowledge of the spatial behaviors of its concepts. The mechanism to map spatial behaviors to zoo concepts is reflected in the definition of the concepts, which are either *derived* or *abstract*.

5.5.1.1.1 Derived Concept

As a general rule, a derived concept represents any zoo-related class of objects that can be instantiated and rendered graphically; e.g., a GIRAFFE, a LAKE, or a PICKUP-TRUCK. A derived concept always inherits from one or more

[3]The structure is also simplified by reduction because many levels of the animal kingdom play no role in this project; i.e., phylum, class, order, etc.

other derived concepts and indirectly from THING, which is the root concept of the ontology. It can also inherit from abstract concepts, which Section 5.5.1.1.2 will discuss. As the name suggests, a derived concept derives through inheritance the contents of all its ancestor concepts (both derived and abstract). As its contents need only define how it differs from its ancestors, this representation is very compact. In fact, roughly 34% of the knowledge base (398 of 1163 non-blank lines) resides in the root concept THING!

This inheritance hierarchy also inherently supports different levels of commitment to the plausible interpretation of uncertainty in a description. As Section 5.3.1 discussed, concepts that are lower in the hierarchy are more specific than those above it. This narrowing property contributes to two cases. In the first case, if an object node in the semantic network links to a concept node that is a leaf node in the knowledge base, then the most specific interpretation is made. For example, Figure 5.3 depicts a reduced branch of the MAMMAL lineage in knowledge base. The description *there is a St. Bernard* would link directly and unambiguously to the concept ST-BERNARD.

In the second case, if a link is to a non-leaf node, then a more general, less committal interpretation is made. For example, the description *there is a dog* is satisfied indirectly by a ST-BERNARD or a PIT-BULL, and the even more general description *there is an animal* by a ST-BERNARD, a PIT-BULL, or a CAT. The choice of animal is actually irrelevant because *any* leaf-node beneath ANIMAL is equally valid. This project does not consider context in its selection; e.g., the interpretation of *there is animal in the lake* could limit the possible candidates to only those animals that reasonably belong in a lake. It is beyond the scope of this work to infer such specifics from unnecessarily ambiguous descriptions.

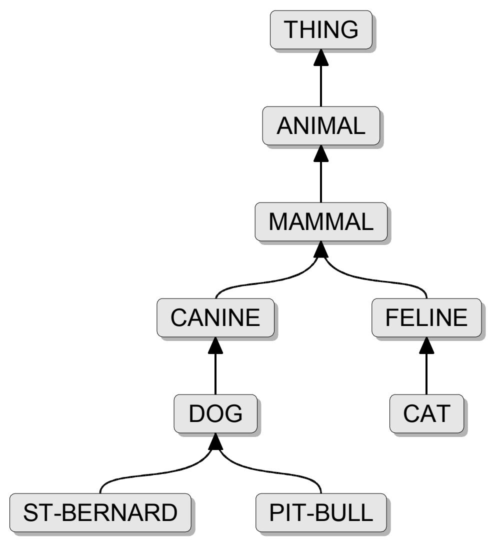


Figure 5.3: Ontology Branch

The knowledge base supports multiple inheritance, so a derived concept may have more than one parent concept. This mechanism is very powerful for representing complexities and irregularities in the real world; e.g., a MULE is a HORSE and DONKEY. Figure 5.4a shows the arbitrary concepts A, B, and C. Concepts A and B define the arbitrary contents x, y, and z (as bold), and concept C inherits them (as italics). Thus concept C implicitly shares the contents of its parents. However, as Figure 5.4b shows, concept C is not required to accept the inherited contents and can explicitly override any of them.

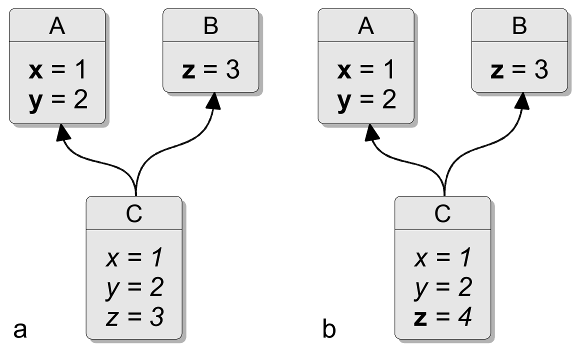


Figure 5.4: Multiple Inheritance

Multiple inheritance unfortunately has the potential to cause ambiguous derivations if two or more parents have conflicting definitions as Figure 5.5 shows [106, 114, 115].

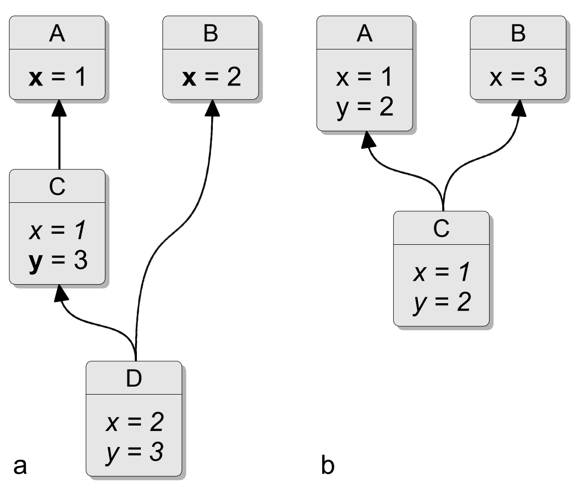


Figure 5.5: Multiple Inheritance Conflicts

The knowledge base implements two mechanisms to resolve such conflicts. For conflicts of the form in Figure 5.5a, the most direct connection to the concept that defines the inherited contents has precedence, so concept D gets its X value from concept B. For conflicts of the form in Figure 5.5b, where both conflicting definitions have equally direct connections, the parent that contributes more non-conflicting information has precedence, so concept D gets its X value from concept A. While these mechanisms are not perfect, they do consistently resolve most conflicts satisfactorily and automatically.[4] Fortunately, a well-designed knowledge base with appropriate use of multiple inheritance tends to avoid most of these issues. The one problematic case in this project was the combination of the abstract concepts (in the next section) GROUND-THING and WATER-THING, which exhibited inconsistent contextual application.

Table 5.1 shows the derived concepts in the knowledge base, each with an arbitrary, unique numeric code that identifies it in the hierarchy in Figure 5.6. They are chosen as representatives for a zoo theme according to the criteria in Section 3.1. Outliers like BLUE-WHALE and RIVER violate these criteria but are included anyway to test the failure limits of this project [101].

[4]Programming languages that support multiple inheritance usually force the programmer to indicate the intended interpretation explicitly. In an automated reasoning system, such a mechanism is not viable.

Table 5.1: Derived Concepts

#	Concept	#	Concept	#	Concept
1	alligator	37	garter-snake	73	park-bench
2	american-alligator	38	gator	74	pen
3	american-crocodile	39	geographic-thing	75	pickup-truck
4	amphibian	40	giant-manta	76	pig
5	anaconda	41	giraffe	77	pine-tree
6	animal	42	golden-eagle	78	pink-salmon
7	ape	43	gray-wolf	79	pit bull
8	aquatic-animal	44	great-egret	80	plant
9	aquatic-plant	45	great-white-shark	81	pond
10	artificial-thing	46	hammerhead-shark	82	primate
11	aspen-tree	47	hippo	83	python
12	atlantic-octopus	48	horse	84	rabbit
13	birch-tree	49	human	85	raft
14	bird	50	iguana	86	red-wolf
15	blue-whale	51	kangaroo	87	redwood-tree
16	bullfrog	52	killer-whale	88	reptile
17	bush	53	lake	89	rhino
18	cactus	54	leopard	90	river
19	cage	55	lily-pad	91	rodent
20	camel	56	lion	92	saguaro
21	canine	57	lizard	93	saint-bernard
22	cat	58	loch-ness-monster	94	salmon
23	cherry-tree	59	mallard-duck	95	sea-monster
24	coho-salmon	60	mammal	96	shark
25	colobus-monkey	61	man	97	snake
26	corral	62	manta	98	snapping-turtle
27	crocodile	63	maple-tree	99	swine
28	dog	64	marsupial	100	tree
29	domestic-cat	65	mexican-wolf	101	turtle
30	elephant	66	mondopod	102	ungulate
31	elk	67	monkey	103	whale
32	emperor-penguin	68	mountain-gorilla	104	white-pelican
33	feline	69	octopus	105	wild-cat
34	fern	70	pacific-octopus	106	willow-tree
35	fish	71	palm-tree	107	wolf
36	fountain	72	panther	108	zebra

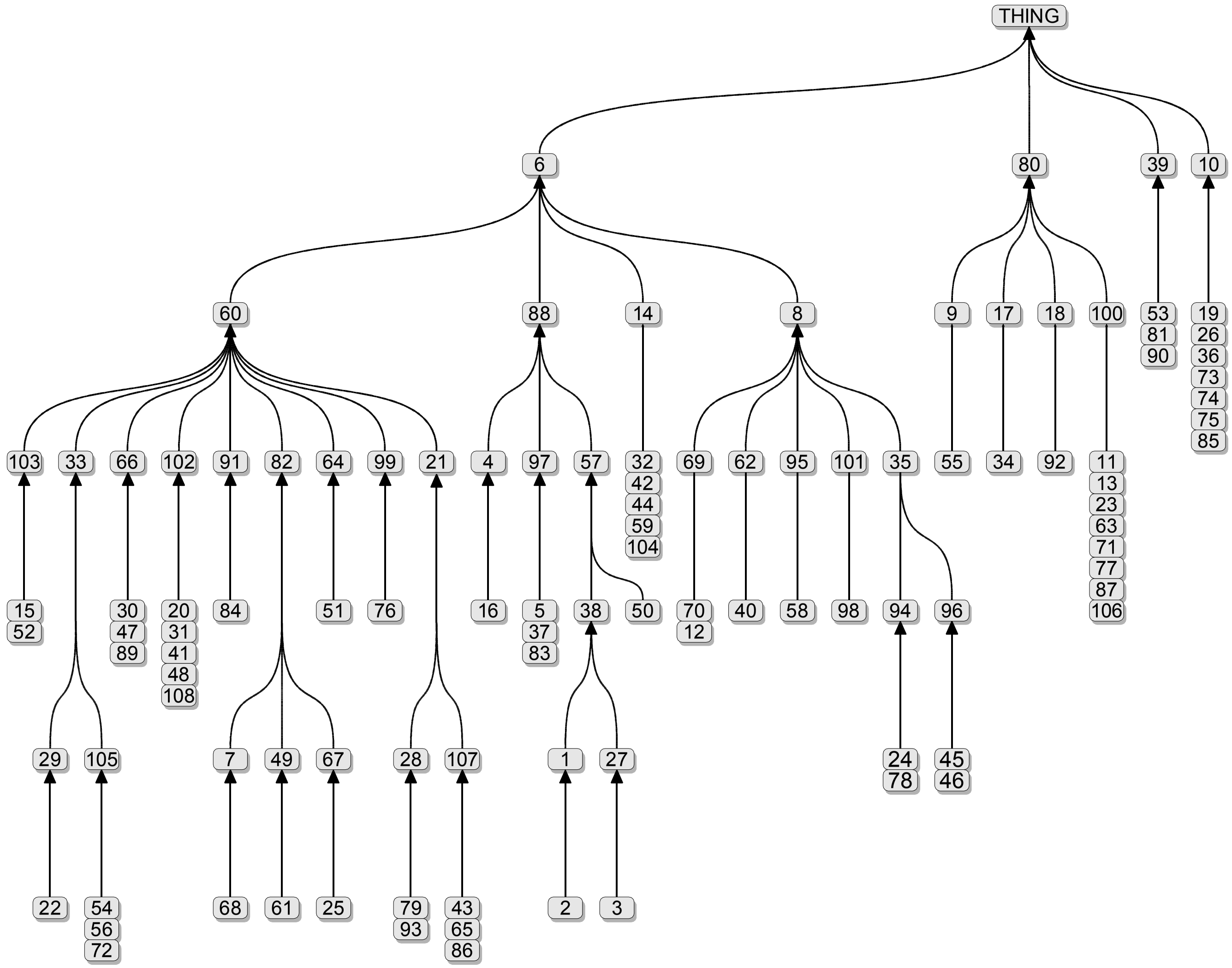


Figure 5.6: Taxonomy of Derived Concepts

5.5.1.1.2 Abstract Concept

An abstract concept follows the opposite general rule from a derived concept: it represents any class of objects that cannot be instantiated or rendered graphically. In this project, such classes are based on spatial behaviors that have no real-world correspondence; e.g., *there is a member of a class of objects that share the spatial behavior of having a front side*. An abstract concept defines a top-level contract for behaviors by which its descendants must abide. It is similar to an abstract class in programming languages. In addition, it cannot inherit from other abstract concepts or derived concepts because it resides at the top level of the ontology. Table 5.2 shows the abstract concepts in the knowledge base.

Table 5.2: Abstract Concepts

Abstract Concept	Description
`THING`	Defines the single ancestor of all derived concepts; contains the default interpretations that they all inherit or override.
`CONTAINER`	Allows other objects to penetrate its volume.
`GROUND-THING`	Belongs on the ground only.
`WATER-THING`	Belongs in a `BODY-OF-WATER` only.
`TREE-ABLE-THING`	Can occupy a `TREE`.
`BODY-OF-WATER`	Allows other objects to be on or below its surface.
`SMALL-ANIMAL`	Indicates a relatively small animal.
`LARGE-ANIMAL`	Indicates a relatively large animal.

5.5.1.1.3 Multidimensional Knowledge Base

The knowledge base is an single ontology that combines the abstract concepts in Table 5.2 with the derived concepts in Table 5.1 to produce a structure similar to the notional example in Figure 5.7. The abstract concepts define the spatial behaviors to which all their derived concepts must conform; e.g., every `CANINE` is a `GROUND-THING`, every `AQUATIC-ANIMAL` is a `WATER-THING`, and a `HIPPO` is both. The derived concepts define the zoo-related taxonomy by inheriting from derived and/or abstract concepts. This solution facilitates overlaying an application-specific structure onto an existing general ontology [71, 41, 124]. It also deflects a common criticism of knowledge-based systems that their haphazard, ad hoc structure bears little or no resemblance to the real world [90, 79, 106].

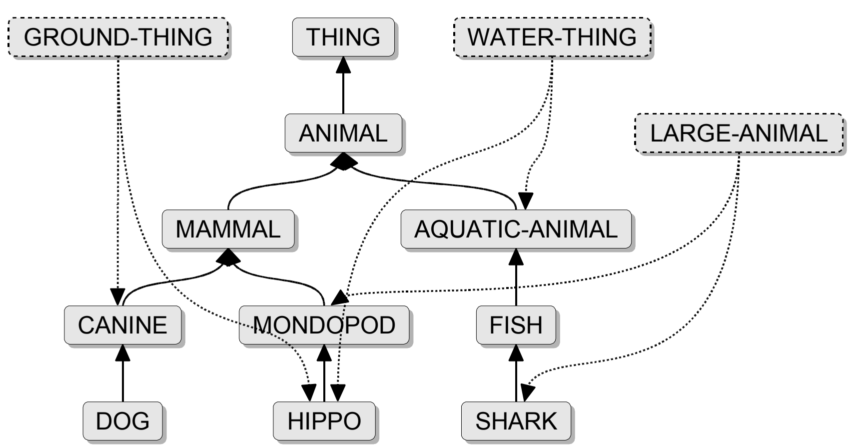


Figure 5.7: Multidimensional Ontology

It is interesting to note how small the collection of abstract concepts is. Although the derived concepts in the animal kingdom vary greatly in breadth and depth (even in this simplified taxonomy), there is surprisingly little variation in their spatial behavior. In other words, not much representation is required to model the spatial behaviors of a wide range of animals at the level that this project considers.

5.5.1.2 Attributes

Attributes in a semantic network correspond to adjectives in a description and map to attribute definitions within a concept in the knowledge base. Attributes define two types of features that configure the interpretation of a concept: *properties* and *attribute intervals*.

5.5.1.2.1 Properties

A property is a type of attribute that is used internally within the knowledge base only. It defines immutable features, either *primitive* or *range* values, that always hold true for the concept in which it resides. For example, the property `has-canonical-front` is assigned a boolean value depending on whether the concept has a front face and is capable of looking in a particular direction; e.g., true for a DOG but false for a TREE. A property cannot be changed or referenced in any way from outside the knowledge base; e.g., *there is a tree with a canonical front*.

5.5.1.2.1.1 Primitive Properties

The properties in Table 5.3 are used mainly in conditional dependencies to process contextual interactions. They are referred to as *primitive properties* because they are assigned a single value, normally a boolean or number. For example, in *the dog is in front of the tree*, the appropriate interpretation of the relation IN-FRONT-OF depends on the knowledge that a TREE has no canonical front. Section 5.6.2 will discuss the details of defining attributes for these properties.

Table 5.3: Primitive Properties

Primitive Property	General Purpose
`has-canonical-front`	Does an object has a canonical front?
`is-container`	Can an object contain another object?
`supports-dimension-comparison`	Does an object generate dimension inferences?

5.5.1.2.1.2 Range Properties

The properties in Table 5.4 are used to determine an appropriate value for each of the three dimensions that an instance of a concept occupies in space. They are referred to as *range properties* because they are assigned an inclusive range of values from minimum to maximum. For example, the absolute height of a `GIRAFFE` is distributed between 3.5 and 6.0 meters, with a suggested minimum and maximum of 3.9 and 5.5 meters, respectively. Section 5.7.1.1 will discuss the details of defining attributes for these properties.

Table 5.4: Range Properties

Range Property
`height`
`width`
`depth`

5.5.1.2.2 Attribute Intervals

An attribute interval is a type of attribute that links a contrasting pair of English adjectives of size to the appropriate dimensions that are defined as range properties for a concept. For example, `short` and `tall` apply to the `height` dimension of a `GIRAFFE`, where the former links to the lower range and the latter to the upper range. Thus the descriptions *there is a short giraffe* and *there is a tall giraffe* should produce giraffes that are roughly 3.9 meters and 5.5 meters in height, respectively. The qualifier *very* is defined to extend this range. Table 5.5 shows the attribute pairings, and Section 5.7.1.2 will discuss the details of defining them.

Table 5.5: Attribute Intervals

Interval	General Application
`short ... tall`	height, for vertically oriented concepts
`short ... long`	length, for horizontally oriented concepts
`narrow ... wide`	width
`small ... big`	height, width, and depth

5.5.1.3 Relations

Relations in a semantic network correspond to prepositions in a description and map to relation definitions within a concept in the knowledge base. Relations define the constraints that apply in the interpretation of spatial interactions for position and orientation between two objects in a description. Section 5.7 will discuss how constraints operate.

5.5.1.3.1 Relative Position Relations

A *relative position relation* specifies the plausible range of positions where an object in a description must appear with respect to the position of another object; e.g., *the dog is east of and near the cat*.

5.5.1.3.1.1 Global Relative Position Relations

The simplest class of position relations, *global relative*, is listed in Table 5.6. It specifies positions based on cardinal and intercardinal compass directions; e.g., *the dog is north of the cat*. These relations are comparatively simple because context plays little role; i.e., north is always north regardless of the particular objects [30].

Table 5.6: Global Relative Position Relations

`north-of`	`directly-north-of`	`northeast-of`
`south-of`	`directly-south-of`	`northwest-of`
`east-of`	`directly-east-of`	`southeast-of`
`west-of`	`directly-west-of`	`southwest-of`

Global relative position relations play a secondary role as well. The knowledge base defines no true *global absolute position relations*, which would specify the plausible range of positions where an object in a description must appear in the world. Nevertheless, the relations in Table 5.7 are available to descriptions because they provide control over the basic layout of objects in the world that is otherwise not be possible with the other relations; e.g., *the dog is in the north*.

Table 5.7: Quasi-Absolute Position Relations

`in-north`	`directly-in-north`	`in-northeast`
`in-south`	`directly-in-south`	`in-northwest`
`in-east`	`directly-in-east`	`in-southeast`
`in-west`	`directly-in-west`	`in-southwest`

The issue that makes this case special is the strict definition of a relation: it establishes a relationship between two objects. In this project, directions and areas on the world, like north, are not objects. To circumvent this syntactic limitation, the relations in Table 5.7 undergo an automatic transformation in parsing from this form in a description to their representation as a legal relation in the semantic network. The mechanism is straightforward: entries from Table 5.7 are syntactically rewritten in terms of entries from Table 5.6, and the special object WORLD-CENTER[5] serves as the missing object. For example, *the dog is in the north* is rewritten as *the dog is north of world center*.

5.5.1.3.1.2 Local Relative Position Relations

The similar but far more complex class of position relations, *local relative*, is listed in Table 5.8. It specifies positions based on the sides and corners—referred to as *facets* here—of objects; e.g., *the dog is in front and left of the cat*. These relations are complicated because facets differ contextually depending on the particular objects. For example, a DOG is defined to have a canonical front, so it has an established front facet, as well as a back, left, right, and so on. A TREE, on the other hand, is defined not to have one, so it has no established facets. Nevertheless, it is obviously possible to describe that *the dog is in front of the tree*. The proper interpretation of these relations

[5] Section 5.5.1.3.1.1 and Section 5.7.2.1.1.2 will discuss the WORLD-CENTER concept/object.

depends on their properties and the context that they form. Section 5.6 will discuss this mechanism.

Table 5.8: Local Relative Position Relations

in-front-of	directly-in-front-of	in-front-left-of	to-side-of
in-back-of	directly-in-back-of	in-front-right-of	
left-of	directly-left-of	in-back-left-of	
right-of	directly-right-of	in-back-right-of	

5.5.1.3.1.3 Relative Distance Relations

The final class of position relations, *relative distance*, is listed in Table 5.9. It specifies how far apart two objects must be based on their perception[6] of distance; e.g., *the dog is midrange from the cat*. These relations are complicated because distances differ contextually depending on the particular objects. For example, what is considered FAR-FROM for a DUCK is probably reasonably NEAR for a GIRAFFE.

Table 5.9: Relative Distance Relations

in	on	midrange-from
inside	adjacent-to	far-from
outside	near	at-fringe-of

[6]Actually, on *humans' perception* of their perception because animals generally cannot communicate their perceptions, of course, even though most certainly are aware of distance [12]. Other objects like trees obviously do not have perceptions at all.

5.5.1.3.2 Orientation Relations

An *orientation relation* specifies the plausible range of directions in which an object in a description must face; e.g., *the dog is facing the cat and the cat is facing the east*.

5.5.1.3.2.1 Absolute Orientation Relations

The simplest class of orientation relations, *absolute*, is listed in Table 5.10. It specifies orientations based on cardinal and intercardinal compass directions and is independent of any other objects; e.g, *the dog is facing north*. These relations are considered simple because context plays little role beyond the requirement of a canonical front; e.g., north is always north regardless of the particular object.

Table 5.10: Absolute Orientation Relations

facing-north	facing-northeast
facing-south	facing-northwest
facing-east	facing-southeast
facing-west	facing-southwest

5.5.1.3.2.2 Relative Orientation Relations

The similar but more complex class of orientation relations, *relative*, is listed in Table 5.11. It specifies orientations based on the position of another object that an object is facing or facing away from; e.g., *the dog is facing the* *cat*. These relations are complicated because context plays a role in what is considered the extent of the front field for particular objects. In general, field size is directly proportional to object size; i.e., larger objects have larger fields. Section 5.7.2.1.1.1 will discuss this issue in detail.

Table 5.11: Relative Orientation Relations

`facing`	`directly-facing`
`facing-away-from`	`directly-facing-away-from`

5.5.2 Role in Inference Generation

Section 5.5.1 discussed in great detail the implicit knowledge in the knowledge base that contributes to producing an interpretation from a description. This section discusses the implicit knowledge that contributes to the complementary process of inferring from an interpretation additional spatial information that the original description did not explicitly state and feeding it back into the semantic network that represents it [26, 4, 77]. In other words, inferences make implicit relations explicit [56, 15, 11]. Section 6.2 will discuss the procedural details and provides a complete example.

The same declarative framework of objects, attributes, and relations applies to the generation of inferences just as it does to the generation of solutions. Not surprisingly, there is usually a direct correspondence between each solution component in Section 5.5.1 and its complementary inference component in this section. However, the correspondences are not necessarily between object and objects, relations and relations, or attributes and attributes, as one might expect.

5.5.2.1 Object Inferences

Inferences do not create new object nodes in a semantic network, so there is no inverse to Section 5.5.1.1. They can only refine existing ones by adding new attribute nodes and new relationship arcs. This behavior is consistent with many basic spatial inferences that humans make [15, 35, 77, 10] and typical of reasoning systems that operate over a static world [106].[7] For example, it would make little sense to infer from the description *there is a dog* that somehow a LAKE plays an unstated role in the scene. Nevertheless, inferring the presence of unstated objects is not at all unusual in more complex descriptions that are beyond the scope of this project; e.g., one can easily infer from *the dog is swimming* that the dog should be in water.[8] In fact, the

[7]It is also convenient for the simulation and analysis processes that Chapter 7 will discuss.

[8]Such inferences are far more complex that those handled in this project because they require reasoning on many more levels over a greater variety of information [104, 114, 121, 26]. For example, swimming actually implies that the action occurs in a liquid, which is not necessarily water. Lava, while certainly not the preferred inference, is a valid liquid!

WordsEye system [116] focuses considerable effort on determining the nature of the environment that implicitly accompanies a description.

5.5.2.2 Attribute Inferences

An attribute inference generates a new attribute node for an object node in the semantic network. Only the *absolute orientation inferences* in Table 5.12 are defined as attribute inferences. The *relative orientation inferences*, which infer objects facing other objects, are generated as relations in Section 5.5.2.3.3.

Table 5.12: Absolute Orientation Inferences

`facing-north`	`facing-northeast`
`facing-south`	`facing-northwest`
`facing-east`	`facing-southeast`
`facing-west`	`facing-southwest`

From a particular solution to a description, attribute inferences specify which cardinal or intercardinal direction each object in it is facing. For example, if the description *the dog is facing the cat and the cat is facing the dog* produces a solution where the DOG is arbitrarily facing north, then the object node of the DOG in the semantic network will receive a new attribute node `facing-north`. Likewise, as the CAT is facing in the opposite direction, it will receive a new attribute node `facing-south`.

Absolute orientation inferences are the direct inverse of absolute orientation relations in Section 5.5.1.3.2.1. Notice that the inverse of these *relations* is actually defined as *attributes*! The reason behind this skewed correspondence lies at the syntactic level of this project. As Section 5.5.1.3.1.1 discussed, directions are not objects, so what appears to be a relation in a description like *the dog is facing north* is actually translated into the semantic network as the attribute node `facing-north`. Therefore, the inverse of this quasi-relation is rightfully an attribute inference.

This confusing transformation deserves additional justification. For descriptions involving one object facing another, say *the dog is facing the cat*, there is clearly a relationship between the object DOG and the object CAT. In contrast, in descriptions involving one object facing a direction, say *the dog is facing north*, there is no such relationship because DOG is the only object. One option in resolving this inconsistency in the behavior of the relation FACING is to treat directions as objects. Unfortunately, this introduces problems with other relations that use directions as locations; e.g., *the dog is in the north*.[9] This

[9]This solution seems viable in the general case because there are established locations for the north and south poles in the world. The lack of east or west poles, however, introduces further inconsistencies that would make the representation even more convoluted.

project instead treats the act of facing objects as a relation and facing directions as an attribute.

Notice also that there is no inverse to attributes of size. A description can state that *there is a big dog* and generate a solution with the appropriate dimensions for the DOG. However, from the solution, unstated attributes of size for the DOG cannot be inferred; e.g., that the DOG is `long`, too. The reason behind this apparent limitation is that any object lacking a specific size in a description defaults to a nominal size in its interpretation [114]. Therefore, all solutions for *there is a dog* always produce roughly the same average-sized DOG. There is no point in inferring that a default dog is of nominal size and adding this redundant information back into the semantic network.

5.5.2.3 Relation Inferences

A relation inference generates a new relationship arc between two object nodes in the semantic network. The relation in this arc may specify new information about their relative dimensions, position, or orientation.

5.5.2.3.1 Relative Dimension Inferences

A *relative dimension inference* individually compares the three dimensions of `height`, `width`, and `depth` between two objects in the solution of a description. The appropriate inferences in Table 5.13 are added back into the semantic network. For example, in *there is a dog and a giraffe*, obviously the GIRAFFE has greater `height`, `width`, and `depth` values than the DOG has, so all inferences in Column A apply to the GIRAFFE and conversely all in Column B to the DOG.

Table 5.13: Relative Dimension Inferences

A	B	C
has-more-height	has-less-height	has-equal-height
has-more-width	has-less-width	has-equal-width
has-more-depth	has-less-depth	has-equal-depth

Isolating comparisons of size into three dimensions removes all context from the inference process because the implementation is purely numerical; e.g., GIRAFFE.height > DOG.height ⇒ GIRAFFE HAS-MORE-HEIGHT DOG. If the desired inference were GIRAFFE BIGGER-THAN DOG, then it would be necessary to determine the definition of BIGGER-THAN in the context of the two; i.e., bigger in precisely which ways? However, this mapping would reintroduce ambiguity into the semantic network and thus undermine the goal of producing a representation that is amenable to computational processing.

Finally, notice that relative dimension inferences have no corresponding relations in Section 5.5.1.3. This situation occurs because comparisons are not allowed in descriptions; e.g., *the dog is bigger than the cat*. Little value for this functionality was found in the proof-of-concept prototypes of this project because the variation in size between two objects of the same type is often imperceptible unless they are presented next to each other. In addition, it was found to be remarkably easy to describe unsatisfiable scenes like *the dog is bigger than the giraffe*!

5.5.2.3.2 Relative Position Inferences

A *relative position inference* adds new information about the position or distance between two objects in the solution of a description. The appropriate relation inference is added back into the semantic network.

5.5.2.3.2.1 Local Relative Position Inferences

A *local relative position inference* determines where one object is located with respect to a facet of another object in the solution of a description. The inferences in Table 5.14 are the direct inverse of the local relative position relations in Section 5.5.1.3.1.2, so the same issues apply in both.

Table 5.14: Local Relative Position Inferences

`local-in-front-of`	`local-directly-in-front-of`	`local-in-front-left-of`
`local-in-back-of`	`local-directly-in-back-of`	`local-in-front-right-of`
`local-left-of`	`local-directly-left-of`	`local-in-back-left-of`
`local-right-of`	`local-directly-right-of`	`local-in-back-right-of`

These inferences are considered local because they apply only if an object supports a local frame of reference by having a true `has-canonical-front` property. For example, in *the dog is facing the tree*, the TREE is inferred to be LOCAL-IN-FRONT-OF the DOG because the DOG has a front facet. Conversely, however, the tree cannot generate any local inferences involving the DOG (or anything else) because it lacks facets.

5.5.2.3.2.2 Global Relative Position Inferences

A *global relative position inference* determines where one object is located with respect to another object in the solution of a description independent of facets. The inferences in Table 5.15 are the direct inverse of both the *global relative position relations* in Section 5.5.1.3.1.1 and, surprisingly, also of the *local relative position relations* in Section 5.5.1.3.1.2.

Table 5.15: Global Relative Position Inferences

A	B
global-in-front-of	south-of
global-in-back-of	north-of
global-left-of	west-of
global-right-of	east-of
global-directly-in-front-of	directly-south-of
global-directly-in-back-of	directly-north-of
global-directly-left-of	directly-west-of
global-directly-right-of	directly-east-of
global-in-front-left-of	southwest-of
global-in-front-right-of	southeast-of
global-in-back-left-of	northwest-of
global-in-back-right-of	northeast-of

This seemingly odd correspondence arises from the treatment of global frame of reference in this project. For example, if any object is in front of an object that does not have a canonical front, then the interpretation is that the first object is located somewhere between the second object and the viewer of the scene. Consistent with the graphical rendering of a three-dimensional scene on a computer monitor, the viewer is always assumed to be located in the south of the world and facing north [8]. Therefore, the first object is located both SOUTH-OF the second object and GLOBAL-IN-FRONT-OF it, as these two relations are equivalent. In fact, for every relation in Table 5.15, Column A and Column B are equivalent. Both forms are defined for completeness and clarity. Global and local frame of reference can apply simultaneously, so it is common to see inferences that, at first glance, appear contradictory. For example, *the dog is facing south and is west of the tree* generates the inferences that the DOG is GLOBAL-LEFT-OF the TREE while the TREE is also LOCAL-LEFT-OF the DOG.

5.5.2.3.2.3 Relative Distance Inferences

The final class of relative position inferences, *relative distance*, determines how far apart two objects are in the solution of a description. The inferences in Table 5.16 are the direct inverse of the *relative distance relations* in Section 5.5.1.3.1.3, so the same issues apply in both.

Table 5.16: Relative Distance Inferences

`in`	`adjacent-to`	`far-from`
`inside`	`near`	`at-fringe-of`
`outside`	`midrange-from`	

The measure of distance is based on the context of the objects, especially on their size. For example, *the giraffe is in the north and the duck is in world center* generates different relative distance inferences for each object. The GIRAFFE is much larger than the DUCK, so the DUCK is perceived (by humans) as NEAR it. The DUCK has quite the opposite behavior, so the GIRAFFE is perceived as MIDRANGE-FROM it. In reality, the absolute distance between them is identical—only the perception differs [12]. Section 5.7.2.1.1.1 will discuss this behavior in detail.

5.5.2.3.3 Relative Orientation Inferences

The third and final class of relative inferences, *relative orientation*, determines where an object in the solution of a description is facing with respect to another object. The inferences in Table 5.17 are the direct inverse of the *relative orientation relations* in Section 5.5.1.3.2.2, so the same issues apply in both. Only objects that have a canonical front can generate a relative orientation inference; e.g., *the dog is facing the tree*, but not *the tree is facing the dog*.

Table 5.17: Relative Orientation Inferences

`facing`	`directly-facing`
`facing-away-from`	`directly-facing-away-from`

5.6 Pragmatic Representation for Extrinsic Context

Section 5.5 discussed semantic representation for intrinsic context, which defines default interpretations for object nodes in a semantic network that are interpreted independently of any other object nodes. This section discusses the complement: in an *extrinsic* or *interactional context*, an object node has a special (or "non-default") interpretation that *is* dependent on other object nodes in the semantic network [69, 58]. For example, *the dog is in the corral* implies that it is (standing) on the ground, but *the dog is in the lake* implies that it is (floating) under the water.[10] The special information for extrinsic contexts is defined in a concept node by a pragmatic representation. This project takes the position that *pragmatics* refers to meaning that is dependent on context [62, 114, 95, 106, 51]. The pragmatic representation builds upon the semantic representation (i.e., semantics + context = pragmatics), so it serves the same purposes and uses the same formalisms that Section 5.5 described. The primary

[10]The dynamic states of *standing* and *floating* are implicit because movement is not supported.

addition is a declarative mechanism to identify contexts and define which special behaviors apply when.

5.6.1 Pragmatic Interpretation by Context

If the interpretation of an object node were always independent of the other object nodes in its relationships, then a pragmatic representation would be unnecessary because only one context could ever exist; i.e., the concept would always have the same meaning. At the other extreme, if the interpretation were unique for every pairing of nodes, then the number of contexts would be c^2, where c is the number of concept nodes in the knowledge base and the exponent derives from the binary nature of the relationships [7]. Fortunately, reality lies somewhere between these two extremes, generally much closer to the former than to the latter. Contexts define these special interpretations through two similar mechanisms: *concept matching* and *concept lineage matching*.

5.6.1.1 Concept Matching

Recall from Section 5.5 that a default (semantic) interpretation applies in the prototypical case where the two object nodes in a relation are interpreted no differently together than they would be apart [35]. For example, the interpretation of the HIPPO is identical in *the hippo is in the corral* and *the hippo is in the cage*. Likewise, the interpretation of IN defaults to the commonsense understanding that the HIPPO is located on the lower surface of the CORRAL and the CAGE; i.e., *standing* on their base. A non-default (pragmatic) interpretation, on the other hand, applies in the special case where one or both objects are interpreted differently when in context with each other [57]. For example, in *the hippo is in the lake*, the HIPPO should appear beneath the upper surface of the LAKE; i.e., *swimming* in it. In contrast, however, in *the raft is in the lake*, the RAFT should appear on the upper surface of the LAKE; i.e., *floating* in it. Figure 9.2 in the results chapter depicts this example.

Concept matching provides a straightforward mechanism to identify and define non-default interpretations. It supports overriding any component in either concept definition except their hierarchical structure in the ontology; i.e., the links to their parents. Figure 5.8 illustrates the pragmatic template for *the golden eagle is in the pine tree.*

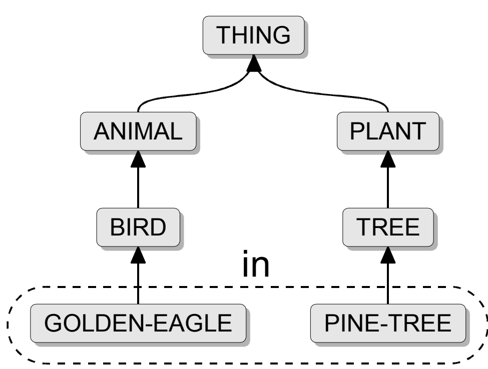


Figure 5.8: Concept Matching

In this example, the GOLDEN-EAGLE should appear somewhere in the upper region of the PINE-TREE, not at the base of its trunk. To define this different position, a *degree-of-freedom adjustment* is associated with the concept definition of the GOLDEN-EAGLE. This adjustment offsets any element of the default position (x, y, and z) and/or attitude (pitch, roll, and yaw) for the object [90, 49]. Thus, to position the GOLDEN-EAGLE up in the PINE-TREE, an arbitrary offset of 3.0 meters is assigned to its y element. The inclusion of the degree-of-freedom adjustment in this project allows it to support limited reasoning in the vertical dimension even though the underlying formalism for reasoning is actually based on only two dimensions. Section 5.7.2 and Section 6.1.1.3 will discuss this issue.

5.6.1.2 Concept Lineage Matching

Concept lineage matching is a minor extension to concept matching to improve the effectiveness and compactness of the representation. It performs the identical function but does so through a more flexible mechanism. With basic concept matching, a unique pragmatic template is required for each pairing of specific concepts. This structure is acceptable for relatively small sets of unusual or exotic non-default interpretations that are best defined in an ad hoc manner. For larger sets that share obvious common behavior, however, the number of individual templates is prohibitive. For example, the spatial behavior that the GOLDEN-EAGLE exhibits in context with a PINE-TREE also holds for the seven other trees in the knowledge base. Concept matching would require additional templates in context with ASPEN-TREE, BIRCH-TREE, CHERRY-TREE, MAPLE-TREE, PALM-TREE, REDWOOD-TREE, and WILLOW-TREE. The obvious commonality between these eight concepts is that they are all trees. As they are all descendants of the same superconcept TREE, it is far more effective to match to the *lineage* than to the individual concepts as Figure 5.9 shows.

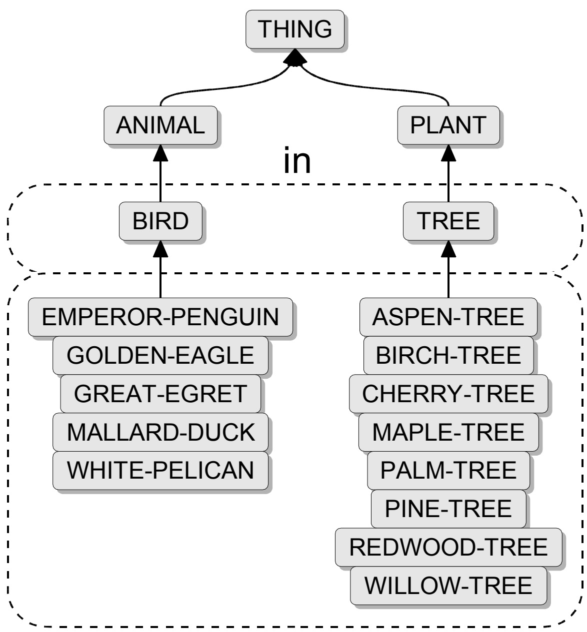


Figure 5.9: Concept Lineage Matching

This mechanism could be extended even further based on the observation that perhaps *any* BIRD exhibits this spatial behavior, not just a GOLDEN-EAGLE. Thus a single pragmatic template would describe this entire class of non-default interpretations: BIRD in context with TREE. With basic concept matching, 40 templates are required because the knowledge base defines five kinds of birds and eight kinds of trees.

Note that it also possible to override a non-default interpretation [130]; i.e., override the override of a default interpretation. For example, not all descendant concepts of the superconcept BIRD may exhibit this spatial behavior in the context of a TREE; e.g., an EMPEROR-PENGUIN would find it difficult![11] Conflicts in multiple inheritance appear more likely to arise as the override nesting deepens, however.

As concept lineage matching supports overriding almost any component in a concept definition, dimensions can be contextually defined as well. For example, a reasonably appropriate class of size can be assigned to each CAGE in *the elephant is in the cage* and *the rabbit is in the cage* based on the templates LARGE-ANIMAL in context with CAGE and SMALL-ANIMAL in context with CAGE, respectively.

[11] In reality, the knowledge base does not define such inane cases because it is unacceptable to describe that *the emperor penguin is in the tree* anyway. In other words, this project is not expected to produce a valid interpretation from an invalid description, even though a human can easily envision and reason over such a scenario.

5.6.2 Pragmatic Interpretation by Conditional Dependency

The mechanism for pragmatic interpretation by context in Section 5.6.1 operates at a high level by concept lineage matching or at an intermediate level by concept matching. Pragmatic interpretation by *conditional dependency* completes this set by operating at a low level and evaluating the contents of concepts. This mechanism is similar to a Horn clause in Prolog, in that a rule applies only if all its antecedents are true [106]. Its propositions define (or redefine) only constraint rules (Section 5.7.2) or inference rules (Section 5.5.2) in a concept, unlike pragmatic interpretation by context, which can define or refine almost any component. A conditional dependency is classified as either an *early dependency* or a *late dependency* based on when it can be evaluated. For clarity and simplified parsing, the grammar of the knowledge base in Appendix A makes a further distinction between a *static dependency function* or *dynamic dependency function*, which Chapter 6 will discuss for the spatial reasoning engine.

5.6.2.1 Early Dependencies

An *early dependency* consists of one or more conditional expressions that are evaluated during solution generation as Section 5.5.1 outlined. Each conditional expression uses one or more of the *static dependency functions* in

Table 5.18 to determine whether a constraint rule applies in a particular situation.[12] These functions are considered static because only the semantic network is used in evaluating them.

Table 5.18: Static Dependency Functions

PROPERTY-IS-TRUE	PROPERTY-IS-FALSE
PROPERTY-IS-EQUAL	PROPERTY-IS-UNEQUAL
PROPERTY-IS-MORE	PROPERTY-IS-LESS
PROPERTY-IS-PRESENT	PROPERTY-IS-ABSENT
ATTRIBUTE-IS-PRESENT	ATTRIBUTE-IS-ABSENT

The most common early dependency uses the PROPERTY-IS-TRUE function to determine whether the has-canonical-front property of a concept is true. This information is used as the basis for determining which frame of reference to apply in solution generation as Section 1.1.5 and Section 2.3 discussed. For example, the FACING relation can apply only if an object has a front, so a DOG, which does, can face a TREE, which does not, but not vice versa. Similarly, the IN-FRONT-OF relation applies local frame of reference if an object has a front and global frame of reference if it does not. Therefore, in *the tree is in front of the dog*, the TREE is located within the line of sight of the DOG, but in *the dog is in front of the tree*, the DOG is south of the TREE.

[12]A *situation* is considered in this project as a degenerate form of a context, where the latter is based on a particular combination of concepts and the former on certain states of their internal definitions [114, 106]

5.6.2.2 Late Dependencies

A *late dependency* consists of one or more conditional expressions that are evaluated during inference generation as Section 5.5.2 outlined. Each conditional expression uses one of the *dynamic dependency functions* in Table 5.19 or one of the static dependency functions in Table 5.18 to determine whether an inference rule applies in a particular situation. These functions are considered dynamic because they depend on the particular, unique solution from many solutions for a description. The static dependency functions can also be evaluated at this time because the semantic network is always available.

Table 5.19: Dynamic Dependency Functions

IS-IN-FIELD	DIMENSION-IS-EQUAL
FIELDS-OVERLAP	DIMENSION-IS-LESS
	DIMENSION-IS-MORE

The most common late dependency uses the IS-IN-FIELD[13] function to determine whether one object is positioned in a certain way with respect to another object. This information is used as the basis for determining which inferences to generate as Section 5.5.2 discussed. For example, the NEAR inference can apply only if one object is reasoned to be within the contextually appropriate vicinity of another.

[13]Section 5.7.2 introduces fields.

Dynamic and static dependency functions can be nested to form compound conditional expressions. However, only intersection can conjoin them, which greatly simplifies their evaluation in the reasoning engine and is also consistent with the conjunctive normal form of a Horn clause [106]. For example, the `LOCAL-IN-FRONT-OF` inference is generated if the `has-canonical-front` property of object B is true and object A is located in the front field of object B. In contrast, the `GLOBAL-IN-FRONT-OF` inference is generated if the `has-canonical-front` property of object B is false and object A is located in the front field of object B. Thus, *the tree is in front of the dog* generates the former and *the dog is in front of the tree* the latter.

5.7 Representation by Constraints

The underlying formalism of all representation and reasoning in this project is based on *interval constraints* and *field constraints* to represent all aspects of the dimensions, position, and orientation of objects. This declarative approach, which Section 5.3.4 introduced, allows the knowledge engineer to focus on the form of the solution rather than on the procedural details of computing it. Constraints generally map qualitative features of a description to quantitative data structures for computational processing by the spatial reasoning engine, which Chapter 6 will discuss.

5.7.1 Interval Constraints

Interval constraints apply to reasoning over dimensions only. They provide the framework to map a vague, underspecified English description of size to plausible numerical values in the contextually appropriate dimensions of height, width, and/or depth for any object. This mechanism utilizes two types, *plausibility interval constraints* and *attribute interval constraints*, and extends the discussion on properties and attributes in Section 5.5.1.2.

5.7.1.1 Plausibility Interval Constraints

A plausibility interval constraint defines a function to calculate a contextually appropriate value for a single dimension of an object. As all objects have three dimensions, each requires three definitions. This section extends the discussion on range properties in Section 5.5.1.2.1.2. A plausibility interval is similar to a *fuzzy membership function* in fuzzy logic [64]. It defines a range of values that reasonably correspond to sizes for the lower, nominal, and upper limits of an object. Its role is to accept from an attribute interval constraint (see Section 5.7.1.2) a contextually appropriate binding from an English adjective of size. For example, *small* might bind to the lower limit of the plausibility interval for the `height` property of an object. The definition of a plausibility interval consists of a *discrete interval*, a *distribution coefficient*, a *bandpass filter*, and a *disproportionality limit*.

5.7.1.1.1 Discrete Interval

A plausibility interval is a continuous function with five discrete points:

1. *Absolute minimum value*: the lower limit of a dimension for the smallest real-world value that an object should ever exhibit.
2. *Suggested minimum value*: the lower limit of a dimension for the smallest real-world value that an object typically exhibits.
3. *Nominal value*: the default or average value of a dimension that an object exhibits if its size is not stated.
4. *Suggested maximum value*: the upper limit of a dimension for the largest real-world value that an object typically exhibits.
5. *Absolute maximum value*: the upper limit of a dimension for the largest real-world value that an object should ever exhibit.

Figure 5.10 shows the discrete interval for a GIRAFFE. The values for animals are much easier to define than for inanimate objects because nature dictates their intervals. Also, many resources conveniently document this information. To simplify the calculations further, all animals are assumed to be adult males unless otherwise specified.

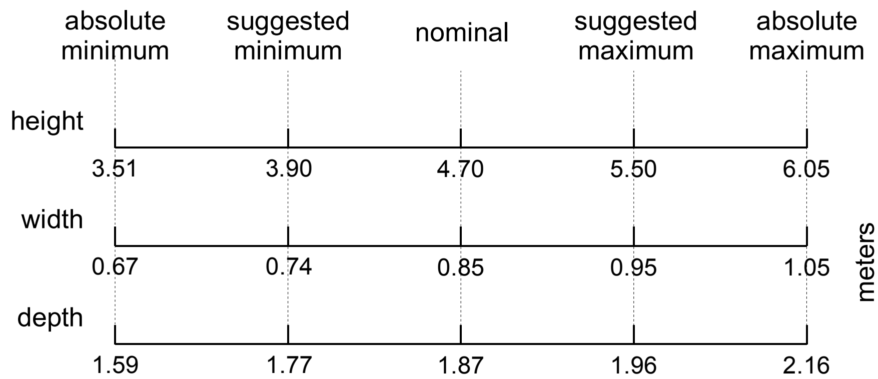


Figure 5.10: Discrete Intervals for GIRAFFE

5.7.1.1.2 Distribution Coefficient

The suggested minimum, nominal, and suggested maximum points on the interval lines in Figure 5.10 correspond respectively to the appropriate English adjectives for *least*, *average*, and *most* size in a particular dimension. Linguistic qualification of size is inherently vague, so the correspondence should not be exactly one to one; e.g., not every average GIRAFFE is precisely 4.70 meters in height. To introduce realistic variation in the size values, a modified random Gaussian distribution is defined such that the mean of the curve is centered at the corresponding discrete point, and the standard deviation is adjustable through a *distribution coefficient* to steepen or flatten the default probability curve as Figure 5.11 shows [101]. The result is a random value that is consistent with the expected size while accommodating linguistic vagueness and exhibiting real-world uncertainty.

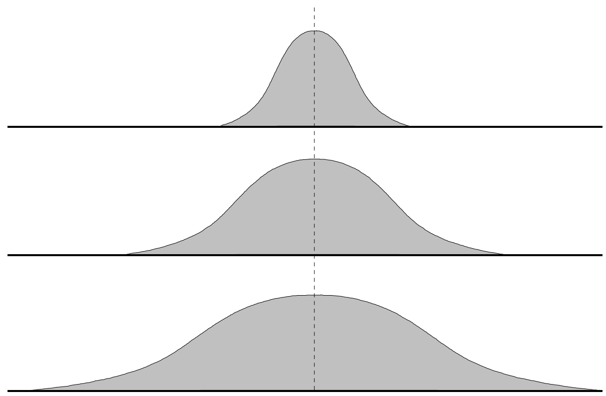


Figure 5.11: Distribution Coefficients

5.7.1.1.3 Bandpass Filter

For all random values in a standard Gaussian distribution, 68 percent are within 1.0 standard deviations of the mean. This property produces a realistic distribution and thus useful values in the majority of cases, but it does not prevent outliers in extreme cases.[14] A value that is too far from the mean encroaches on the next discrete point and corrupts its correspondence to the English adjective. A *bandpass filter* solves this problem by applying a simple, adjustable clamping function over the result of the Gaussian random function. Thus, any random value outside this local interval is set to the nearest legal value as Figure 5.12 shows. The same role is played over the global interval by the absolute minimum and maximum values in Figure 5.10 to prevent sizes in excess of the defined real-world limits.

[14]The curve extends to infinity in both directions.

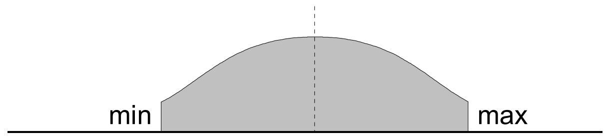


Figure 5.12: Bandpass Filter

5.7.1.1.4 Disproportionality Limit

The final component of a plausibility interval, the *disproportionality limit*, influences the plausibility intervals for the two other dimensions of an object. It dictates how far one dimension can deviate from nominal before the other dimensions must be adjusted to maintain realistic proportionality in an object. For example, *the snake is long* implies that the `length` dimension of the SNAKE is near its suggested maximum. The `height` and `width` dimensions are not addressed in this description, so they would default to roughly their nominal values. The resulting proportions may not correspond well with the real-world spatial behavior of a snake: greater length usually implies greater height and width as well, so calculating the length only would produce an unrealistic, spaghetti-like snake!

5.7.1.2 Attribute Interval Constraints

An *attribute interval constraint* defines the binding of a contrasting pair of English adjectives of size to one or more plausibility interval constraints, each of which defines how to generate a reasonable value for a single dimension of an object based on the interpretation of this adjective. This section extends the discussion on attribute intervals in Section 5.5.1.2.2.

Contrasting adjectives of size—basically antonyms—define a simple interval from least to most of the quality they describe; e.g., *short ... tall* for *height*, or *small ... big* for *height*, *width*, and *depth*. Each linguistic interval in Table 5.5 maps directly onto the appropriate plausibility intervals for height, width, and depth such that the first adjective binds to their suggested minimum values and the second to their suggested maximum values in Figure 5.13, respectively. Adjectives of size in this project also may be qualified with *very* to intensify their interpretation by nudging the suggested values closer to their absolute limits.

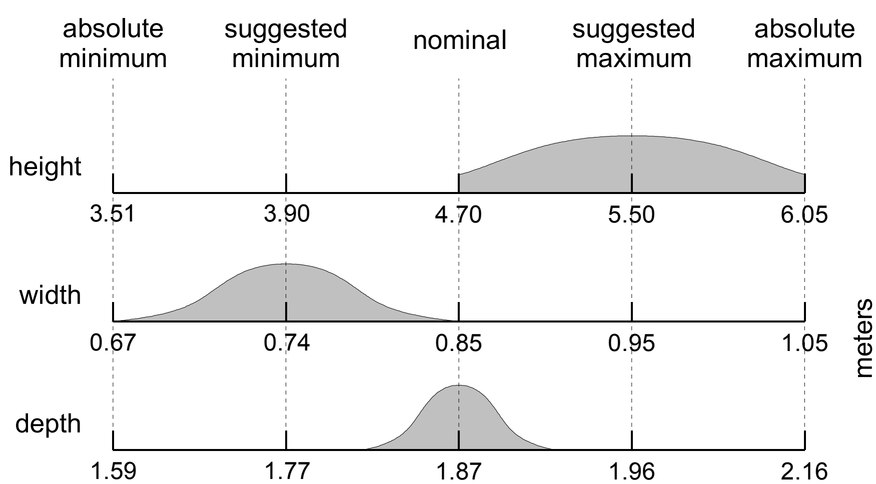


Figure 5.13: Attribute Interval Constraints

Adjectives of size may be ambiguous in their application to dimensions; e.g., *short* ... *long* for *length* in SNAKE, but *short* ... *tall* for *height* in GIRAFFE. As long as the different definitions reside in different concepts as they do here, no ambiguity exists because the interpretations are independent of each other.

5.7.2 Field Constraints

Field constraints apply to reasoning over position and orientation only. They provide the framework to map a vague, underspecified, uncertain English description of where an object is located and/or is facing to contextually appropriate numerical values of horizontal position and 360-degree directional orientation in a two-dimensional virtual world.[15] A field constraint consists of two components: a *geometry* to define its absolute interpretation and a *topography* to define its probable interpretation.

5.7.2.1 Geometry

The *geometry* of a field constraint limits where one object can appear with respect to another in a relation. For example, in *the tree is in front of the dog*, the TREE must be located somewhere within the front field of the DOG. This field extends from the center of the DOG (D) outward as a cone or *conical*

[15]Although the world is internally represented and graphically rendered in three dimensions, the spatial reasoning engine does not take full advantage of the vertical dimension [35]. Section 6.1.1.3 will discuss this issue.

frustum [97]. However, as the geometry is a two-dimensional top-view projection, this cone is actually flat as depicted from a top view in Figure 5.14.

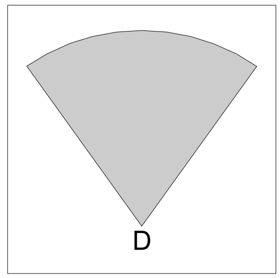


Figure 5.14: Two-Dimensional Frustum

This projection lends itself directly to a polar representation of cells that are referenced as the intersection of a sector and a cylinder as Figure 5.15 shows. A position is considered plausible if and only if it is located within the area of the enabled cells.

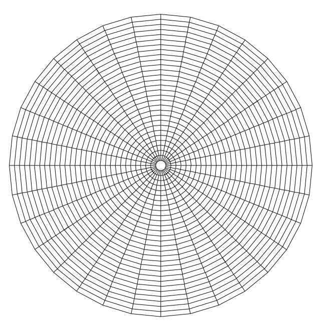


Figure 5.15: Field Projection

Empirical testing suggests that 32 sectors and 100 cylinders[16] provide adequate resolution in this project. The world is 100 meters square, so each

[16]For clarity, the projections in this chapter do not show all the cylinders.

cylinder is by default 1.0 meter thick. The width of each sector varies depending on the cylinder it intersects; i.e., smaller near the center and greater toward the fringe. This property is advantageous because it maps realistically to real-world uncertainty as a function of distance [35, 56, 15]. For example, the depiction of *the cat* (C) *is in front of the dog* (D) *and <u>near</u> the dog* in Figure 5.16a allows little freedom in how far the CAT can be to either side of the centerline of the DOG; whereas *the cat is in front of the dog and <u>far</u> from the dog* in Figure 5.16b allows significant freedom.[17]

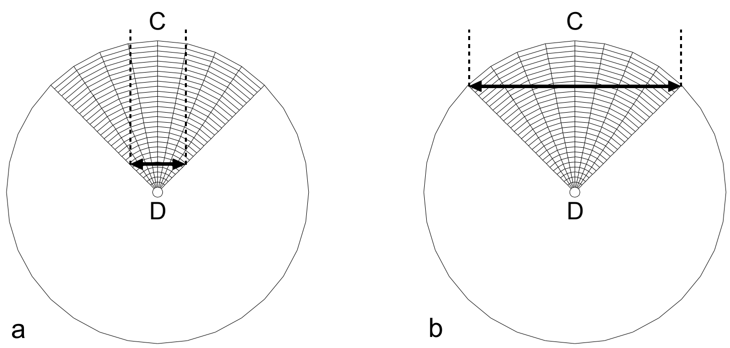


Figure 5.16: Uncertainty as a Function of Distance

More cells would provide finer resolution, but such level of detail is unnecessarily precise for the imprecise nature of the issues that this project investigates. Similarly, fewer cell would provide coarser resolution with less control over the interpretations. In theory, this polar projection supports almost infinite variety in

[17]More precisely, the *relative* freedom in degrees is constant because the same angle subtends the arc over the near and far distances. The *absolute* freedom in meters varies as a function of distance.

the definition of fields.[18] In practice, however, only two classes of geometries, *facets* and *rings*, were found necessary to define all desired spatial relations.

5.7.2.1.1 Facet Geometry

A facet is basically a pie slice that originates from the center of the projection, where the object that defines it is located. Each facet in the definition of a concept plays at least one of the spatial roles for local relative position relations in Section 5.5.1.3.1 and for relative position inferences in Section 5.5.2.3.2. For example, the front facets in Figure 5.17a and Figure 5.17b associate with the canonical front of a concept and play a role in their IN-FRONT-OF and DIRECTLY-IN-FRONT-OF relations, respectively, as well as in the FACING and DIRECTLY-FACING relations.

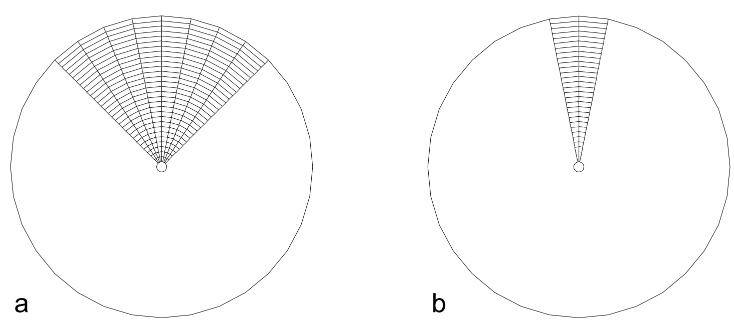


Figure 5.17: Facet Geometry of FRONT and DIRECT-FRONT

[18]The 3,200 cells (32 sectors × 100 rings) support ${}_{3200}C_{3200}$ combinations!

5.7.2.1.1.1 Real-World Foundation

The definition of each facet is based on a *human's* perception of how a concept behaves spatially, not on the concept's perception.[19] This speciocentric bias comes from the observation that humans describe scenes from their own point of view for consumption by other humans [62, 46, 106]. A facet is not a *field of view* as defined by the physiological characteristics of the eyes of an animal [12, 106]. For example, a GOLDEN-EAGLE has a narrow focus because its eyes are located at the front of the head and face forward to improve stereoscopic vision; whereas an IGUANA has a wide focus because its eyes are located at the sides of the head and face outward to improve peripheral vision.

In this project, such issues of physiology play no role in how each object is perceived spatially by humans because the viewer of a scene is merely an external observer. Field of view would come into play, however, if the viewer were part of the scene; e.g., *the dog is in front of* _me_. It would also if the objects could be queried on their perception of the world; e.g., in the vicarious sense, *If I were the iguana, what would I be able to see*? Such a level of representation would be useful in a simulation of the interactive dynamic behaviors of objects, but it is not for this project [127].

[19] Assuming that it is even capable of perception. Obviously inanimate objects and plants are not, so this distinction applies just to animals.

Not all physiological or physical characteristics can be ignored in spatial reasoning with fields, however. The size of an object plays a significant role in contextually determining how large its fields should be. The geometry defines the shape of the field but not its size because this information derives from the particular dimensions that an object is reasoned to have. Thus, if the identical front facet is shared by two objects with significantly different dimensions, for example, an ELEPHANT (E) and a RABBIT (R), the facets on the surface of the world differ contextually as Figure 5.18a and Figure 5.18b respectively show.

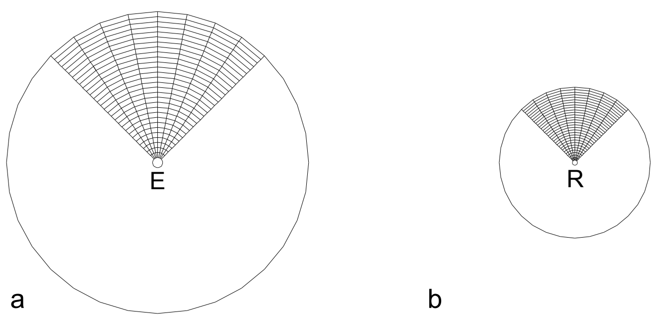


Figure 5.18: Front Facet for ELEPHANT and RABBIT

This contextual interpretation is consistent with real-world spatial reasoning [56]. The ELEPHANT, as a large object, naturally accommodates more area "in front of" it than the RABBIT can as a small object. As Figure 5.19a demonstrates, the RABBIT can shift laterally by a large amount while remaining in front of the ELEPHANT. In contrast, the ELEPHANT at the same relative distance (midrange) in Figure 5.19b can shift laterally by only a slight amount while remaining in front of the RABBIT. Only the scale differs between the two, not the number of cylinders.

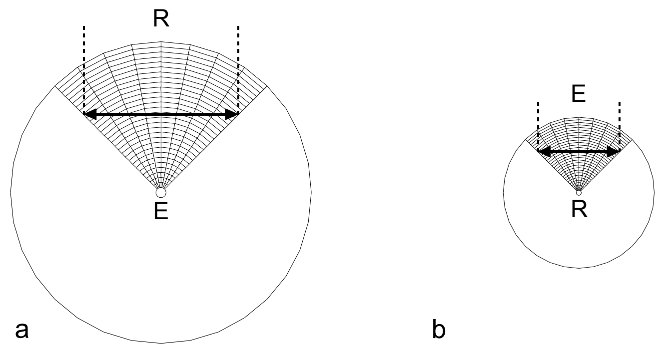


Figure 5.19: Lateral Freedom of ELEPHANT and RABBIT

5.7.2.1.1.2 Orientation Anchor

The final component of a facet geometry is its *orientation anchor*. So far in this discussion, the centerline of a projection has been assumed to align with the centerline of the object that defines it as Figure 5.20a shows. This *relative orientation*, which most fields use, anchors the 12-o'clock position of the projection to the 12-o'clock position of the object. Thus, for example, the front field of a DOG will always rotate with it to remain "in front." The set of eight fields for cardinal and intercardinal directions, or *compass rose*, in the world use an *absolute orientation* that never rotates [85]. The centerline of these projections always points north such that the 12, 3, 6, and 9-o'clock positions correspond to north, east, south, and west, respectively, as Figure 5.20b shows.

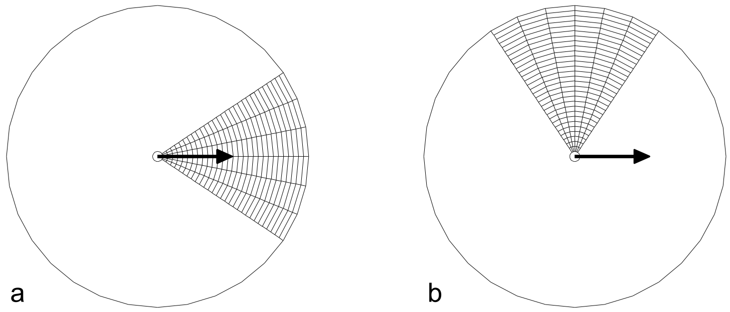


Figure 5.20: Relative and Absolute Orientation

Thus, for example, the east field of a DOG will always face east regardless of where the DOG is facing, and any object positioned in this field is interpreted as EAST-OF the DOG. Furthermore, as the center of the projection is co-located with the center of the DOG, the contextual interpretation depends on the position of the DOG. This floating-position behavior applies to all objects except one, WORLD-CENTER, which is a special object that is always implicitly present in a description but cannot be referenced. Its position and absolute orientation is fixed in the exact center of the world, but in all other respects, it behaves like any other object. Its role is to define the compass rose of the world for the global relative position relations in Section 5.5.1.3.1.1; e.g., *the cat is in the northeast*.

5.7.2.1.2 Ring Geometry

A ring surrounds the center of a projection and defines a measure of relative distance from the object there that defines it. Each ring in the definition of a concept plays a role in the relative distance relations in Section 5.5.1.3.1.3

and the relative distance inferences in Section 5.5.2.3.2.3. For example, the two rings in Figure 5.21a and Figure 5.21b correspond to the default NEAR and FAR-FROM relations, respectively.

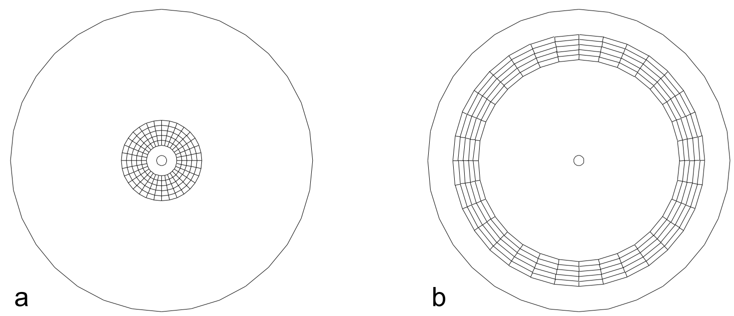


Figure 5.21: Ring Geometry for NEAR and FAR

The same real-world foundation that Section 5.7.2.1.1.1 discussed for facets applies to rings as well. The most important point is that the width of the cylinders that comprise a ring varies contextually with the dimensions of the object they surround. In other words, for the same definition of a ring, the absolute inside and outside radii in meters may differ. Figure 5.22a and Figure 5.22b demonstrate this effect on a FAR ring for an ELEPHANT (E) and a RABBIT (R), respectively.

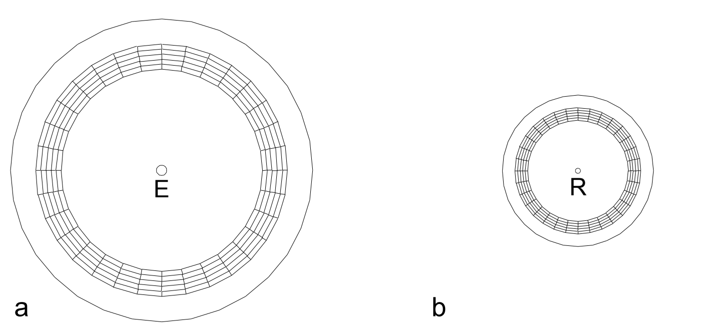


Figure 5.22: FAR Rings for ELEPHANT and RABBIT

This contextual interpretation is consistent with real-world spatial reasoning [56]. The ELEPHANT, as a large object, naturally accommodates a more distant and liberal horizon than the RABBIT can as a small object. As Figure 5.23a demonstrates, the RABBIT can shift in range by a large amount while remaining far from the ELEPHANT. The ELEPHANT, as Figure 5.23b demonstrates in contrast, can shift in range by only a slight amount while remaining far from the RABBIT.

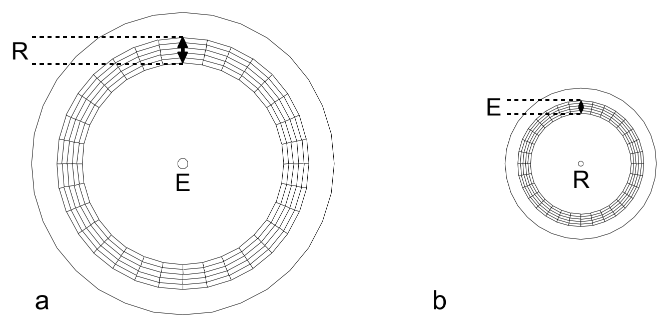


Figure 5.23: Distance Freedom for ELEPHANT and RABBIT

The contextually appropriate width of the cylinders is determined by scaling the first cylinder to the *bounding cylinder* of the object [15, 87]. This cylinder is defined as the minimum radius that fully encloses the object. In addition to establishing the scale factor, it also serves as the interface between the *interior* and *exterior regions* of the object. The first cylinder can thus be associated with the IN, ON, and INSIDE relations as Figure 5.24a shows, and the second through 100th are available as appropriate for the OUTSIDE, ADJACENT-

TO, NEAR, MIDRANGE-FROM, FAR-FROM, and AT-FRINGE-OF relations as Figure 5.24b shows.

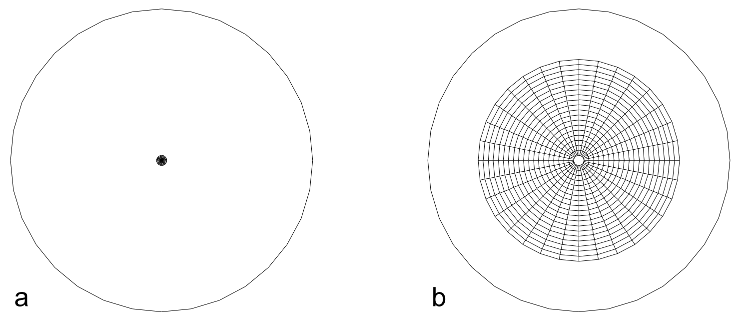


Figure 5.24: INTERIOR and EXTERIOR Rings

5.7.2.2 Topography

Whereas the *geometry* of a field constraint strictly dictates which positions are *sanctioned*, the *topography* loosely suggests which positions are *recommended* [27]. In other words, the topography overlays a probability distribution on the geometry. The basis of this distribution is the observation that interpretations favor positions in the "core" of a field over those at the periphery [103, 94]; e.g., "in front of" is more likely to be interpreted as somewhere directly down the middle of a facet than off to a side.

Each field defaults to a Gaussian distribution and accepts different standard deviations to steepen or flatten the curve as Figure 5.11 illustrates. Uniform, triangular, concave, and convex distributions are also supported for experimentation purposes, but they were found to be less effective and more difficult to justify than a straightforward bell curve. The standard Gaussian distribution is one dimensional, which makes it inadequate for producing random positions over the two dimensions of a field. To extend it into a second dimension, a discrete array of Gaussian random functions is projected along the perpendicular axis of the field [68]. A uniformly distributed random number selects the array element, which corresponds to one dimension, then a Gaussian-distributed random number selects a position on the other dimension. This algorithm produces the tent-like two-dimensional distribution in Figure 5.25 (tilted for clarity). The axes are intentionally undefined here because, as the next two sections will discuss, the uniformly distributed axis is automatically morphed to fit the geometry of a field.

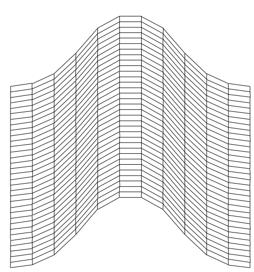


Figure 5.25: Two-Dimensional Gaussian Distribution

5.7.2.2.1 Facet Topography

Facet topography maps directly onto the facet geometry in Section 5.7.2.1.1. Its role is to favor random positions down the center of the facet over those at the lateral periphery without biasing distance. The distribution in Figure 5.25 is automatically morphed along the longitudinal axis, and for each element in the array of Gaussian random functions, the upper and lower limits are sandwiched between the bounds of the lateral axis. The result is the wedge-shaped, tent-like, two-dimensional distribution in Figure 5.26a for a typical `front` facet and in Figure 5.26b for a typical `direct-front` facet.

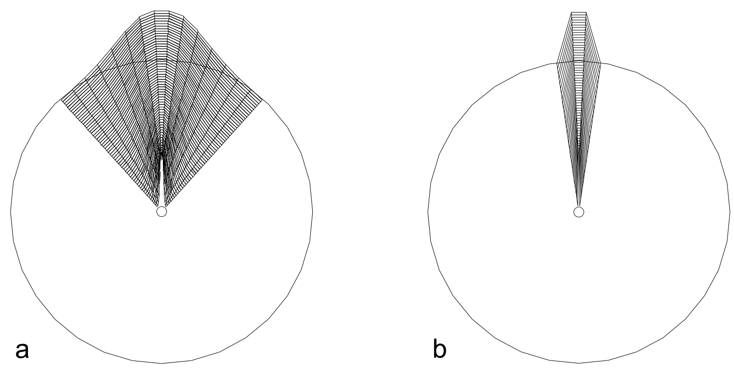


Figure 5.26: Facet Topography for `FRONT` and `DIRECT-FRONT`

This contextual interpretation is consistent with real-world spatial reasoning because freedom in uncertain lateral positions increases as a function of the distance [56]. In other words, the farther an object is from the center of the projection, the farther it may appear left or right of the centerline while still satisfying the relation "in front of." The description has no effect on which positions along the centerline are more favored as it lacks such information.

5.7.2.2.2 Ring Topography

Ring topography maps directly onto the ring geometry in Section 5.7.2.1.2. Its role is to favor random positions uniformly around the ring between the minimum (proximal) radius and maximum (distal) radius. The distribution in Figure 5.25 is automatically morphed around the ring to sandwich each element in the array of Gaussian random functions. The result is the donut-shaped, tent-like, two-dimensional distribution in Figure 5.27a for a typical `midrange` ring and in Figure 5.27b for a typical `adjacent` ring. All topography figures are tilted here for perspective.

This contextual interpretation is consistent with real-world spatial reasoning because ordinary distance descriptions are inherently imprecise and open to wide interpretation in both distance and azimuth from the center of the projection [56]. Only the freedom in distance plays a role, however, because any azimuth is equally plausible. Bounding this impreciseness as described provides a reasonable estimate of distance without biasing the azimuth. In other words, a `midrange` ring might constrain an object to appear 20 meters plus or minus 5 meters from the center, but it has no effect on which positions along the particular circle are more favored. Descriptions of distance lack such information.

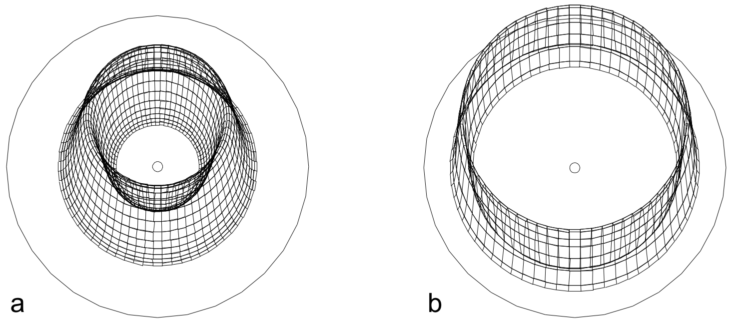


Figure 5.27: Ring Topography for MIDRANGE and FAR

5.7.2.3 Spatial Interaction with Fields

Section 5.7.2 covered the declarative details of what fields are and how they are defined. This section extends that discussion into interactions between fields, which is what occurs in most aspects of spatial reasoning over position and orientation in this project. Without loss of generality, fields can be represented as overlapping planes and manipulated using standard *intersection*, *union*, *symmetric difference*, and *complement* set operations [26, 56, 93, 15, 106, 81, 111]. All these operations have been implemented and evaluated in this project. However, only intersection is currently supported by the input parser for the English description because the others were found unnecessary to satisfy the goals.

5.7.2.3.1 Field Intersection

Intersection operates on any combination of fields in which *all* spatial constraints must be satisfied simultaneously; e.g., *the dog is north of the lake <u>and</u> in front of the cat <u>and</u> midrange from the cat <u>and</u> facing away from the tree*. Only the cells that are available in all projections contribute to the intersected result, as Figure 5.28a shows for *the dog is in front of the cat <u>and</u> midrange from the cat*. The intersection operation applies first to the geometry of each projection to produce a new geometry. The topography is then recalculated over this new geometry as the *product* of the probabilities at each cell over the old geometries. Thus, the new geometry sanctions which cells are available in the interpretation, and the new topography suggests which are preferred [106]. The volcano-like topography in Figure 5.28b is consistent with real-world spatial reasoning because it simultaneously combines the uncertainty of lateral (sideways) position with the uncertainty of longitudinal (distance) position [56]. The compositional peak is most preferred; whereas the perimeter is least preferred.

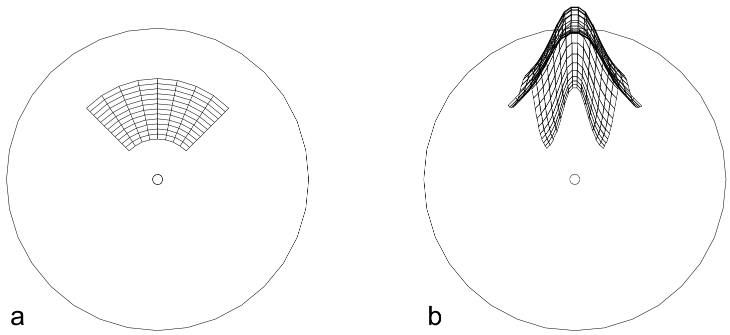


Figure 5.28: Intersection of FRONT and MIDRANGE

5.7.2.3.2 Field Union

Union operates on any combination of fields in which *any* spatial constraints must be satisfied simultaneously; e.g., *the dog is north of the lake* <u>*or*</u> *in front of the cat* <u>*or*</u> *midrange from the cat* <u>*or*</u> *facing away from the tree*. Only the cells that are available in at least one of the projections contribute to the unioned result, as Figure 5.29a shows for *the dog is in front of the cat* <u>*or*</u> *midrange from the cat*. The union operation applies to the geometry the same way as described for the intersection operation. The topography, however, is then recalculated over this new geometry as the *maximum* of the probabilities at each cell over the old geometries. The topography in Figure 5.29b is consistent with real-world spatial reasoning because it simultaneously addresses the uncertainty of lateral position independent of the uncertainty of longitudinal position [56]. The ridge of the facet or of the ring is most preferred; whereas the perimeter of each is least preferred. There is no compositional peak because the two fields are independent.

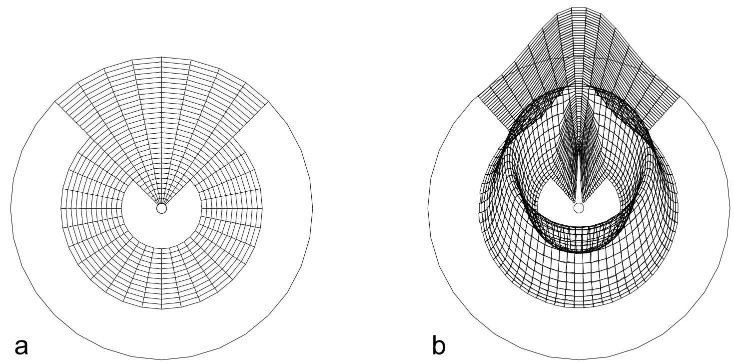


Figure 5.29: Union of FRONT and MIDRANGE

5.7.2.3.3 Field Symmetric Difference

Symmetric distance, also known as *exclusive-or*, operates on any combination of fields in which *one and only one* spatial constraint can be satisfied; e.g., *the dog is either north of the lake <u>or</u> in front of the cat <u>or</u> midrange from the cat <u>or</u> facing away from the tree*. Only the cells that are available in exactly one of the projections contribute to the symmetric-difference result, as Figure 5.30a shows for *the dog is <u>either</u> in front of the cat <u>or</u> midrange from the cat <u>but not both</u>*. The symmetric-difference operation applies to the geometry the same way as described for the intersection and union operations. The topography, however, is then recalculated over this new geometry as the *actual* probability at each cell from the selected old geometry.

The topography in Figure 5.30b is generally[20] consistent with real-world spatial reasoning because it simultaneously addresses either the uncertainty of lateral position or the uncertainty of longitudinal position, but not both [56]. As with the union operation, the ridge of the facet or of the ring is most preferred; whereas the perimeter of each is least preferred. There is no compositional peak because the two fields are mutually exclusive.

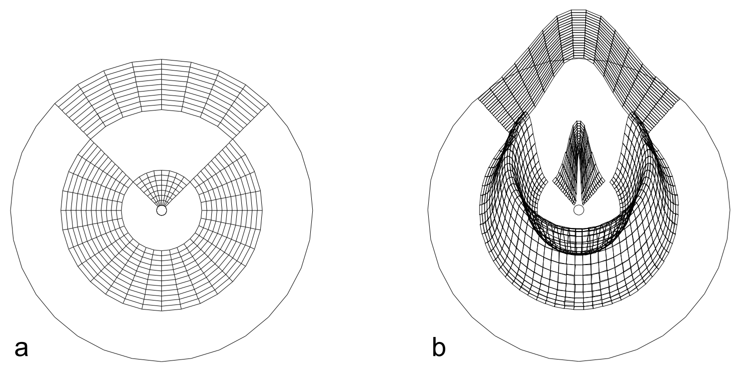


Figure 5.30: Symmetric Difference of FRONT and MIDRANGE

5.7.2.3.4 Field Complement

Complement operates on a single field such that the *one* spatial constraint must be satisfied by its opposite interpretation; e.g., *the dog is <u>not</u> north of the lake*. Only the cells that are *unavailable* in the projection contribute to the complemented result, as Figure 5.31 shows for *the dog is <u>not</u>*

[20]The precipitous drop at the boundaries between the two fields should be smooth gradients. The topography generator does not take this transition into account because symmetric difference is not officially supported in this project.

in front of the cat and for *the dog is <u>not</u> midrange from the cat*, respectively. The complement operation applies to the geometry only because the topography is uniform. This layout is consistent with real-world spatial reasoning because it addresses the uncertainty of lateral and longitudinal position in terms of the opposite interpretation [56]. That is to say, *not in front of* implies a position uniformly anywhere on the projection except *in front*, and *not midrange from* implies uniformly anywhere except *midrange*. As the preferred positions on each projection were the ones that were discarded, there is no topography.

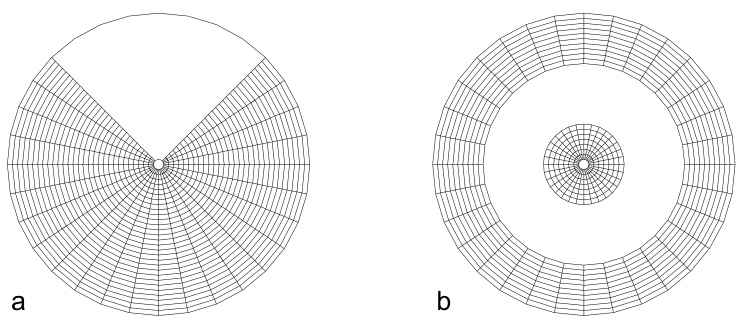


Figure 5.31: Complement of FRONT and MIDRANGE

5.7.2.3.5 Other Field Operations

Finally, as the solution to spatial interactions over fields in this project is closed under intersection, union, symmetric difference, and complement, DeMorgan's Laws can be used to manipulate fields even further. However, this functionality is of theoretical interest only because ordinary descriptions of spatial scenes in the real world seldom—if ever—exhibit such complex logical interactions [119, 26].

5.7.2.4 Catalog of Fields

Every relation and inference in this project binds to one of the 34 fields defined in the knowledge base. As there are 50 relations and 56 field-based inferences, many fields play more than one role. Nevertheless, all are easily categorized as *local*, *global*, or *distance* fields. The following three sections illustrate the default geometry of all fields. Concepts may refine them according to their needs, but the differences are generally minor.

5.7.2.4.1 Local Fields

Local fields are tightly associated with an object in the world. Each projection centers on the object and orients itself toward the front of the object. The fields in Figure 5.32 are used by the local relative position relations in Table 5.8, the local relative position inferences in Table 5.14, and the global relative position inferences in Table 5.15.

5.7.2.4.2 Global Fields

Global fields are associated with either an object in the world or the special WORLD-CENTER object (see Section 5.5.1.3.1.1 and Section 5.7.2.1.1.2).

Each projection centers on the object and orients itself toward the north. The fields in Figure 5.33 are used by the global relative position relations in Table 5.6, the quasi-absolute position relations in Table 5.7, the absolute orientation relations in Table 5.10, the absolute orientation inferences in Table 5.12, and the global relative position inferences in Table 5.15.

5.7.2.4.3 Distance Fields

Distance fields are associated with an object in the world. Each projection centers on the object but has no orientation because it is symmetrical. The fields in Figure 5.34 are used by the relative distance relations in Table 5.9 and the relative distance inferences in Table 5.16.

5.7.2.5 Comparison with Other Field Models

The notion of fields in other projects varies considerably in name and definition depending on their use and emphasis; e.g., potential fields, typicality potential fields, continuum measures, acceptance areas or volumes, containment areas, influence areas, etc. [91, 30, 66] Nevertheless, they all share the properties of being either two- or three-dimensional geometric representations of space as Figure 5.35 from [50, 50, 59] and Figure 5.36 from [23, 59, 59] respectively illustrate. They may also impose preference upon certain positions with the fields and a "haze" factor for uncertainty about fuzzy bounds [15].

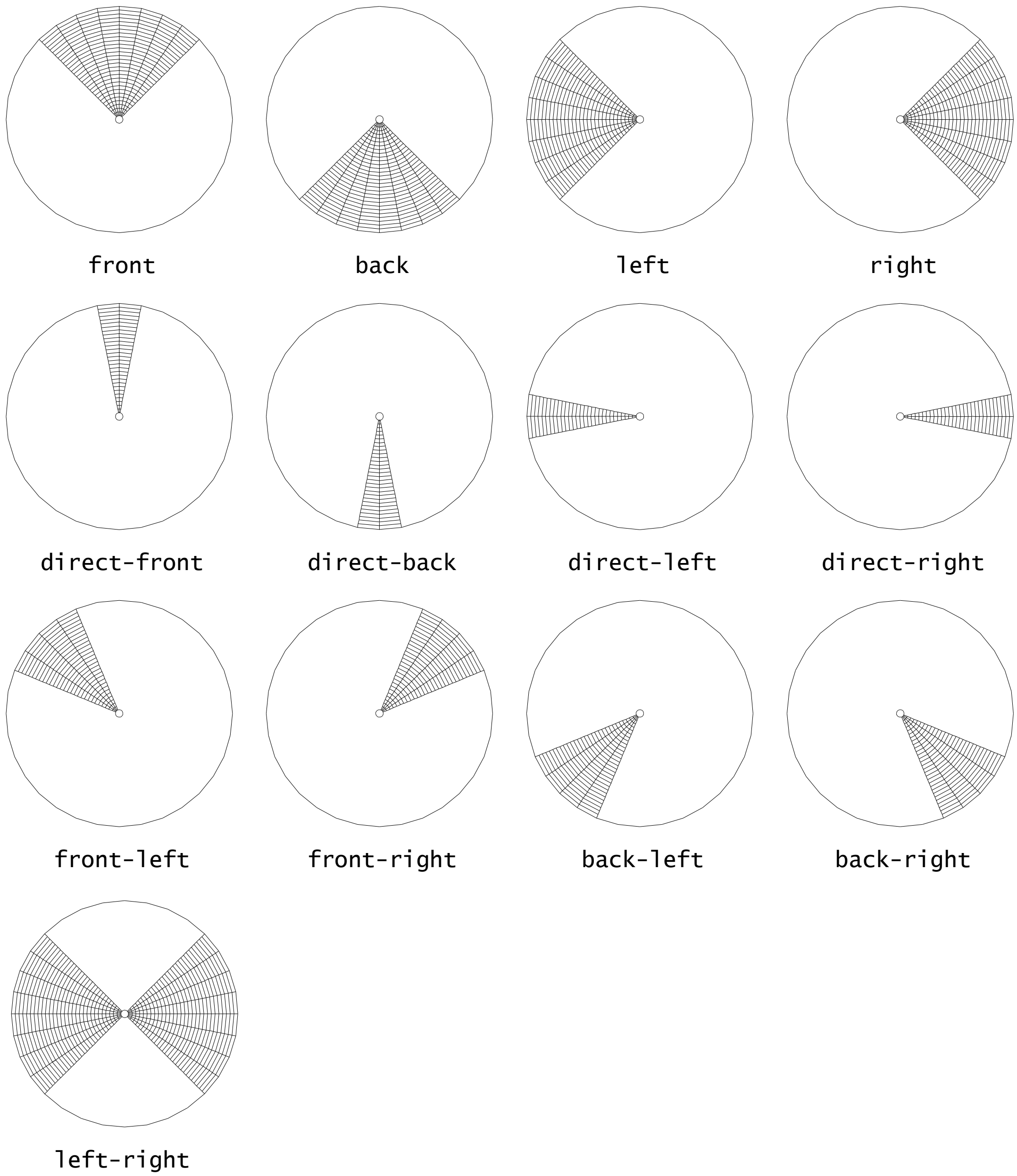


Figure 5.32: Local Fields

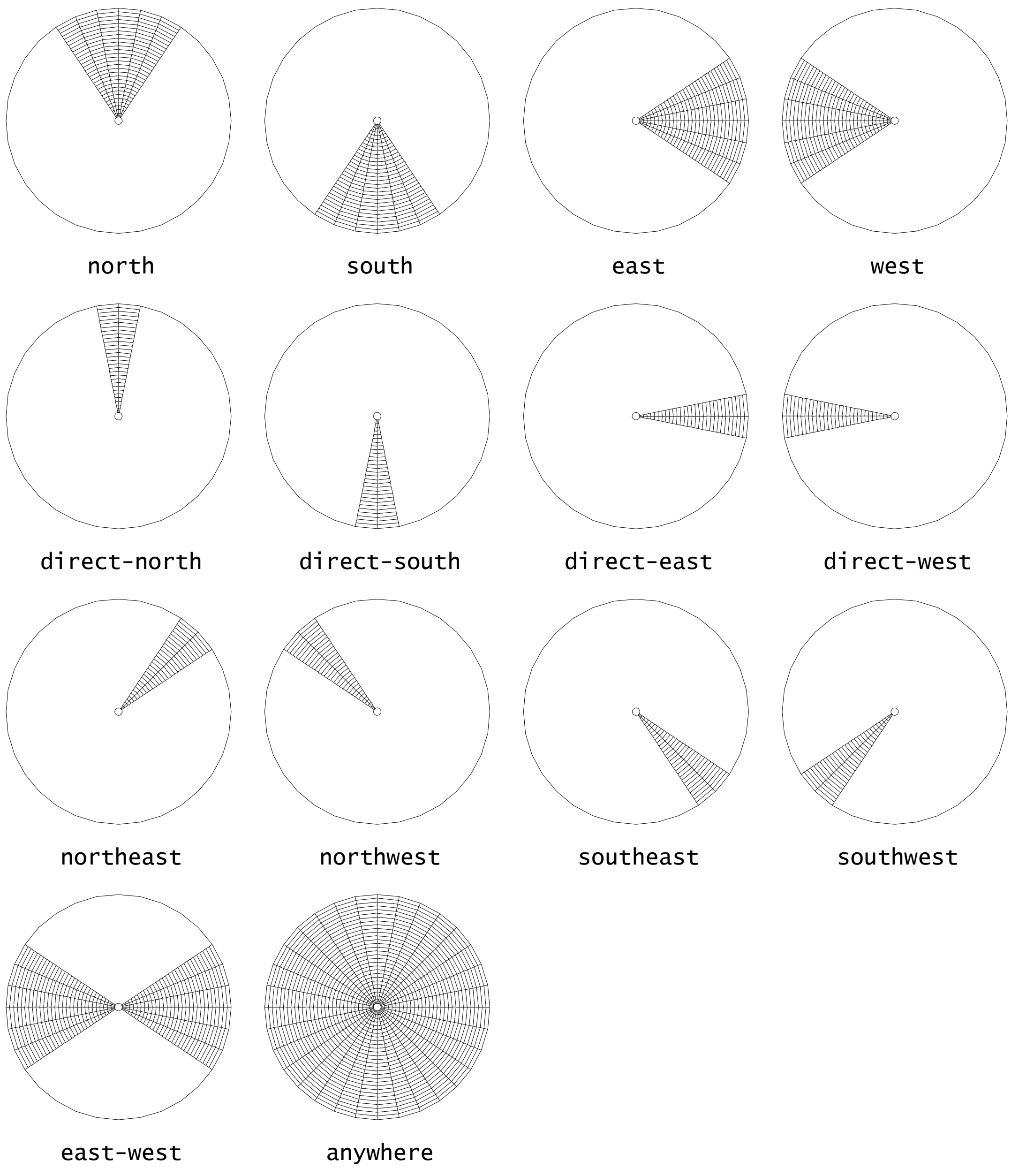


Figure 5.33: Global Fields

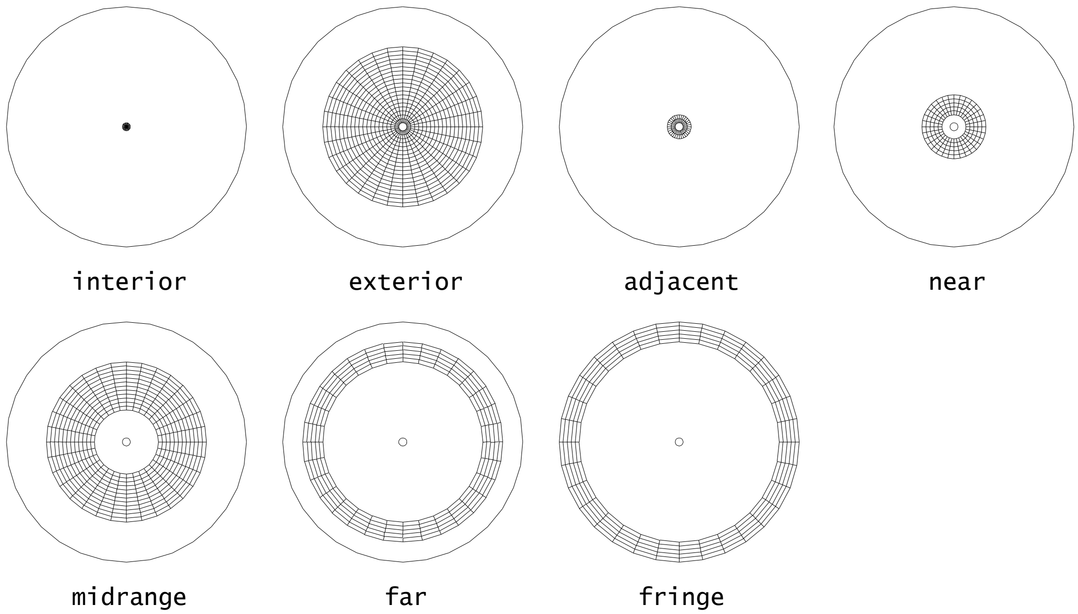


Figure 5.34: Distance Fields

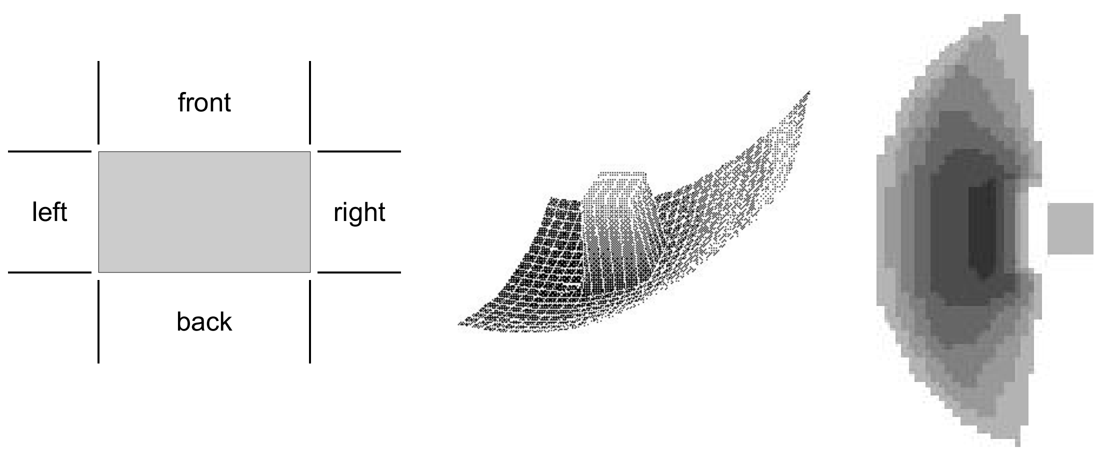


Figure 5.35: Sample Two-Dimensional Fields

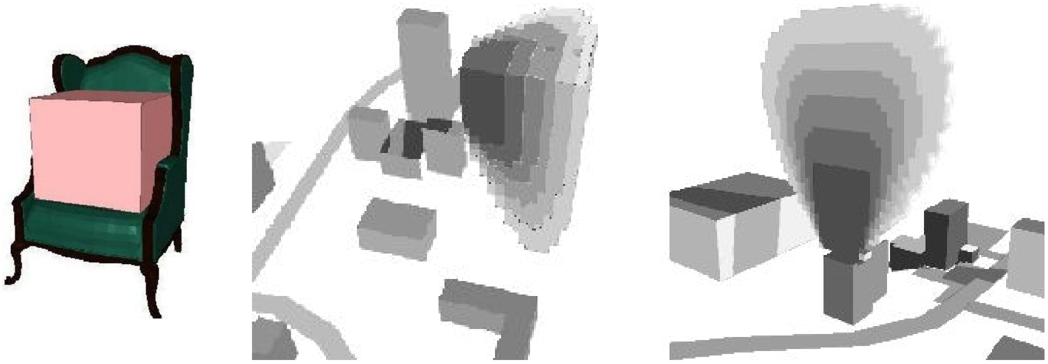


Figure 5.36: Sample Three-Dimensional Fields

Regardless of their form, it is difficult to argue against the usefulness, straightforward nature, and obvious intuitiveness of fields [85]. While it is true that establishing which fields a system should implement and defining their shape is somewhat of an art [90], they do not have to be a "rather cumbersome" approach as Hernández [55] claims. Granted, some formalisms arguably bear little resemblance to the problem, as Figure 5.37 exemplifies (from [36] and [93], respectively), but, as this project shows, a straightforward polar projection of cells need not exhibit such a weakness.

```
(< ?X ?Y
 (AND (Point-2d ?X)
      (Point-2d ?Y)
   (Or (< (X-Coord ?X)
          (X-Coord ?Y))
    (And (= (X-Coord ?X)
            (X-Coord ?Y))
     (< (Y-Coord ?X)
        (Y-Coord ?Y))))))
```

$$d_x = (x - x_0)$$
$$d_y = (y - y_0)$$
$$P_{\frac{X}{Y}} = P_{prox,\frac{X}{Y}} + P_{dir,\frac{X}{Y}}$$
$$P_{prox,\frac{X}{Y}} = \frac{K_{prox,\frac{X}{Y}}}{2} \sqrt{(d_x^2 + d_y^2 - L_{prox,\frac{X}{Y}})^2}$$
$$P_{dir,\frac{X}{Y}} = \frac{K_{dir,\frac{X}{Y}}}{2} d_x^2$$

Figure 5.37: Sample Definitions

To be fair, notation and elegance in knowledge representation are unimportant or even irrelevant—-content, expressiveness, and usefulness are what count most—but a non-intuitive representation with no clear visualization is harder for humans to define and manipulate [78, 112, 27, 53. 90, 54]. As for the criticism of a lack of theoretical elegance in "simplistic" approaches (such as geometric fields), Egenhofer [35] counters succinctly: "[i]f it is simple and solves the problem, then it is good."

5.8 Synopsis of Knowledge-Base Architecture

This chapter on the representation of implicit spatial knowledge so far spans 73 sections over 85 pages! The breadth and depth of its vast contents are difficult to grasp, especially from a declarative perspective alone because procedural details play an important role as well. This final section presents a synopsis of the chapter and leads into the next chapter on the spatial reasoning engine that processes the semantic network and knowledge base.

5.8.1 Form of Representation

The knowledge base is a declarative representation that defines constraints, as well as rules for when they apply. It is a structure of constraints in which each object defines a conceptual template for its corresponding instance(s) in the real world. The structure is hierarchical such that concepts are composed of the more general concepts above them (Figure 5.1). Two hierarchical organizations map the same concepts for different purposes (Figure 5.7): derived concepts map to a simplified taxonomy of the actual animal kingdom; abstract concepts map to a contrived framework of spatial behaviors and interpretations. Each concept defines its single semantic interpretation in terms of default attributes and relations that apply in the same way regardless of any context that other concepts in a description might impose. A concept may also define one or more pragmatic interpretations in terms of non-default attributes and relations that apply differently in specialized contexts. The mechanisms for identifying and applying these differences are concept matching, concept lineage matching, and conditional dependencies.

5.8.2 Role in Spatial Reasoning

The combined role of the semantic network and the knowledge base (Figure 5.2) is to build from a description a set of contextually appropriate constraints that the spatial reasoning engine can mechanically process without considering linguistic and knowledge-related complexities. This processing generates two types of results: solutions are plausible spatial interpretations of the objects in a description; inferences are additional spatial details that derive from a particular solution and contribute to a better understanding of the description.

5.8.2.1 Solution Generation

The role of the knowledge base in generating solutions is to build constraints that appropriately limit the plausible values for the dimensions, position, and/or orientation of objects in context. A constraint defines either an interval or a field. An interval constraint applies to dimensions by binding English adjectives of size to numerical functions that calculate plausible values for height, width, and/or depth (Figure 5.13). A field constraint applies to position and orientation by specifying the shape of a geometric area (Figure 5.28a) that an object must inhabit and which subareas within it are more likely in context (Figure 5.28b). Table 5.20 summarizes the constraints that this project supports.

Table 5.20: Summary of Constraint Classes

Constraint Class	Role
Attribute interval	Dimensions of an object
Global relative position relation	Position of one object with respect to position of another object in terms of the compass
Local relative position relation	Position of one object with respect to position of another object in terms of the intrinsic facets of the latter
Relative distance relation	Distance of one object from another object with respect to position and spatial conceptualization of the latter
Absolute orientation relation	Direction of an object in terms of the compass
Relative orientation relation	Direction of one object with respect to position of another object

5.8.2.2 Inference Generation

The role of the knowledge base in generating inferences is to apply rules that contextually derive from a particular solution new attributes and relations for the dimensions, position, and/or orientation of its objects with respect to each other [26]. Any new attributes or relations go back into the original semantic network to augment its basic interpretation with commonsense information that a human might infer from the solution; i.e., a form of *situation semantics* [114, 106]. Table 5.21 summarizes the inferences that this project supports, in addition to all the entries from Table 5.20.

Table 5.21: Summary of Inference Classes

Inference Class	Role
Relative dimension relation	Dimensions of one object with respect to dimensions of another object

6 SPATIAL REASONING

Together, the semantic network and knowledge base respectively supply the explicit and implicit knowledge that is needed to interpret a description [15]. Their combined role is to construct a set of interval and field constraints that declaratively define the form of an appropriate solution, as well as a set of inference rules to apply to it. All the complex issues of underspecification, uncertainty, vagueness, context, and frame of reference have already been addressed by the time these sets reach the spatial reasoning engine. The only processing that remains is to generate a solution that satisfies these constraints, then to generate any inferences that it satisfies in turn. The spatial reasoning engine has no interest in decisions that contributed to these sets. Therefore, it can blindly operate in the same mechanical way over *any* constraints and inferences for *any* description and does not need an intricate maze of complex, contextually dependent, conditional rules to perform its task.

6.1 Solution Generation

The purpose of and representation for solution generation in spatial reasoning have been addressed from various perspectives already[1]; only the procedural details of its implementation remain. The reasoning that generates a

[1]See Section 5.5.1 and Section 5.8.2.1.

solution in this project is *static*: the form of the solution is already known in advance, and nothing in the form changes during the reasoning or as a result of it [106]. If this reasoning were *dynamic*, the decoupling of the spatial reasoning engine from the knowledge base would not be possible because the representation would change during the reasoning and thus require reinterpretation with respect to the complexities that only the knowledge base can resolve [106, 114]. Humans do both static and dynamic reasoning to generate solutions to spatial descriptions [63].

6.1.1 Process Outline

The "reasoning" in this project is actually treated as a *constraint satisfaction problem*, which is similar in many respects to the way humans solve analogous problems [73, 77, 90, 11]. A typical approach to solving such a problem for the position and orientation of its objects employs the following general steps:

1. Place together into a pool to the side of the solution area all the objects that play a role.
2. Select one object from the pool as the initial object and add it to the solution area.

3. Select one object from the pool as the working object and add it to the solution area.
4. Satisfy all the constraints between the working object and the initial object by moving only the working object.
5. Select one object from the pool as the working object and add it to the solution area.
6. Satisfy all the constraints between the working object and the other objects in the solution area by moving only the working object. If this is not possible, return the working object and another object to the pool and repeat 5.
7. Repeat 3 until all the pool is empty, or give up after a certain number of attempts.

This (oversimplified) approach is greedy: it tests a potential solution for one object at a time against the solutions for the previous objects that have already been satisfied [5, 101]. If the current solution works, then it is accepted, and the next unsatisfied object is considered. If it does not, then some part of the previous solutions is changed in the hope that this change will satisfy the current object when it is eventually reconsidered. The process of advancing and retreating, known as *backtracking*, is a common approach for constraint satisfaction problems [73, 5, 56]. However, it is not efficient beyond a certain number of objects and/or total constraints (generally 7 and 11 in this project, respectively) because a significant amount of work is duplicated in a blind search for an overall solution that satisfies all constraints simultaneously. It also does not exhibit the property of *graceful degradation*, which humans do: if an overall solution cannot be found, often a less-than-perfect one can [72]. This process, known as *satisficing*, gradually relaxes the constraints to find an imperfect yet acceptable solution instead of none [56, 15, 11]. In contrast, if this approach fails to find a solution after a certain (arbitrarily defined) number of attempts, it generates nothing at all, declares that no solution exists, and exits. Furthermore, it justifies this behavior as a variation on the *closed-world assumption*: if it does not find a timely solution, then it assumes that none exists [53, 26, 4]. This check prevents an infinite loop in case a description is inadvertently unsatisfiable.

There are many other approaches to constraint satisfaction that are more efficient and effective [5]. A single key fact justifies why this one is acceptable in this project: humans cannot reasonably process more than a few objects and constraints in a spatial description, so there is no reason to do better [63]. In other words, realistic descriptions of spatial scenes do not exceed the capabilities of this admittedly handicapped approach.

6.1.1.1 Semantic Network as Dependency Graph

Recall that the semantic network contains the explicit definitions of objects in a description, as well as their attributes and interrelations. By the time it reaches the spatial reasoning engine, it also contains formalized constraint rules for the implicit contextual interpretation of all these components. This complete form of the semantic network also inherently serves as a *dependency graph* that shows which objects *directly* and *indirectly* affect others [56, 73]. For example, Figure 6.1 depicts the semantic network for *the dog is south of the tree and near the cat; the cat is right of the dog; the elk is facing away from the lake; the lake is midrange from the elk*.

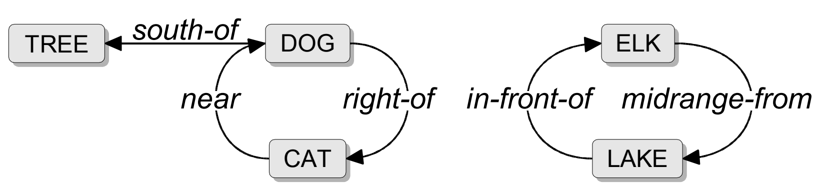


Figure 6.1: Dependency Graph

The TREE, DOG, CAT are interdependent in the following ways:

- The DOG is directly dependent on the TREE.
- The DOG is directly dependent on the CAT and vice versa.

- The CAT is indirectly dependent on the TREE by transitive closure.

Likewise, the ELK and CAMEL are interdependent in the following ways:

- The ELK is directly dependent on the LAKE and vice versa.

However, the TREE, DOG, CAT in the first subgraph are independent of the ELK and LAKE in the second subgraph, and vice versa. In fact, for any semantic network, all subgraphs are always independent of each other and therefore can be solved as independent subproblems through a *divide-and-conquer strategy* [101]. This process of fragmenting a semantic network into subgraphs, called *partitioning*, forms the high-level organization of the constraint satisfaction algorithm in this project [73]. It applies only to constraint satisfaction for the *interobject* constraints of position and orientation.

The foundation of this reasoning from relation arc to constraint rule to numerical position and orientation has already been addressed in Section 4.2 and Section 5.4. The implementation of it in the spatial reasoning engine is relatively simple and straightforward:

1. For an unsolved partition, satisfy its position and/or orientation constraints as the next two sections will discuss.

1a. If the solution to the current partition conflicts with the solution to any previous partition, repeat Step 1 for the current partition up to a predefined number of times.

1b. If a certain number of attempts does not generate a non-conflicting solution to the current partition, backtrack and re-solve a previous partition.

2. Repeat Step 1 until all partitions have been solved, or give up after a certain number of attempts.

Note the apparent contradiction between Step 1a and the assertion that partitions are independent: if it is true that all partitions are always independent, then how could there ever be a "conflict" between them? The answer lies in the definition of a conflict. It is true that the objects within a partition are independent of the objects outside that partition *in terms of interdependent constraints*. All such *static pre-constraints* are known when the dependency graph is built and remain constant, so conflicts can never occur at this stage of constraint satisfaction; i.e., partitions that conflict *before* a solution is generated for them are never built. However, it is possible—actually quite common—for partitions to conflict *after* a solution is generated for them. Most objects have rules that define their abstract spatial behavior; e.g., an ALLIGATOR derives from the abstract concepts LAND-THING and WATER-THING, which together indicate that it is equally at home in either environment. This definition serves as a *dynamic constraint* to verify that an object does not violate a customary interpretation. For example, if a particular solution to the first partition in Figure 6.1 placed the DOG in the LAKE, the interpretation would be considered odd because a DOG typically does not belong there. However, no *static constraint* between the two partitions asserts that such a solution is illegal. Only after the solution has been generated does the conflict arise, which forces the constraint satisfaction algorithm in Step 1b to reconsider the "independent" partitions from a global perspective.

All concepts define a general behavior toward the *noninterpenetration constraint rule* that implicitly applies to everything in the world. Every concept must declare through its (usually inherited) `is-container` property whether it allows any other object to occupy the same space that it does. This constraint prevents objects from inappropriately embedding in each other; e.g., the ELK is impaled on the TREE! Certain concepts like LAKE and CAGE actually require interpenetration for their interpretation, so they are defined as *containers* and are exempt from this post-constraint.[2]

[2]Noninterpenetration is the only post-constraint in this project.

6.1.1.2 Processing Dimension Constraints

Dimension constraints are the simplest to satisfy because they are *intraobject* constraints—fully self-contained rules that are not dependent on other objects. Every object must be assigned a value for each of its height, width, and depth dimensions in order for it to exist in three dimensions. Dimension constraints are also the first to be processed because the size of an object affects the size of its fields, which play a role in subsequently solving position and orientation constraints.

The foundation of this reasoning from attribute node to attribute interval to plausibility interval to numerical value has already been addressed in Section 5.4 and Section 5.7.1. The implementation of it in the spatial reasoning engine is relatively simple and straightforward:

1. For each object node in the semantic network, visit each of its attribute nodes.

1a. Using the attribute interval of the object, bind the attribute to the plausibility interval of the object.

1b. Generate the contextually appropriate random value for each dimension to which the attribute applies.

2. For any dimension that is not specified with an attribute node, use the nominal interpretation in the plausibility interval of the object to generate the contextually appropriate random value.

6.1.1.3 Processing Position and Orientation Constraints

Position and orientation constraint rules require that one object appropriately appear within a field of another. The aspects of the objects that can be manipulated to satisfy such a constraint are the position and/or orientation of either or both of them. For instance, *the dog is left of the cat* can be satisfied by moving the DOG and/or by moving and/or rotating the CAT! The result is a three-dimensional (x, y, z) coordinate in the world that indicates the static position of the objects and/or an azimuth (in degrees) from north where the object or objects face in a solution. Be forewarned: this process is complicated and/or confusing![3]

First, in reality, as Section 5.6.1.1 and Section 5.7.2 discussed, the reasoning in this project is considered *two-and-a-half dimensional* because the vertical component (y) is controlled by a contextual degree-of-freedom adjustment in the knowledge base and not by the spatial reasoning engine. It

[3]For the sake of clarity and to avoid near redundancy, read "position" throughout this discussion as "position and/or orientation." The same description basically applies to both.

was concluded that fields are better defined in this project as two-dimensional footprints than as three-dimensional regions because birds are basically the only vertically capable objects that are consistent with the zoo theme, and their behavior is relatively constant and thus uninteresting. Moreover, English has only two major spatial prepositions (with variations) for vertical specification that fit the scope of this project: *above* and *below* [57, 121]. In other words, the substantial cost (in both development and run time) of true three-dimensional reasoning did not justify the minor return. Other systems have made similar concessions [18, 119, 54, 56, 35]. In any case, the vastly simpler implementation with reduced vertical capabilities still satisfies the goals of this project. The primary limitation is the following: the vertical dimension is available in the *implicit* knowledge representation for reasoning but not in the *explicit* knowledge representation for describing a spatial scene. For example, this project can generate solutions and inferences that place a HIPPO beneath the surface of a LAKE, but a person cannot write a description that states *the hippo is under the lake*.

The mechanism for determining the position and/or orientation of one or both objects such that they satisfy a constraint uses field membership as

Section 5.7.2 discussed. For example, (6.1) paraphrases a logical form of *the dog is is left of the cat.*

$$\texttt{DOG left-of CAT} \Rightarrow \texttt{DOG.position} \in \texttt{CAT.field-left} \tag{6.1}$$

True to the declarative nature of this project, any number of coordinates and/or orientations for DOG and/or CAT can satisfy this constraint. In Figure 6.2, the three interpretations respectively illustrate how the position of the CAT (C) alone dictates the position of the DOG (D), how the orientation alone does so, and how both do so simultaneously.

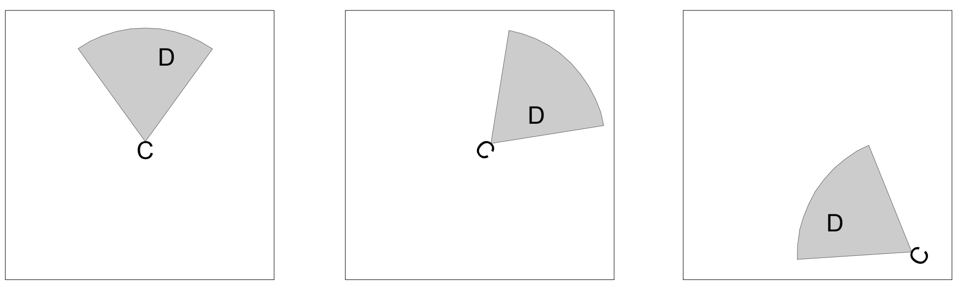


Figure 6.2: Interpretations for Position and Orientation

For two objects and one constraint, the task seems simple. However, the complexity increases quickly with more objects. For example, (6.2) paraphrases a logical form of *the dog is left of the cat, and the cat is in back of the tree.*

$$\begin{array}{l}\texttt{DOG left-of CAT} \wedge \texttt{CAT in-back-of TREE} \Rightarrow \\ \quad \texttt{DOG.position} \in \texttt{CAT.field-left} \wedge \\ \quad \texttt{CAT.position} \in \texttt{TREE.field-north}\end{array} \tag{6.2}$$

Although the underlying formalism of reasoning in this project is not intentionally[4] based on first-order logic, the two share semi-analogous proof mechanisms. The first mechanism, *forward chaining*, starts with a head object and individually solves constraints in the forward direction [5]; i.e., first DOG, then CAT, and finally TREE. In other words, first find a solution for DOG, then based on that solution, find a solution for CAT, then based on that solution, find a solution for TREE. The second mechanism, *backward chaining*, starts with a tail object and individually solves constraints in the backward direction [5]; i.e., first TREE, then CAT, then DOG. In other words, first find a solution for TREE, then based on that solution, find a solution for CAT, then based on that solution, find a solution for DOG. Consistent with the *fail-first principle*, the order of the nodes is from most constraining to least constraining [5, 73]. This heuristic functions as a proactive truth maintenance system to reduce the likelihood that later solutions will force the revision of earlier constraints [106].

In forward chaining, "based on" means to anchor the field of the current object at the position that was selected for it, then randomly[5] select a position within this field for the next object as Figure 6.3 shows. The possible positions

[4]All symbolic approaches to knowledge representation derive from first-order logic [130]. This project simply does not advertise or promote this foundation.

[5]More precisely, *randomly with respect to the topography* of the field that defines its probability distribution.

for the next object are limited to the gray area. In abstract terms, *If object A is here, then object B must appear somewhere within here.*

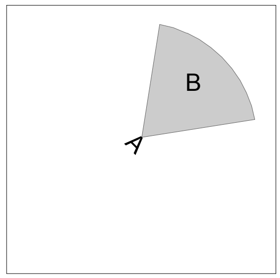


Figure 6.3: Forward Chaining

In backward chaining, "based on" means to position the field of the next object randomly such that it satisfies the position that was selected for the current object and use the anchor of this field as the position for the next object as Figure 6.4 shows. In this inverse constraint mode, the possible positions for the next object are limited to the hatched area. In abstract terms, *Where would object A (with unknown position) have to be in order for the known position of object B to be within the field of object A?*

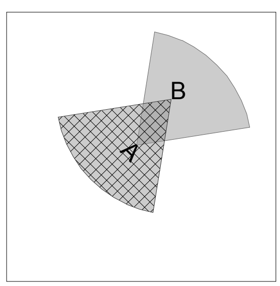


Figure 6.4: Backward Chaining

Almost any nontrivial description has several objects with several interdependencies that cannot be solved by either forward or backward chaining alone. For example, in *the dog is in front of the cat and the cat is in back of the dog*, the position of the DOG depends on the position of the CAT, which, in turn, depends on the position of the DOG! Such circular dependencies require simultaneous forward and backward chaining.

There is one aspect of solving orientation constraints that is mercifully simple in comparison with the others: absolute orientation refers to compass directions and therefore does not impinge as much on the states of other objects. Each object contains the set of compass fields in Figure 5.33 and internally implements a "probe" called a *pipper*[6] that extends out from the front of the object such that it always resides in one of these fields. The process of constraining the orientation of an object to a compass field requires only that the orientation be rotated randomly such that the pipper is in the appropriate field. Figure 6.5 shows two interpretations for *the dog is facing south*.

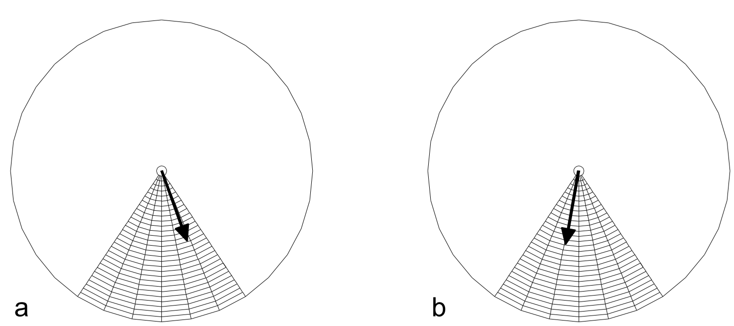


Figure 6.5: Pipper in SOUTH

[6]A military designation for the line-of-sight aimpoint on a targeting reticule.

The implementation of constraint satisfaction for position and orientation in this project is horrendously complex because it must take into account changes and revisions that propagate across parts of the current solution that are already committed; e.g., if *the tree is left of the dog*, too, then repositioning the DOG must reposition the TREE, as well as re-satisfied any dependencies that the TREE imposes on other objects, and so on. Thus, in the worst case, a single change may ripple throughout the entire solution and invalidate every object in every partition along the way! For this reason, constraint satisfaction by backtracking is not a fast approach for spatial reasoning over complex descriptions [73]. In fact, results in this project may take minutes to appear. Alternative search mechanisms like hill-climbing in [133] may improve this performance.

6.1.2 Synopsis of Constraint Satisfaction

Constraint satisfaction problems typically employ either backtracking or constraint propagation as their solution [73, 5]. This project uses a combination of both due to the way it decomposes the task from a high level to a middle level to a low level. At the high level, the problem is to reason over spatial constraints of position and orientation for the entire description. At the middle level, the subproblem is to reason over principally independent partitions. At the low level, the sub-subproblem is to reason over interdependent objects. Backtracking at the middle level attacks each partition in series; if the current partition cannot be satisfied, then the approach systematically backtracks to a previous partition, discards its solution, rebuilds it from scratch, and continues forward again. Constraint propagation at the low level attacks each object in series; if the current object pair cannot be satisfied, then it systematically backtracks to a previous object pair, adjusts some part of its solution, propagates any changes forward, and continues forward again from some intermediate point. Together, the processing of the middle and low levels contributes to the high-level solution. This combination of backtracking and constraint propagation coexists well because the middle level of partitions has little control over adjusting its parameters; whereas the low level of objects has significant control.

6.2 Inference Generation

The purpose of and representation for inference generation in spatial reasoning have been addressed from various perspectives already[7]; all that remains are the procedural details of its implementation. The reasoning that generates inferences in this project is *dynamic*: specific inferences are not known until a solution is generated. If this reasoning were *static*, the inferences

[7]See Section 5.5.2 and Section 5.8.2.2.

would be known in advance from commonsense expectation [56, 114]. Humans do both dynamic and static reasoning to generate inferences from spatial solutions [63].

6.2.1 Process Outline

The approach for generating inferences is far simpler than for generating solutions. It is treated as an "extreme" case of constraint propagation in which a nearly complete graph is built with relation arcs between every pair of objects if they satisfy any inference rules in context [15]. Every combination of two objects is considered, the set of contextually appropriate inference rules for each object is evaluated, and the satisfied ones generate attribute nodes and/or relation arcs that go directly back into the original semantic network. This process serves a second specialized purpose as well: to verify that a solution is consistent with the constraints that define it. This mechanism plays no role in the reasoning process, but it does in the simulation process that Chapter 7 will discuss.

The sample description in Figure 3.1, for example, corresponds to the reduced form in Figure 4.1, which directly translates to the semantic network in Figure 4.2. Figure 6.6 specifies this explicit knowledge more succinctly in terms of the attributes and relations that the description specifies. Before inference generation, only this information is known.

```
tree small
tree in-front-left-of giraffe
zeus in-front-of giraffe
zeus at-fringe-of tree
giraffe big
giraffe in-front-of tree
```

Figure 6.6: Explicit Knowledge

After inference generation, the additional knowledge in Figure 6.7 is known! All these relations are added back into the original semantic network, thereby augmenting the understanding of the description further [114]. Although the augmented structure is not used beyond this point in this project, it is available to follow-on work and other applications.

6.2.1.1 Processing Attribute Inferences

The only attribute inference rules in this project infer the compass direction that an object is facing. The appropriate inference in Table 5.12 is generated for an object simply by determining which one of its global fields in Figure 5.33 contains its pipper.

6.2.1.2 Processing Relation Inferences

Relation inference rules consider all combinations of pairs of objects with respect to their set of contextually appropriate inference rules. These inferences apply to dimensions, position, distance, and orientation.

```
tree southwest-of world-center
tree far-from world-center
tree local-in-front-of giraffe
tree local-in-front-left-of giraffe
tree global-in-back-of giraffe
tree global-directly-in-back-of giraffe
tree north-of giraffe
tree directly-north-of giraffe
tree outside giraffe
tree near giraffe
tree has-more-height giraffe
tree has-less-width giraffe
tree has-less-depth giraffe
tree local-in-front-of zeus
tree local-directly-in-front-of zeus
tree global-left-of zeus
tree west-of zeus
tree outside zeus
tree near zeus
tree has-more-height zeus
tree has-less-width zeus
tree has-less-depth zeus

giraffe south-of tree
giraffe directly-south-of tree
giraffe at-fringe-of tree
giraffe facing tree
giraffe has-more-width tree
giraffe has-more-depth tree
giraffe has-less-height tree
giraffe south-of world-center
giraffe far-from world-center
giraffe local-left-of zeus
giraffe local-in-front-left-of zeus
giraffe global-in-front-left-of zeus
giraffe southwest-of zeus
giraffe outside zeus
giraffe midrange-from zeus
giraffe facing zeus
giraffe directly-facing zeus
giraffe has-more-height zeus
giraffe has-more-width zeus
giraffe has-more-depth zeus

zeus east-of tree
zeus at-fringe-of tree
zeus facing tree
zeus directly-facing tree
zeus has-more-width tree
zeus has-more-depth tree
zeus has-less-height tree
zeus south-of world-center
zeus far-from world-center
zeus local-in-front-of giraffe
zeus local-directly-in-front-of giraffe
zeus global-in-back-right-of giraffe
zeus northeast-of giraffe
zeus outside giraffe
zeus near giraffe
zeus has-less-height giraffe
zeus has-less-width giraffe
zeus has-less-depth giraffe

world-center global-in-back-of giraffe
world-center north-of giraffe
world-center at-fringe-of giraffe
world-center global-in-back-of zeus
world-center north-of zeus
world-center at-fringe-of zeus
```

Figure 6.7: Inferred Knowledge

6.2.1.2.1 Dimension Relation Inferences

Dimension inference rules determine how two objects are related in their three dimensions of height, width, and depth. As the result of the comparison for each must satisfy the trichotomy property of *less than*, *equal to*[8], or *greater than*, every object combination always produces one dimension inference for each row in Table 5.13 [21].[9]

6.2.1.2.2 Position and Distance Relation Inferences

Position and distance inference rules respectively infer where two objects are positioned with respect to each other and how far apart they are. In both cases, the appropriate inferences in Table 5.14, Table 5.15, and/or Table 5.16 are generated for each object simply by determining which one of its fields in Figure 5.32, Figure 5.33, and/or Figure 5.34 contains the other object.

6.2.1.2.3 Orientation Relation Inferences

Orientation inference rules infer where two objects are facing with respect to each other. The appropriate inference from Table 5.17 is generated for each

---

[8] Equality uses a $\pm 5\%$ proximity threshold to avoid the direct comparison of real numbers internally.

[9] Except in combination with `WORLD-CENTER`, which is not considered a comparable object. To avoid a nonsensical comparison, the inference rules for dimension relations have a conditional dependency that determines applicability based on the `supports-dimension-comparison` property of both objects.

object simply by determining whether its `front` or `back` fields in Figure 5.32 contains the other object.

6.2.2 Scene Recognition for Description Generation

A surprising discovery emerged during the write-up of this project: it inherently supports limited scene *recognition*, from which it can generate coarse natural language descriptions! This process is the converse:

a. For any scene that conforms to the specifications in Chapter 3, the dimensions, positions, and orientations of its objects map directly to a semantic network. Without relations, all nodes are independent.

b. The inference component of the spatial reasoning engine considers the entries in all combinations with each other with respect to their definitions in the knowledge base.

c. The resulting inferences form a rudimentary English description.

This capability is serendipitous and has received little attention beyond basic verification of its existence. Cursory inspection, however, suggests that such inference generation does not take full advantage of the knowledge base to resolve contextual issues in recognition because this project considers only generation. Future work will investigate this area.

7 SIMULATION AND ANALYSIS

The spatial reasoning engine generates numerical results for the dimensions, positions, and orientations of all objects in a scene. These values derive from constraints, which are a declarative representation that specifies valid numerical ranges of solutions as well as probability distributions for preferred solutions. This mechanism produces results that address the issues of vagueness and uncertainty in spatial descriptions. The randomized or probabilistic nature of this process means that the interpretation is different every time the project runs for the same description. In other words, it produces a valid but different interpretation each time from an infinite set of valid interpretations [47, 23, 63].

Although *any* valid interpretation of a description officially satisfies the goals of this project, a single interpretation does not depict a true measure of its behavior or performance. For example, consider an analogous example for the game darts. The declarative form of a “valid” throw is that the dart strikes the board (or close to it). The stochastic or uncertain nature of the action dictates that the dart does not always strike exactly where the player wants it to go. Thus, for each valid throw, the dart strikes the board but not in exactly the same place. If only one dart is thrown—analogous to generating only one interpretation—then these results are ambiguously representative of the following performances:

- A good player may have thrown well.
- A good player may have thrown poorly.
- A poor player may have thrown well.
- A poor player may have thrown poorly.

In other words, a single, valid throw demonstrates little about true performance overall—the player may have had a lucky throw, an unlucky throw, or a typical throw, so the conclusion that he or she is respectively a good player, bad player, or mediocre player has little merit. Thus, although the results can be measured, confidence in the conclusion cannot.

After 50 throws, however, a more representative picture emerges because the contribution of true performance dominates that of pure chance. More throws thus strengthen confidence in the conclusion. They also serve to identify patterns in the behavior of the processes that contribute to performance [114]. In numerical analyses, *accuracy* and *precision* are a standard measure of performance [89]. Accuracy refers to how close the actual results are to the expected results; whereas precision refers to how close the actual results are to each other. The former is a metric of correctness and the latter of consistency.

Figure 7.1 illustrates the four combinations of high and low accuracy and precision with respect to 50 darts (using a non-standard dartboard for clarity).

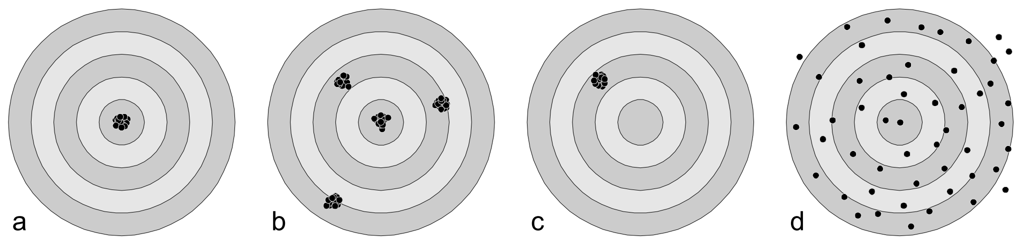


Figure 7.1: Accuracy and Precision

In Figure 7.1a, one can conclude with high confidence that the player is good at darts. In Figure 7.1b, the player is inconsistently good and needs to stabilize performance. In Figure 7.1c, the player is consistently poor and needs to improve performance. And in Figure 7.1d, the player is inconsistently poor and perhaps needs to figure out what the goal of the game is!

7.1 Monte Carlo Simulation

This process of evaluating performance at darts, which is analogous to spatial reasoning in this project as well, forms a *Monte Carlo simulation* [89, 45]. It consists of a framework for collecting the results of multiple runs and feeding them into data-reduction and analysis components to draw conclusions from the aggregate results. This methodology is frequently applied to stochastic processes for independent events in many disciplines, but seldom in artificial intelligence for knowledge representation and reasoning [42]. Nevertheless, in this project, it provides a powerful, analytical mechanism to investigate not only the performance of this approach but also the underlying cognitive and linguistic foundations of spatial representation and reasoning [118, 127, 35].

At this point in the chapter, a disclaimer is appropriate: simulation and analysis are not part of the proposed work for this project. They emerged during its development as a natural extension and a stepping stone toward future work. As such, this chapter is intended only to present a reasonable discussion on the mechanics of the simulation and analysis and the issues they likely addresses. It is an introduction to the potential only and therefore lacks strong justification of its details and conclusions.

7.2 Data Reduction

The spatial reasoning engine can be run multiple times on any description to produce a *set* of solutions. The same objects appear in each solution, but they differ between solutions in their dimensions, positions, and/or orientations. *Data reduction* is the process of culling and organizing these objects in a useful way for analysis. This project does so through *distillation*, which clusters the solutions into one or more relatively disjoint subsets by similarity [106]. Any combination of similarity tests for dimensions, position, and/or orientation can be configured for distillation and analysis. The similarity threshold for each is measured in terms of a standard deviation from the mean interpretation between all objects in all solutions. This relative metric for similarity is inherently based on the nature of the solution as opposed to an absolute metric with a "magic number" that would arbitrarily specify some rigid threshold like 5.0 meters [56].

7.2.1 Dimensional Similarity

Similarity in dimensions is generally the least interesting metric of the three due to circularity in their behavior: the plausibility intervals that constrain the values of dimensions use a defined probability distribution that, unsurprisingly, the objects exhibit in every solution! Nevertheless, dimensions are available for isolating different interpretations of size; e.g., to investigate issues of vagueness. For example, Figure 7.2 depicts 9 GIRAFFEs of average (or default) height to illustrate their comparative dimensions. The height component of the plausibility interval of the GIRAFFE (Section 5.5.1.2.1.2 and Section 5.7.1.1.1) specifies that its mean height lies at 4.70 meters.[1] The bandpass filter on the interval (Figure 5.12) is relatively liberal at ± 1.18 meters, which allows considerable variation in nominal height. Consequently, for a small sigma, three disjoint interpretations of these solutions are easily possible: one for the cluster of GIRAFFEs around the mean, and one each for the outlying clusters.

[1]For simplicity, the width and depth components are ignored here.

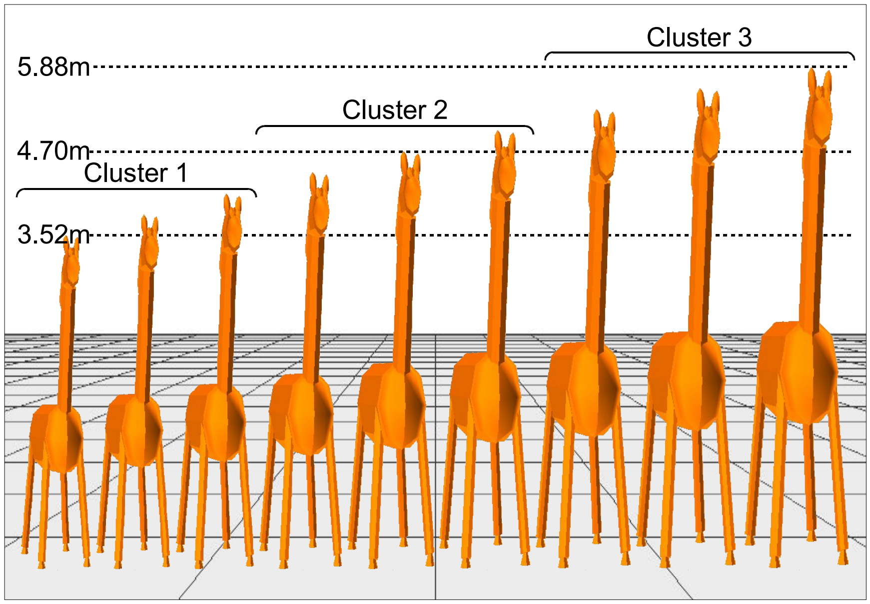


Figure 7.2: Dominant Dimensional Interpretations

7.2.2 Positional Similarity

Similarity in position uses the coordinates of objects to cluster by proximity [93]. For example, in *the dog is to the side of the cat*, the overwhelming uncertainty revolves around whether the DOG (D) is to the *left* or *right* of the CAT (C),[2] either of which is equally valid. As Figure 7.3 illustrates, an arbitrary seven to the left form one cluster, and another three form the other. Notice that although all the DOGs are roughly the same distance from the CAT and thus have similar sigmas, they are not all the same relative distance from *each other*. This secondary contribution to the measure of similarity triggers the creation of separate clusters.

[2]Assume the position and orientation of the cat are fixed.

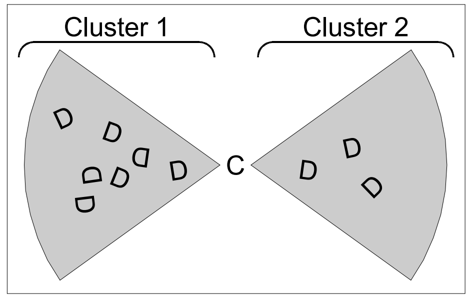


Figure 7.3: Dominant Positional Interpretations

7.2.3 Orientational Similarity

Similarity in orientation uses the azimuth of objects to cluster by angular proximity [93]. For example, in *the dog is facing the cat*, the uncertainty revolves around the acceptable deviation between the front of the DOG and a direct line from its center to the CAT. As Figure 7.4 shows, different sigmas might place an arbitrary seven CATs into one, two, or three clusters.

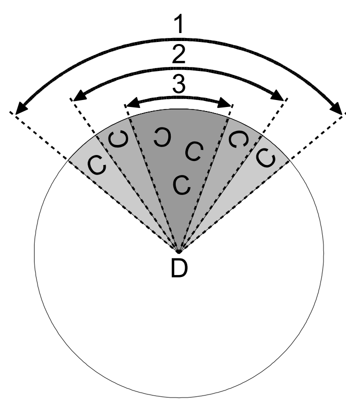


Figure 7.4: Dominant Orientational Interpretations

7.3 Analysis

Distillation supports a test-and-evaluation framework for this project that can be used both to improve performance and to test hypotheses [45]. The former aspect is more developed than the latter because it contributes to the goals. Hypothetical and theoretical analyses, on the other hand, better contribute to future work. This Monte Carlo framework implements the essential analytical functionality with certain limitations [89]:

- It configures the initial conditions of an experiment in terms of control and test variables. However, the input parser does not currently support absolute descriptions for precise manipulation; e.g., *the dog named Control is at position 10x10 and facing 270 degrees*. Such fine tuning is supported internally in the source code, however.
- It supports consistency and reproducibility of results through a configurable randomization seed.
- It oversees an arbitrary number of independent simulations and collects their results. However, memory considerations currently limit the maximum number of runs to roughly 25 depending on the complexity of the description. The graphical rendering engine imposes this limitation, not the simulation framework itself.
- It supports analytical methods for drawing conclusions from the results. However, the actual analyses require external statistical processing.

7.3.1 Analysis of Knowledge Base Configuration

The knowledge base provides extremely detailed control over the interpretation of its contents. Thus, it is not difficult to define or refine particular interpretations. Rather, it is difficult to determine which of the many parameters to address. If the performance of a simulation is not consistent with real-world interpretations, then one or more of them needs adjustment. Analysis of the knowledge base configuration provides a mechanism to tweak definitions in a systematic manner. For a constant description, one set of solutions serves as a baseline, and subsequent sets serve as excursions from the baseline with one perturbed experimental parameter. In this way, it should be possible to isolate and rank the contributing factors to any observed behavior. In fact, this mechanism played a considerable role in refining the results of this project.

7.3.2 Analysis of Interpretation Commonality

Due to the stochastic nature of this simulation framework, multiple runs over the same description often produce significant variation in the results. Clusters demonstrate commonality among different solutions and facilitate "averaging" over a cluster to produce a single interpretation that represents it [45]. For all objects in a cluster, their dimensions, position, and/or orientation contribute to a single, new object in a new cluster, which then takes the place of the entire existing cluster as Figure 7.5 exemplifies [22]. This process summarizes interpretations. It also provides an analytical method to investigate how much freedom is acceptable in such interpretations before their average no longer represents the commonalities of the whole cluster. For example, a relatively large sigma may be acceptable for position but not for orientation, from which one might conclude that variation over position is less important than variation over orientation.

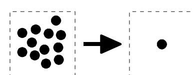


Figure 7.5: Averaged Cluster

In general, determining the validity of interpretations or judging their consistency with the real world is subjective [111, 35, 50]. Clustered averaging, however, takes advantage of the objective power of inferences [10]. Section 6.2 discussed how inferences—new attributes and relations—are generated from a solution. The important characteristic of inferences for analysis is that they also redundantly generate *existing attributes and relations* from a solution as a by-product. Normally these inferences are discarded because they contribute no new information to the semantic network of a description; e.g., *the dog is in front of the cat* produces the tautological inference DOG IN-FRONT-OF CAT, which, of course, is already in the semantic network because it is exactly what was specified. With an average solution, however, such inferences play a critical role in verification. For example, 25 runs would produce 25 valid solutions with DOG IN-FRONT-OF CAT. If the average interpretation of these solutions also produces the inference DOG IN-FRONT-OF CAT, then the sigma that specified the freedom for averaging is acceptable. On the other hand, if the average interpretation *fails* to produce this inference, then the sigma is unacceptable because the actual results are not consistent with the expected results. This test-and-evaluation framework supports automated, iterative refinement of the sigmas by driving a set of Monte Carlo simulations with different sigmas to find the boundary between acceptable and unacceptable. Such results are beyond the scope of this project, however.

7.3.3 Analysis of Spatial Discovery

The observation that interpretations exhibit certain inherent commonalities can contribute to more than just the process of averaging clusters for summarization. It can also potentially detect so-called *emergent properties* of unspecified or unknown commonalities [106, 87, 64]. Basically, if a set of objects with predefined behaviors is allowed to run on its own, over time, do any unexpected or previously unknown patterns emerge [114]? It appears possible to use this test-and-evaluation framework to identify such interrelations between objects [35, 43]. For example, if an arbitrary description explicitly interrelates three objects A, B, and C over 1,000 runs, do any of the following unspecified interrelations consistently emerge?

1. Does A NEAR B always imply B NEAR A?
2. Does A NORTH-OF B and B NORTH-OF C always imply A NORTH-OF C?
3. Does A FACING-NORTH and B FACING-SOUTH always imply B FACING A?

In fact, none[3] of these claims is true, but proving[4] so is non-trivial. This test-and-evaluation framework allows solutions to be compared against formal mathematical relations like symmetry, reflexivity, transitivity, asymmetry, and so on to discover constant patterns [35, 56, 111]. Such investigation is beyond the scope of this project, however, the focus of which is to develop the necessary framework, but not actually to utilize it or report its results.

[3]In commonsense terms, (2) is indeed true, but it is possible to manipulate the fields such that the relation in the consequent is NORTHEAST-OF or NORTHWEST-OF, which does not entail NORTH-OF in this reasoning formalism.

[4]Or at least convincingly demonstrating. Linguistic issues cannot be "proved" in the formal sense due to the inconsistency of language [80].

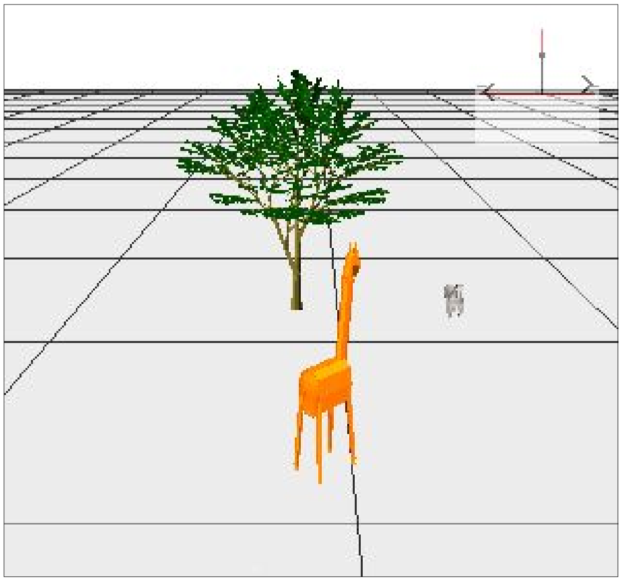
8 GRAPHICAL RENDERING

The internal representations of the semantic network, knowledge base, spatial reasoning engine, and simulation framework have a human-readable text form that was very helpful during the development of this project to inspect and debug their contents [56]. This unified formalism could also serve as the final output from this project because it contains all the results of interest. Obviously, however, text output of an arcane internal representation lacks the intuitive appeal of a visual presentation for a spatial description.

The graphical rendering engine is a powerful visualization tool that translates these arcane structures into a corresponding three-dimensional virtual world. It serves simultaneously as both a display tool to view the results and an evaluation tool to inspect their underlying representations [56]. The graphics are a minor component in this project, however, because they play no role in the issues of representing and reasoning over knowledge of spatial interactions. Consequently, they are relatively primitive by modern standards. This decision is easily justified by the lack of precise detail in descriptions anyway as there is no basis for rendering ornate graphics from generic statements like *there is a big dog*.[1] For example, Figure 8.1 graphically renders the description in Figure 3.1.

---

[1]A major component of the WordsEye system actually does try to fill in unspecified objects that plausibly appear in an environment [116]; e.g., a kitchen should have counters, cabinets, a sink, etc.



The scene contains a tree, a zebra named Zeus, and a giraffe.

Zeus is in front of the giraffe.
Zeus is at the fringe of the tree.
The giraffe is in front of the tree.
The tree is in front and left of the giraffe.

The tree is small.
The giraffe is big.

Figure 8.1: Sample Graphical Rendering

8.1 Architecture

The architecture of the graphical rendering engine maps directly onto the framework of the Monte Carlo simulation that Chapter 7 discussed. The *playground–sandbox–toy* model[2] in Figure 8.2 corresponds to a set of independent solutions that are generated from a single description, for which each solution contains copies of same objects but with different values.

---

[2]The basis of this naming scheme is to reduce confusion within the components of this project. The *simulation–solution–object* model applies only to the reasoning components; whereas the *playground–sandbox–toy* model applies only to the rendering components. Both play confusingly similar roles but are completely decoupled internally. Independent naming decouples them externally as well.

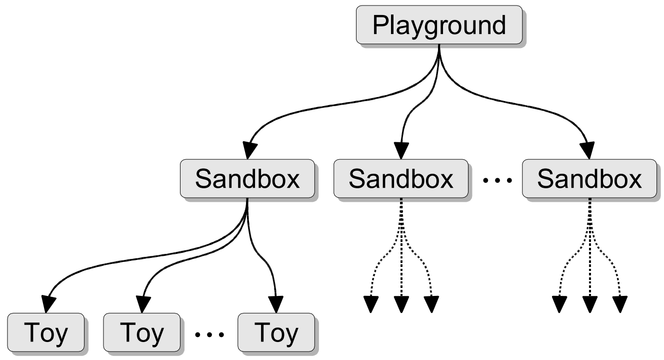


Figure 8.2: Architecture

8.1.1 Playground

The playground is a container for all the independent solutions in the sandboxes. It provides no functionality beyond the initial placement of the sandbox windows. As the graphical output in this project receives relatively little emphasis, there is only this rudimentary graphical user interface to manipulate the system. It is for output only; the input of a description and the configuration of the simulation and analysis are through a text-file interface known as a *vignette*. Appendix C provides a complete example.

8.1.2 Sandbox

A sandbox is a self-contained virtual world for one interpretation of a description. As the entire "world" encompasses just a small zoo in this project, the flat-earth representation in Figure 8.3 is appropriate [35]. WordsEye [117] uses a similar platform that floats in space.[3] A sandbox is a true three-dimensional representation of the world that overlays its objects and the viewer onto a 100-meter square platform with 10-meter grid resolution. The viewer[4] appears initially south of the world facing north, which is the default deictic and extrinsic frame of reference. He or she possesses six degrees of freedom for ghost-like mobility to fly around and through objects. The optional head-up display (inset) reflects the current position and attitude.

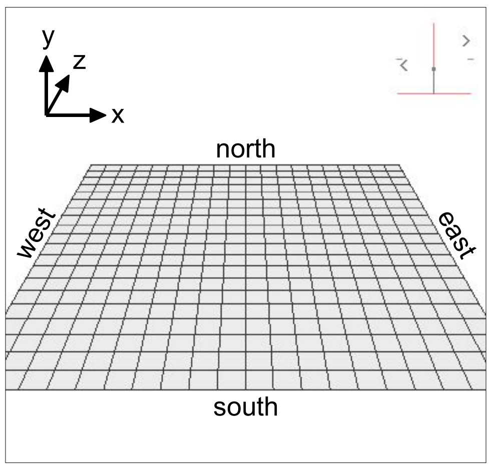


Figure 8.3: Sandbox

Each raw or clustered interpretation from a Monte Carlo simulation appears in its own independent sandbox in the playground. The viewer can fly through

---

[3]The platform plays no role in the reasoning here because it is not a true object; it is merely a graphical artifact for reference and scale. As such, objects may occasionally appear beyond it in space, especially when the AT-FRINGE-OF distance relation is used.

[4]The viewer is never shown because he or she is assumed to be see the world through his or her own eyes; i.e., from in front of the computer monitor.

each independently to view the different interpretations from different perspectives. To facilitate comparing and contrasting them, sandboxes also support synchronization, in which all sandboxes configure the viewer based on the active sandbox.

8.1.3 Toy

A toy is the manifestation of an object in an interpretation. Each graphically reflects the dimensions, position, and orientation that the spatial reasoning engine calculated for it. These values map to six degrees of freedom (i.e., x, y, z, pitch, roll, and yaw), but the pitch and roll components play no role because they are not considered in the reasoning process. For evaluation purposes, each toy can optionally render a *metaoverlay* of the fields that play a role in its interpretation to illustrate the interaction between itself and other toys. Figure 8.4a shows only field geometry, which indicates all the possible interpretations of position and/or orientation for the constraint that generated this toy. Figure 8.4b overlays the field topography to show the probability distribution over the possible interpretations. Finally, each toy can interactively display its internal details, which provide insight into the underlying representations that contributed to it from the first stage of processing to this final stage.

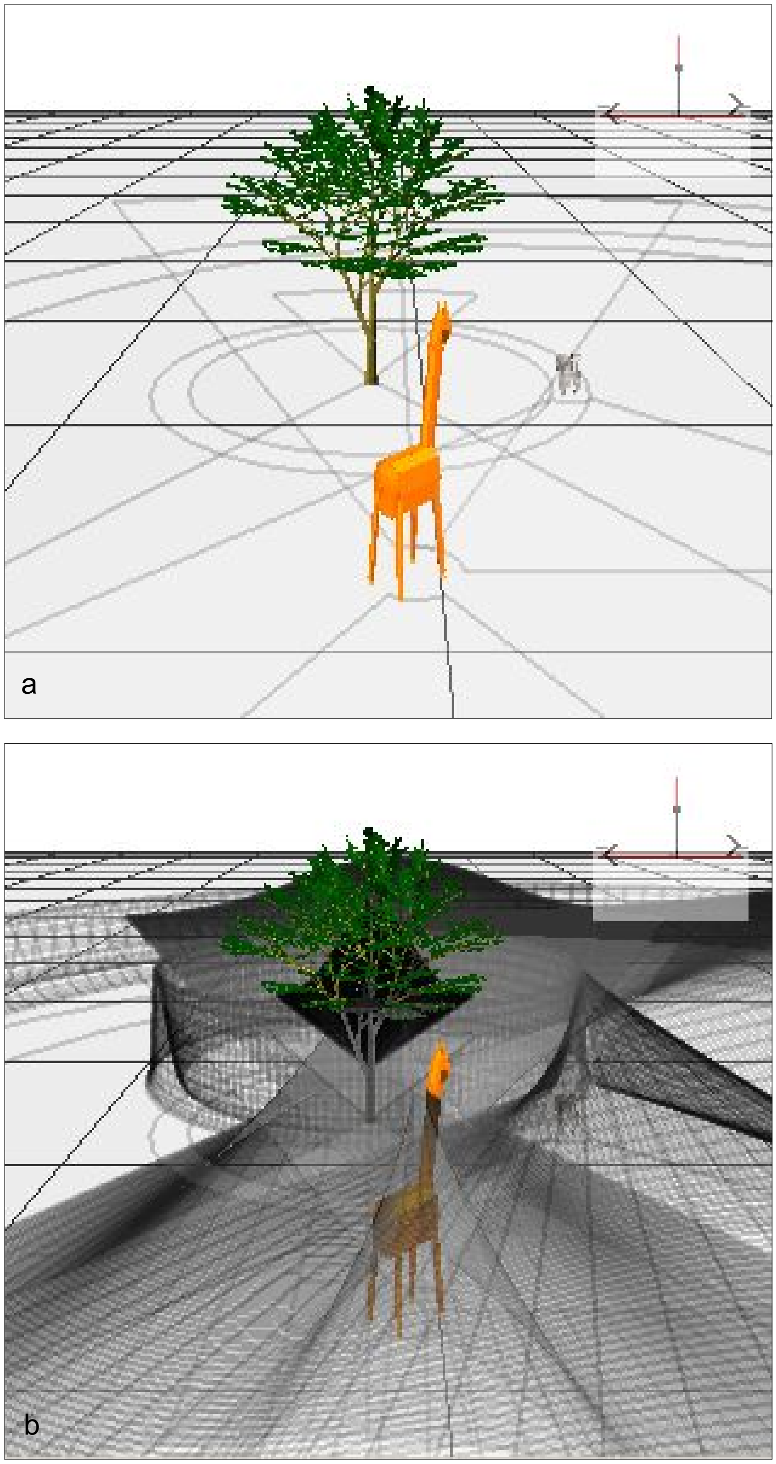


Figure 8.4: Metaoverlays for Geometry and Topography

8.2 Implementation Issues

The implementation of the graphical rendering engine is unnecessarily detailed and out of scope for this discussion. However, one related issue is appropriate because it plays a role in aspects of the knowledge representation. The knowledge base and spatial reasoning engine are completely decoupled from the graphics in all respects except one: each derived concept in the knowledge base must indicate the name of the model that represents it graphically. One could consider this entry as knowledge of how something is supposed to look.

8.2.1 Model Mapping

The low-level details of the graphics are maintained separately in a table that maps the model name to the file containing the polygon definitions of the corresponding three-dimensional model. This project supports only models that are defined in the popular 3D Studio MAX and Wavefront formats. Popularity is an important consideration because almost all the models were freely acquired from the World Wide Web. Specialized and obscure models like a corral, lake, non-coiled snake, and (inexplicably) a giraffe had to be created using a basic shareware editor. The table not only maps a model name to a model file, but it also configures how the model appears in a particular mapping. As there are

108 derived concepts in the knowledge base and only 69 unique models in the project library, obviously some concepts must share the same model as Figure 8.5 shows.

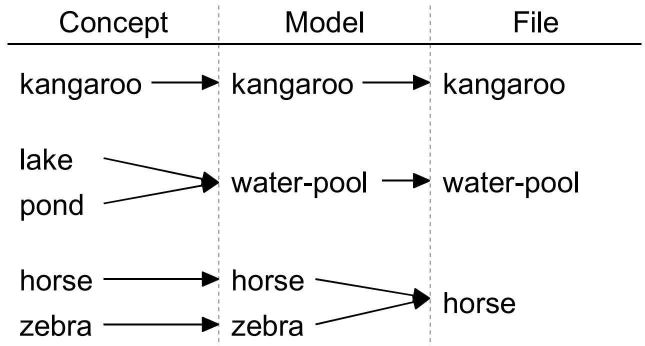


Figure 8.5: Concept Mapping

This property is exploited to reuse the same models with slight configuration changes, normally the color. This implementation eliminates the need to find specific models for visually similar concepts whose real-world differences in appearance are irrelevant for this project. For example, a LAKE and a POND differ only in their dimensions, so they share the same model and file. Similarly, the HORSE and the ZEBRA are share the same file (which explains why the ZEBRA in Figure 8.4 has no stripes), but they require different color configuration.

8.2.2 Model Normalizing

The motivation behind the mapping table extends beyond the convenience of model reusability. It also serves the essential role of normalizing the models to a single, standardized internal form. The models in this project were acquired from many sources and therefore have little or no consistency in their scale, origin, orientation, and so on. To ensure that they render according to the specifications of the spatial reasoning engine, each model was manually normalized so that it stands with its base on the ground, has its origin in the center, faces forward, and fits tightly within a one-meter cube. Normalization for most models is straightforward, but certain irregular shapes require a judgment call. For example, trees are normalized such that the trunk fits within the cube regardless of the width of the canopy. This determination is consistent with the real-world description of trees, which tend to refer to width in terms of the trunk. Visual inconsistencies, however, are still common due to limitations in the graphics. For example, a big tree should have more leaves than a small tree. Unfortunately, the number of leaves is fixed by the model, so a big tree is instead rendered with bigger leaves!

9 RESULTS AND DISCUSSION

The practical, application-oriented goal of this project is to translate text descriptions of spatial scenes to corresponding graphical renderings. The theoretical, research goal is to investigate the underlying processes and develop approaches to resolving major issues. The degree of success of the latter goal is reflected in the former, which this chapter addresses.

The results in this project are presented in the form of *vignettes*—simple, descriptive sketches that showcase its features. The determination of whether a solution to a vignette is consistent with a real-world interpretation is subjective [56]. A stronger evaluation of the results would require psychometric studies and statistical analyses over a reasonably large sample of determinations by different human subjects [56, 50, 35, 15, 86, 116]. This approach was unanimously rejected by all parties at the proposal stage due to the unnecessary, additional complexity it would bring to the project. However, it is possible to establish a reasonable measure of evidence to estimate informal confidence in the results [45, 101]. As underspecification, uncertainty, and vagueness in descriptions contribute in general to an infinite number of valid solutions to a single description, obviously it is not possible to define unambiguously the subjective form of a "correct" solution [118, 10, 47]. However, a more relaxed metric has potential: if a solution is not arguably *incorrect*, then it is assumed to be correct. The human spatial mind is very discerning and can readily detect most inconsistencies [56]. Furthermore, as Section 5.5.2 discussed on inference generation, this project inherently supports the objective verification that its results do not violate their defining constraints. Together, these subjective and objective metrics form a reasonable basis for presenting the samples of positive and negative results in the following two sections, respectively.

9.1 Positive Results

The issues in this project have been throughly addressed from various perspectives. Each has been intricately decomposed into sub-issues with individual solutions that can recombine in countless ways. The great number and large, graphical nature of these solutions preclude anything resembling an exhaustive treatment of their results. In light of this inherent limitation, this section presents selected results that showcase interesting features and behaviors that are representative of the success of the project as a whole.

The distance relations `INSIDE` and `OUTSIDE`, unlike `ADJACENT-TO`, `NEAR`, `MIDRANGE-FROM`, `FAR-FROM`, and `AT-FRINGE-OF`, have a clearly defined interface between themselves, which corresponds to the bounding cylinder of their object (see Section 5.7.2.1.2). Thus, as Figure 9.1 illustrates for *the horse is inside* *the corral and the zebra is outside the corral*, ring A serves as the interface. The EXTERIOR field (Figure 5.34) that the relation OUTSIDE uses extends only to ring B, and not to the horizon as might be expected, however. This behavior is consistent with real-world spatial reasoning because the focus of these relations is on the interface, which imposes preference on its vicinity [103, 94, 56].

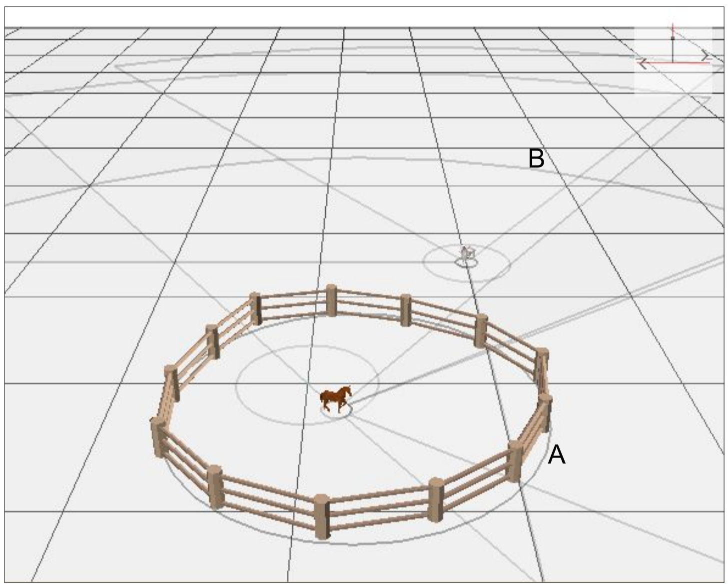


Figure 9.1: Relations INSIDE and OUTSIDE for CORRAL

A LAKE also defines an interface, but a vertical one; i.e., between positions on and below its surface. Figure 9.2 illustrates this behavior for (a) *the raft is in the lake* and (b) *the hippo is in the lake*. For (c) *the hippo is in the raft and the raft is in the lake*, both the HIPPO and the RAFT belong on its surface, which is also the interpretation that applies if they are on land.

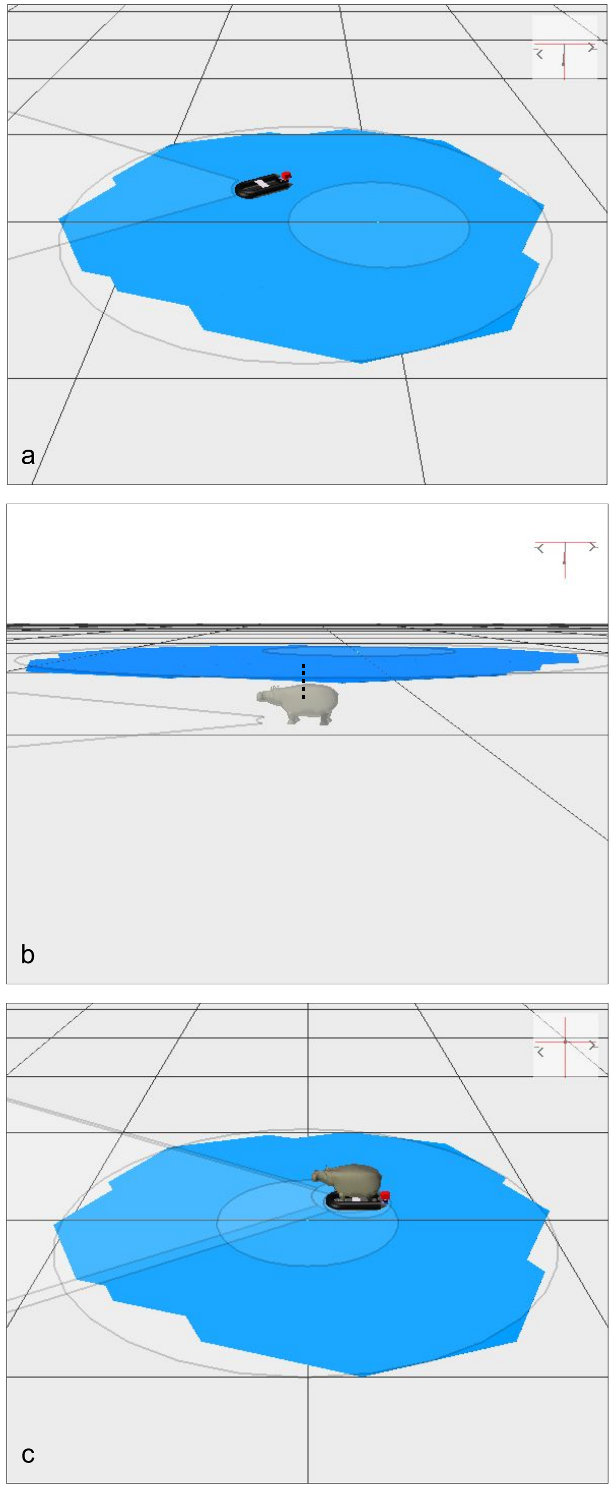


Figure 9.2: Relation IN for RAFT, HIPPO, and LAKE

Concept matching in Section 5.6.1.1 discussed the spatial interaction and pragmatic interpretation for *the golden eagle is in the pine tree*, which Figure 9.3 illustrates. This example also serves as a negative result, which the next section will discuss.

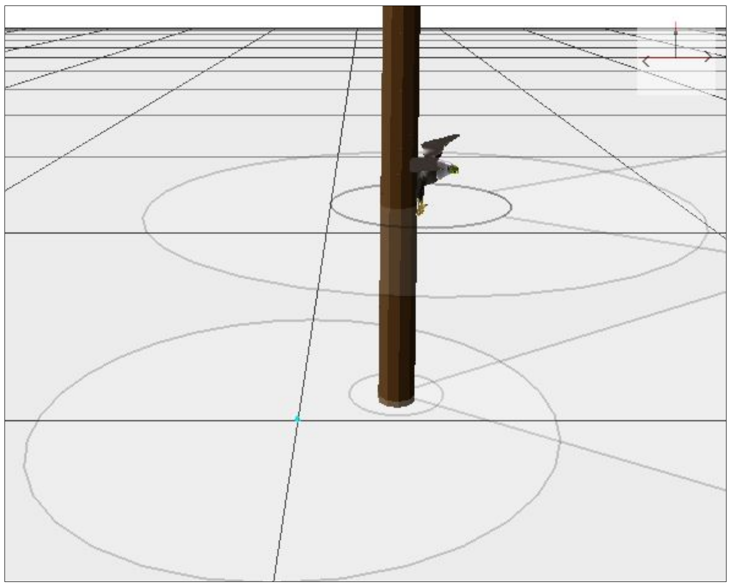


Figure 9.3: Relation IN for GOLDEN-EAGLE and PINE-TREE

Concept matching facilitates contextually specialized modifications to almost any part of a concept definition. Usually it applies to relations, but dimensions occasionally benefit as well. For example, the appropriate, general dimensions of a CAGE depend on the object it encloses. Thus, any concept that derives from the abstract concepts LARGE-ANIMAL or SMALL-ANIMAL deserves a large or small CAGE, respectively. Figure 9.4 illustrates this behavior for (a) *the monkey is in* *the cage* and (b) *the gorilla is in the cage*.[1] This example also serves as a negative result, which the next section will discuss.

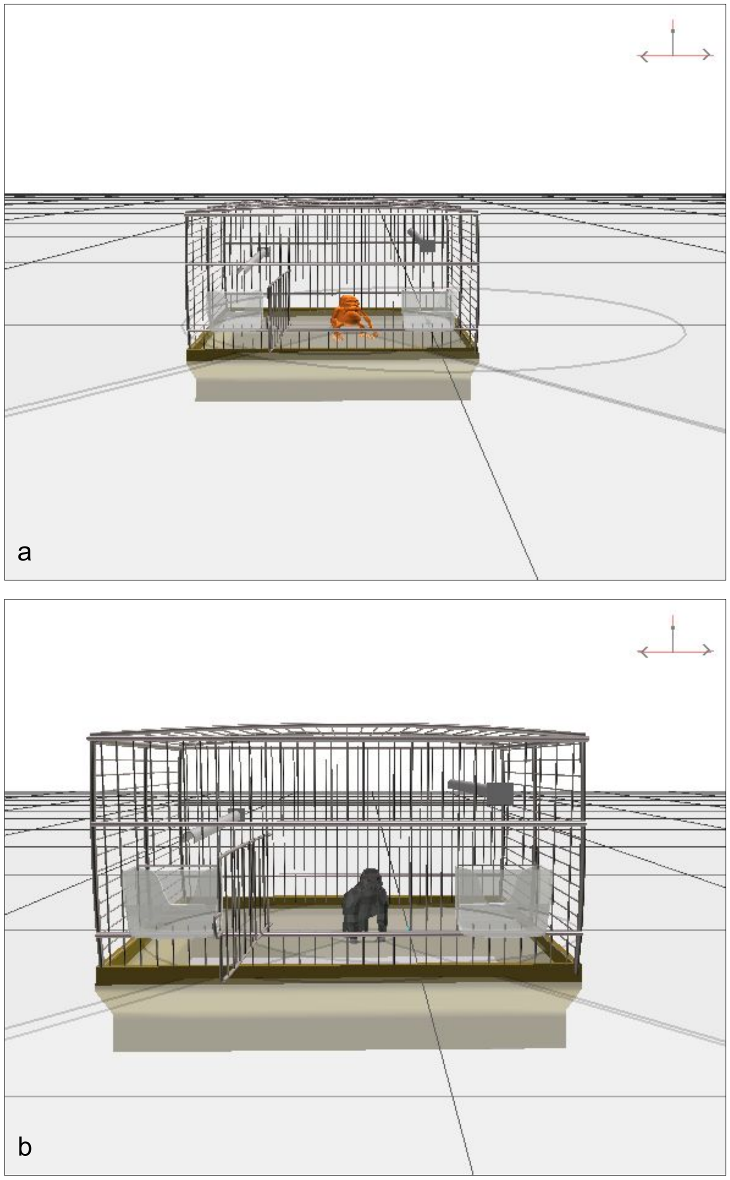


Figure 9.4: Concept CAGE for SMALL-ANIMAL and LARGE-ANIMAL

[1]The cage is actually an oversize birdcage! Section 8.2.1 discussed various ways to map models to extend their usefulness.

As Section 5.5.1.2 discussed, dimensions are contextually dependent on objects even if they do not interact with other objects. Figure 9.5 comparatively illustrates this behavior for *Roger* (R) *and Peter* (P) *are giraffes; Peter is short and Roger is tall. Son* (S) *and Kerry* (K) *are anacondas; Son is short and Kerry is long.*

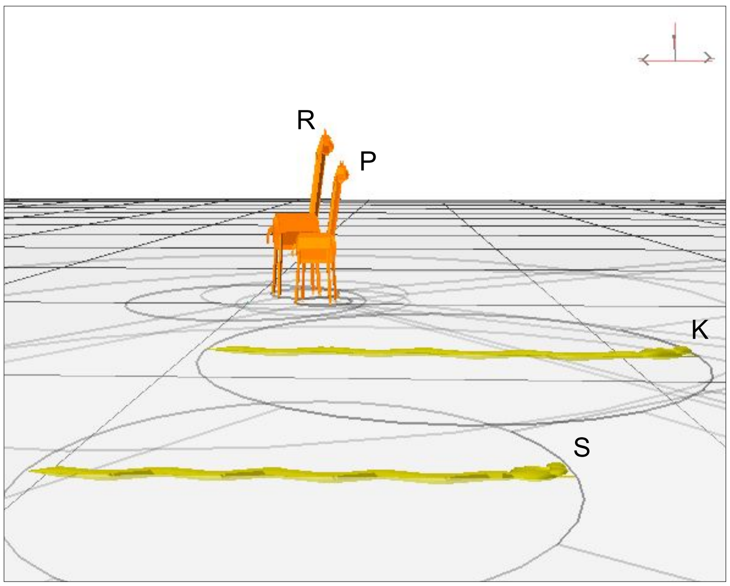


Figure 9.5: Dimensions SHORT, TALL, and LONG for GIRAFFE and ANACONDA

Section 5.6.2.1 discussed how the IN-FRONT-OF relation uses the `has-canonical-front` property of the second object to determine the frame of reference. Figure 9.6 illustrates this difference for (a) *the tree is in front of the dog* and (b) *the dog is in front of the tree*. Note that (a) also results from *the dog is facing the tree*.

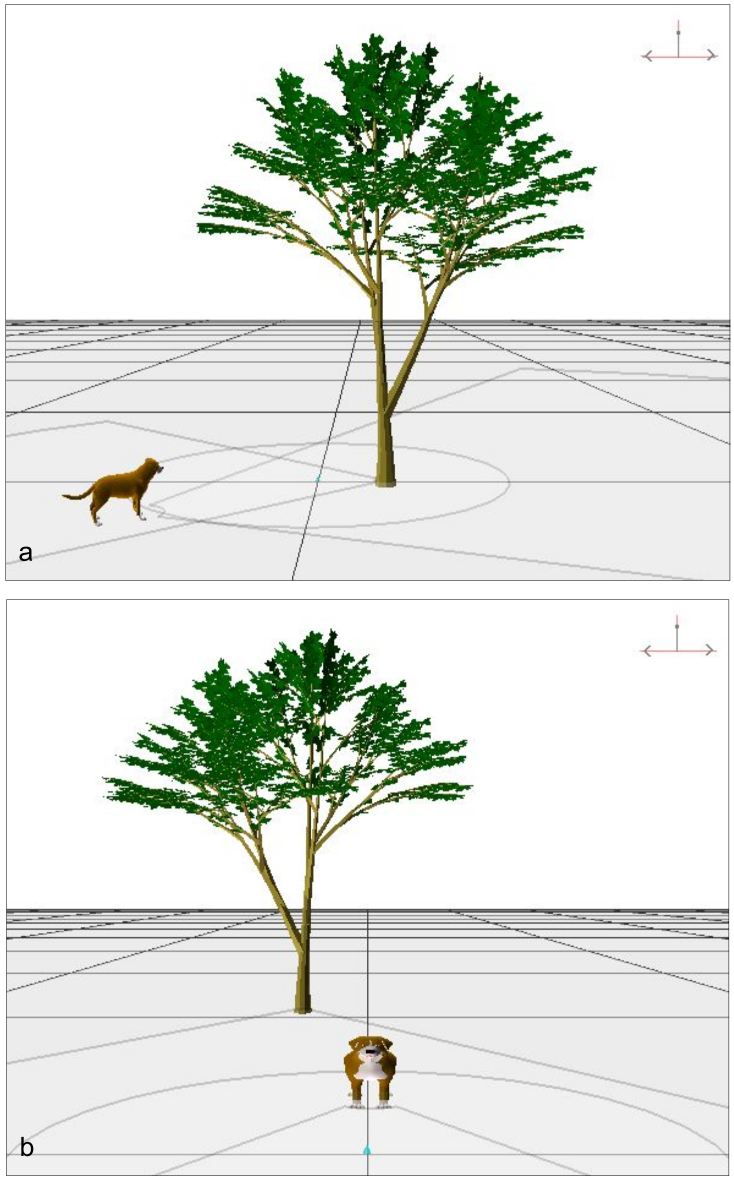


Figure 9.6: Relation IN-FRONT-OF for TREE and DOG

The relation TO-SIDE-OF is the most ambiguous in this project because it is the union of the LEFT-OF and RIGHT-OF relations for intrinsic frame of reference or of WEST-OF and EAST-OF for deictic (see Section 5.7.2.3.2 and

Section 2.3). Figure 9.7 illustrates one interpretation for *the dog is to the side of the gorilla*.

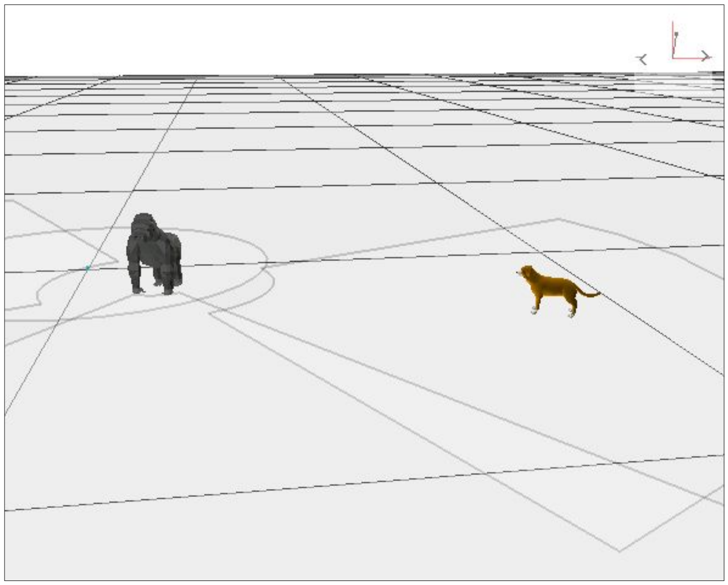


Figure 9.7: Relation TO-SIDE-OF for GORILLA and DOG

Section 5.7.2.3 discussed various spatial interactions of fields in terms of set operations. Figure 9.8 illustrates four intersections in *the rhino is north of the lake, in front of and midrange from the elephant, and facing away from the maple tree*.

Section 5.7.2.1.1.1 discussed how the size of objects heavily influences the scale of their fields. Figure 9.9 illustrates this behavior for (a) *the turtle is near the elephant* and (b) *the elephant is near the turtle*.

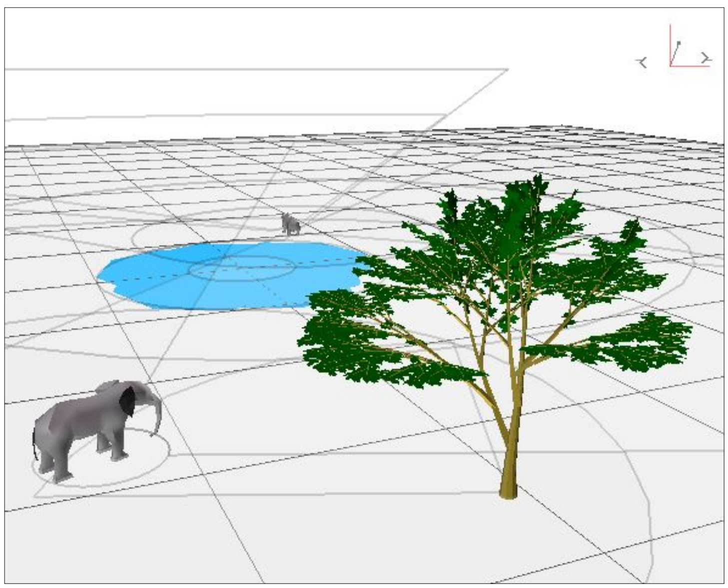
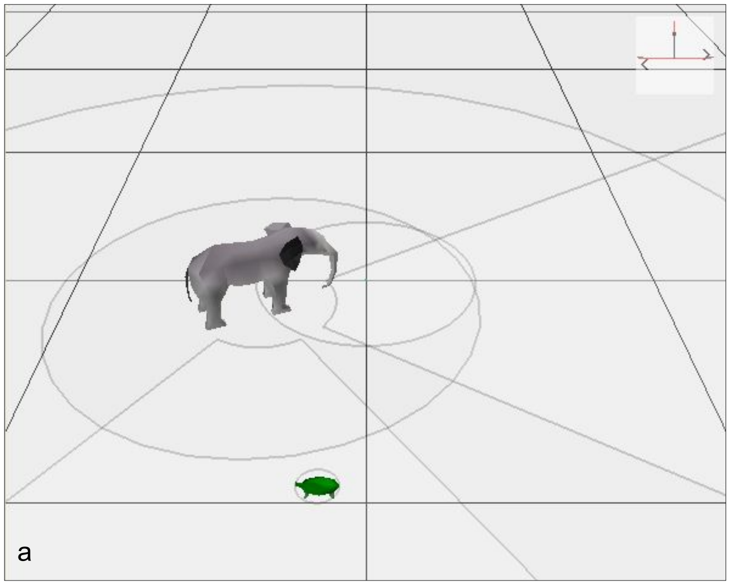


Figure 9.8: Field Intersections



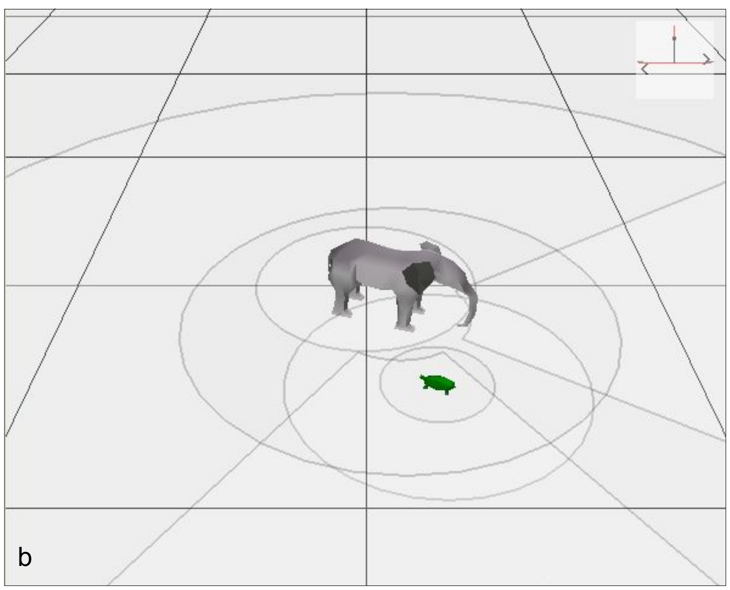


Figure 9.9: Relation NEAR for ELEPHANT and TURTLE

Finally, Section 6.1.1.1 discussed how the components of descriptions may form independent partitions of objects that reside in the same interpretation but do not interact. Figure 9.10 illustrates this behavior for (partition 1) *the dog is south of the tree and near the panther, and the panther is right of the dog* and (partition 2) *the elk is midrange from and facing away from the lake and near the maple tree*. This example also serves as a negative result.

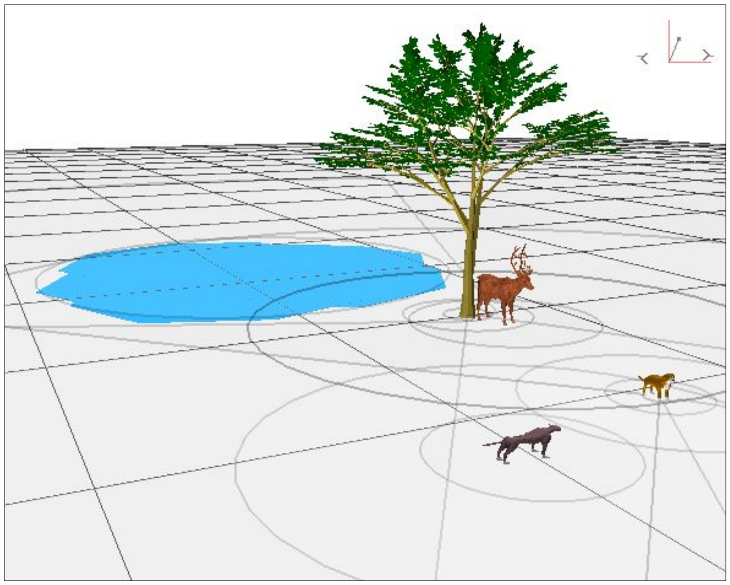


Figure 9.10: Independent Dependency Graphs

9.2 Negative Results

The combination of tightly defined constraints, an all-or-nothing constraint satisfaction algorithm, and a strict postchecking mechanism unsurprisingly guarantees that all results will reflect their defined behavior. Thus, only incorrect or inadequate definitions can produce negative results. All known cases are *errors of omission*, in which behaviors that should be present are not (through fault of the implementation and/or the knowledge engineer); none are *errors of commission*, in which the results are not justified by the premises [100]. As this project performs above and beyond its proposed requirements, the contents of this section are more appropriately considered issues and limitations rather than incorrect results.

Several negative results were acknowledged in the previous section. In Figure 9.3, the eagle is in the tree, but no branches support it. In fact, the eagle is actually partially embedded in the trunk. This visual inconsistency stems from the decoupled nature of the reasoning and rendering components of this project. The knowledge base has no information about the detailed, compositional structure of the graphical models and therefore cannot enforce such fine-grained interpretations.

In Figure 9.4, the size of the cage reflects the size class of its contents. This mechanism functions correctly for a single object in the cage; e.g., either the monkey or the gorilla. However, it does not currently resolve conflicts between multiple objects. For example, if the cage contains both a monkey and a gorilla, common sense dictates that the cage must reflect the size class of the larger of the two objects. Similarly, two gorillas should make the cage roughly twice as large. Finally, physical size alone is not a good indicator for the size of the cage; e.g., an eagle is a small animal, but its real-world behavior demands a large cage for freedom of mobility.

In Figure 9.10, the elk is midrange from the lake. Although the depiction appears reasonable, the underlying representation is slightly flawed. A lake is a large, ill-defined, geographical feature that technically violates the criteria for objects in Section 3.1. The unseen problem it introduces is that objects are processed in the reasoning engine as *point sources*, or single points in space, and not as the shape they really exhibit. Thus, a lake is represented by a point in its center, and the elk is actually midrange from *it*, not from the shore. The noninterpenetration postconstraint (Section 6.1.1.1) dictates that the elk should not appear in the lake, so the position of the elk reflects the closest valid position to the center of the lake that is not actually in it. A person would probably say that the elk is *near* the lake.

The polar projection of rings and sectors (Section 5.7.2.1) is appropriate for the animals and plants in this project. However, it does not extend well to non-circular objects because of mismatches between polar and Cartesian coordinate systems. For example, no suitable fields exist for a river because it is primarily a rectangular object. In order to support the opening sentence of this dissertation, "[p]icture yourself on a boat in a river," a bow-tie field similar to the LEFT-RIGHT field in Figure 5.33 is used. Another concession of this nature is the use of a round corral and lake.

Another unseen issue arises due to shape. All objects are approximated as vertically oriented cylinders (Section 5.7.2.1.2), but not all objects conform to such an abstraction. The anacondas in Figure 9.5, for example, are poorly represented this way because their length dictates the diameter of their cylinder, even though they occupy only a narrow chord of it. This limitation is wasteful in terms of available space for valid positions because bounding cylinders normally cannot interpenetrate. Thus, Figure 9.5 represents the tightest parallel depiction of two snakes. One easy solution is to define snakes with a true `is-container` property so they ignore the noninterpenetration rule, but this hack would undermine the validity of the knowledge base because snakes are not actually containers, of course [27]. The bounding cylinder introduces further problems as well because it assumes the same diameter from bottom to top. Clearly the space under the giraffes can accommodate the anacondas, but this solution is not available to the reasoning engine. A finer-grained bounding mechanism such as constructive solid geometry or a volumetric representation is more appropriate [26, 98, 87].

9.3 Future Work

While this project arguably presents a successful approach to addressing its stated issues and achieving its goals, it is hardly a complete and final solution to the larger problem of representing and reasoning over spatial descriptions. Two practical considerations deserve attention: *scalability* and *extensibility*.

9.3.1 Scalability

Scalability is the measure by which this approach can expand its *depth* to handle more of the *current* domain. Such expansion would likely require only additions to the knowledge base but no new functionality to the spatial reasoning engine. For example, new relations for PARALLEL-TO and PERPENDICULAR-TO can be defined in terms of the current formalism of fields. In this respect, this project exhibits good scalability [14]. In fact, this exact approach was used to add the last-minute relation TO-SIDE-OF as the union of the LEFT-OF and RIGHT-OF relations. Scalability in this context applies to increasing the number of available concepts, attributes, and relations of existing types. It does not apply to increasing the number of objects that the constraint satisfaction engine can handle. As Section 6.1.1 discussed, humans have inherent limitations in the maximum number that can be reasonably processed in a spatial description. Improving this performance is a different issue.

9.3.2 Extensibility

Extensibility is the measure by which this approach can expand its *breadth* to handle *other* domains. Such expansion would definitely require additions to the knowledge base and new functionality to the spatial reasoning engine. Extensibility applies to increasing the number of available concepts, attributes, and relations of new types. For example, Egenhofer's 9-intersection model [34] and the region connection calculus of Randell, Cui, and Cohn [99] each define dozens of tricky relations that would not fit easily into the current formalism of fields on a polar projection [108]. In this respect, this project exhibits limited extensibility [14]. However, such expansion is not believed to be totally inconsistent with this formalism, so "limited extensibility" in this respect is more a judgment of immediate versus long-term, potential expansion.

10 CONCLUSION

It terms of the proposed goals of this project, it is a complete success beyond expectation as the following accomplishments summarize:

- The research side identified various interdisciplinary issues in computational text understanding that play a role in interpreting spatial descriptions. It focused on five that were hypothesized as most important: underspecification, uncertainty, vagueness, context, and frame of reference.
- The development side built a multidimensional, knowledge-based, weak artificial-intelligence system to investigate these issues.
- The input component of the system accepts spatial descriptions in a restricted form of English and translates this explicit knowledge into a semantic-network representation of object nodes, attribute nodes, and relation arcs that serves as the primary data structure throughout the system.
- A complex knowledge base defines a wide range of implicit knowledge to cope with the stated issues in understanding a description. It builds a collection of declarative constraints that define the form of a valid solution.

- The spatial reasoning engine processes the constraints from the semantic network and the knowledge base to generate solutions with contextually appropriate dimensions, positions, and orientations for the objects in a description. It also generates new knowledge from a solution in the form of inferences, which contribute to a better computational understanding of a description.
- The process of generating inferences also exhibits unexpected potential to recognize scenes and produce limited, corresponding natural language descriptions.
- The Monte Carlo simulation framework collects multiple, independent solutions for several types of analyses that contribute to improving the performance of the system and to discovering unknown interactive behaviors in spatial descriptions.
- The graphical rendering engine serves as both a display and an evaluation platform to depict one or more three-dimensional, virtual worlds that correspond to different interpretations of a description.

The success of this project as a knowledge representation and automated reasoning system is evaluated in terms of many requirements [14, 40, 27, 130, 87]. In particular, this system satisfies the following:

- *Generality*: it represents and reasons over a broad range of spatial knowledge within its domain.
- *Granularity*: it operates at various levels of abstraction and detail.
- *Competence*: it produces correct results.
- *Inference*: it possesses the capability to answer a wide range of questions.
- *Explanation of inference*: it can list the rules that it used to produce an answer.
- *Meta-reasoning*: it is aware of what it knows and does not know.
- *Expressiveness*: it allows the knowledge engineer to say what he or she wants within the domain.
- *Naturalness or perspicuity*: it is intuitive, syntactically friendly, and relatively straightforward to configure and use.
- *Semantic clarity*: it has a clear, well-defined semantics.
- *Transformability*: it can be used other purposes.
- *Contexts and knowledge encapsulation*: it maintains a coherent structure of related knowledge.
- *Graphics*. it provides intuitive, visual access to its internals.
- *Efficiency*: it is a reasonably small and fast implementation.

These accomplishments contribute to a better understanding of several issues. First, knowledge representation benefited from the following observations:

- A shallow ontology of concepts is adequate to represent animals and plants because there is surprisingly little variation in their spatial behavior.
- The shallowness of the ontology may account for why few problems arose from the support of multiple inheritance.
- Mapping the separate ontology of spatial behaviors onto the existing taxonomy of the animal kingdom reduces the amount of design work and highlights structural inconsistencies early in the process.
- The encapsulated, object-oriented structure of the knowledge base provides a clean, intuitive framework to define implicit knowledge.
- The declarative paradigm of knowledge representation cleanly decouples the form of knowledge from the implementation of the mechanisms that process it.
- The knowledge base is scalable to accommodate additional knowledge with the current domain.
- The knowledge base is reasonably extensible to other domains.

Second, spatial reasoning benefited from the following observations, in descending order of importance:

- *Underspecification*: without the implicit knowledge of rules and guidelines for acceptable interpretation, spatial reasoning would be nearly impossible.
- *Uncertainty*: any reasoning mechanism must account for the wide range of valid interpretations of any spatial description.
- *Frame of reference*: different concepts exhibit different behaviors with respect to other objects and the viewer. Proper placement and alignment depends on a correct or plausible interpretation.
- *Context*: although it plays an important role in non-default interpretations, the vast majority of interpretations adhere to the defaults. Default interpretations applied to non-default contexts may appear out of place, but they are still reasonably consistent with a correct interpretation.
- *Vagueness*: the dimensions of objects need only be proportional and reasonably on target, so precise computation of them is unnecessary. It is difficult to perceive minor differences in size under normal circumstances.

Finally, constraint satisfaction proved itself as a viable approach to spatial reasoning:

- Declarative constraints are a clean, concise representation for the form of a valid solution.
- Although constraint satisfaction is far too slow for real-time, dynamic reasoning, it is effective for static reasoning within this restricted domain.
- Interval constraints provide suitable control over the range of dimensions of objects.
- Geometric and topographical fields are simple but powerfully flexible constraints for uncertain positions and orientation. They also appear consistent with the way humans perceive and reason about space.
- Only facets and rings are necessary to handle the majority of common spatial relations.
- Only field intersection is necessary to process most spatial descriptions. Union, symmetric difference, and complement are mainly of theoretical value as they have few realistic counterparts in the spatial descriptions.
- Three-and-a-half degrees of freedom are acceptable to manipulate the majority of objects in typical ways. Full three-dimensional reasoning has definite applications, but the cost-benefit ratio must be considered.
- An adjustable Gaussian distribution realistically handles uncertainty and variation of dimensions, position, and orientation.

APPENDICES

A. KNOWLEDGE BASE GRAMMAR

The following annotated grammar specifies the syntax and semantics of the subset of the knowledge base in Appendix B. It modifies Extended Backus-Naur Form slightly to indicate the data type of each terminal as a subscript where it is not obvious: `s` for string and `n` for real or integer number.

```
<KNOWLEDGE_BASE> := ('(' <ABSTRACT_CONCEPT> | <DERIVED_CONCEPT> ')')*
```

The knowledge base is an inheritance hierarchy or ontology of abstract and derived concepts.

See Section 5.5.1.1.

```
<ABSTRACT_CONCEPT> :=
   'ABSTRACT_CONCEPT' concept_name_s <CONCEPT_BODY>
```

An abstract concept defines a top-level concept that can be inherited by any DERIVED_CONCEPT. It cannot inherit from other concepts or be instantiated.

See Section 5.5.1.1.2.

```
<ATTRIBUTE_INTERVAL> :=
   'ATTRIBUTE-INTERVAL'
   lower_bound_attribute_s [lower_bound_adjustment_n] '...'
   upper_bound_attribute_s [upper_bound_adjustment_n] ':' dimension_s+
```

An attribute interval associates a range of English adjectives with one or more physical dimensions of a concept.

See Section 5.5.1.2.2.

```
<CONCEPT_BODY> :=
   ['(' <MODEL> ')'] ['(' <DOF_ADJUSTMENT> ')']
   ('(' (<PROPERTY> | <RELATION> | <ATTRIBUTE_INTERVAL> |
         <CONTEXT> | <FIELD> | <INFERENCE> | <LATE_DEPENDENCY>) ')')*
```

A concept body defines the components of a concept or context.

See Section 5.3.2.

<CONSTRAINT> := 'CONSTRAINT::FIELD-MUST-CONTAIN' (referent$_s$ | field$_s$)+

A constraint specifies that a field of one object must contain another object for a relationship to be satisfied.

See Section 5.7.2.

<CONTEXT> := 'CONTEXT' (context_name$_s$ ['+'])+ <CONCEPT_BODY>

A context is an embedded concept definition that overrides the main concept definition when an object is used in a relationship with another specified object.

See Section 5.6.1.

<DEPENDENCY_ARGUMENTS> :=
   (identifier | referent | literal | number | boolean)+

Dependency arguments pass constant or variable data into dependencies for evaluation. A `referent` is an identifier with a `?` prefix. A `literal` is any string enclosed in single quotes.

See Section 5.6.2.

<DERIVED_CONCEPT> :=
   'DERIVED-CONCEPT' concept_name$_s$
   'IS-A' concept_name$_s$+ <CONCEPT_BODY>

A derived concept defines components to extend the definitions inherited from any number of abstract and derived concepts. All derived concepts inherit from `thing` and can be instantiated.

See Section 5.5.1.1.1.

<DOF_ADJUSTMENT> := 'DOF_ADJUSTMENT' x$_n$ y$_n$ z$_n$ pitch$_n$ roll$_n$ yaw$_n$

A degree-of-freedom adjustment defines how the position and attitude of the toy representing an object is offset in three-dimensional space. The position and attitude offsets are given in terms of the coordinate system in Figure 8.3.

See Section 5.6.1.1 and Section 6.1.1.3.

```
<DYNAMIC_DEPENDENCY> :=
   ('DYNAMIC-DEPENDENCY::IS-IN-FIELD'         |
    'DYNAMIC-DEPENDENCY::DIMENSION-IS-LESS'    |
    'DYNAMIC-DEPENDENCY::DIMENSION-IS-MORE'    |
    'DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE'     |
    'DYNAMIC-DEPENDENCY::PROPERTY-IS-FALSE'    |
    'DYNAMIC-DEPENDENCY::PROPERTY-IS-EQUAL'    |
    'DYNAMIC-DEPENDENCY::PROPERTY-IS-UNEQUAL'  |
    'DYNAMIC-DEPENDENCY::PROPERTY-IS-LESS'     |
    'DYNAMIC-DEPENDENCY::PROPERTY-IS-MORE'     |
    'DYNAMIC-DEPENDENCY::PROPERTY-IS-PRESENT'  |
    'DYNAMIC-DEPENDENCY::PROPERTY-IS-ABSENT'   |
    'DYNAMIC-DEPENDENCY::ATTRIBUTE-IS-PRESENT' |
    'DYNAMIC-DEPENDENCY::ATTRIBUTE-IS-ABSENT')
   [<DEPENDENCY_ARGUMENTS>]
   (('(' <INFERENCE> ')')+ | ('(' <DYNAMIC_DEPENDENCY> ')'))
```

A dynamic dependency specifies a set of conditions that can be evaluated only after a solution has been generated.

See Section 5.6.2.2.

```
<EARLY_DEPENDENCY> :=
   <STATIC_DEPENDENCY_FUNCTION> [<DEPENDENCY_ARGUMENTS>]
   ('(' <INFERENCE> | <CONSTRAINT> ')')+
```

An early dependency is evaluated before all constraints have been satisfied to generate early inferences and/or constraints.

See Section 5.6.2.1.

```
<FIELD> := 'FIELD' names definitions
```

A field defines the geometry and topography of a region.

See Section 5.7.2.

```
<INFERENCE> :=
   ('INFER-ATTRIBUTE' attribute_names) |
   ('INFER-RELATIONSHIP' relation_names referentr*)
```

An inference adds an attribute node or relation arc to the semantic network based on knowledge that was inferred from a solution.

See Section 5.5.2.

```
<LATE_DEPENDENCY> :=
   <DYNAMIC_DEPENDENCY> |
   (<STATIC_DEPENDENCY_FUNCTION> [<DEPENDENCY_ARGUMENTS>]
    '(' <LATE_DEPENDENCY> ')')
```

A late dependency is evaluated after all constraints have been satisfied to generate late inferences.

See Section 5.6.2.2.

```
<MODEL> := 'MODEL' model_names
```

A model binds a derived concept to a three-dimensional, graphical model of polygons in the model library.

See Section 8.2.

```
<PROPERTY> :=
   'PROPERTY' property_name
   (literal | number | boolean |
     ('(' 'RANGE' absolute_lower_boundn suggested_lower_boundn
      defaultn suggested_upper_boundn absolute_upper_boundn ':'
      variationn bandpassn proportionality_percentagen ')'))
```

A property defines a data element for a concept as either a primitive value or a range of values. A `literal` is any string enclosed in single quotes.

See Section 5.5.1.2.1.

```
<RELATION> :=
   'RELATION' relation_names
   ['(' <MODEL> ')'] ['(' <DOF_ADJUSTMENT> ')']
   ('(' (<EARLY_DEPENDENCY> | <CONSTRAINT> | <INFERENCE>) ')')*
```

A relation defines how to interpret a relationship between two objects.

See Section 5.5.1.3.

```
<STATIC_DEPENDENCY_FUNCTION> :=
   'STATIC-DEPENDENCY::PROPERTY-IS-TRUE'      |
   'STATIC-DEPENDENCY::PROPERTY-IS-FALSE'     |
   'STATIC-DEPENDENCY::PROPERTY-IS-EQUAL'     |
   'STATIC-DEPENDENCY::PROPERTY-IS-UNEQUAL'   |
   'STATIC-DEPENDENCY::PROPERTY-IS-LESS'      |
   'STATIC-DEPENDENCY::PROPERTY-IS-MORE'      |
   'STATIC-DEPENDENCY::PROPERTY-IS-PRESENT'   |
   'STATIC-DEPENDENCY::PROPERTY-IS-ABSENT'    |
   'STATIC-DEPENDENCY::ATTRIBUTE-IS-PRESENT'  |
   'STATIC-DEPENDENCY::ATTRIBUTE-IS-ABSENT'
```

A static dependency function specifies a condition that can be evaluated either before or after a solution has been generated.

See Section 5.6.2.1.

B. SUBSET OF KNOWLEDGE BASE

```
(ABSTRACT-CONCEPT thing

  ;---[ PROPERTIES AND ATTRIBUTES ]------------------------------------

  (PROPERTY supports-dimension-comparison true)

  (PROPERTY is-container false)

  (PROPERTY has-canonical-front true)

  (PROPERTY height (RANGE 1.0 1.0 1.0 1.0 1.0 : 0.25 1.0 1.0))
  (PROPERTY width  (RANGE 1.0 1.0 1.0 1.0 1.0 : 0.25 1.0 1.0))
  (PROPERTY depth  (RANGE 1.0 1.0 1.0 1.0 1.0 : 0.25 1.0 1.0))

  (ATTRIBUTE-INTERVAL small  ... big  : height width depth)
  (ATTRIBUTE-INTERVAL narrow ... wide : width             )
  (ATTRIBUTE-INTERVAL short  ... tall : height            )

  ;--[ FIELDS ]--------------------------------------------------------

  (FIELD field-front        [front : 02.29-50.04 : : ])
  (FIELD field-front-left   [front : 02.27-50.30 : : ])
  (FIELD field-front-right  [front : 02.03-50.06 : : ])
  (FIELD field-back         [front : 02.13-50.20 : : ])
  (FIELD field-back-left    [front : 02.19-50.22 : : ])
  (FIELD field-back-right   [front : 02.11-50.14 : : ])
  (FIELD field-left         [front : 02.21-50.28 : : ])
  (FIELD field-right        [front : 02.05-50.12 : : ])
  (FIELD field-left-right   [front : 04.21-15.28 04.05-15.12 : 0 : ])

  (FIELD field-direct-front [front : 02.32-50.01 : : ])
  (FIELD field-direct-back  [front : 02.16-50.17 : : ])
  (FIELD field-direct-left  [front : 02.24-50.25 : : ])
  (FIELD field-direct-right [front : 02.08-50.09 : : ])

  (FIELD field-interior     [north : 01.01-01.32 : : ])
  (FIELD field-exterior     [north : 02.01-20.32 : : ])
  (FIELD field-adjacent     [north : 02.01-03.32 : : ])
  (FIELD field-near         [north : 04.01-15.32 : : ])
  (FIELD field-midrange     [north : 16.01-30.32 : : ])
  (FIELD field-far          [north : 31.01-47.32 : : ])
  (FIELD field-fringe       [north : 48.01-60.32 : : ])

  (FIELD field-north        [north : 02.30-95.03 : : ])
  (FIELD field-south        [north : 02.14-95.19 : : ])
  (FIELD field-east         [north : 02.06-95.11 : : ])
  (FIELD field-west         [north : 02.22-95.27 : : ])
  (FIELD field-northeast    [north : 02.04-95.05 : : ])
  (FIELD field-northwest    [north : 02.28-95.29 : : ])
  (FIELD field-southeast    [north : 02.12-95.13 : : ])
  (FIELD field-southwest    [north : 02.20-95.21 : : ])
  (FIELD field-east-west    [north : 04.06-15.11 04.22-15.27 : 0 : ])

  (FIELD field-direct-north [north : 02.32-95.01 : : ])
  (FIELD field-direct-south [north : 02.16-95.17 : : ])
  (FIELD field-direct-east  [north : 02.08-95.09 : : ])
  (FIELD field-direct-west  [north : 02.24-95.25 : : ])

  (FIELD field-anywhere     [north : 03.01-10.32 : : ])
```

```
;---[ GLOBAL RELATIVE POSITION RELATIONS ]---------------------------

(RELATION north-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-north     ?self))

(RELATION south-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-south     ?self))

(RELATION east-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-east      ?self))

(RELATION west-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-west      ?self))

(RELATION northeast-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-northeast ?self))

(RELATION northwest-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-northwest ?self))

(RELATION southeast-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-southeast ?self))

(RELATION southwest-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-southwest ?self))

(RELATION directly-north-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-north     ?self))

(RELATION directly-south-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-south     ?self))

(RELATION directly-east-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-east      ?self))

(RELATION directly-west-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-west      ?self))

;---[ LOCAL RELATIVE POSITION RELATIONS ]----------------------------

(RELATION in-front-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-front ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-south ?self)))

(RELATION in-back-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-back ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-north ?self)))

(RELATION left-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-left ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-west ?self)))

(RELATION right-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-right ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-east ?self)))
```

```
(RELATION to-side-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-left-right ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-east-west ?self)))

(RELATION in-front-left-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-front-left ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-southwest ?self)))

(RELATION in-front-right-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-front-right ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-southeast ?self)))

(RELATION in-back-left-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-back-left ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-northwest ?self)))

(RELATION in-back-right-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-back-right ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-northeast ?self)))

(RELATION directly-in-front-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-direct-front ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-direct-south ?self)))

(RELATION directly-in-back-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-direct-back ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-direct-north ?self)))

(RELATION directly-left-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-direct-left ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-direct-west ?self)))

(RELATION directly-right-of
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-direct-right ?self))
  (STATIC-DEPENDENCY::PROPERTY-IS-FALSE ?b.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-direct-east ?self)))

;---[ RELATIVE DISTANCE RELATIONS ]---------------------------------

(RELATION in
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-interior ?self))

(RELATION inside
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-interior ?self))
```

```
(RELATION outside
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-exterior ?self))

(RELATION on
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-interior ?self))

(RELATION adjacent-to
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-adjacent ?self))

(RELATION near
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-near     ?self))

(RELATION midrange-from
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-midrange ?self))

(RELATION far-from
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-far      ?self))

(RELATION at-fringe-of
  (CONSTRAINT::FIELD-MUST-CONTAIN ?b.field-fringe   ?self))

;---[ RELATIVE ORIENTATION RELATIONS ]-------------------------------

(RELATION facing
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-front ?b)))

(RELATION directly-facing
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-direct-front ?b)))

(RELATION facing-away-from
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-back ?b)))

(RELATION directly-facing-away-from
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-direct-back ?b)))

;---[ ABSOLUTE ORIENTATION RELATIONS ]-------------------------------

(RELATION facing-north
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-north ?self.pipper)))

(RELATION facing-south
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-south ?self.pipper)))

(RELATION facing-east
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-east ?self.pipper)))

(RELATION facing-west
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-west ?self.pipper)))

(RELATION facing-direct-north
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-direct-north
                                    ?self.pipper)))
```

```
(RELATION facing-direct-south
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-direct-south
                                    ?self.pipper)))
(RELATION facing-direct-east
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-direct-east
                                    ?self.pipper)))
(RELATION facing-direct-west
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-direct-west
                                    ?self.pipper)))
(RELATION facing-northeast
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-northeast
                                    ?self.pipper)))
(RELATION facing-northwest
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-northwest
                                    ?self.pipper)))
(RELATION facing-southeast
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-southeast
                                    ?self.pipper)))
(RELATION facing-southwest
  (STATIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
    (CONSTRAINT::FIELD-MUST-CONTAIN ?self.field-southwest
                                    ?self.pipper)))

;---[ LOCAL RELATIVE POSITION INFERENCES ]--------------------------

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-front
    (INFER-RELATIONSHIP local-in-front-of            ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-back
    (INFER-RELATIONSHIP local-in-back-of             ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-left
    (INFER-RELATIONSHIP local-left-of                ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-right
    (INFER-RELATIONSHIP local-right-of               ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-front-left
    (INFER-RELATIONSHIP local-in-front-left-of       ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-front-right
    (INFER-RELATIONSHIP local-in-front-right-of      ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-back-left
    (INFER-RELATIONSHIP local-in-back-left-of        ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-back-right
    (INFER-RELATIONSHIP local-in-back-right-of       ?self ?any)))
```

```
(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-front
    (INFER-RELATIONSHIP local-directly-in-front-of   ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-back
    (INFER-RELATIONSHIP local-directly-in-back-of    ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-left
    (INFER-RELATIONSHIP local-directly-left-of       ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-right
    (INFER-RELATIONSHIP local-directly-right-of      ?self ?any)))

;---[ GLOBAL RELATIVE POSITION INFERENCES ]--------------------------

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-south
    (INFER-RELATIONSHIP global-in-front-of           ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-north
    (INFER-RELATIONSHIP global-in-back-of            ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-west
    (INFER-RELATIONSHIP global-left-of               ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-east
    (INFER-RELATIONSHIP global-right-of              ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-southwest
    (INFER-RELATIONSHIP global-in-front-left-of      ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-southeast
    (INFER-RELATIONSHIP global-in-front-right-of     ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-northwest
    (INFER-RELATIONSHIP global-in-back-left-of       ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-northeast
    (INFER-RELATIONSHIP global-in-back-right-of      ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-south
    (INFER-RELATIONSHIP global-directly-in-front-of  ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-north
    (INFER-RELATIONSHIP global-directly-in-back-of   ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-west
    (INFER-RELATIONSHIP global-directly-left-of      ?self ?any)))
```

```
(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-east
    (INFER-RELATIONSHIP global-directly-right-of     ?self ?any)))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-north
  (INFER-RELATIONSHIP north-of          ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-south
  (INFER-RELATIONSHIP south-of          ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-east
  (INFER-RELATIONSHIP east-of           ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-west
  (INFER-RELATIONSHIP west-of           ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-northeast
  (INFER-RELATIONSHIP northeast-of      ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-northwest
  (INFER-RELATIONSHIP northwest-of      ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-southeast
  (INFER-RELATIONSHIP southeast-of      ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-southwest
  (INFER-RELATIONSHIP southwest-of      ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-north
  (INFER-RELATIONSHIP directly-north-of ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-south
  (INFER-RELATIONSHIP directly-south-of ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-east
  (INFER-RELATIONSHIP directly-east-of  ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-direct-west
  (INFER-RELATIONSHIP directly-west-of  ?self ?any))

;---[ RELATIVE DISTANCE INFERENCES ]---------------------------------

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-interior
  (INFER-RELATIONSHIP in             ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-interior
  (INFER-RELATIONSHIP inside         ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-exterior
  (INFER-RELATIONSHIP outside        ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-adjacent
  (INFER-RELATIONSHIP adjacent-to    ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-near
  (INFER-RELATIONSHIP near           ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-midrange
  (INFER-RELATIONSHIP midrange-from  ?self ?any))

(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-far
  (INFER-RELATIONSHIP far-from       ?self ?any))
```

```
(DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self ?any.field-fringe
  (INFER-RELATIONSHIP at-fringe-of   ?self ?any))

;---[ RELATIVE ORIENTATION INFERENCES ]------------------------------

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?any ?self.field-front
    (INFER-RELATIONSHIP facing   ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?any ?self.field-back
    (INFER-RELATIONSHIP facing-away-from ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?any ?self.field-direct-front
    (INFER-RELATIONSHIP directly-facing   ?self ?any)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?self.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?any ?self.field-direct-back
    (INFER-RELATIONSHIP directly-facing-away-from ?self ?any)))

;---[ ABSOLUTE ORIENTATION INFERENCES ]------------------------------

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self.pipper ?self.field-north
    (INFER-ATTRIBUTE facing-north    )))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self.pipper ?self.field-south
    (INFER-ATTRIBUTE facing-south    )))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self.pipper ?self.field-east
    (INFER-ATTRIBUTE facing-east     )))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self.pipper ?self.field-west
    (INFER-ATTRIBUTE facing-west     )))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self.pipper ?self.field-northeast
    (INFER-ATTRIBUTE facing-northeast)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self.pipper ?self.field-northwest
    (INFER-ATTRIBUTE facing-northwest)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self.pipper ?self.field-southeast
    (INFER-ATTRIBUTE facing-southeast)))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE ?any.has-canonical-front
  (DYNAMIC-DEPENDENCY::IS-IN-FIELD ?self.pipper ?self.field-southwest
    (INFER-ATTRIBUTE facing-southwest)))

;---[ RELATIVE DIMENSION INFERENCES ]--------------------------------

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
 ?self.supports-dimension-comparison
  (DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
   ?any.supports-dimension-comparison
    (DYNAMIC-DEPENDENCY::DIMENSION-IS-MORE height ?self ?any
      (INFER-RELATIONSHIP has-more-height ?self ?any))))
```

```
(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
 ?self.supports-dimension-comparison
  (DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
   ?any.supports-dimension-comparison
    (DYNAMIC-DEPENDENCY::DIMENSION-IS-MORE width  ?self ?any
      (INFER-RELATIONSHIP has-more-width  ?self ?any))))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
 ?self.supports-dimension-comparison
  (DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
   ?any.supports-dimension-comparison
    (DYNAMIC-DEPENDENCY::DIMENSION-IS-MORE depth  ?self ?any
      (INFER-RELATIONSHIP has-more-depth  ?self ?any))))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
 ?self.supports-dimension-comparison
  (DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
   ?any.supports-dimension-comparison
    (DYNAMIC-DEPENDENCY::DIMENSION-IS-LESS height ?self ?any
      (INFER-RELATIONSHIP has-less-height ?self ?any))))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
 ?self.supports-dimension-comparison
  (DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
   ?any.supports-dimension-comparison
    (DYNAMIC-DEPENDENCY::DIMENSION-IS-LESS width  ?self ?any
      (INFER-RELATIONSHIP has-less-width  ?self ?any))))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
 ?self.supports-dimension-comparison
  (DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
   ?any.supports-dimension-comparison
    (DYNAMIC-DEPENDENCY::DIMENSION-IS-LESS depth  ?self ?any
      (INFER-RELATIONSHIP has-less-depth  ?self ?any))))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
 ?self.supports-dimension-comparison
  (DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
   ?any.supports-dimension-comparison
    (DYNAMIC-DEPENDENCY::DIMENSION-IS-EQUAL height ?self ?any
      (INFER-RELATIONSHIP has-equal-height ?self ?any))))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
 ?self.supports-dimension-comparison
  (DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
   ?any.supports-dimension-comparison
    (DYNAMIC-DEPENDENCY::DIMENSION-IS-EQUAL width  ?self ?any
      (INFER-RELATIONSHIP has-equal-width  ?self ?any))))

(DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
 ?self.supports-dimension-comparison
  (DYNAMIC-DEPENDENCY::PROPERTY-IS-TRUE
   ?any.supports-dimension-comparison
    (DYNAMIC-DEPENDENCY::DIMENSION-IS-EQUAL depth  ?self ?any
      (INFER-RELATIONSHIP has-equal-depth  ?self ?any))))

) ; thing
```

This appendix omits the definitions of 103 of the 108 derived concepts in Table 5.1 for space reasons. It is important to note that they are significantly smaller than this definition of the abstract concept THING because they inherit (and thus do not duplicate) its contents. For example, the following definitions specify the lineage that a MAN is a HUMAN is a PRIMATE is a MAMMAL is an ANIMAL is a THING:

```
(DERIVED-CONCEPT animal
  (IS-A thing)
  ) ; animal

(DERIVED-CONCEPT mammal
  (IS-A animal)
  ) ; mammal

(DERIVED-CONCEPT primate
  (IS-A mammal)
  ) ; primate

(DERIVED-CONCEPT human
  (IS-A primate)
  ) ; human

(DERIVED-CONCEPT man
  (IS-A human)
  (MODEL man)
  (PROPERTY height (RANGE 1.48 1.64 1.73 1.82 2.00 : 0.25 0.17 0.43))
  (PROPERTY width  (RANGE 0.57 0.63 0.72 0.81 0.89 : 0.25 0.07 0.18))
  (PROPERTY depth  (RANGE 0.25 0.28 0.36 0.44 0.48 : 0.25 0.04 0.09))
  ) ; man
```

C. SAMPLE VIGNETTE

```
; ──────────────────────────────────────────────────────────────────
; This file defines a vignette, which consists of a description and a
; configuration that specifies how to interpret it.
; ──────────────────────────────────────────────────────────────────

; [DESCRIPTION]
;
; the name of this vignette

  vignette_name: viewer-in-east

; a brief description of this vignette

  vignette_description: this is a sample vignette

; the English description to render graphically.  Each line starts
; with a vertical bar |.

  | The scene contains an animal.

; [DEPICTION]
;
; whether to indicate clustering through colors that override the
; model colors

  show_clustering: false

; whether to display the region geometry meshes of each toy

  show_geometry: false

; whether to display the region topology meshes with the geometry;
; applies only for show_geometry=true

  show_topography: false

; whether to generate inferences

  generate_inferences: true

; whether to render everything in wireframe

  wireframe: false

; the pixel width and height of each sandbox window

  sandbox_width:  400
  sandbox_height: 400

; [SIMULATION]
;
; the number of simulations to run

  num_simulations: 1

; whether to disable nondeterministic behavior for dimensions

  force_determinism: false

; the random seed; use negative for automatic generation

  random_seed: 1
```

```
; the maximum number of propagation attempts within a partition
; before backtracking to an earlier partition
  max_partition_iterations: 1000000

; the maximum number of backtracks to earlier partitions
  max_partition_backtracks: 10

; [DISTILLATION]
;
; whether to return the composite interpretation of the aggregate
; sets as ghosts
  return_composite: false

; whether to return all interpretations or just one; overrides all
; distill_by_? settings
  return_all_otherwise_one: false

; whether position deviation contributes to distillation
  distill_by_position: false

; the minimum sigma difference between object positions to spawn a
; new interpretation; applies only for distill_by_position=true
  max_position_sigma: 1.0

; whether attitude deviation contributes to distillation
  distill_by_attitude: false

; the minimum sigma difference between object attitudes to spawn a
; new interpretation; applies only for distill_by_attitude=true
  max_attitude_sigma: 1.5

; whether dimensions deviation contribute to distillation
  distill_by_dimensions: false

; the minimum sigma difference between object dimensions to spawn a
; new interpretation; applies only for distill_by_dimensions=true
  max_dimensions_sigma: 1.0

; [KNOWLEDGE BASE]
;
; the fully qualified filename of the knowledge base
  knowledge_base_filename:
  /home/dtappan/java/source/KnowledgeBase/knowledge-base.dkb
```

REFERENCES

[1] Adorni, G., Di Manzo, M., and Ferrari, G., "Natural Language Input for Scene Generation," in Proc. First Conference of European Association for Computational Linguistics, pp. 175–182, 1983.

[2] Adorni, G., Di Manzo, M., and Giunchiglia, F., "Natural Language driven Image Generation," in Proc. COLING, pp. 495–500, 1984.

[3] Allen, J., "Maintaining Knowledge about Temporal Intervals," Communications of the ACM, vol. 26, no. 11. pp. 832–843, 1983.

[4] Allen, J., Natural Language Understanding, Redwood City, CA: Benjamin Cummings, 1995.

[5] Barták, R., "Guide to Constraint Programming," in Proc. Week of Doctoral Students, Part IV, Prague: MatFyzPress, pp. 555–564, 1999.

[6] Ben-Avi, S., "Conversation with SHRDLU," lecture notes, available at www.ee.cooper.edu/courses/course_pages/past_courses/EE459, 1997.

[7] Bennett, B., "Some Observations and Puzzles about Composing Spatial and Temporal Relations," in Proc. ECAI Workshop on Spatial and Temporal Reasoning, pp. 65–72, 1994.

[8] Bergen, B., "Image Schemas; Color; Frame Semantics," lecture notes, available at www.icsi.berkeley.edu/~bbergen/cogsci110/lectures/lecture12.html, 1999.

[9] Bierwisch, M. and Lang, E., eds., Dimensional Adjectives: Grammatical Structure and Conceptual Interpretation, Berlin: Springer, 1989.

[10] Boggess, L., "Spatial Operators in Natural Language Understanding: the Prepositions," in Proc. Seventeenth annual southeast regional conference, pp. 8–11, 1979.

[11] Car, A., "Hierarchical Spatial Reasoning: a GeoComputational Method," in Proc. Third International Conference on GeoComputation, 1997.

[12] Carello, C. and Turvey, M., "The Ecological Approach to Perception," lecture notes, available at ione.psy.uconn.edu/~psy254/CT_02.pdf, 2002.

[13] Church, K., “A stochastic parts program and noun phrase parser for unrestricted text,” in Proc. Second Conference on Applied Natural Language Processing, pp. 136–143, 1988.

[14] Clark, P., “Requirements for a Knowledge Representation System,” working notes, available at www.cs.utexas.edu/users/pclark/working_notes/010.pdf, 1996.

[15] Claus, B., Eyferth, K., Gips, C., Hörnig, R., Schmid, U., Wiebrock, S., Wysotzki, F., “Reference Frames for Spatial Inference in Text Understanding,” in Lecture Notes in Computer Science, no. 1404, Berlin: Springer, 1998.

[16] Clay, S. and Wilhelms, J., “Put: Language-Based Interactive Manipulation of Objects,” IEEE Computer Graphics and Applications, vol. 16, no. 2, pp. 31–39, 1995.

[17] Coenen, F. and Visser, P., “A Core Ontology for Spatial Reasoning,” Department of Computer Science, University of Liverpool, 1998.

[18] Cohn, A., “Qualitative Spatial Reasoning,” lecture notes, available at www.qrg.nwu.edu/MONET%20Summer%20School%202000/TC-B7.pdf, 2000.

[19] Cohn, A., “The Challenge of Qualitative Spatial Reasoning,” ACM Computing Surveys, vol. 27, no. 3, pp. 323–325, 1995.

[20] Collins, M., “Head-Driven Statistical Models for Natural Language Parsing,” Ph.D. Dissertation, University of Pennsylvania, 1999.

[21] Cormen, T., Leiserson, C., and Rivest, R., Introduction to Analysis of Algorithms, New York, NY: MIT Press, 1990.

[22] Cox, N., “Analysing circular data in Stata,” in North American Stata Users’ Group Meetings, no. 5.3, 2001.

[23] Coyne, B. and Sproat, R., “WordsEye: An Automatic Text-to-Scene Conversion System,” in Proc. SIGGRAPH, pp. 487–496, 2001.

[24] Crystal, D., Cambridge Encyclopedia of Language, Cambridge, England: Cambridge University Press, 1987.

[25] Crystal, D., Cambridge Encyclopedia of the English Language, Melbourne, Australia: Cambridge University Press, 1996.

[26] Davis, E., Representations of Commonsense Knowledge, Morgan Kaufmann: San Mateo, CA, 1990.

[27] Davis, R., Shrobe, H., and Szolovits, P., "What is Knowledge Representation?" AI Magazine, vol. 14, no. 1, pp. 17–33, 1993.

[28] Düntsch, I., Wang, H., and McCloskey, S., "Relations algebras in qualitative spatial reasoning," Fundamenta Informaticae, vol. 39, no. 3, pp. 229–248, 1999.

[29] Dupuy, S., Egges, A., Legendre, V., and Nugues, P., "Generating a 3D Simulation of a Car Accident from a Written Description in Natural Language: the CarSim System," in Proc. ACL Workshop on Temporal and Spatial Information Processing, pp. 1–8, 2001.

[30] Dzikovska, M. and Byron, D., "When is union really an intersection? Problems interpreting reference locations in a dialogue system," in Proc. Fourth workshop on the semantics and pragmatics of dialogue, 2000.

[31] Egenhofer, M. and Overmyer, S., "Multi-Modal Spatial Querying," in Proc. Seventh International Symposium on Spatial Data Handling Multi-Modal Spatial Querying, 1996.

[32] Egenhofer, M. and Shariff, A., "Metric Details for Natural-Language Spatial Relations," ACM Transactions on Information Systems, vol. 16, no. 4, pp. 295–321, 1998.

[33] Egenhofer, M., "Deriving the Composition of Binary Topological Relations," Journal of Visual Languages and Computing, vol. 5, no. 2, pp. 133–149, 1994.

[34] Egenhofer, M., "Reasoning about Binary Topological Relations," in Advances in Spatial Databases, Gunther, O., and Schek, H., eds., pp. 144–160, Berlin: Springer, 1991.

[35] Egenhofer, M., and Mark, D., "Naive Geography," in Spatial Information Theory: A Theoretical Basis for GIS, Frank, A. and Kuhn, W., eds., pp. 1–15, Berlin: Springer-Verlag, 1995.

[36] Emdanat, S. and Vakalo, E., "An Ontology for Conceptual Design in Architecture," in Proc. Third Conference on Computer Aided Architectural Design Research in Asia, pp. 425–434, 1998.

[37] Esterline, A., Dozier, G., and Homaifar, A., "Fuzzy Spatial Reasoning," in Proc. International Fuzzy Systems Association, 1997.

[38] Fellbaum, C., ed., WordNet: An Electronic Lexical Database, Cambridge, MA: MIT Press, 1998.

[39] Firth, J., "A Synopsis of Linguistic Theory 1930–1955," in Studies in Linguistic Analysis, Palmer, F., pp. 1–32, London: Longman, 1957.

[40] Fox, M., Chionglo, J., and Fadel, F., "A Common-Sense Model of the Enterprise," in Proc. Second Industrial Engineering Research Conference, pp. 425–429, 1993.

[41] Fraczak, L., "From route descriptions to sketches: a model for a text-to-image translator," in Proc. Association for Computational Linguistics, pp. 299–301, 1995.

[42] Gapp, K-P., "Basic Meanings of Spatial Relations: Computation and Evaluation in 3D Space," in Proc. Twelfth National Conference on Artificial Intelligence, pp. 1393–1398, 1994.

[43] Glasgow, J. and Malton, A., "A Semantics for Model-Based Spatial Reasoning," Technical Report 94–360, Department of Computing and Information Science, Queen's University, 1994.

[44] Glasgow, J. and Papadias, D., "Computational Imagery," in Mind Readings, Thagard, P., ed., pp. 157–205, Cambridge, MA: MIT Press, 1998.

[45] Graham, A., Choi, C., and Mullen, T., "Using Fit-Constrained Monte Carlo Trials to Quantify Confidence in Simulation Model Outcomes," in Proc. Hawaiian International Conference on System Sciences, pp. 96–104, 2002.

[46] Green, G., Pragmatics and Natural Language Understanding, Hillsdale, NJ: Lawrence Erlbaum, 1989.

[47] Guarino, N. and Giaretta, P., "Ontologies and Knowledge Bases: Towards a Terminological Clarification," in Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, Mars, N., ed., pp. 25–32, Amsterdam: IOS Press, 1995.

[48] Guarino, N., "Formal Ontology, Conceptual Analysis and Knowledge Representation," International Journal of Human Computer Interaction, vol. 43, no. 5–6, pp. 625–640, 1995.

[49] Guarino, N., "The Ontological Level," in Philosophy and Cognitive Science, Casati, R., Smith, B., and White, G., eds., pp. 443–456, Vienna: Hölder-Pichler-Tempsky, 1995.

[50] Gupta, K. and Nautiyal, S., "Artificial Intelligence in Engineering," progress report, available at www.cse.iitk.ac.in/~amit/courses/768/98/snauti/, 1998.

[51] Hausser, R., Computation of Language, New York, NY: Springer-Verlag, 1987.

[52] Hausser, R., Foundations of Computational Linguistics, Italy: Springer-Verlag, 1999.

[53] Hayes, P., "Knowledge Representation," available at www.cs.umbc.edu/771/papers/hayesp.html, 2001.

[54] Hernández, D. and Mukerjee, A., "Representation of spatial knowledge," in Tutorial Notes of IJCAI, 1995.

[55] Hernández, D., "Qualitative vs. Quantitative Spatial Knowledge," in Qualitative and Quantitative Approaches to Spatial Inference and the Analysis of Movements, Technical Report, Fakultät für Informatik, TU Berlin, Schmid, U. and Wysotzki, F., eds., pp. 9–12, 1998.

[56] Hernández, D., Qualitative Representation of Spatial Knowledge, Berlin, Germany: Springer-Verlag, 1994.

[57] Herskovits, A., "On the Spatial Uses of Prepositions," in Proc. Association for Computational Linguistics, pp. 1–5, 1980.

[58] Herskovits, A., Language and Spatial Cognition: An Interdisciplinary Study of the Prepositions in English, Cambridge, MA: Cambridge University Press, 1986.

[59] Herzog, G., “Coping with Static and Dynamic Spatial Relations,” in Proc. Time, Space, and Movement: Meaning and Knowledge in the Sensible World, pp. 47–59, 1995.

[60] Iwanska, L. and Shapiro, S., Natural Language Processing and Knowledge Representation, Menlo Park, CA: AAAI Press, 2000.

[61] Jackendoff, R., “Semantics and Cognition,” Cambridge, MA: MIT Press, 1983.

[62] Jannedy, S., Poletto, R., Weldon, T., eds., Language Files, Columbus, OH: Ohio State Press, 1994.

[63] Johnson-Laird, P., Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness, Cambridge, MA: Cambridge University Press, 1983.

[64] Kaehler, S., “Fuzzy Logic—An Introduction,” available at www.seattle robotics.org/encoder/mar98/fuz/fl_part1.html, 1998.

[65] Katz, G., “Natural Language Parsing for a Text-to-Scene Conversion System: The WordsEye System and its Foundations,” www.gkatz.net/ lawrence/wordseye.pdf, 2003.

[66] Kettani, D. and Maamar, Z., “Towards a Qualitative Spatial Model for Military GIS,” in Proc. International Conference on Advances in Intelligent Systems: Theory and Applications, pp. 160–164, 2000.

[67] Knauff, M., and Johnson-Laird, P., “Visual and Spatial Representations in Relational Reasoning,” in Proc. Twenty-Second Annual Conference of the Cognitive Science Society, pp. 759–765, 2000.

[68] Knuth, D., Art of Computer Programming, Volume II, 3rd ed., Reading, MA: Addison Wesley, 1997.

[69] Kopytko, R., “What is wrong with modern accounts of context in linguistics?” Vienna English working papers 12.1, pp. 45–60, June 2003.

[70] Kruse, R., Schwecke, E., and Heinsohn, J., Uncertainty and Vagueness in Knowledge Based Systems, New York, NY: Springer-Verlag, 1991.

[71] Kuipers, B., "A Hierarchy of Qualitative Representations for Space," Spatial Cognition, vol. 1404, pp. 337–350, 1998.

[72] Kuipers, B., "Modeling Spatial Knowledge," in Advances in Spatial Reasoning, Chen, S., ed., pp. 171–198, Norwood, NJ: Ablex, 1990.

[73] Kumar, V., "Algorithms for Constraint-Satisfaction Problems: A Survey," AI Magazine, vol. 13, no. 1, pp. 32–44, 1992.

[74] Lakoff, G., On Women, Fire, and Dangerous Things: What Categories Reveal about the Mind, Chicago, IL: University of Chicago Press, 1990.

[75] Langacker, R., Foundations of Cognitive Grammar, Volume I, Theoretical Prerequisites, Stanford, CA: Stanford University Press, 1987.

[76] Lennon, J. and McCartney, P., "Lucy in the Sky with Diamonds," 1967.

[77] Lindsay, R., "Images and Inference," Cognition, vol. 29, pp. 229–250, 1988.

[78] Mahesh, K. and Nirenburg, S., "A Situated Ontology for Practical NLP," In Proc. Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence, 1995.

[79] Mahesh, K., "Ontology Development for Machine Translation: Ideology and Methodology," Technical Report MCCS 96–292, Computing Research Laboratory, New Mexico State University, 1996.

[80 Mangus, M., "Description in Linguistic Theory," available at www.conknet.com/~mmagnus/Method, 1995.

[81] Manos, K., Panayiotopoulos, T., and Katsionis, G., "Virtual Director: Visualization of Simple Scenarios," in Proc. Second Hellenic Conference on Artificial Intelligence, SETN-02, pp. 3–14, 2002.

[82] Mark, D. and Freundschuh, S., "Spatial Concepts and Cognitive Models for Geographic Information Use," in Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems, Nyerges, T., Mark, D.,

Laurini, R., and Egenhofer, M., eds., Dordrecht: Kluwer Academic Publishers, pp. 21–28, 1995.

[83] Mark, D., "Cognitive Perspectives on Spatial and Spatio-Temporal Reasoning," in Geographic Information Research Bridging the Atlantic, Craglia, M. and Couclelis, H., pp. 308–319, London: Taylor and Francis, 1997.

[84] Mark, D., "Human Spatial Cognition," in Human Factors in Geographical Information Systems, Medyckyj-Scott, D. and Hearnshaw, H., eds., pp. 51–60, London: Belhaven Press, 1993.

[85] Mark, D., "Spatial Representation: A Cognitive View," in Geographical Information Systems: Principles and Applications, Second edition, Maguire, D., Goodchild, M., Rhind, D., and Longley, P., eds., vol. 1, pp. 81–89, 1999.

[86] Mark, D., Skupin, A., and Smith, B., "Features, Objects, and other Things: Ontological Distinctions in the Geographic Domain," in Spatial Information Theory, Montello, D., ed., pp . 488–502, Berlin: Springer, 2001.

[87] Markman, A., Knowledge Representation, New Jersey, NJ: Erlbaum, 1999.

[88] Montello, D., "Human Cognition of the Spatial World," NCGIA Core Curriculum in GIS, National Center for Geographic Information and Analysis, University of California, Santa Barbara, Unit 006, available at www.ncgia.ucsb.edu/giscc/units/u006/u006.html, 1997.

[89] Mooney, C., Monte Carlo Simulation, Newbury Park, CA: Sage, 1997.

[90] Mukerjee, A., "Neat vs Scruffy: A survey of Computational Models for Spatial Expressions," draft paper, available at www.cs.albany.edu/~amit/, 1996.

[91] Mukerjee, A., Gupta, K., and Nautiyal, S., "Conceptual Description of Visual Scenes from Linguistic Models," Journal of Image and Vision Computing, vol. 18, no. 2, pp. 173–187, 1999.

[92] Neufeldt, V., Guralnik, D., eds., Webster's New World Dictionary of American English, New York, NY: Simon & Schuster, 1988.

[93] Olivier, P. and Tsujii, J., "A Computational View of the Cognitive Semantics of Spatial Prepositions," in Proc. Association for Computational Linguistics, pp. 303–309, 1994.

[94] Olivier, P., Maeda, T., and Tsujii, J., "Automatic Depiction of Spatial Descriptions," in Proc. Twelfth National Conference on Artificial Intelligence, AAAI-94, pp. 1405–1410, 1994.

[95] Pinker, S., The Language Instinct: How the Mind Creates Language, New York, NY: Harper Collins, 1994.

[96] Qin, Y. and Simon, H., "Imagery and Mental Models," in Diagrammatic Reasoning: Computational and Cognitive Perspectives, Glasgow, J., Narayanan, N., and Chandrasekaran, B., eds., Menlo Park, CA: AAAI, MIT Press, 1995.

[97] Rade, L. and Bertil W., Beta Mathematics Handbook, 2nd ed., Boca Raton, FL: CRC Press, 1990.

[98] Raffaetà, A., Renso, C., and Turini, F., "Qualitative reasoning in a Spatio-Temporal Language," in Proc. Workshop on Complex Reasoning on Geographical Data, pp. 105–118, 2001.

[99] Randell, D., Cui, Z., and Cohn, A., "A Spatial Logic Based on Regions and Connection," in Proc. Principles of Knowledge Representation and Reasoning, Third International Conference, pp. 165–176, 1992.

[100] Rauh, R., Hagen, C., Schlieder, C., Strube, G., and Knauff, M., "Searching for Alternatives in Spatial Reasoning: Local Transformations and Beyond," in Proc. Twenty-Second Annual Conference of the Cognitive Science Society, pp. 871–876, 2000.

[101] Raynor, W., The International Dictionary of Artificial Intelligence, New York, NY: Glenlake Publishing, 1999.

[102] Renz, J. and Nebel, B., "Efficient Methods of Qualitative Spatial Reasoning," in Proc. Thirteenth European Conference on Artificial Intelligence, 1998.

[103] Retz-Schmidt, G., "Various views on spatial prepositions," AI Magazine, vol. 9, pp. 95–105, 1998.

[104] Richardson, D., Spivey, M., Edelman, S., and Naples, A., "Language is spatial: experimental evidence for image schemas of concrete and abstract verbs," in Proc. Twenty-Third Annual Meeting of the Cognitive Science Society, pp. 873–878, Mawhah, NJ: Erlbaum, 2001.

[105] Rohlfing, K., "Aspects in Understanding Spatial Relational Terms," Ph.D. dissertation, Bielefeld University, 2002.

[106] Russell, S. and Norvig, P., Artificial Intelligence: A Modern Approach, Upper Saddle River, NJ: Prentice-Hall, 1995.

[107] Schulz, B., "Development of an Interface and Visualization Components for a Text-to-Scene Converter," Master's Thesis, Master's Thesis, Lund Institute of Technology, 2002.

[108] Shariff, A. and Egenhofer, M., "Natural-Language Spatial Relations Between Linear and Areal Objects: The Topology and Metric of English-Language Terms," in International Journal of Geographical Information Science, vol. 12, no. 3, pp. 215–245, 1998.

[109] Sharma, J., Flewelling, D., and Egenhofer, M., "A Qualitative Spatial Reasoner," in Proc. Sixth International Symposium on Spatial Data Handling, pp. 665–681, 1994.

[110] Shinyama, Y., Tokunaga, T., and Tanaka, H., "Processing of 3-D Spatial Relations for Virtual Agents Acting on Natural Language Instructions," in Proc. Third International Workshop on Human-Computer Conversation, 2000.

[111] Smith, B. and Mark, D., "Ontology and Geographic Kinds," in Proc. Eighth International Symposium on Spatial Data Handling, pp. 308–320, 1998.

[112] Sowa, J., "Semantic Networks," available at www.jfsowa.com/pubs/semnet.htm, 2002.

[113] Sowa, J. and Majumdar, A., "Task-Oriented Semantic Interpretation," available at www.jfsowa.com/pubs/tosi.htm, 2003.

[114] Sowa, J., Knowledge Representation: Logical, Philosophical, and Computational Foundations, Pacific Grove, CA: Brooks/Cole, 2000.

[115] Sowa, J., Principles of Semantic Networks, San Mateo, CA: Morgan Kaufmann, 1991.

[116] Sproat, R., "Inferring the Environment in a Text-to-Scene Conversion System," in Proc. International Conference on Knowledge Capture, pp. 147–154, 2001.

[117] Sproat, R., "Language, Graphics, and Meaning," available at www.research.att.com/~rws/newindex/wegen.pdf, 2002.

[118] Srihari, R., "Computational Models for Integrating Linguistic and Visual Information: A Survey," Artificial Intelligence Review, vol. 8, pp. 349–369, 1995.

[119] Stopp, E., Gapp, K-P., Herzog, G., Langle, T., and Luth, T., "Utilizing Spatial Relations for Natural Language Access to an Autonomous Mobile Robot," in KI-94: Advances in Artificial Intelligence, Nebel, B. and Dreschler-Fischer, L., eds., pp. 39–50, Berlin: Springer, 1994.

[120] Svensson, H. and Åkerberg, O., "Development and Integration of Linguistic Components for an Automatic Text-to-Scene Conversion System," Master's Thesis, Lund Institute of Technology, 2002.

[121] Talmy, L., "How Language Structures Space," in Toward a Cognitive Semantics, Volume 1, Talmy, L., pp. 176–254, Cambridge, MA: MIT Press, 2000.

[122] Talmy, L., Toward a Cognitive Semantics, Volume 1, Cambridge, MA: MIT Press, 2000.

[123] Taylor, T., Gagné, C., and Eagleson, R., "Cognitive Constraints in Spatial Reasoning: Reference Frame and Reference Object Selection," in American Association for Artificial Intelligence Technical Report SS-00-04, pp. 168–172, Menlo Park, CA: AAAI Press, 2000.

[124] Towns, S., Callaway, C., and Lester, J., "Generating Coordinated Natural Language and 3D Animations for Complex Spatial Explanations," in Proc. Fifteenth National Conference on Artificial Intelligence, pp. 112–119, 1998.

[125] van Deemter, K. and Peters, S., eds., Semantic Ambiguity and Underspecification, Stanford, CA: CSLI Publications, 1996.

[126] Wahlster, W., "Text and images," in Survey of the State of the Art in Human Language Technology, Cole, R., Mariani, J., Uszkoreit, H., Zaenen, A., and Zue, V., eds., pp. 348–352, Dordrecht: Kluwer, 1996.

[127] Waltz, D., "Understanding Scene Descriptions as Event Simulations," in Proc. Association for Computational Linguistics, pp. 7–11, 1980.

[128] Wazinski, P., "Coping with Topological and Directional Relations Based on 3-Dimensional Geometric Representations," in Proc. Workshop on Spatial Concepts: Connecting Cognitive Theories with Formal Representations, Tenth European Conference on Artificial Intelligence, 1992.

[129] Weizenbaum, J., "ELIZA—A computer program for the study of natural language communication between man and machine," Communications of the ACM, vol. 9, no. 1, pp. 36–45, 1966.

[130] Welty, C., "An Integrated Representation for Software Development and Discovery," Ph.D. dissertation, Vassar College, 1995.

[131] Winograd, T., "Procedures as a Representation for Data in a Computer Program for Understanding Natural Language," Ph.D. dissertation, Massachusetts Institute of Technology, 1971.

[132] Winograd, T., Understanding Natural Language, New York, NY: Academic Press, 1972.

[133] Yamada, A., Yamamoto, T., Ikeda, H., Nishida T., and Doshita S., "Reconstructing spatial images from natural language texts," in Proc. Fourteenth International Conference on Computational Linguistics, pp. 1279–1283, 1992.

[134] Zhan, F., "Approximate analysis of binary topological relations between geographic regions with indeterminate boundaries," Soft Computing, vol. 2, no. 2, pp. 28–34, 1998.