Exploring Natural User Abstractions For Shared Perceptual Manipulator Task Modeling & Recovery

by

Seng Lee Koh
M.S. Information Technology Systems, University of Strathclyde, 2001
M.S. Modeling And Simulation, University of Central Florida, 2011

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Modeling And Simulation
in the College of Engineering And Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2018

Major Professor: Joseph LaViola Jr.

ABSTRACT

State-of-the-art domestic robot assistants are essentially autonomous mobile manipulators capable of exerting human-scale precision grasps. To maximize utility and economy, non-technical end-users would need to be nearly as efficient as trained roboticists in control and collaboration of manipulation task *behaviors.* However, it remains a significant challenge given that many WIMP-style tools require superficial proficiency in robotics, 3D graphics, and computer science for rapid task modeling and recovery. But research on *robot-centric* collaboration has garnered momentum in recent years; robots are now planning in *partially observable* environments that maintain geometries and semantic maps, presenting opportunities for non-experts to cooperatively control task behavior with autonomous-planning agents exploiting the knowledge. However, as autonomous systems are not immune to errors under perceptual difficulty, a human-in-the-loop is needed to bias autonomous-planning towards recovery conditions that resume the task and avoid similar errors.

In this work, we explore interactive techniques allowing non-technical users to model task behaviors and perceive cooperatively with a service robot under robot-centric collaboration. We evaluate stylus and touch modalities that users can intuitively and effectively convey natural abstractions of high-level tasks, semantic revisions, and geometries about the world. Experiments are conducted with 'pick-and-place' tasks in an ideal 'Blocks World' environment using a Kinova JACO six degree-of-freedom manipulator. Possibilities for the architecture and interface are demonstrated with the following features; (1) Semantic 'Object' and 'Location' grounding that describe function and ambiguous geometries (2) Task specification with an unordered list of *goal predicates*, and (3) Guiding task recovery with implied scene geometries and trajectory via symmetry cues and configuration space (C_{space}) abstraction. Empirical results from four user studies show our interface was much preferred than the control condition, demonstrating high learnability and ease-of-use that enable our non-technical participants to model complex tasks, provide effective recovery assistance, and teleoperative control.

To Lily, Peter, and Poh Lee, whose encouragement dragged this work across the finish line.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Joseph J. LaViola Jr., the most for his continual support and professional advice. Many thanks to the esteemed members of my dissertation advisory committee, Dr. Hassan Foroosh, Dr. Shaojie Zhang, and Dr. Si Jung “SJ” Kim, for their valuable feedback and guidance. Special thanks to past and present members of the Interactive Systems and User Experience Research Cluster of Excellence (ISUERCE), especially Kevin Pfeil, Andres Vargas Gonzalez, and Corey Pittman for their support and help on the research contained within this dissertation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Robots For the Home

Robot manipulators have been extensively used in the manufacturing industry since the 60s'. Common uses are found in assembly, palletizing, cargo handling, while complex examples include metal-working such as welding, deburring, grinding, polishing and spray-painting [Mil87]. They are also found in medical and military operations, such as remote surgery and assistive grasping, while the military use mobile manipulators typically for search-&-rescue and bomb-disposal under hazardous environments. Without a doubt, domestic robot manipulators have enormous potential, but they are difficult to utilize for two main reasons. First, household environments are dynamic and unstructured; without full observability, manipulator actions become error-prone and unsafe, necessitating re-programming or a re-plan of the affected tasks. It becomes unfeasible for home use as the domestic user may not be properly equipped to handle the stoppages. Second, users at home are very unlikely to be technically adept to task a robot; programming a manipulator competently for domestic chores inside *unstructured* and *partially observable* environments can be a huge challenge to overcome for non-technical users. Without a wide variety of tasks, the robot becomes under-utilized and no longer economical. These facts necessitate most robots at home today to be generally inexpensive ‘turn-key’ solutions[1] (switch-on-and-run) with little to no manipulation abilities, executing standard and simple pre-programmed tasks that require no intervention from a user.

Robot Manipulators For the Home

Robot manipulators offer more potential for performing a wide variety of tasks. They are operationally generic, capable of manipulating novel objects, and grasping tools for a variety of tasks (e.g. cleaning with a broom). Significant examples of mobile manipulators can be found in the PR2[2] and HERB[3]. In an on-line survey about where and how users would like to use robot manipulators [WMO12], household chores are the most common task category (46%) where cooking and cleaning activities are prominent. And coming in after workplace tasks are *desktop* or working-surface activities (e.g. writing a letter, clearing and cleaning a desk). The survey results underscored high expectations perceived of domestic mobile manipulators, but they seem commonplace only in manufacturing. Besides the possible prohibitive cost of owning a service robot[4], household chores are complex problems that require constant attention from a skilled roboticist. Deceptively simple manipulator tasks, such as taking apart an Oreo cookie[5], can require many iterations of planning and simulation before fruition. Industry manipulators usually operate under a controlled and fully-known environ-

[1] http://www.robotshop.com/en/personal-domestic-robots.html

[2] https://www.willowgarage.com/pages/pr2/overview

[3] www.cmu.edu/herb-robot

[4] In this work, ‘service robot’ shall be interchangeable with ‘service manipulators’ or ‘mobile manipulators’.

[5] http://www.cmu.edu/homepage/computing/2013/winter/cookie-vs-creme-a-robotic-twist.shtml

ment under expert guidance; whereas household robots, on the other hand, are tasked by normal users using 'the same plan that may not work under different circumstances'. As the Oreo cookie demonstration shows, home users *implicitly* know what *actions* robot manipulators should do in a task, and more importantly, what is the final *goal* (that accomplishes the task) by relating to their own experience of doing that same task. Positing the hypothesis that a user's superior cognition can enhance service robot productivity, the main challenges are to (1) Identify user interfaces that allow users to transfer their task experience, and (2) Design systems that assist robots to *negotiate* task-spaces by leveraging human observations.

Perceptual Modeling For Autonomous Operation

Robots that operate autonomously need constant adaptation to their changing environment in order to perform their actions reliably. Meeting this requirement requires constant learning of geometric and semantic models which represent their perception of the operating space [KMM12]. Building and updating the perceptual models require geometric data sensing of the physical space, followed by recognition of shape and semantics of the artifacts inferred in that space.

Geometric Mapping

Synthetic mapping is an important resource for planning and decision making in robots. The progress of physical geometry acquisition with laser range-finders or time-of-flight techniques such as LIDAR (Light Detection And Ranging), concomitant with Robot Localization (SLAM) and 3-D surface reconstruction has allowed mobile manipulators to navigate *collision-free* around their operating environment [BD06]. Geometric mapping models are also very useful for robot teleoperation in shared control scenarios. By modeling user teleoperative behaviors with context to their detected geometries, machine intelligence can be trained to analyze and augment manipulator trajectories helpful to the controlling user even as they maintain control of speed and direction using avateering [KPL14], [DS13]. Perceiving 3-D models of geometries and navigational maps, however, are insufficient conditions for autonomous agents to execute complex household tasks [PSB11].

Semantic Mapping

Robots also need to infer meaningful relationships between objects and environment spaces in order to *plan* the required manipulative *actions*. Geometric 3-D models would require useful semantic labels such that they can be expressed through a language understood by both humans and robots about their manipulative transforms in a task [SYC14]. Semantic information can additionally provide manipulation rules and *constraints* that apply to the recognized artifacts. For example, an agent would infer it should not attempt to grasp a glass object, or whether handle geometries should be located for grasping, or to desist from stacking any more books into a fully-occupied shelf, or not to serve coffee with a bowl even though it is physically permissible. Modern information technologies could be leveraged to enhance robot awareness and understanding about their task space; the practical improvements will be significant if they can assist more effective decision making in automatic task and grasp planning. As state-of-the-art object recognition and semantic frameworks are not yet viable for production-oriented application [MCS10], [HCB13], [PPT12], optimal decision making will require a combination of both human and machine reasoning [PSB11]. Effective human assistance can give autonomous agents competitive advantages by overcoming their limitations in *both* spatial and semantic *cognition*.

Sketch-based Abstractions For *Robot-centric* Manipulator Control

Mobile Natural User Interfaces (NUI) with sketch-based modalities can allow users to explore ideas visually and *anywhere*; with inclusion of other modalities such as touch, voice, and arm and hand gestures, messages can be communicated intuitively to intelligent agents that analyze the abstract input's perceptual features and taking appropriate actions through a service robot. Although NUIs' as communication channels have been actively studied in HRI, current literature are essentially user-centric, or *human-centric*, collaborative models; delegating 'low-level' processes to autonomy that are *agnostic* to the task *goal*, while human users control high-level supervisory functions that *plan* and *monitor* the required actions performed by autonomy [CBH11]. Multi-modal systems for Human-Robot Interaction (HRI) [CWF10], [TFC12] follow three requirements for supervisory control [She92]; Autonomy, High-level commands, and Situational Awareness. A typical interaction scenario would have the human user maintain situational and decision control of the task by providing a plan, or course of action, whose workload is partitioned out to 'low-level' autonomies, or robot(s), *trusted* [CBH11] to complete the actions; when any robot reports an exception, the user either teleoperates or re-assign alternative actions to recover task progress, rather than the autonomy (or robot) that performs the action. Instead, human knowledge could be applied through *robot-centric* collaborative models with sketch-based modalities to assist 'low-level' autonomies [PSW00] at (1) Information Acquisition and (2) Analysis, (3) Decision Selection, and (4) Task Implementation. It has been shown that humans can supplement, or partially replace, robots in object selection (category: Information Analysis) when the state-of-the-art is inadequate [PSB11]. This dissertation work therefore differs from supervisory systems; besides presenting intuitive channels that relay high-level commands, the work explores abstractions that support *robot-centric* interactions cooperating with robot autonomies in task, grasp, and motion trajectory behaviors using Parasuraman's [PSW00] four categories of autonomy. We introduce example scenarios under each category in the following sections where user intervention could be helpful in assisting a normally autonomous process;

Assisting Information Acquisition

Complete information about the robot and environment states is crucial for motion and grasp planning, but sensors typically acquire incomplete information about the task surroundings. Geometrical mapping with laser scans contain hidden or mismatched geometries with the ground-truth due to noisy sensors, self-occlusions, or less than ideal conditions of surface materials [BJL11]. Gaps appear inside spatial mesh reconstructions due to under-sampling, inaccurate, or missing point clusters (Figure 1.1). Humans are excellent at predicting unobserved or undetectable scene structures using controlled scene continuation based on users' prior visual experience and presented evidence [BF05], [BJL11]. A simple, intuitive strategy is for users to assist in acquiring geometric information based on their own visual observations of the same objects or space. For example, a robot could present a user its perceived spatial model of the object they would be grasping for cooperative modeling (e.g. a 3-D point cloud or mesh); and rather than correcting, or predicating, the shape or size of the presented model, a user instead provides geometric *cues* an intelligent agent can validate and extrapolate further spatial information ordinarily not detected by hardware sensors. Referring back to Figure 1.1, gaps in the point cloud could reasonably be restored by cloning existing geometries and relocating the copies into the gaps. One example geometric cue is an axial plane of symmetry, which can be extremely memory and time-consuming to model through automatic means [BJL11], [QMG15]; users can present this cue very quickly to a simple algorithm that fill the gaps by extrapolating a circle of points around the presented axis (Chapter 5). With a simple user cue or *hypothesis*, a robot can enhance the quality of its grasps with a small time cost but no loss to its grasp-planning autonomy, as compared against doing the same on its own that may not reach either a feasible solution or terminal condition.

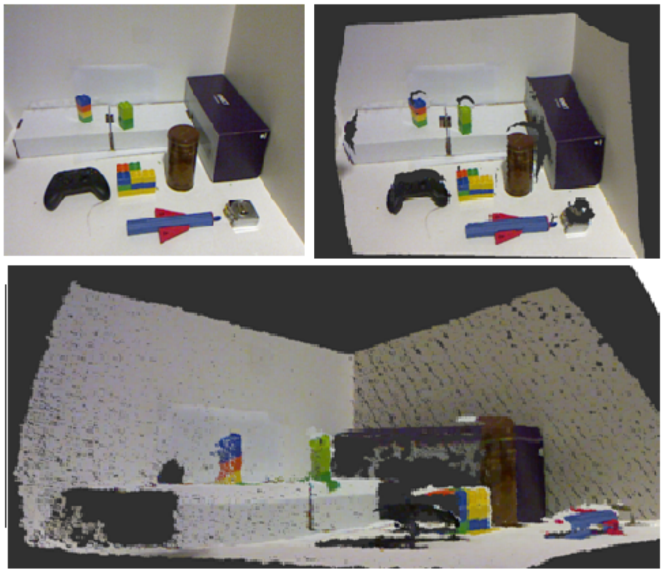


Figure 1.1: (Clockwise from Top-Left) Matching RGB image; point cloud from a typical viewpoint looking into an experimental desk set-up; side view of the cylinder and cube surface scans.

Assisting Information Analysis

Unprocessed data in robotics literature are provided as raw sensor readings, such as point clouds from LIDAR scans, or kinematic data from accelerometers. For point clouds, they are further organized into meshes, analogous to measuring 3-D position, orientation, and velocity with any inertial-based systems such as accelerometers. Given a set of meshes, robots need to perform object *segmentation* within the scanned geometries in order to infer the targets localized by the task instruction[6]. Reliable recognition and selection of objects, especially in cluttered scenes, is a hard task that largely remains unsolved in full autonomy [PSB11]. Groups of objects can appear as a single continuous mesh, especially in cluttered scenes with objects of similar textures or texture-less objects [HBP13]. An intuitive strategy is for users to assist through sketched annotations. Figure 1.2 illustrates an example of how it could be performed; sketched partitions, lassos, and labels can help to outline, disambiguate, or localize the target object's spatial boundaries and geometries inside the scene (e.g. Block, Shelf), or alternatively, provide training data for machine-learned models. Secondly, a user could also interact locally within the manipulators' task space with pointing gestures and voice commands to annotate the object, [GSR14] (e.g. user provides a label by pointing towards an object with arm gestures followed by a voice call-out of 'Vase'), while wearing any Mixed Reality Head Mounted Display (HMD), such as the HTC Vive[7], Hololens[8], or Google Cardboard[9]; this alternative interaction will be left as future project work for interested readers. Localized vertices by user annotations can also back-project to pixels in the 2-D image, isolating the sub-window at image plane coordinates and their descriptors which can be used for alternative recognition with 2-D features instead, such as SIFT and SURF, when there is absence of sufficient or accurate features in 3-D [TS10], [MCS10]. Labels also

[6] For example, an instruction of "Placing This Book on That Shelf", requires locating targets 'Book' and 'Shelf' in the 3-D world before any manipulation can take place.

[7] http://www.htcvive.com/us

[8] http://www.microsoft.com/microsoft-hololens/en-us

[9] http://vr.google.com/cardboard

function as semantic descriptors to predicates in the expression of *quadruples* inside classical planning [SYC14].

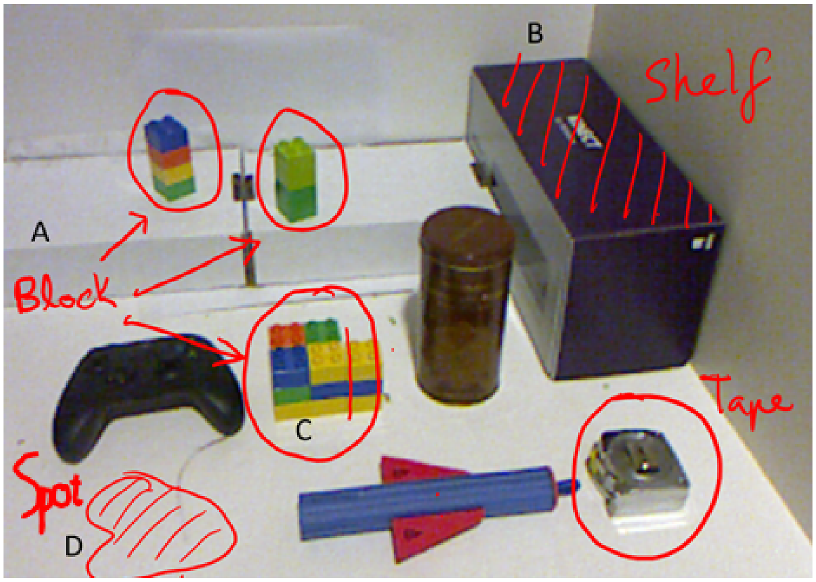


Figure 1.2: (A) 'Block' labels three objects (B) 'Shelf' labels on top of the card box (C) Line stroke that represents a plane-cut separation of smaller block with the larger block (D) Lasso and line strokes that represent arbitrary location 'Spot'

Decision-Making Through Shared Perceptual Spaces

A task action by a robot has to be consistent with the decision it makes before committing that action [PSW00]. Humans collaborating inside robot-centric team tasks are inadvertent co-participants of the robot's perceptual and action choices. For example, robots often have more than one choice or solution, sometimes having no solutions at all, to motion and grasps it could use for an action. Users can bias grasp planning by recommending spatial or mesh regions of interest (ROI) around the targeted object; ROI could represent one of the following: (1) where manipulators should reference for grasp points in tool center alignment (TCP) [BJL11], (2) as approach vectors using surface-point normals and (3) as alternative landing points when placing the object. ROI themselves could also be labeled with semantics (e.g. ‘Hard’ or ‘Soft’) that suggest a recommended grasping force. For path-planning, users can either sketch virtual obstacles into a scene that limit motion trajectory choices of robot manipulators [GSR14], or indicate passable way-points in the absence of a motion plan; a situation which is a common occurring motion-planning challenge inside confined workspaces [SES14a].

Modeling Tasks & Voice Commands

High-level tasks can be represented, or modeled, with goal-states achieved as a result of executing ‘low-level’ actuator actions from the current-state, known a-priori by default [SYC14]. User annotations can provide symbols, predicates, and effects (or goal-states) to a task executive that assemble the high-level task undertaken by the service robot. For example, a user could model a task with the single effect of placing a Lego block on a shelf (‘Place Block on Shelf’) as a *goal predicate* by linking between an annotated Lego block and specific shelf (Figures 1.2 & 1.3); the annotations help planning agents *semantically* ground the block and shelf as a new *object* and *location* respectively using spatial descriptors isolated by the annotative strokes (Figure 1.2). Multiple goal predicates can be further *coalesced* together

(Figure 1.3) into a compound task[10]; modeling the task as an *unordered list* of goals that abstracts away user consideration of the actions and iterations needed to an autonomous pipeline (Chapters 3, 4). Using Figure 1.3 as an illustrative example, the user is not required to consider whether a clean shelf should be achieved ahead of the blocks being relocated to the shelf. Additionally, Figure 1.3 further illustrates how a user can tag a descriptive verb, which also imply the grammar for a voice command, to a compound task of three sub-goals.

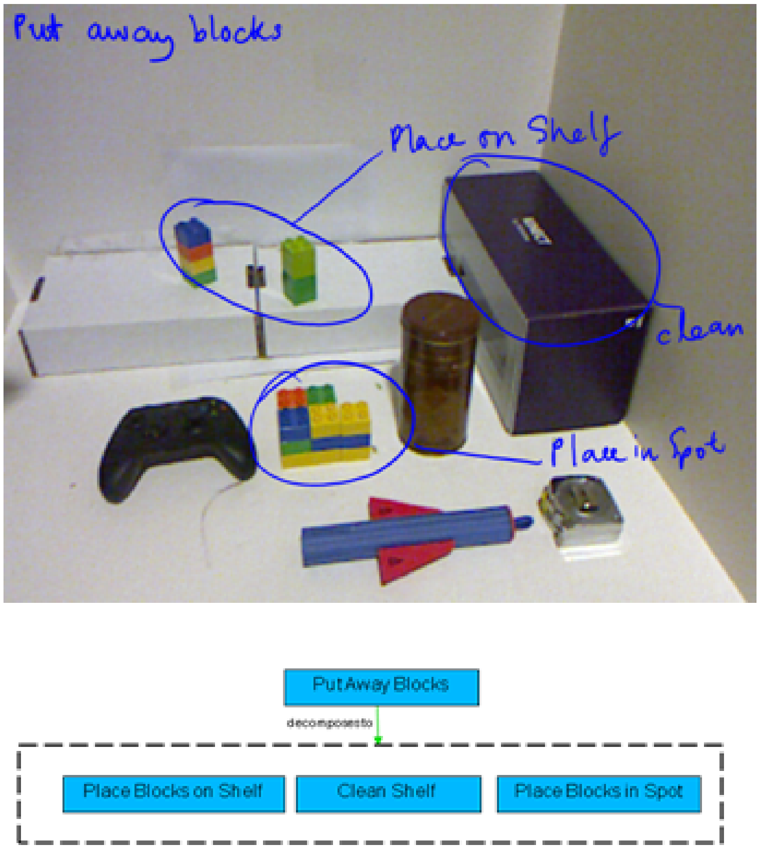


Figure 1.3: A compound task with the description ‘Put Away Blocks’ comprising of three sub-tasks (1) Place two lego blocks on the shelf (2 Place a large lego block at a location labeled ‘Spot’. (3) Wipe the shelf. Task descriptions can be used as voice commands.

[10] A task with multiple sub-goals, or effects.

Service Robot Utility

Service robots can become commonplace at home when their *utility* is maximized through technological *accessibility*. A wider variety of manipulation tasks becomes available when non-technical users are granted effective access to a service robot's controller architectures governing autonomous task, grasp, and motion planning (Chapters 3 to 5). Robots can become more efficient when their task autonomy can be recovered at home or *away* without expert robotics knowledge. These however, remain significant challenges given that in many state-of-the-art WIMP-style tools, a superficial understanding in robotics, 3D graphics, and computer science is still needed for many task modeling and recovery activities.

Thesis Statement

With stylus and touch modalities, a user can convey natural abstractions of high-level tasks, semantic revisions, and environment geometries to a service robot without breaking autonomy. When the service robot is unable to accomplish a task through autonomous means, teleoperative recovery is possible with natural modalities of body gestures and speech.

Research Objectives

This work introduces a mobile and teleoperative interface schematic to allow cooperative modeling and recovery of a high-level task performed by a service robot under a shared perceptual autonomy. The main hypothesis posited is that free-form modalities of sketch, touch, and body gestures not only provide an intuitive channel for new instruction, it presents a mobile, natural, and powerful device for collaborating effective task behavior and recovery assistance amenable for non-expert users. This is an important design requirement; even though users are not expected to be technically qualified, they can maximize utility by cooperating anywhere with a service robot's autonomous planning *without* unnecessary intervention and severe neglect. Shraft [SM06] cites three requirements for robotics to be practical in small-scale production for small and medium-sized enterprises (SMEs), (1) new users who can program the robot in a day, (2) easy re-programming under slightly varied task environments, and (3) the time required to program new tasks should be significantly reduced compared with the default. Though these requirements were addressed for industry, the context is highly relevant; in order for domestic robotics to be practical, the modeling *and* recovery of complex manipulation tasks would need to be made accessible for non-technical users.

A first question we would need to ask is how a user could interact with a service manipulator as a robotics novice; and perhaps, the interaction could also be used outside the home maximizing mobility and usage? We shall examine state-of-the-art *pipeline* architectures in literature that implement task autonomies, and how stylus with touch modalities and gesture recognition available in commodity mobile devices and wearables[11] can be accommodated to provide rapid modeling and recovery assistance in these autonomies. Entailing questions we ask are *where* and *when* in a general pipeline architecture could the non-expert user provide cooperative and teleoperative inputs that are both effective and easily understood.

The second question we would explore is whether implementing *robot-centric* interactions for our user to the service robot's task, grasp, and motion planning pipeline could provide both productive and qualitative benefits in comfort, ease of use, and learnability with our non-technical participants. We shall examine how a manipulation task is programmed or recovered in literature, and how explicit programming can be *minimized* by abstracting the user away from the actions that perform the task. We shall further examine manipulator grasp and motion trajectory planning to evaluate our alternative interactions of recovering task, grasp, and trajectory autonomies that minimize the need for teleoperative recovery. Furthermore, we shall examine how teleoperation can be immersive and intuitive when needed as a last resort. If the cooperative and teleoperative interactions we provide demonstrate significant user performance benefits, we can leverage the prevalence of mobile devices to provide a portable and intuitive interface that non-experts are amenable with to control and recover service robots.

[11] Our ideal interface is a constitution of any mobile device with optional stylus support, and a HMD leveraging on the mobile device hardware, such as Google Cardboard.

Approach

To address these questions, we devised a service framework (Chapter 3) under the RAP system (Robots, Agents, and People) [SJP02] for an empirical platform to evaluate our proposed interactions through a prototype (Figure 3.3). The framework encapsulates our model *architecture* supporting a service robot's manipulation planning from the high-level task to the actuator primitives of grasping and trajectories needed for each action. With reference to the framework, we conducted four studies, each proposing a *human-in-the-loop* collaboration within the model architecture, to evaluate performance and perception from a novice user perspective.

In our first user study (Chapter 4), we explore and evaluate abstract sketches for symbol grounding and semantic revisions in both modeling and recovery of *high-level* manipulator tasks. Though our framework is generic enough to be used under any manipulation planning domain, we limit our discussion to 'pick and place' tasks in a 'Blocks World' scenario [GN92], [NGT04]. The assumption facilitates simple color-based feature recognition (for the blocks) and removes the complex requirement for low-level grasp planning. The main purpose was to evaluate cooperative modeling and recovery with automated task planning in an unstructured and partially observable domain through stylus-based modalities available on commodity mobile devices. We compared the interaction against desktop-based modeling with a simple visual *programming* language that require *full* knowledge about the task domain. Our prototype that supported the interaction was well-received by a sample of non-expert participants, significantly outperforming the control in quantitative measures of task completion time and mean error. We observed also that recovering the task with symbol grounding will be insufficient to ensure a successful manipulation.

For the second study (Chapter 5), we focus on how users could address the grasp-planning challenge of *partially cognizant* 3-D geometries after either an interactive grounding (from the first study) or automated recognition. We evaluate the use of symmetry as a global descriptor, or cue, for a user to express an estimated *hypothesis* about the actual 3-D shape. Using simulated single-view depth measurements of common household object meshes taken inside a virtual environment, we assume the user is providing grasp recovery by expressing a hypothesis that allow extrapolation of further spatial primitives through symmetry completion rules. We compared similarity of the extrapolated point clouds to the ground-truth model, receiving a mean normalized Hausdorff distance of 4.16% across seven tasks from a sample of twelve participants with little experience in 3-D modeling and computer graphics. Participants appreciated the ease and utility of symmetry descriptors that could express *global* spatial aspects of the object shape.

In our third study, we moved our focus to interactions that participate in autonomous trajectory planning. In any 'pick-and-place' manipulation, it is quite common for a motion planner to return no feasible trajectories to the end-effector poses either in a 'pick' (pre-grasp) or 'place' action (Chapter 5, Section *'Theoretical Framework: Collaborating with C_{space} Exploration in Motion Planning'*). Reusing the empirical set-up of the first study, we evaluate the interaction of artifact point-selection; each labeled artifact servicing as a trajectory *way-*

point, or region, that stimulate further configuration free space C_{free} samples. It is posited that teleoperative recovery either with interactive markers (Chapter 5, Figure 5.4) or avateering (Chapter 6) can be minimized by improving the odds of finding a feasible path through a user's cooperative influence in a motion plan. However, full teleoperative recovery is still unavoidable; there will be times when a user must take full control of the manipulator due to safety, robot malfunction, or task expedition.

In the fourth study, we examine the use of skeletal-joint tracking in teleoperating the grasp by leveraging on human-robot anthropomorphic similarities, termed avateering. Avateering offers a more natural, free, and intuitive approach by teleoperating through egocentric immersion and proprioception [KPL14]. However, free avateering is often inadequate in precision grasps due to latency, viewpoint, human error, and lack of informative tactile feedback. An intelligent agent can enhance the user's teleoperative input through a predictive-arbitration loop according to the observed trajectories around scene geometries. We found the avateering metaphor greatly benefited from an assistive agent providing non-obtrusive arbitration *without* disrupting the teleoperative experience and expectation of the user.

The main contributions include the following:

1: Natural User Interfaces of stylus, touch and body gesture modalities applied to Sheridan's Four-Stage model of human information processing to assist autonomous planning. The system comprises of: (1) cooperative stylus and touch inputs on 2-D video and 3-D models in the task environment that augment a service robot's geometric and semantic understanding for effective grasp and motion recovery, (2) annotations that transfer user expectations into abstract models of new tasks, and (3) gestures & voice input for remote teleoperative recovery. Users can transfer new knowledge, communicate new tasks, and become collaborative partners in helping robots gain better scene understanding for optimal planning.

2: A study of the effects of the proposed interface on users' qualitative and quantitative performance when collaborating with automatic methods in high-level task planning and recovery (e.g. sequences of pick-and-place tasks in a workspace with unknown artifacts). Whenever the robot encounters physical and semantic obstacles that stall a task, the current plan is usually invalidated in order to initiate a new plan accommodating the environment changes; or the user acquires full teleoperative control. The user studies provide valuable insight on the proposed system, evaluating cooperative planning models that promote plan re-use or minimal modification such that tasks under unstructured and partially observable environments are able to complete with less attention.

Reader's Guide

Chapter 2 provides a review of the literature in shared autonomy and sketch-based NUIs for Human-Robot Interaction (HRI) and Human-Robot Team (HRT) collaboration. In Chapter 3, we propose the system architecture and user interface specifications for robot-centric collaboration. Chapter 4 provides a discussion of our theoretical framework behind cooperative modeling and recovery. The chapter also details a user study of our prototype which is an implementation of our theoretical framework for picking and placing (‘pick and place’) tasks in a ‘Blocks World’ planning domain. Chapter 5 explores the design and evaluation of two further cooperative abstractions that allow users to provide recovery for grasp and motion planning. Chapter 6 explores the requirements and design of autonomous agents for assisted teleoperation, followed by a user study evaluating the agents’ effect on free avateering. Finally, we discuss our findings, opportunities for future work, and concluding remarks in Chapter 7.

CHAPTER 2
GUIDING ROBOTS IN SHARED PERCEPTUAL TASKS

Perceptual Shared Control in Human-Robot Teams (HRT)

Studies of team dynamics between Human and Robot Agents collaborating as teams have been growing steadily recently, with psychologists and roboticists in much agreement about the productive benefits mixed collaborative teams have over human-only or only fully autonomous teams. Humans and robots can help each other by leveraging upon and compensating for each other's cognitive or compute strengths and weaknesses. The polar-opposite characteristics between human (cognition) and robot (compute) have implicitly defined or constrained tasks that can be shared or done individually; tasks are assigned whether they could be better performed either by higher cognition or compute ability. Concomitantly, research about human-robot team design has also become an interesting topic due to the cognitive-compute disparities between humans and robots [She92], [PSW00].

HRI research spans across multiple fields, such as adjustable autonomy [SPT01], mixed initiative control [BFB05], and cognitive workload modeling [FY07] that study task collaboration between machine and human agents ranging from full autonomy to human teleoperation [DS13]. The works report findings where human users apply their cognition advantage enabling autonomous processes to perform, such as abstract perception and reasoning [PSB11], modeling collaborative strategies that prefer human-centricity. Humans agents are imposed supervisory roles which grant high-level task assignment and control, monitoring and assisting with sub-task executions delegated to reactive and compute-intensive agents. However, current prior art machine learning have enabled intelligent agents to model towards human-like analysis and decision-making, performing admirably in tasks requiring human-scale cognition and manipulation [KMD10]. It is still an open problem that autonomous processes have cognitive limitations that require human agents for essential intervention (or the task remains incomplete) and information (or task state either remain unknown or incorrectly perceived), though the ideal collaboration scenario is to model HRTs' towards robot-centricity [Fon01].

Robot-centric teams model robots to behave more like fellow human peers in the team, requiring little or no human assistance with their assigned high-level tasks or to be instructed for new task instances during collaboration. Additionally, their task behavior can be augmented by users during collaboration in order to improve overall team performance. There is surprisingly little research in prior robotics literature that study and develop systems for robot-centric models. Perhaps mainly due to accepted paradigms of cognitive-compute differences between human and machine agents, nearly all well-studied team scenarios have been the opposite. Robots in human-centric teams complement human tasks either by observing and reacting to their actions, or by predicting their behavior temporally and react according to the prediction [IPK03], [SWW11], [LTS13]. Another common human-centric collaborative theme is based on human-agents acting as leaders; directing and supervising robots at sub-task level without requirement for any artificial agent to be aware of the overall team goal. When reporting progress, robots communicate back their sub-task status in set-timed intervals, or when encountering difficulty that may require the leader to teleoperate or re-assign [CBH11], [PSA01]. There is also very little reported work regarding human agents aiding and co-participating unobtrusively with autonomous robots at their sub-tasks. Human co-participation, ideally, should be kept minimal and non-explicit; a user should assist with implicit guidance, rather than manually taking over all autonomous function, by leveraging on the robot's perceptual intelligence and compute advantages. The robot should also interrupt the user for assistance only when it is really essential in order to divert valuable human attention elsewhere.

Regardless the model, HRT research centers around the accepted idea that control and execution can be shared and exchanged between humans and robots for the same task, since both taxonomies are observing the same workspace, but perceiving and understanding that space differently. The levels and types of perceptual collaboration in robot-centric schemes can be modeled after Parasuraman's four categories of autonomy, occurring during a robot's (1) Information Acquisition, (2) Information Analysis, (3) Decision and (4) Action Selection [PSW00].

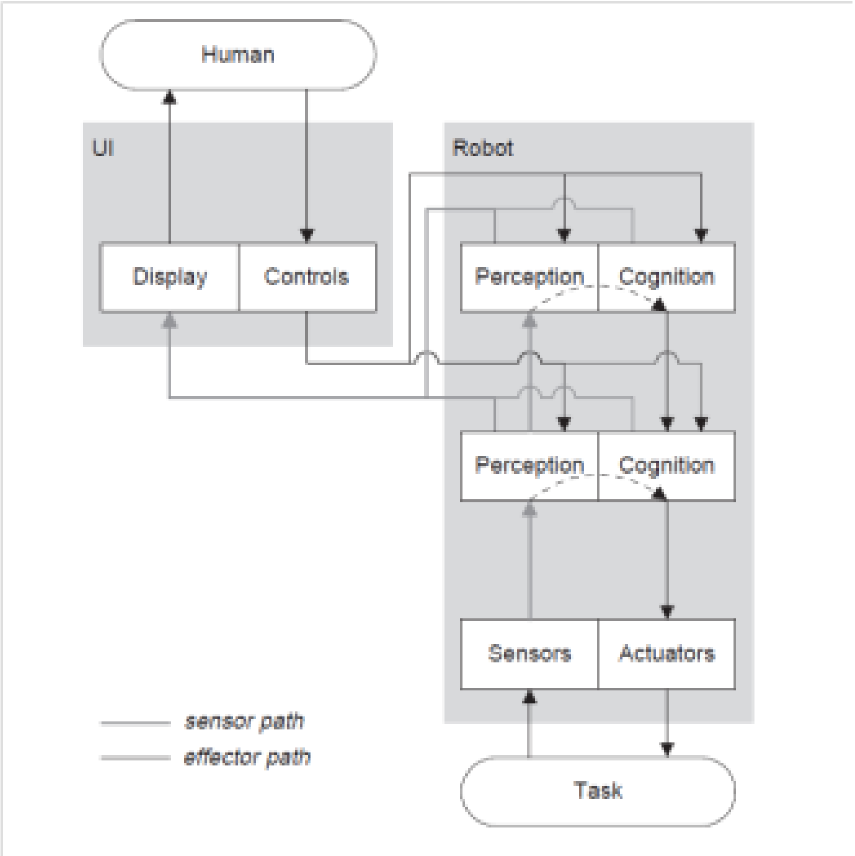


Figure 2.1: Collaborative control system model (Fong, 2001)

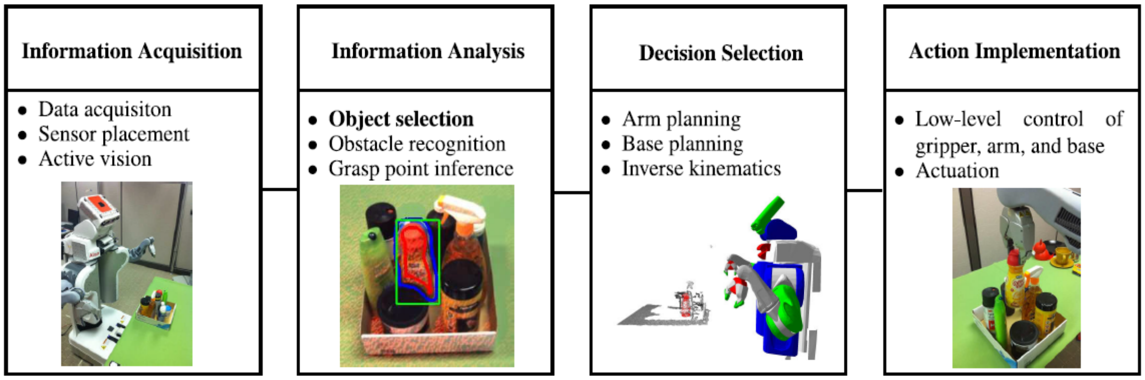


Figure 2.2: Retrieving an object based on Sheridan's 4-stage model of human information processing. Assisted selection is highlighted where human input intervenes (Pitzer, Styer, Bersch, DuHadway & Becker, 2011)

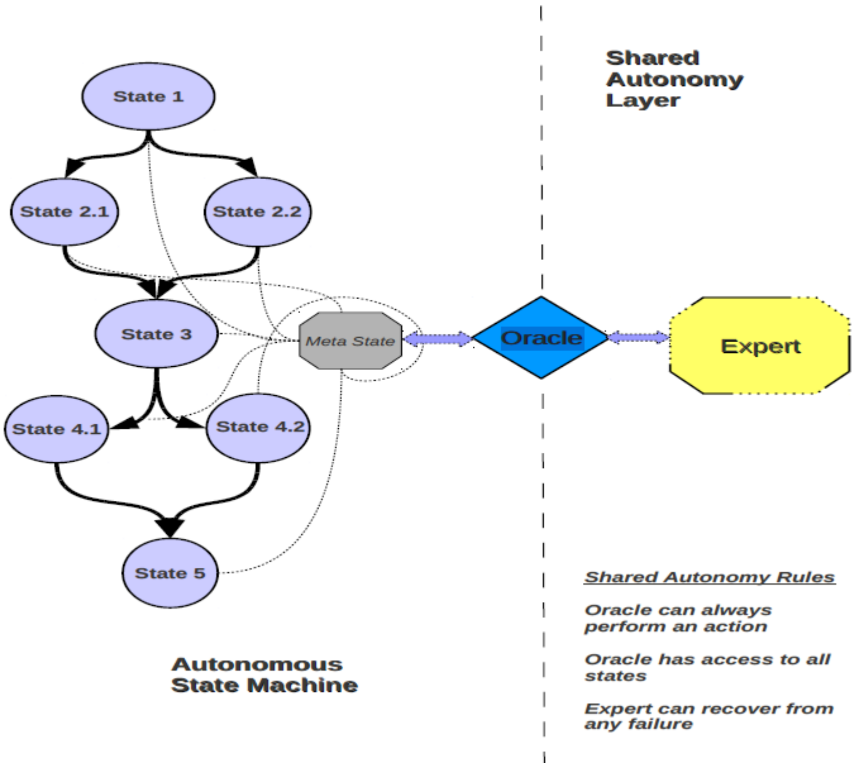


Figure 2.3: Oracle (Sankaran, Pitzer, & Osentoski, 2012)

Information Acquisition And Analysis Under Incomplete Observations

With adequate sensing frameworks [MCS10], [PTP12], robots are highly efficient and more than capable, in comparison with their human counterparts, at acquiring raw data and information required to accomplish their assigned tasks. Well-known examples range from tangibles such as robot pose from odometry, point clusters [BJL11], [NGP13], [QMG15], and 2D imagery (pixel features) for SLAM applications [BD06], [HM11], to custom intangibles such as gestures and human intent [IPK03] as quantitative signals via natural input modalities [PKL13], [KPL14], [SSC10]. Sensing and generating information understood by users for robot interaction is viable due to the presence of measurable and consistent empirical training data in noisy environments [DSK12]. Autonomous acquisition of structured, tangible state data and information can be considered as a solved problem; and should the need arise with seemingly intelligible data, machine learning techniques and heuristics could make up for the technical limitations of sensor hardware. Clear examples can be seen in state-of-the-art SLAM applications, where statistical techniques such as Kalman (EKF-SLAM) and Particle filters (Monte-Carlo localization), Bayesian Graphs, and Feature-based Matching are applied to register observed sensor scans that optimize belief of robot-pose and landmarks of explored space [FWR10], [HM11]. Bohg [BJL11], Kroemer [KAE12], Navarro [NGP13], and Quispe [QMG15] studied methods for generating 3-D models of household items automatically that can be under incomplete observation. Autonomous robot teams, however, are still limited by sensor noise and mobility constraints; they are never exempt from having sufficient and reliable information about their environment that allow good decision-making about their task actions [Fon01]. Users could assist autonomies here, by inferring together with agents useful spatial information that rectify model predictions. In describing collaborative teleoperation control, Fong et al. [Fon01] addressed the main limitations of supervisory or fully autonomous robot teleoperation, where the only choices a robot has when encountering task difficulty are "to continue performing poorly or to stop". Collaborative or shared control, however, allows the robot to compensate its own inadequacy through human cooperation; requesting supplemental information from a co-participating human that *augments* its task autonomy. Fong illustrated two loop closures, perception and cognition (Figure 2.1), where human co-participants assist a robot in resolving its perceptual and cognitive difficulties. Illustrating with a teleoperative demonstration, a human controller helps to validate obstacles deemed impassable by the robot to be passable, allowing it to plan better routes to its assigned destinations.

Besides physical information such as 3D spatial geometries, autonomous gathering and analysis of semantic information in a robot's workspace have been gaining attention in recent literature [PTP12], [RMB09], [SBB12], [ZC14], enabling machine intelligence to inquire labels and relationships between common household artifacts that predict tractable actions. By constraining to indoors environments (e.g. kitchen), Rusu, Pangeric et al. studied autonomous object pre-categorizations of structures (e.g. walls, floors, ceiling), box-like containers (e.g. drawers, cabinets, cupboards) and planar areas (e.g. tables, counters, shelves) using machine learning. Further child-categorizations are parameterized with features; for instance, objects with a unique number of handles for grasping allow differentiation, such as between cupboards (with two handles) and drawers (with a single handle). Point cloud scans are post-processed with segmentation and labels, generating modifiable Semantic Object Maps (SOM+) that cache physical and logical relationships between objects and structures, utility (e.g. oven for heating food), and articulation models which can be learned on-line while a robot manipulates the object by their handles. Dynamic or changing environments, however, still present an enormous hurdle to the storage and generation of useful semantic information, requiring inexhaustible lists of heuristics, or learning statistical models a-prior at multiple scales that need huge ground databases updated constantly and training time to enable effective object classification [PTP12], [RMB09]. Instead, human agents can provide such information to robots along task planning and execution according to task-specific identifiers and context. Besides knowledge of workspace semantics, robots need continuous localization and differentiation about novel objects, structures, and locations stipulated by the task instructions (e.g.“Place This Book on That Shelf”). Reliable object classification and recognition, especially inside cluttered scenes, is a hard problem that largely remains unsolved under full autonomy. At Pitzer et al. [PSB11] collaborative task of selection, the robot relied on human assistance for object recognition when it has ‘exhausted’ all possible avenues. Pitzer demonstrated that robots in a shared autonomy (Figure 2.2), as compared against state-of-the-art recognition under full autonomy, performed more robustly with user help when isolating an object to grasp. Users become helpful in a graph-cuts image segmentation algorithm, enclosing an image sub-window where the target object resides and marking the object's foreground and background areas. Similarly, free-form sketches on 2-D images can be meaningful cues at both spatial and semantic levels for object or structural recognition, providing users an intuitive modality to assist robots as a secondary resource. One objective for the work in this dissertation, is to explore how abstract sketches could be effective spatial and semantic cues that augment robotic perceptual models during task and grasp planning.

Plan & Action Selection Under Incomplete Observations

Any action or task undertaken by autonomous agents need to be consistent with the choice of plan [PSW00]. Planning is a well-studied problem in Artificial Intelligence (AI) and Machine Learning (ML); they are commonly associated with graph-search algorithms of discrete maps demarcating state or plan spaces, such as best-first (e.g. A*) or first-solution (e.g. greedy depth-first) searches that compute a heuristic cost requiring complete 'off-line' a-priori information [KF11]. Sampling-Based Planners (SBP), such as Probabilistic RoadMaps (PRMs) and Rapidly-exploring Random Trees (RRTs) on other hand, expand and process graphs on-the-fly through samples in robot configuration space (C_{space}), providing substantial computation and memory savings even though the solution may more often be sub-optimal [ES14]. In autonomous tasking for robot manipulators, the common scenarios are planning low-level primitives, requiring specialized motion planners for trajectories and grasps that employ SBPs' with graph-search [VAD12], while configurable classical-based planners such as SHOP2, UCPOP, Fast-Forward, and GraphHTN are more practical for planning in abstract spaces that process STRIPS-like instructions representing actions in environment space [NGT04]. Under 'real-world' planning circumstances however, robots compute multiple plans as contingencies for partial observations; or sometimes, no plan or action can be feasible when prescribed with constraints or time-schedules that become Constraint Satisfaction Problems (CSPs). In the former case, robots could explore and mine the task-space further to reinforce useful information or alleviate a 'no-plan' occurrence, a phase that can be extremely time consuming and computationally expensive [KLM96]. Human-in-the-loop interaction or intervention, however, have shown to help robots expedite plan exploration via natural instruction or teach tokens that prune or filter the search space. Using Programming by Demonstration (PbD), or Learning from Demonstration (LfD) [CDS10], recent examples can be found at Seidel and Morante [SES14a], [MVJ14] where a robot manipulator generates novel and valid actions using taught motion segments of prior human-guided control. Conversely, other examples can be found where human agents specify spatial and motion constraints that imply forbidden actions that agents would need to plan around [FON12], [GSR14].

There is little reported literature, however, that report qualitative findings of cooperative interaction with a non-prehensile grasp planner; the literature in general reports the study of automatic methods that provide (i) object shape recognition, and (ii) grasp points or posture modeling [MKC03], [BK10], [BJL11], before applying self-correcting pressure on the grasp. Miller et al. presents a system where users first select a starting grasp position and approach for a particular arm, followed by the pre-grasp pose from a ranked list of possible poses. Bohg et al. defined a grasp-planning pipeline that progresses as follows: (i) Generation of grasp points on the object that align the hand, or tool, center point (TCP), (ii) Computing an approach vector to the grasp points parameterized as a 3D angle parameter, and (iii) Determining the wrist orientation for the pre-grasp. A user could potentially assist grasp planning agents unobtrusively by intervening along the pipeline; for instance, they could recommend mesh regions of interest around and on the object where manipulators should reference for grasp points during tool center alignment (TCP), approach vectors for moving to the pre-grasp pose, or to assist the agent in forming and correcting object shape hypotheses.

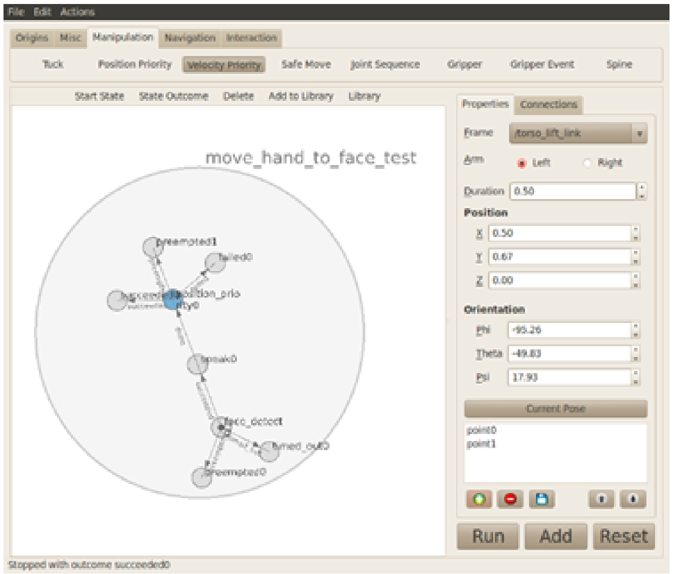
The ideal collaborative scenario, however, is to necessitate human intervention only upon request by any autonomous agent so that human supervision and attention is minimized. In Sankaran et al., a shared autonomy was developed that allows a user to intervene only after the agent, or robot, has exhausted all attempts of recovery on its own [SPO12]. The collaboration requires an 'oracle' agent to mediate between a human 'expert' and the robot's task executive, selectively querying for user intervention only after progressing through a flow-chart of recovery measures that fail to progress the task. The oracle constantly refers to a state-machine that represents the high-level task plan for a status update as the robot progresses through each action [BRJ11]. Alternatively, the expert can opt to supersede the original plan with teleoperation only when an unrecoverable failure has occurred (Figure 2.3). Sankaran has shown that in a shared autonomy, human co-participation as a secondary source of semantic information is essential for a successful plan execution, and teleoperating through an unrecoverable task failure is not an uncommon occurrence given the complexities of task planning in a dynamic and unstructured environment.

Task Transfer & Learning

A robot must be able to perform every foreseeable iteration of actions required by a task beforehand. These iterations are acquired either through autonomous planning (Section *'Plan & Action Selection Under Incomplete Observations'*), or by programming executives. Every task can be modeled as a sequence of actions or a state machine [SPO12], and many atomic tasks that represent a high-level task can be modeled as a Hierarchal Task Network [EKR13], [NGT04]. Using human dialogue, She et al. [SYC14] demonstrated how robots can learn new high-level actions or tasks from known actions. The study defined a set of manipulator-only atomic actions, such as opening and closing of the gripper, and 8 variants of manipulator moving actions described in generic terms of constraints under which the manipulator can move around the workspace. Any action can be represented as a STRIPS quadruple (P,O,I,G), and during teaching a user instructs the robot to execute each known action in a sequence as it senses the initial and goal states of that action, and finally submitting the new quadruple into a knowledge database. With Knowledge Representation such as STRIPS, automated planners such as SHOP2 can compose state-machines (or task executives) that dictate a robot's actions based on its observations to changes in the task space.

On other hand, users can express the state-machines visually, which can be convenient with modelers such as Robot Operating System's (ROS) RCommander with the underlying ROS SMACH (State MACHine) engine [NCH13] (Figure 2.4). Chen et al. [CCC13], for instance, built an interactive user interface (Figure 2.5) on top of tasks built with RCommander that allow users with disabilities to control a PR2 robot as their surrogate for object manipulation and social interaction. These techniques, however, still require explicit expression of the task semantics from users; With automated planners, users are required to represent the knowledge as quadruples before transfer to a robot. Modelers such as RCommander mitigate the requirement by letting users visually graph out the state-machine. However, the tool assumes a full plan is known prior, requiring users to express the state-machine fully that includes recovering from failure. Work from She et al. is interesting given that it can allow users to impart new tasks without requiring direct and explicit expression on the task executive from the user. However, it assumes the taught task to be in total-order, whereas most plans in real-life can be partially-ordered [EKR13].

There has been very little work in prior literature about user interfaces that could allow a user to impart new tasks or actions implicitly; it is possible that without requiring direct expression by a user about the task's prerequisites, a planner could generate the actions (or state-machine) needed for the task to complete its goals under observed environment changes by both machine and human users. Meeting this requirement is very interesting, as it can allow users to program behaviors into a manipulator without requiring to know and implement much of the low-level technical details of both machine and task. Instead, free-form sketches can be used to implicitly express HTNs', using its annotations that hold semantic information which planners can work with. User annotations can implicitly provide predicates, intermediate and end-goal states, and induced state environment changes or effects to an executive that assemble the high-level task to be acquired. For instance when indicating a manipulator to put a lego block on the shelf, a user can mark and label both the book and shelf, which allows a robot to isolate specific areas in the workspace to acquire spatial and semantic knowledge about the objects and locations involved in the task (Figure 1.2). Tasks can be also merge into larger task trees through user interaction and re-validated by the planner. If no feasible action could be found by any planning agent for the task to proceed, the action could be implied through user recommendation or intervention. Automated planners can leverage on implied HTNs to generate task-executives users can collaborate on in a shared autonomy based on Parasuraman's proposed function classes.



Frame /torso_lift_link

Arm ● Left ○ Right

Duration 0.50

Position

X 0.50

Y 0.67

Z 0.00

Orientation

Phi -95.26

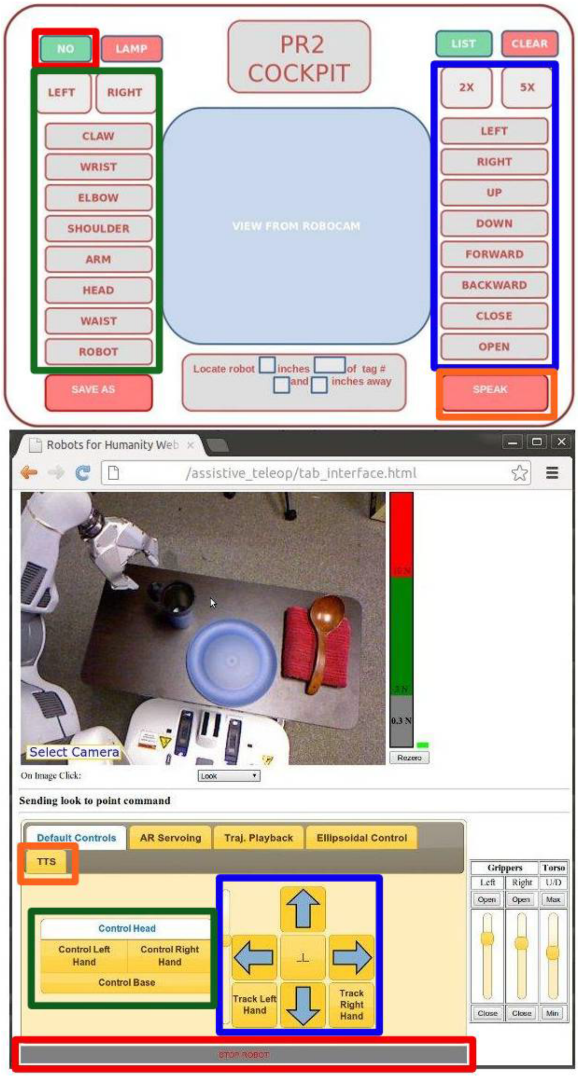
Theta -49.83

Psi 17.93

point0
point1

Stopped with outcome succeeded0

Figure 2.4: RCommander User Interface (Nguyen et al., 2013)



Sending look to point command

Figure 2.5: Interface Layout (Chen et al., 2013)

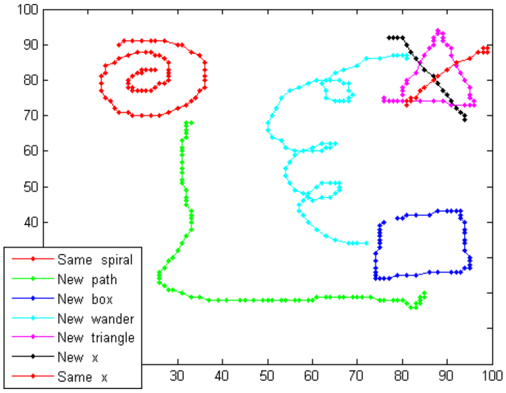


Figure 2.6: Example Sketch with six gestures, each gesture symbolizing an action, waypoint, path, or interesting landmark (Shah et al., 2010)

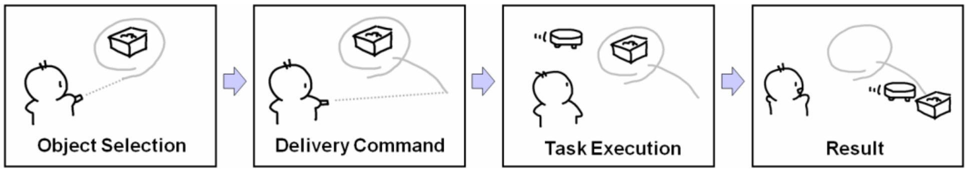


Figure 2.7: Robot Control using Laser Gestures (Ishii et al., 2010)

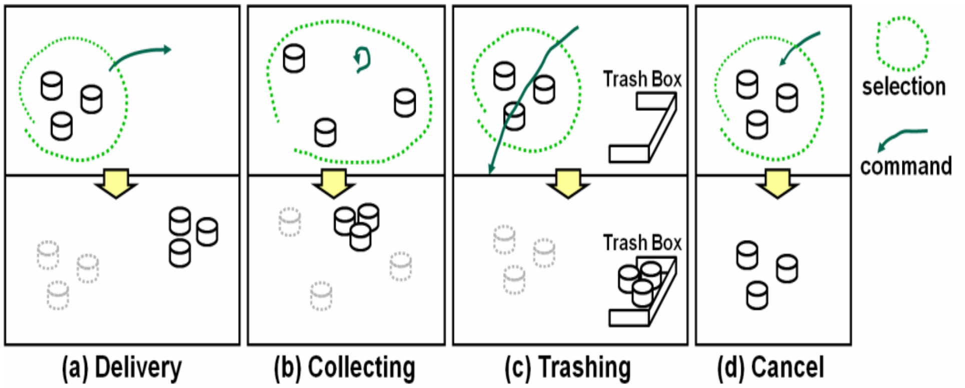


Figure 2.8: Command Gestures (Ishii et al., 2010)

Natural Abstractions for Human-Robot Interaction

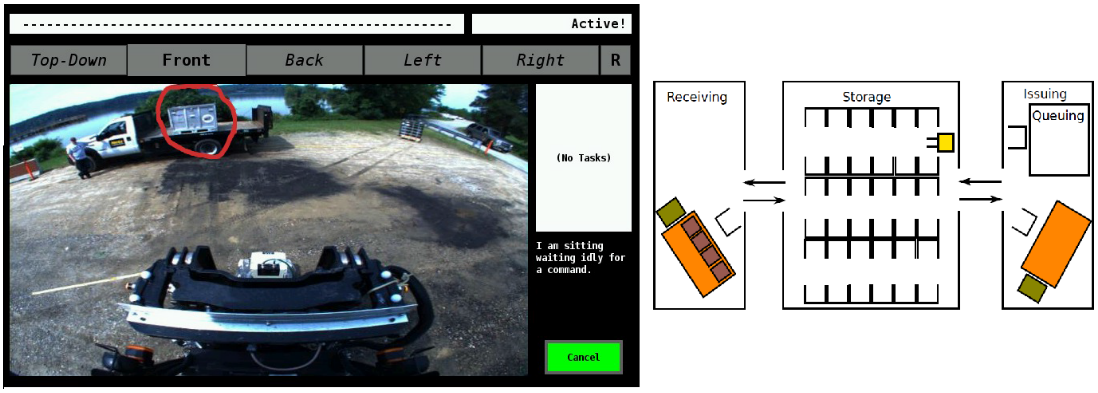
Using sketch and touch modalities as part of a Natural User Interface (NUI) to convey natural abstractions that interact with robots is a widely studied topic from prior work [CWF10], [SSC10], [SAB07], [SHI09], [LSI11]. The vast literature, however, can be classified as either action or task-oriented works. But varied hybrids between the two categories can be observed in some, though with more emphasis on the former. Action-oriented applications typically *prescribe* robots on an optimal course of action(s) which maximizes probability of the robot completing its assigned task ('user-knows-best'), while task-oriented approaches allow for greater autonomy but tend to require user intervention between execution due to greater risk, especially when operating under changing task conditions.

Action-Oriented Sketches

Works by Skubic [SAB07], Sakamoto [SHI09], Sugiura [SSW10], Ishii [IZI09], and Shah [SSC10] let users indicate explicitly on a mobile device the goals and way-points robots have to navigate through, or what actions to perform that achieve the goals. There can be autonomy present at lower-level activities, such as obstacle avoidance when teleoperating a robot to a goal on the map [SAB07]. Using sketched landmarks on a map, Skubic et al. issues implicit navigation commands to a team of robot(s) when routes to one or many landmarks are sketched. Sakamoto [SHI09] and Sugiura [SSW10] used a closely-similar approach with the exception of sketches are made out on RGB image-streams instead of a digital canvas. AR markers[1] are used to detect the robots from a overhead live-video stream, allowing a user to reference the robot's 2D location on the image when assigning paths or actions that move a robot around the floor for vacuuming [SHI09], or assign cooking actions on a stove [SSW10]. Similarly, Shah et al. use ink gestures recognized with a trained variable duration Hidden Markov Model (VDHMM), each mapping an action to perform or an event that a robot encounters (Figure 2.6). Ink gestures events include way-points, paths, and points of interest (POI) before a robot reaches the final goal and completing the task. Instead of sketching on a mobile device, Ishii on the other hand uses laser-pointed gestures to indicate 'move-and-place' tasks for a robot (Figuire 2.7). Ishii's laser gestures demarcate objects for delivery and collection, trashing actions, or to cancel the last executing action.

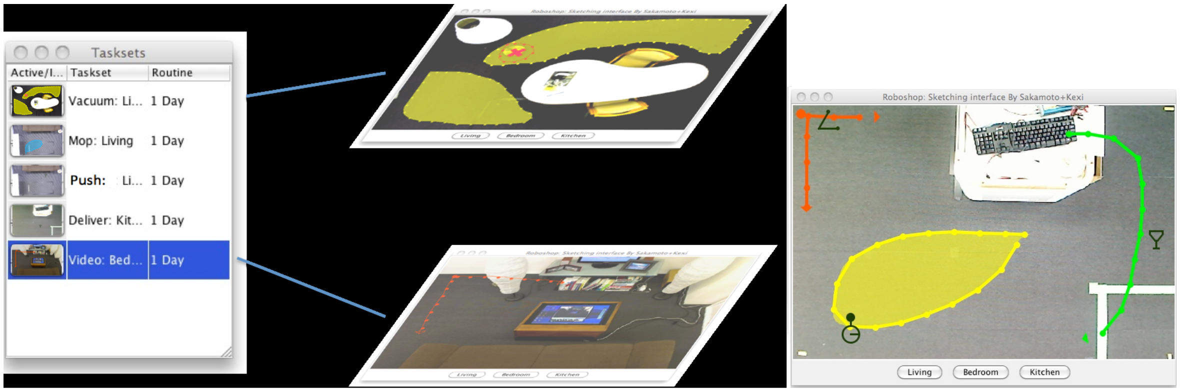
Action-oriented works that use sketch or natural gestures are interesting as they provide users an intuitive way to express the actions needed for a robot to complete the task. Upon encountering task difficulty, users can re-use the same intuitive commands for the robot to execute alternative actions that recover from the action causing the problem. Though user-knows-best strategy could be a best approach for many difficult tasks, it usually draws away a lot of valuable human attention due to a lack of trust in automation, forgoing the possibility that users might overestimate or overvalue their own decisions in comparison to solutions from autonomy [CBH11].

[1] http://artoolkit.org/



I am sitting waiting idly for a command.

Figure 2.9: (Left)User Stroke selecting the pallet (Right)Warehouse receiving area (Correa et al., 2010)



Active/I...	Taskset	Routine
	Vacuum: Li...	1 Day
	Mop: Living	1 Day
	Push: Li...	1 Day
	Deliver: Kit...	1 Day
	Video: Bed...	1 Day

Figure 2.10: (Left)Layer Palette (Right)Tasks Layers for a room overlaid upon each other (Liu et al., 2011)

Task-Oriented Sketches

Task-oriented applications could be found at Correa [CWF10] and Liu [LSI11], using ink strokes to indicate what needs to be done (e.g. move X from Y to Z) but delegating entirely how it is done to the robot. Correa et al. developed an application where a user demarcates the pallets to be moved off a truck with an autonomous forklift using sketch and speech commands. Upon receipt of the selected pallet, the forklift proceeds with the transportation task to a storage lot inside the warehouse, where the choice of lot can be optionally selected by the user. In transit, the forklift is aware of its surroundings, such as pallet recognition, fixed and moving obstacles, people, and the available storage lots at the warehouse (Figure 2.9). Liu et al. developed a prototype for sketch layering that imitates the image editing framework of Adobe Photoshop, which they call Roboshop; Each layer represents one task for a specific room that can comprise of multiple actions, such as vacuuming, mopping, or push/grab-and-deliver objects (Figure 2.10). Multiple layers can be overlaid over one another, presenting all the tasks to be executed for that room. Any task layer can be saved, reloaded, and rescheduled for future use.

Even though literature have shown application of sketch or ink gestures to be highly usable and intuitive for expressing tasks instead of actions, support for user-assisted task recovery is either absent or discharged with action-oriented interaction. The forklift in Correa's study either awaits for reassignment instruction, or switches to manual mode that lets the user take over complete control. This recovery strategy can generally be observed inside nearly every multi-modal system that supervise robot teams in collaborative task control [MTA00], [CBH11]. It is a safe and effective scheme, but robot-centricity is lost since the robots do not take any part during recovery planning. Liu hardly covered task recovery roles and strategies during collaboration, assuming the robots tasked by Roboshop are operating in closed-world environments. But the simplified assumptions were valid given that the work's emphasis was the evaluation of interface usability. Another objective for the work in this dissertation is to provide this role for users under semi-autonomy, other than demonstrating the use of sketch for task-oriented interaction.

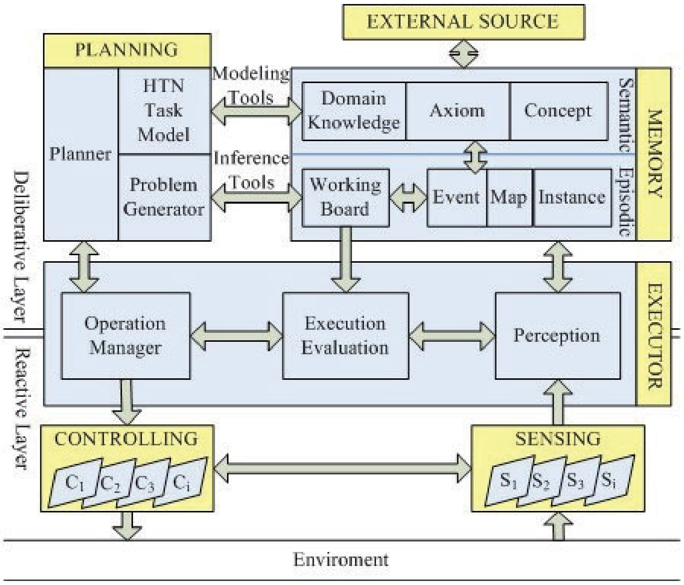


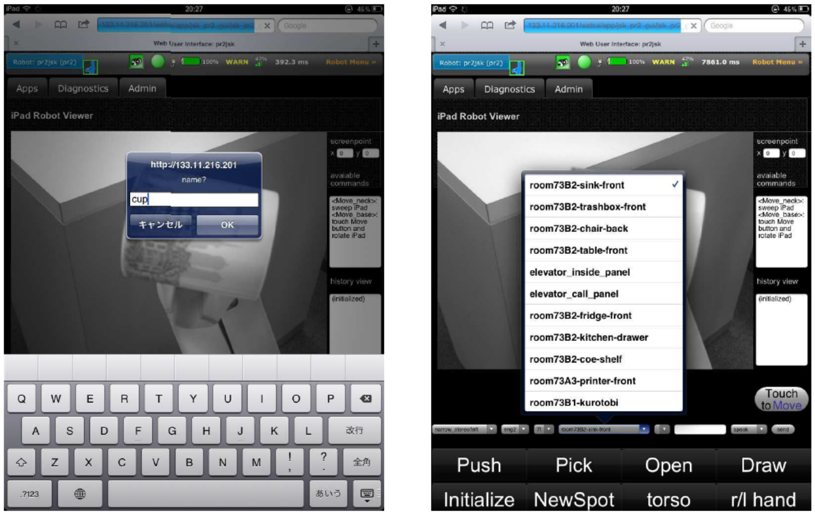
Figure 2.11: Memory-Based HTN planner control architecture (Zhang et al., 2011)

Task-Oriented Interfaces under Incomplete Knowledge

Past literature have shown that fully autonomous systems can perform quite well despite having incomplete information under partially observable dynamic environments. Studies from Karaman [KF11], Zhao [ZC14], Bohg [BK10], Navarro [NGP13], and Quispe [QMG15] have shown interesting workarounds with respect to motion and grasp planning when having incomplete information about the world state. Analogously, Weser [WOZ10], Zhang, and Zeng et al. described the approaches and developed a deliberative-reactive framework [ZZL11], [ZZL12] when dealing with incomplete world knowledge in the context of fully autonomous task planning (Figure 2.11). Weser et al. offered two ideas; the first strategy is to put the robot in a perceive-replan pattern, where any unknown object or location shall be replaced with a default plan of exploring the unknown entity until recognition occurs (e.g. search for Lego block), or adding a book-keeping literal to the plan (e.g. Lego block not found) if recognition fails so as to prevent an infinite perceive-replan loop. The second strategy attempts to circumvent the need for re-planning, by generating a solution for the task based on a prediction of the perceived unknown world state (e.g. placing Lego block on a shelf by assuming that the shelf is usually empty). If the plan according to the predicted world state fails, the symbolic and continuous representations of the world state are automatically modified, initiating a new perceive-replan process of the first strategy.

Instead of letting robots explore and perceive an unknown object under Weser's strategies, robots could obtain new symbols of unknown instances by human observation. Mikita et al. [MAK12] proposes constructing conditional plans, and populating these plans with the required symbols via human recognition when the robot encounters unknown objects while executing the task. For instance in the task of clearing a table, the final goal-state is for any object that was on the table to be placed elsewhere in order to have a clear table. From the task description, a conditional plan can be constructed but some of the objects that require moving may be unknown to the robot. Mikita's strategy is to proceed on with the object pick, and with it in the robot's camera view while in the state of perceiving, delegate recognition to the user via an input labeling of the object (Figure 2.12). Upon receipt of new knowledge about the unknown entity, the robot is able to move the newly-recognized object to a known storage location (e.g. all cups are kept in a kitchen cabinet).

Besides the implementation of a collaborative shared perceptual autonomy, Mikita's work is interesting in the context of this dissertation; it proposes an intuitive technique that enable users to assist robots in task planning via an implicit modification to the task plan through semantic labeling of objects unrecognized by the robot. The implementation, however, has two major drawbacks; firstly, prior conditional task plans would need to be prepared before the robot can be useful. Also, this approach asserts the requirement that in any plan, recognized objects either by automated means or user grounding have a known location it needs to go, or move away from. For example in Mikita's experiment, the recognized cup instantly goes to the sink, and the plastic bottle to trash. The Planning Domain Description Language (PDDL) that describes the task plan and action behaviors are encoded in Lisp, before conversion to a state machine readable with ROS SMACH. The work in this dissertation sets out to discharge this requirement, by demonstrating the use of free-form sketch that implicitly expresses the required task behavior, which can also be saved and re-loaded for future use. Secondly, the system does not make provision for changing semantics, such as the fact that movable objects themselves can be locations, or sometimes, they could be viewed like obstacles as if they were part of the table. For instance, decorative objects on a table, such as a flower vase or a picture frame, are items that usually should not be moved. Utility objects, such as letter trays and stationery holders, are movable items found commonly on a study table that are better associated as locations. Work in this dissertation sets out to study the use of free-form sketches to assist robots in differentiating between objects that can be manipulated or behaving as locations.



Input text

Choose from a list

Figure 2.12: Tablet screen of human supervisor input (Mikita et al., 2012)

CHAPTER 3
SYSTEM ARCHITECTURE & INTERFACE DESIGN

Introduction

We describe our theoretical service framework and interface specifications where stylus and touch modalities are presented for robot-centric collaboration. On system request, interactions for task modeling and recovery bias autonomous-planning at two manipulation levels; (1) *high-level*[1] task, and (2) *low-level* actuator primitives of grasping and motion planning. Otherwise, the robot is unsupervised and left alone to decide actions that perform the task.

Design Motivation

Service robots typically work in partially observable environments that constrain planning. It is impractical to assume symbols generated from a known environment will be sufficient to enable execution (AI Frame Problem). But the observability requirement can be relaxed through human knowledge that help robots to learn the appearance by grounding

[1] In this work, we define a high-level task as a sequence of one or more high-level actions. A high-level action may require one or more primitive actions that include base motion, manipulator trajectories, and grasping.

symbols to their unknown perceptual instances. Corroboratively, the trend towards semantic ontologies for knowledge expansion can produce unintended over-fitting. To scale an action corpus (Chapter 4, Section *‘Theoretical Framework’*) across ‘common’ artifacts that share utility concepts[2], ontologies are exploited to infer deeper relationships from reasoning rules (Figure 3.4). As illustration with a simplified ontology from [TB13] (Figure 3.1), observations from a serving tea demonstration could provide the following action corpus in Planning Domain Definition Language (PDDL) notation:

Listing 3.1: Serving Tea

```
(:action bring-tea :parameters (?x ?y ?z)
  :precondition (and (FoodVessel ?x) (FoodVessel ?y)
                (FurniturePiece ?z)
                (or ((has-tea ?x) (has-tea ?y)))
                (clear ?z))
  :effect (and (at-furniture ?x ?z) (at-furniture ?y ?z)))
(:action pour-tea  :parameters (?x ?y)
  :precondition (and (DrinkPreparationDevice ?x)
                (FoodVessel ?y) (not (has-tea ?y)))
  :effect (has-tea ?y))
```

These actions are agreeable with instances *‘teapot’* and *‘cup’* as they have concepts (or concept relations) *‘DrinkPreparationDevice’* for *‘teapot’*, *‘FoodVessel’* for both. Instance *‘kettle’* could be a *‘DrinkPreparationDevice’*, and *‘glass’* *‘FoodVessel’*, but interpretation is subjective (e.g. medication bottles, vases as *‘FoodVessel’*). Some instances can also assume

[2] We define an artifact to be an object or space that maps between symbols of an action to their perceptual representation. Concepts are resource *triples* describing relationships between artifacts tied by an *is-a* relation (https://www.w3.org/TR/rdf-concepts/).

separate concept relations (e.g. *DrinkPreparationDevice('teapot')*, *FoodVessel('teapot')*). An effective solution is for users to assert or retract semantic facts (or *ABox* statements) about instances that enable or disable actions with respect to personal task demands[3]. A secondary consideration for semantic modification is perceptual adaptability; allowing uncommon, arbitrary artifacts and spaces[4] left out of standard ontologies to assume concepts outside of common-sense knowledge.

Second, authoring task behaviors for robotics in an unstructured world can be demanding for non-technical users, even with visual languages reported at [NCH13], [PHJ17]. Using the corpus from Listing 3.1, prior robot observations of the space before or after serving tea include whether the desk is full, or if the desk presents a cup with no tea. One iteration includes pouring the tea *before* bringing to the desk. The amenable approach is to minimize authoring by abstracting users *away* from the corpus, modeling the task as *effects* parameterized with semantic instances from an ontology[5].

Third, the partial observability of spatial geometries in the world can be disruptive to primitive manipulation planning in a tractable task plan. Using back the corpus from Listing 3.1, either action would be impossible without finding stable grasps on the teapot if the robot perceives the handle geometries to be missing, or when there are no feasible manipulator trajectories towards the *pre-grasp* pose [CCM13], [PGS17]. Autonomous reconstitution is hard [BJL11], [QMG15], especially for unknown or novel artifacts that require

[3] Chapter 4, Section *'Theoretical Framework'*, *'Collaborating With Semantic Assertions'*

[4] *Instances* for resting items, like *'desk'*, *'shelf'*, *'bulletin-board'*, are debated to be enclosed planes which take arbitrary shape and size.

[5] Chapter 4, Section *'Theoretical Framework'*, *'Collaborating In Tasks'*

user grounding. State-of-the-art trajectory planning have traded completeness (exhaustive path search) for improved efficiency (probabilistic searches) [ES14], [DCQ16], resulting in false-negatives. However, there are effective cooperative approaches to reconstitute the geometries or teleoperate (Chapter 5).

To clarify development goals that describe how users would use stylus and touch collaborating natural abstractions to a service robot, the following sections from *'Put-That-There'* to *'Grasp/Place-That-There-Differently'* illustrate 'pick-and-place' use cases that motivate the interface and architecture specifications.

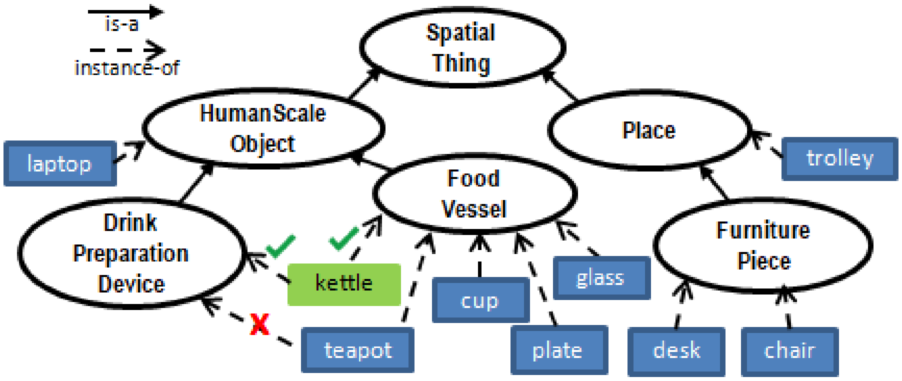


Figure 3.1: Layout of the conceptual ontology for the corpus *'bring-tea'*, *'pour-tea'*, *'bring-toast'*, and *'remove-to-trolley'*. Instances are represented in blue boxes. Red 'X' indicates a semantic retraction, rendering any instance of 'teapot' to be ignored for manipulation when *'pour-tea'* activates.

'Put-That-There'

A user observes security video streams of his home on a mobile app on his phone while at work. He sees the kettle on the coffee table, and would like it to be placed on the stove. The user achieves this by tapping on the kettle in the stream followed by the stove. The user has modeled a reusable predicate (*'Put-That-There'*) comprising of the 'kettle' object, and 'stove' as the location.

'Put-Them-There'

The user has modeled and saved multiple 'Put-That-There' predicates previously. He observes from the mobile app that two or more predicates can complement one other. Taking on one case where objects 'plate' and 'cup' go on his study desk for meals, he proceeds to merge the two predicates and saving as a coalesced predicate (*'Put-Them-There'*).

'Put-That/Them-There-Differently'

The user schedules the service robot to execute a saved predicate (*'Put-That-There'* or *'Put-Them-There'*), but the state of the world or environment keeps changing. For example, the user would like his laptop computer to be brought to the study desk when he gets home, but a plate with other eating utensils are taking up the space needed. However there is nothing to be concerned about; before getting the laptop, the robot proceeds to make space on the desk by relocating the plate and utensils to the sink.

However in more common scenarios which occur under partial observability, the service robot will be undecided and does nothing (a stall occurs) if it detects any reasonably large geometries unregistered to its knowledge base that are on the study desk. The service robot alerts the user on his phone, who proceeds to resolve the perceptual ambiguity by lassoing out the 'obstacles' and providing semantic descriptions.

'Grasp-That-There'

The service robot stalls when grasp-planning is unable to detect a stable grasp due to ambiguous or incomplete geometries [BK10], [NGP13], [QMG15]. For example, the service robot alerts the user when it perceives large clusters of points to be missing from a cup when setting or clearing his study desk. The user responds to help reconstitute the geometries back to a usable cluster or shape that resumes the grasp.

'Grasp/Place-That-There-Differently'

The service robot stalls when motion-planning is unable to locate valid trajectories for its manipulator(s) to navigate towards or place an object. For example, the user is alerted when the robot perceives unknown clutter on the desk to be too obstructive to reach for the cup without making contact with its manipulators. The user responds either by manual teleoperation, or instruct motion-planning to explore alternative configurations for a feasible trajectory.

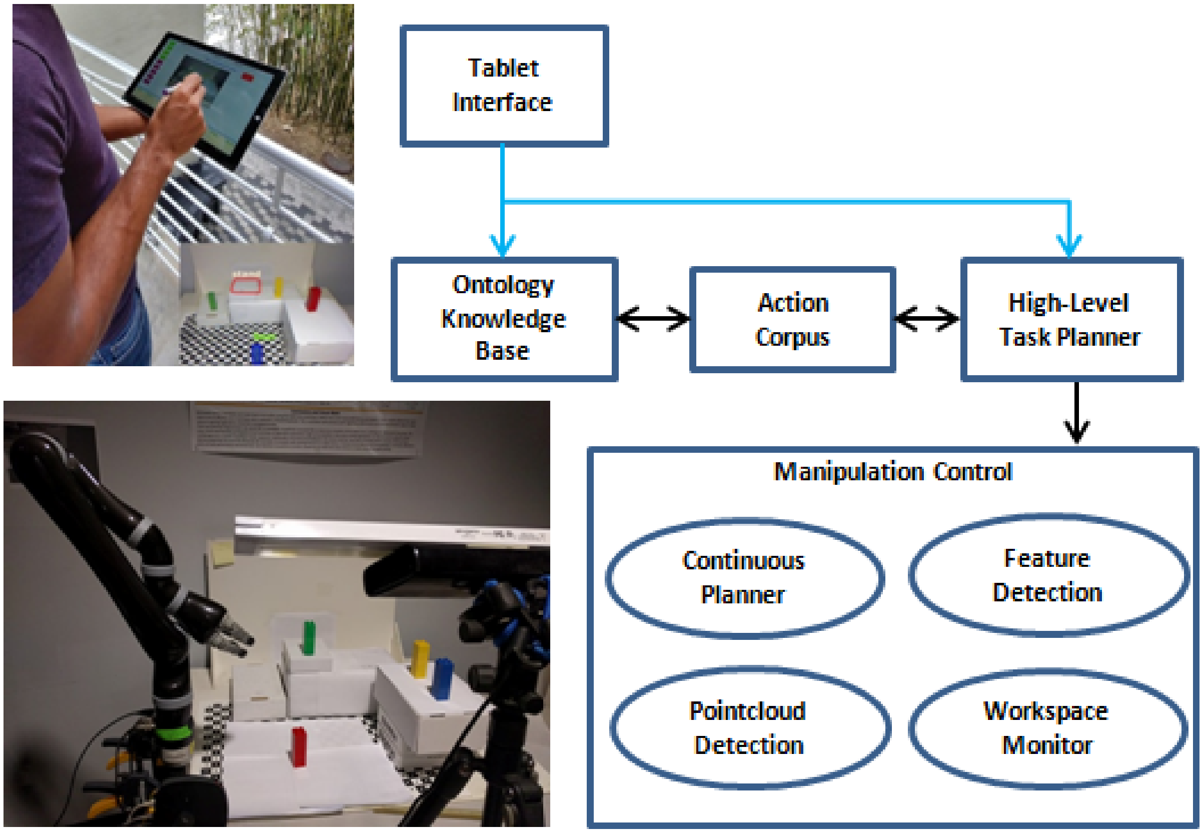


Figure 3.2: System Architecture: Blue Arrows represent interaction presented when modeling new tasks and semantic assertions for collaboration using the tablet.

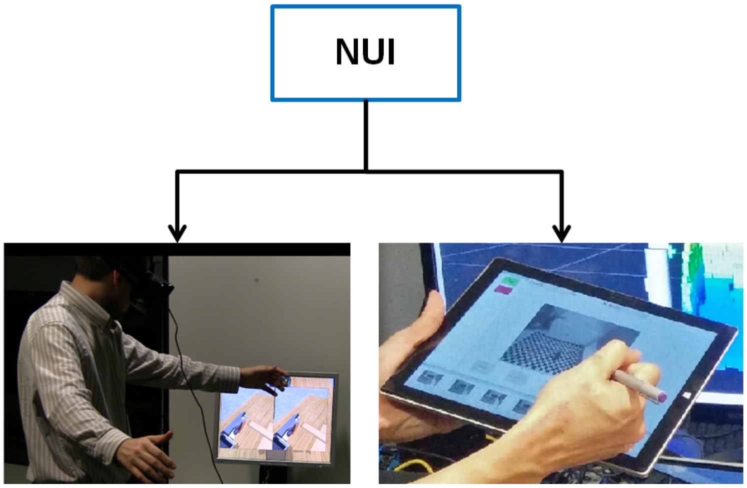


Figure 3.3: Our Natural User Interface (NUI) Prototype

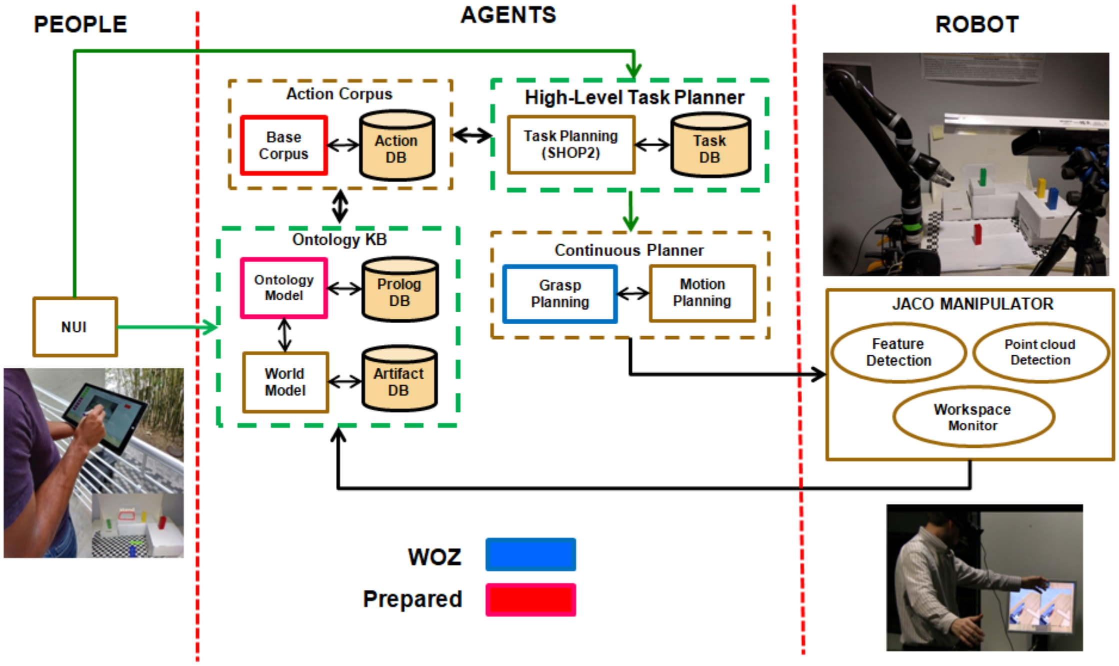


Figure 3.4: Data-Flow Diagram (DFD) of the System Architecture as a RAP Multi-Agent-System (MAS): System Blocks in red were prepared before-hand for our user studies; Grasp Planning is a simulated system, or Wizard of Oz (WOZ) implementation, performing perfect grasp postures

System Architecture

Figure 3.2 illustrates the system architecture composed of five primary blocks; our NUI client that includes a Head-Mounted-Display (HMD) and a mobile device such as a tablet (Figure 3.3), Ontology Knowledge Base (KB), Action Corpus, High-Level Task Planner, and Manipulation Control. The *Ontology Knowledge Base* module is used to persist semantic rules and facts of the known perceptual instances according to the concept ontology. We used Protégé to build our concept relation taxonomies according to OWL/RDF standards. Apache Jena with SPARQL is used to maintain and query the ontology database. Besides persisting snapshots of the current state of the world for reasoning, SWI-Prolog is also used for building and reasoning the semantic rules and facts about the artifacts (e.g. *FoodVessel(cup), DrinkPreparationDevice(teapot)*) and providing updates to the database upon a semantic assertion or retraction (Chapter 4, Section *'Theoretical Framework'*). The *Action Corpus* module generates further actions by reasoning from the Ontology KB module. We used SHOP2 for our *High-Level Task Planning.* The planner references the corpus for methods, and reasons the Ontology KB for predicate states. The *Manipulation Control* module converts the high-level task plan into lower-level manipulator actions, leveraging Robot Operating System[6] (ROS) specialized continuous planners for trajectories and grasps. Manipulation Control also commands the sensor array that provides perceptual feedback about the world back to the Ontology KB with feature extraction and processed point clouds. Figure 3.4 illustrates the data-flow interactions across key components between blocks under

[6] http://www.ros.org

the RAP system (Robots, Agents, and People) [SJP02]. Through the NUI, a user interacts with agents that maintain ontology representation and tasks modeled as Hierarchical Finite-State-Machines (HFSM) behind-the-scenes. A HFSM illustrates how a robot switches to different states between actions that require motion or grasp planning according to the current world belief. The NUI component provisions four main collaborative features: (1) Augmenting semantic and spatial representations of the world, (2) Augmenting continuous planning, (3) Assisting Task Behavior and Modeling, and (4) Agent-assisted teleoperation. Sensor models that detect exogenous events or exceptions that occur during an action are reported to the NUI, allowing the user to intervene either with teleoperation, or with stylus and touch modalities (Figure 3.3). In the user studies, our service robot is modeled by a six degrees-of-freedom (DOF) JACO arm manipulator[7] with a Kinect RGB-D Sensor (Figure 3.2).

Interface Design

Our NUI comprise of two child interfaces for interaction; a Mixed-Reality Head-Mounted-Display (HMD) with skeletal-joint tracking that supports user-avateering in teleoperation (Chapter 6), and Second, a tablet or any suitable mobile device (Figure 3.3). To present semantic assertions and task collaboration (Chapter 4, Section *'Prototype & Experiment Design'*) with stylus and touch, the tablet interface is divided into four perspectives

[7] http://www.kinovarobotics.com/service-robotics/products/robot-arms

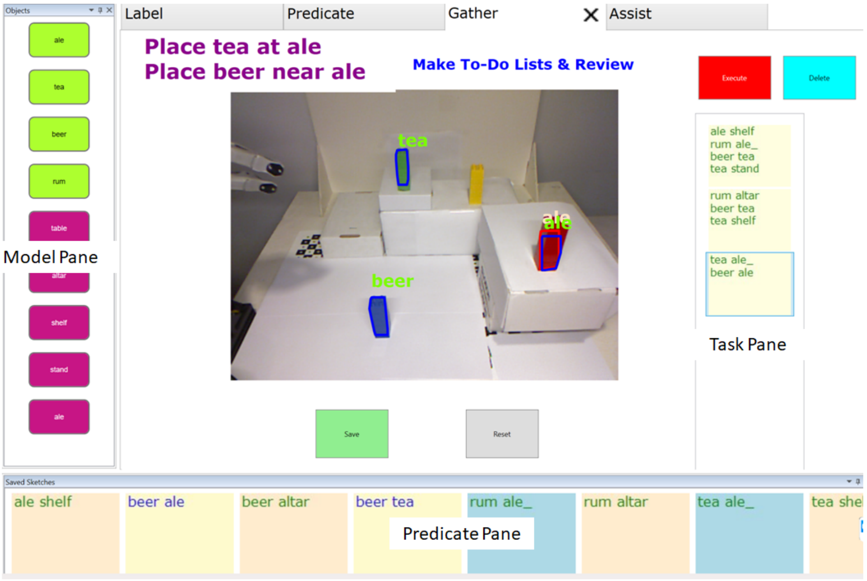
for a distinct interaction; (1) Symbol and Semantic Concept Grounding, (2) Predicate Construction, (3) Predicates Coalescence, and (4) Collaboration and Assistance. In the following sections, we list general design specifications independent to any service manipulation domain; a specific implementation for 'pick-and-place' was developed for our user studies at Chapter 4 however, and we shall use snapshots of the experiment prototype for illustration.

Symbol & Concept Grounding ('Label')

Lasso strokes of different-color stylus ink are used for grounding and recognition of unknown perceptual instances to the Ontology KB (Figure 3.4). Each color represents a different concept relation of the ontology. For example, following Figure 3.1, there would be six colors available to a palette menu. For our user studies, only two colors are used (Chapter 4, Section *'Prototype & Experiment Design'*). The instance is processed according to the semantic relation encoded in the color of the stroke, followed by asserting further concept relations by cascading upwards. Similarly, semantic retractions from the user would also be published to the Ontology KB.

Predicate Construction ('Predicate')

A list of predicate functions, according to the effects of each action from the Action Corpus (Figure 3.4), would be available for selection when constructing a predicate. For example, if the corpus is taken from Listing 3.1, the available names would be *'at-furniture'* and *'has-tea'*. Before selecting the desired effect function (or predicate function) from the corpus, touch modality is used to select the concept(s) to populate the predicate function parameters either through the live image or from the Model Pane. The populated predicate can be saved and previewed in the Predicate Pane (Figure 3.5).



Place tea at ale
Place beer near ale

Figure 3.5: Layout of the 'Gather' Perspective: Model Pane (left) displays known concepts about the world. Predicate Pane (lower) displays Object-Location, Object-Object effect predicates. Task Pane displays each task as a list of effect predicates (right)

Coalescing A Goal ('Gather')

A task is modeled by expressing a conjunction of predicates[8] as a goal. The task is modeled by selecting the predicates one after another from the Predicate Pane, appending to

[8] Section *'Interface Design', 'Predicate Construction ('Predicate')'*

a list that appears as an overlay. Once the user is satisfied with the model, the conjunction can be saved and previewed at the Task Pane (Figure 3.5). To submit a task to the High-Level Task Planner (Figure 3.4), a selection is made from the same pane and publishing a request to execute the task.

Collaboration & Assistance ('Assist')

Real-time monitoring of an executing task shall be observed as a sequence of primitive manipulator actions processed by Manipulation Control (Figure 3.4). The current executing action is highlighted at each stage (Figure 3.6), providing a status of the last executing state if the task stalls and requires manipulation assistance.

Conclusion: Natural Abstractions For Cooperative Planning

Non-technical users can be efficient with service robots by cooperating with autonomous planning. Collaborative robot-centric scenarios through natural modalities (Section *'Design Motivation'*) have shown how task specifications and recovery for robot manipulation in a dynamic and unstructured world can be modeled with actuation abstractions to the service framework. As a final resort, tasks that remain unrecoverable after cooperative interaction can be teleoperated either fully, or to an amenable state that allow task autonomy to recover.

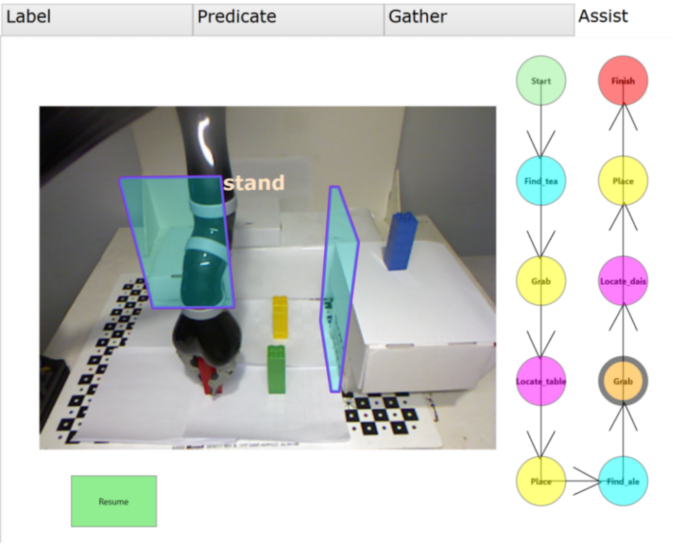


Figure 3.6: Current executing action is highlighted at each stage of the task - node indicates a ‘grab’ action. ‘stand’ is offered as an intermediary for motion planning to manipulate past the virtual obstacle

CHAPTER 4
PLANIT! HIGH-LEVEL TASK COLLABORATION

Introduction

In chapter 3, we addressed architecture and interface considerations amenable to non-technical users for cooperative task modeling and recovery of service robots. We shall be examining in this chapter our theoretical framework behind cooperative modeling and perceptual assistance that abstracts the action corpus away from the user (Section *'Theoretical Framework'*). To evaluate user perception and quantitative performance, we developed a prototype for 'pick and place' tasks in a 'Blocks World' planning domain (Section *'Prototype & Experiment Design'*), and compared our interface against a desktop control that uses a very simple visual language. The remaining chapter discusses the empirical setup and results of the user study.

Design Motivation

Visual languages such as Behavior Trees [PHJ17] and Hierarchal Finite State Machines (HFSM) [NCH13] were proposed to simplify task authoring in robotics. However, it does not abstract away manual programming; a user still performs an analytical consideration for *all* iterations of the task in a *fully known* domain beforehand. But manual programming can be simplified further with kinesthetic teaching and natural language techniques [SDB13], [LOM14], [SES14b], [SCC14], [FM14], parsing observations of user demonstrations into a ***corpus*** (Figure 3.4) of usable actions which can be generalized into PDDL or Lisp-like languages [BKS12], [KSL15], [KB17]. These natural techniques can be useful for non-experts to populate the *corpus* that function as a database of reusable building blocks to a task in their free time. However, the corpus lack scale and adaptability to be applied on artifacts that are outside the demonstration domain. With a corpus, our approach encourages *rapid* modeling by not requiring users to consider the iterations and actions since they are cooperating with automated task planning. The corpus is *adapted* and incorporated to a framework expressing the actions with semantic literals[1] defined from an ontology [JQN12], [PPT12], [TB13]. Users can provide perceptual grounding and semantic revisions to task models[2] on a mobile app [CWF10], [LSI11], [PSB11], [MAK12].

Theoretical Framework

In this work, semantic concepts are set apart from *labels* commonly associated with autonomous recognition (e.g. teapot, cup, desk); though labels can be treated like concepts, they are highly *specific* instances describing appearance. On other hand, semantic

[1] Section *'Theoretical Framework'*, *'Action Templates with Conceptual Ontologies'*

[2] Section *'Theoretical Framework'*, *'Collaborating With Semantic Assertions'*

concepts describe *roles* and *definitions* (*T-Box* statements) about what and how the artifact is used (e.g. *'FoodUtensil'*, *'FurniturePiece'* is-a *'Place'*) [TB13]. Importantly, the latter informs actions or manipulations which are tractable for that artifact. Given a corpus, a user can model a new task by stating a goal, or *effects*, as a logical conjunction of predicates. Each predicate is parameterized by artifacts that can assume several *different* and separate semantic relations under the same goal (*'Collaborating With Semantic Assertions'*).

Action Templates with Conceptual Ontologies

An action parameterized by semantic concepts of a taxonomy behave closely to a *template* that preserves the parameter subtypes; scaling is possible by adapting the action to be *covariant* on literal subclasses. For illustration, consider the following action in PDDL notation that references the simplified ontology schema (Chapter 3: Figure 3.1):

Listing 4.1: Removing any *'HumanScaleObject'* to a service trolley

```
(:action remove-to-trolley :parameters (?x ?y ?z)
  :precondition (and (HumanScaleObject ?x) (Place ?z)
               (FurniturePiece ?y) (at-furniture ?x ?y))
  :effect (at-place ?x ?z))
```

Listing 4.1 can illustrate the action of removing the teapot from the desk to the service trolley (or serving-cart). This is made apparent by switching out the concept relations with their instance equivalents:

```
(:action remove-to-trolley :parameters (?x ?y ?z)
  :precondition (and (teapot ?x) (trolley ?z) (desk ?y)
                (at-furniture ?x ?y))
  :effect (at-place ?x ?z) )
```

Traversing down the taxonomy and substituting the parameters with concept relations deeper in the tree permute a total of five actions with different signatures. We highlight an interesting case:

Listing 4.2: Parameter substitution on *covariant* action *'remove-to-trolley'*

```
(:action remove-to-trolley :parameters (?x ?y ?z)
  :precondition (and (FoodVessel ?x) (FurniturePiece ?z)
                (FurniturePiece ?y) (at-furniture ?x ?y))
  :effect (at-place ?x ?z))
```

Alternative to the service trolley, the generated action can move any item with a *'FoodVessel'* relation between furniture (e.g. 'desk', 'chair'). The effect predicate *'at-place(x,z)'* remains true since the concept relation (*'is-a'*) flows upwards from *'FurniturePiece'* to *'Place'*.

Collaborating With Semantic Assertions

Enabling semantic modification over known artifacts provides an intuitive method for implied manipulation control. Suppose the world model (Chapter 3: Figure 3.4) could provide recognition of detected instances, the following facts are defined for *'teapot'* and *'cup'* as examples of detected instances x and y:

$$\begin{aligned}
teapot(x) \rightarrow DrinkPrepartionDevice(x) \wedge HumanScaleObject(x) \\
\wedge FoodVessel(x) \wedge SpatialThing(x) \\
cup(y) \rightarrow FoodVessel(y) \wedge HumanScaleObject(y) \wedge SpatialThing(y)
\end{aligned} \tag{4.1}$$

Consider also the *Serving Tea* corpus from Listing 3.1 is included; if there is preference for the teapot to be unused for pouring tea, retracting the semantic fact *'DrinkPreparationDevice'* from the teapot would disable the action *'pour-tea'*. Any retraction will cascade upwards along the relations; for instance, if *'FoodVessel'* is retracted from the cup, it is trimmed away from the ontology schema and no longer holds any concept semantics other than its own recognition label (i.e. *'cup'*). The remaining *'FoodVessel'* relation held by the teapot would still allow the action *'bring-tea'* to manipulate it.

Suppose a task would prefer the kettle to pour tea, but there is no record in the world model for automated recognition (highlighted green, Chapter 3: Figure 3.1), human knowledge can ground and persist that unknown perceptual instance (i.e. z) by providing identity (i.e. 'kettle') and the necessary semantic relations:

$$kettle(z) \rightarrow DrinkPrepartionDevice(z) \wedge HumanScaleObject(z) \wedge SpatialThing(z) \tag{4.2}$$

Similarly, asserting the semantic relation *'FoodVessel'* would enable *'bring-tea'* to move the kettle along with the serving cup.

Any task can be represented, or modeled, as a minimal set of *effects* to fulfill. Referring back to the *Serving Tea* corpus, a task of serving tea is modeled by stating the following predicates that represent a goal:

$$at\text{-}furniture(teapot, desk) \wedge at\text{-}furniture(cup, desk) \wedge has\text{-}tea(cup) \tag{4.3}$$

Casting each artifact according to the concept relations predicated at the preconditions models a more meaningful expression:

$$\begin{aligned} &at\text{-}furniture(FoodVessel(teapot), FurniturePiece(desk)) \\ &\wedge\, at\text{-}furniture(FoodVessel(cup), FurniturePiece(desk)) \\ &\wedge\, has\text{-}tea(FoodVessel(cup)) \end{aligned} \tag{4.4}$$

If the task also demands a plate of toast, the following predicate is appended to (4.4) (assuming the plate has toast):

$$at\text{-}furniture(FoodVessel(plate), FurniturePiece(desk)) \tag{4.5}$$

With reference to the first predicate in (4.4), *‘teapot’* can alternatively be re-cast as a *‘DrinkPreparationDevice’*, modeling a conceptually different effect that disables *‘bring-tea’*:

$$at\text{-}furniture(DrinkPreparationDevice(teapot), FurniturePiece(desk)) \tag{4.6}$$

Since the *'FoodVessel'* relation is never an ancestor of *'DrinkPreparationDevice'*, it is impossible for template covariance (Section 4) to generate the necessary action that accomplishes the re-casted predicate. However, the following action would if it is included to the corpus:

Listing 4.3: Bringing tea with a *'DrinkPreparationDevice'* artifact

```
  :precondition (and (DrinkPreparationDevice ?x)
                (FoodVessel ?y) (FurniturePiece ?z)
                (has-tea ?x) (has-teabags ?x) (clear ?z))
  :effect (and (at-furniture ?x ?z) (at-furniture ?y ?z)))
```

From a combined corpus from Listings 3.1 and 4.3, it becomes apparent that a user can imply the actions needed from the corpus to fulfill the task by modeling *different* concept relations assigned to the *same* effect. Alternatively, the goal can also be modeled as:

$$\begin{aligned}
&\textit{at-furniture}(DrinkPreparationDevice(teapot), FurniturePiece(desk)) \\
&\quad \wedge \textit{at-furniture}(FoodVessel(teapot), FurniturePiece(desk)) \\
&\quad \wedge \textit{at-furniture}(FoodVessel(cup), FurniturePiece(desk)) \\
&\quad \wedge \textit{has-tea}(FoodVessel(cup))
\end{aligned} \tag{4.7}$$

The interesting occurrence is of having *'teapot'* assuming separate *'DrinkPreparationDevice'* and *'FoodVessel'* concept relations under the *same* goal or effect. The *'teapot'* must have tea with tea-bags for the goal to fulfill. However, if the task demands only a cup of tea be served to the desk, stating the following predicates as the goal would suffice:

$$\begin{aligned}
&\textit{at-furniture}(FoodVessel(cup), FurniturePiece(desk)) \\
&\quad \wedge \textit{has-tea}(FoodVessel(cup))
\end{aligned} \tag{4.8}$$

The first predicate of eqn.(4.4) is unnecessary, since the second predicate is a sufficient requirement to ensure the action *'bring-tea'* is required; modeling a task only requests user consideration for the goal-states of the artifacts concerned, allowing autonomy to handle the remaining workspace which are superfluous objectives from the user's view.

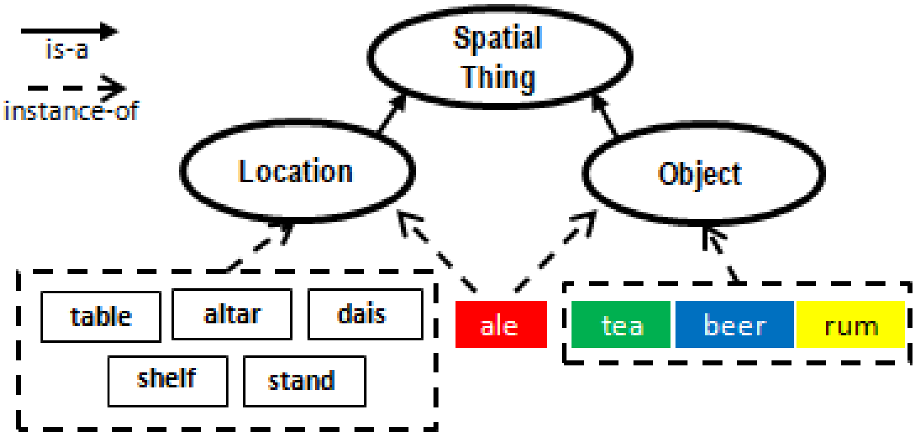


Figure 4.1: 'Blocks World' conceptual ontology

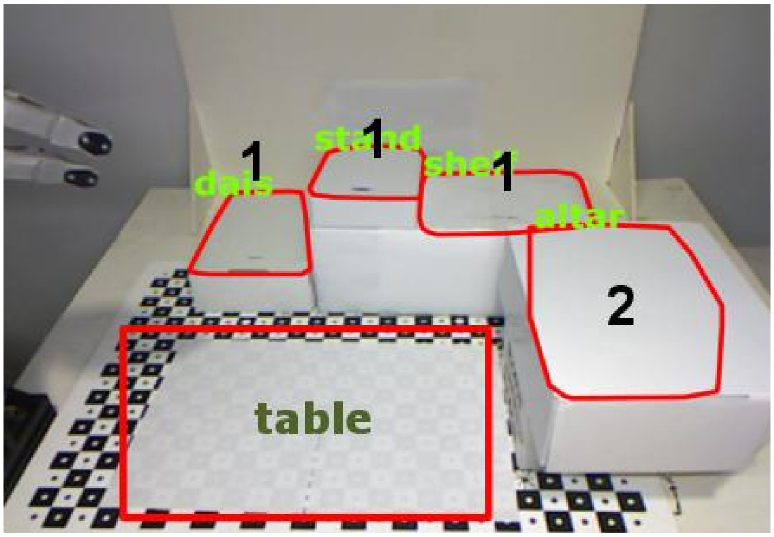


Figure 4.2: Artifacts assigned as *Locations*-Only for the study. Numbers indicate available capacity each location accommodates, e.g. 'altar' has space for two objects, remaining has one each.

Prototype & Experiment Design

Experiment Domain & Ontology

To evaluate how non-experts would perceive collaborative modeling[3], a main consideration was for a simple, clear, and controlled domain to introduce users to our system. The domain should be capable of partially observable and unstructured scenarios. 'Blocks World' [GN92], [NGT04] for *pick-and-place* tasks meets these requirements, facilitating a simple ontology of *Object* and *Location* concept relations that a color block can be either or both (Figure 4.1). Immovable artifacts that can only be *Location* concepts are included (Figure 4.2).

Experiment Setup

The JACO robot arm is used as the robot platform for development work with ROS Indigo as the middle-ware (Chapter 3, Figure 3.2). A set of four different colored blocks was prepared for stacking in our environment shown in Figure 4.2. Except for 'table', the 'Blocks World' environment is totally unknown to the robot when the study commences. A total of eight *'pick-and-place'* tasks of variable difficulty is planned such that as the study progresses, a user gradually populates the instances in the ontology to completion as shown in Figure 4.1. Each block is assigned a unique name and color feature (Figures 4.1 & 4.3) for

[3] Section *'Theoretical Framework'*, *'Collaborating With Semantic Assertions'* & *'Collaborating In Tasks'*

easy perceptual recognition, and instances which have *Location* relations have unique names and poses (Figure 4.2). Except *'table'*, there is a stack limit for every artifact, including colored blocks (limit 1) if they have *Location* concept relations asserted (e.g. *'ale'*)

Learnt Action Corpus

Modeling under 'Blocks World' allows the consideration for a simple corpus of learnt actions from literature [SCC14]. From She et al., two actions learnt from natural language demonstration are prescribed:

Listing 4.4: Blocks World' Action Corpus (Object-Location) artifact

```
(:action clear-top :parameters (?x ?y)
  :precondition (and (Object ?x) (table ?y)
              (has-space ?y))
  :effect (at-location ?x ?y) )
(:action stack :parameters (?x ?y)
  :precondition (and (Object ?x) (Location ?y)
              (has-space ?y))
  :effect (at-location ?x ?y) )
```

Examining Listing 4.4, it follows that modeling the task would be a logical conjunction of *'at-location' predicates* or *effects*, each parameterized by a *Object-Location* pair. For example, to model a task that moves colored blocks *'ale'* to the *'altar'*, and *'beer'* onto *'ale'* (Figures

4.1, 4.2), the goal is stated as:

$$\begin{aligned} &\textit{at-location}(\textit{Object}(\textit{ale}), \textit{Location}(\textit{altar})) \\ &\wedge \textit{at-location}(\textit{Object}(\textit{beer}), \textit{Location}(\textit{ale})) \end{aligned} \tag{4.9}$$

The following action is added to the corpus in order to model an effect parameterized by an *Object-Object* pair:

Listing 4.5: 'Blocks World' Action Corpus (Object-Object) artifact

```
(:action stack-near :parameters (?x ?y)
  :precondition (and (Object ?x) (Object ?y)
  :effect (at-obj-location ?x ?y) )
```

Predicate *at-obj-location* returns true when either *Object(x)* supersedes the current location of *Object(y)*, or placed at the same location with *Object(y)*.

Interface Setup

Microsoft Windows Presentation Framework (WPF) was used to develop our *Tablet Interface* prototype, deployed on a Microsoft Surface-Pro 3 tablet with an i7 processor with 8GB of RAM (Figures 4.3, 3.5). Predicate construction is either an *at-location* predicate parameterized by an *Object-Location*, or an *at-obj-location* predicate parameterized by an

Object-Object pair. As these parameters are distinguishable, the interaction to select the predicate name[4] is not required.

'Label'

Unknown perceptual instances are recognized and grounded either with an *Object* or *Location* concept relation. This is accomplished by inking a name, followed by either a blue or red-colored ink lasso around the unknown artifact that present possible 2D image boundaries. Once finished, control is handed to geometry extraction to retrieve corresponding 3D points. If the ink used is red, a plane model segmentation is performed to retrieve a plane representative of a *Location* concept relation. A cluster extraction is performed instead if the user ink is blue, retrieving the largest point cluster within the presented boundaries representative of an *Object* concept relation. Lastly, a 2D convex hull extraction is performed on the corresponding image pixels, finalizing the image bounds of the artifact. For accomplishing a *Location* semantic assertion to an instance that has an *Object* concept relation, the user would first select an artifact from the model pane followed by the red-ink menu option from the radial menu (Figure 4.3) to initiate the assertion. To retract, the *Location* concept is selected from the model pane, and a delete event is published to the Ontology module after tapping on 'Delete' located below the image (Figure 4.3).

[4] Chapter 3, Section *'Interface Design'*, *'Predicate Construction ('Predicate')'*

'Predicate' Construction

A predicate is defined either with an *Object-Location* that parameterizes the predicate *'at-location'*, or an *Object-Object* pair that parameterizes the predicate *'at-obj-location'*, following the effects defined by the action corpus (Listings 4.4 and 4.5).

'Gather' A Task

The goal (eqn. 4.9) is modeled by selecting the predicates one after another from the Predicate Pane, appending to a list that appears as an overlay (Chapter 3, Figure 3.5). To submit a saved task to the High-Level Task Planner, a user selects a task from the Task Pane and submits the request by tapping 'Execute' located above the pane.

'Assist' & Collaborate

For our experiments, three possible causes of task failure could occur; (i) when the high-level plan encounters unknown artifacts that prevents an action in the plan from proceeding, (ii) when grasp planning fail to locate stable object grasps, and (iii) motion planning fail to find trajectories between primitive actions. The first exception is resolved with Symbol & Concept Grounding (Chapter 3, Section *'Interface Design'*). Resolving the second and last exceptions shall be addressed at Chapter 5.

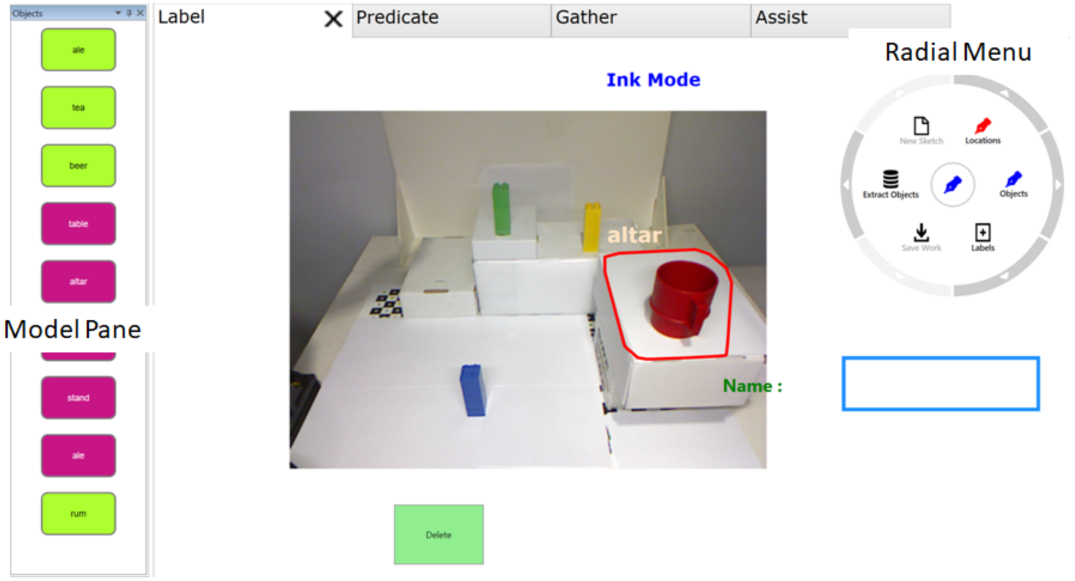


Figure 4.3: Layout of the 'Label' Tab Pane: Stylus ink is used here for symbol & concept relation grounding of the environment.

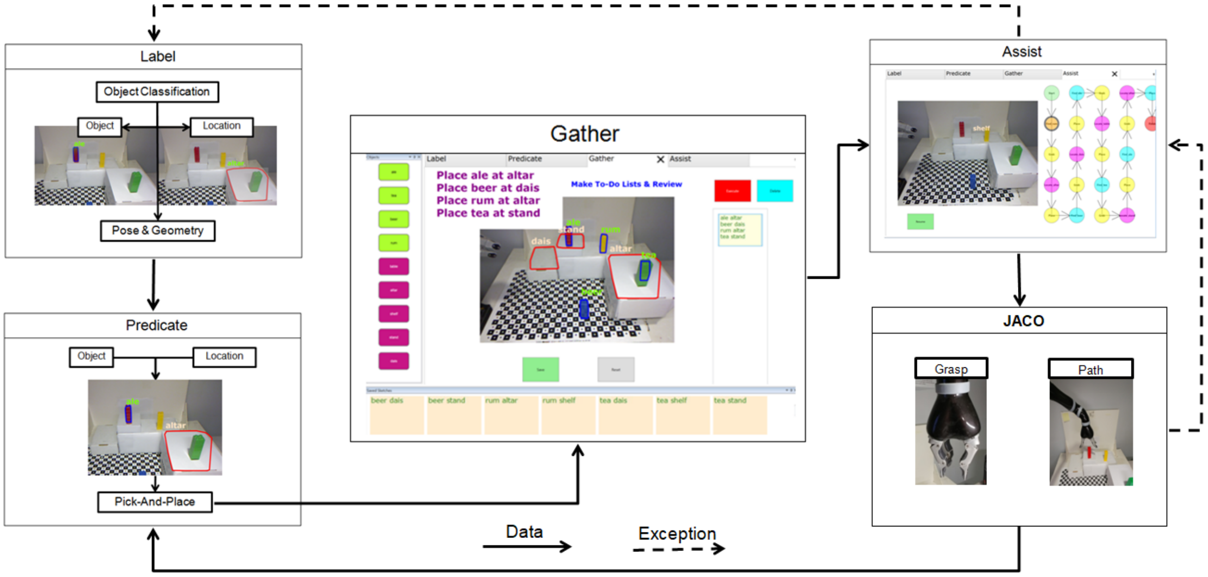


Figure 4.4: Overview of the four interaction panels: (1) 'Label' - Object and Location symbols are identified. (2) 'Predicate' - Pick-And-Place predicates are instantiated either as an *Object-Location*, or *Object-Object* tuple. (3) 'Gather' - Modeling a Task with Predicate Coalescence . (4) 'Assist' - Published user alerts in the event of a manipulation stall.

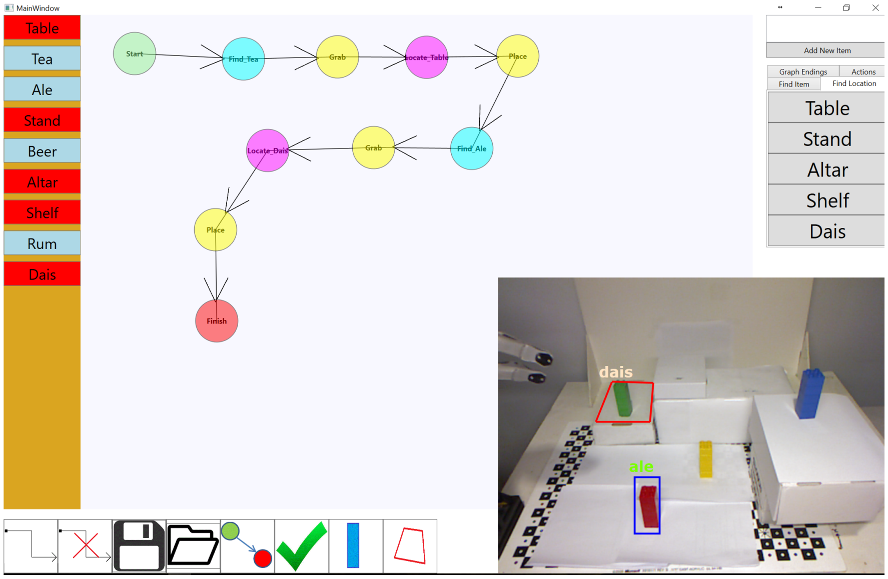


Figure 4.5: desktop control in the within-subjects study

Object (tea) : Location(stand)

Object (ale) : Location (shelf)
Object (beer) : Location (altar)

Object (tea) : Location (stand)
Object (beer) : Object (tea)

Object (tea) : Object (ale)
Object (beer) : Location (ale)

Figure 4.6: Trial 1 - The four easy tasks

Empirical Study

To evaluate our tablet system performance and measure user perception and preference, we performed a within-subjects user study that compares our interface against a desktop control. The control was built for non-technical users to learn quickly how to build 'Blocks World' manipulator tasks with simple visual *programming*, represented as a chain of inter-linked manipulator activity nodes that facilitate drag-n-drop using mouse and keyboard (Figure 4.5). The tool panel layout for the control was made to be as closely similar to the tablet, providing common WIMP-style interaction for editing artifacts, manipulator nodes, connector links, and saving work. Unlike the tablet system, the control requires a user to have full observability in order to resolve the *sequence* of manipulator actions needed to fulfill the task. For example, to declare the manipulator to place *'tea'* onto *'table'* requires a creation of four linked nodes (Figure 4.5).

Subjects

For evaluating our tablet interface with non-experts, we recruited 20 participants from outside major fields in Science, Technology, Engineering, and Mathematics (STEM); the average age was 23.7 with a standard deviation of 3.7. There were 16 females and 4 males. On a 7-point Likert scale, the sample reported very low experience with manipulators (M = 1.9, SD = 1.3) and medium-low experience with pen/touch interfaces for leisure (M =

3.3, SD = 1.7) and for work (M = 2.6, SD = 1.7). Only 6 knew at least one programming language on a novice level, and the mean experience with the most familiar was 2.8 (SD 1.3).

Experiment Tasks

The experiments consists of two main trials; each with four tasks giving a total of eight. A secondary objective for the first trial (*Trial 1*) was to provide a gentle introduction to the experiment and system. The trial provides four easy tasks for the user to clear; each task can be completed in three *'pick-and-place'* moves or less. Before the user starts modeling a task, the blocks were arranged in a set manner to simulate unstructured environments. For example (Fig. 4.6), the first task requires the user to move the green block 'tea' to the location 'stand' (refer to Figure 4.2 for the experiment locations). Before the first task commenced, block 'tea' was set at the location 'table' as the starting position, and there were no blocks occupying the location 'stand'. However for the starting positions for the third task, 'tea' was set at the location 'altar', and the red block 'ale' was set at the location 'stand'. The third task required the user to move 'ale' off the 'stand', before proceeding to move 'tea'. It is near impossible for the modeled task using the control to be reused when the starting positions of the blocks change. For the tablet interface however, this is not a concern. For the control application, the user would use between eight to twelve nodes to resolve the task sequence. For the tablet system, the user would model a task, or goal, using between one to two predicates (Fig. 4.6). The fourth task requires a user to accomplish a

Location semantic assertion to an *Object* concept (*'ale'*), modeling a task where *'ale'* assumes two different concept relations (Figure 4.6):

The second trial (*Trial 2*) provides four hard tasks to clear; each require at least three *'pick-and-place'* moves or more. For the control, the user would require between twelve to twenty-four nodes, while the user would model a task requiring between two to four predicates for the tablet system. Two of the tasks were essentially the same for the tablet system; the difference between the tasks was the last predicate that is missing from one of the two (eqn. 4.10).

We set a partial observability scenario with the same hard task (eqn. 4.10). When the task is submitted for execution using the tablet system, the task will stall as the yellow block 'rum' has not been identified at this point of the study. The user would need to provide symbol grounding through *'Label'* interaction (Section *'Prototype & Experiment Design'*) in order to resolve the stall. This scenario would not be a concern when using the control as full knowledge for the task is needed.

$$\begin{aligned} & \textit{at-location}(Object(ale), Location(stand)) \\ \wedge\ & \textit{at-location}(Object(tea), Location(shelf)) \\ \wedge\ & \textit{at-location}(Object(beer), Location(table)) \end{aligned} \tag{4.10}$$

Study Procedure

Our recruited participants reported to our laboratory and presented with the client apparatus. They were seated out of sight from where the manipulator is operating. A demographics survey was given; after completion, instructions were given orally about completing the study. The user was assigned an order of system usage in a counterbalanced design. Then, the participant was shown how each system worked and given a simple tutorial to accelerate familiarization. Next, the user was asked to clear *Trial 1*[5] with a system. After completion, the participant repeated this process with the other system. A questionnaire was given at the conclusion of the first trial to solicit feedback regarding both systems. The user subsequently commenced the second trial (*Trial 2*), clearing the trial using both systems, and another questionnaire was given to again garner user feedback. Lastly, a final questionnaire was given in order to determine if there was a clear preference between the two systems. Each user was given $15 for their participation time. The study lasted approximately 90 minutes.

User Feedback

We wanted to evaluate if users favor our interface more than the control; our questionnaires were geared towards understanding user perception. We asked our users which

[5] Section *'Prototype & Experiment Design'*, *'Experiment Tasks'*

method was more easy to learn, intuitive, fun, likable, useful, and easy to use. Additionally, we asked the user to pick which system was better for various factors - easy completion, intuition, usability, comfort, fun, and overall. We also recorded the total time it took to complete each trial for both systems, as well as the errors made during use.

Results & Discussion

Qualitative Analysis

For *Trial 1* (easy), our participants seemed to favor our interface over the control. We performed Wilcoxon Signed Rank Tests to find statistical differences, and we used Holm's Sequential Bonferroni Adjustment method [Hol79] to control Type I errors. We found statistical differences for each of our measured questions except for "Easy to Learn" which has no significant difference (Table 4.1). We repeated this analysis with *Trial 2* (hard), and statistical differences were found between our interface with the control for all questions (Table 4.1).

We performed χ^2 tests on our final post-questionnaire results in order to determine if there were statistical differences and again used Holm's Sequential Bonferroni Adjustment. We found a significant difference for Intuition ($\chi^2(1, N = 22) = 8.909, p < .05$) and for Ease of Use ($\chi^2(1, N = 22) = 6.545, p < .05$). Final results are shown in Table 4.2.

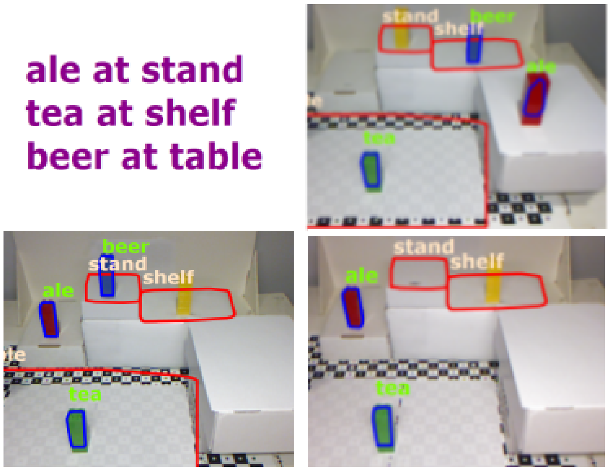


Figure 4.7: Three different task behaviors needed in order to accomplish the same goal (top-left)

Table 4.1: Statistical Tests

	Trial 1		Trial 2	
Question	Z	*p*-val	Z	*p*-val
LabelEasy	-2.264	.024	-1.991	.047
SetupEasy	-3.048	.002	-3.216	.001
LearnEasy	-1.496	.135	-2.488	.013
Intuitive	-3.078	.002	-2.801	.005
Fun	-2.953	.003	-2.445	.014
Liked	-3.756	.000	-3.345	.001
Easy	-2.776	.005	-3.497	.000
Useful	-2.632	.008	-2.919	.004

Table 4.2: Post-Survey Statistical Tests

Question	χ^2	df	*p*-val
Easier Completion	6.545	1	.011
More Intuitive	8.909	1	.003
Easier Use	6.545	1	.011
Comfort Use	2.909	1	.088
Fun Use	0.727	1	.394
Final Pick	4.545	1	.033

Task Completion Time

We performed a Paired Samples T-Test to compare mean task completion times (Figure 4.8). For *Trial 1* (easy), there was a significant difference between the control (Desktop : M = 23.65, SD = 5.51) and our interface (Tablet: M = 12.3, SD = 2.75); $t(19) = 11.083, p < .001$. Significant differences are also found for *Trial 2* (hard) tasks between the control (Desktop: M = 22.7, SD = 4.19) and our interface (Tablet : M = 10.4, SD = 3.05); $t(19) = 12.564, p < .001$.

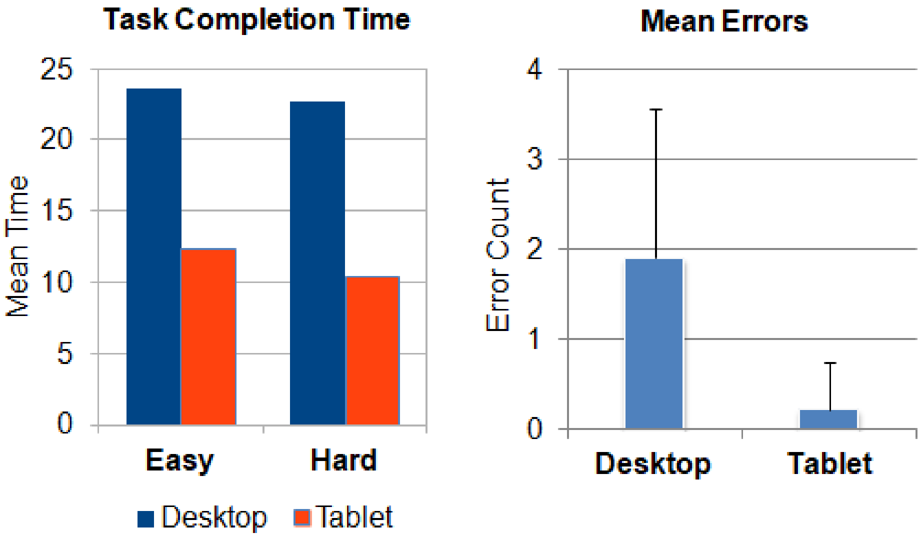


Figure 4.8: *Trial 1* (Easy) & *Trial 2* (Hard) Quantitative Results

Task Errors

We performed a Paired Samples T-Test to compare frequency of errors between the conditions (Figure 4.8). For Trial 1, there was a significant difference between the control (Desktop: M = .909, SD = .92) and our interface (Tablet : M = .136, SD = .35); $t(21) = 3.93, p < .05$. For Trial 2, we again found significant differences (Desktop: M = 1.0, SD = 1.2) (Tablet: M = .09, SD = .29); $t(21) = 3.578, p < .05$.

Discussion

The results expectedly show our interface significantly outperformed and was more consistent across quantitative measures. Besides a smaller mean error (Figure 4.8), it has a small variance compared to the control condition. The cause of the large variance can be attributed to majority of users making careless mistakes in the hard tasks (Trial 2), where the task that was modeled with the control cannot be re-used for subsequent problems, even though the goals remained the same (Figure 4.7). Many users provided less optimal task behaviors than the task planner, though some commented they could do better if allowed to repeat the study. Our interface remained consistent and relatively error-free across usage since the task is modeled as a goal agnostic to structural change. Our interface was well-received by participants, with users finding the interactions easier and more intuitive than the control (Tables 4.1 & 4.2). From feedback, users found no difficulty discerning the differences when instantiating *Object-Object* and *Object-Location* predicates, or when modeling the task. They found the interaction when asserting the additional *Location* relation required ('ale') intuitive and clear. They appreciated the re-usability of the tasks for our interface, especially for the hard tasks (Trial 2).

CHAPTER 5
GRASPIT! MOVEIT! GRASP & MOTION COLLABORATION

Introduction

In chapters 3 & 4, we addressed how symbol grounding with semantic revisions can be encapsulated into abstract sketches. We have shown they can be both intuitive and useful for non-experts to present artifact semantics and appearance in cooperative task[1] modeling. But the collaborative strategy so far with high-level planning (Chapter 4, Section *'Theoretical Framework'*) will not be enough to ensure successful manipulation. Semantics aside, we can observe upon closer examination that 3-D appearance learning is based on segmented, but *partially cognizant* geometries[2] (Chapter 3, Section *'Interface Design'* & Chapter 4, Section *'Prototype & Experiment Design'*). High sparsity and noise further corrupt the scan quality of environment point clouds or depth images with commodity range finders. But robots need to plan manipulator grasps and trajectories through incomplete and degraded geometries at Manipulation Control for any tractable task (Chapter 3, Figures 3.2 & 3.4). In this chapter, we shall explore further abstractions non-technical users can understand and exploit to extend the capabilities of Manipulation Control agents for *recovery* (Section

[1] In this chapter, a task shall be defined as a sequence of manipulator trajectories and grasps.

[2] 'geometries' shall be interchangeable with 'point clouds' or 'meshes'.

'Theoretical Framework'); specifically, we explore interactive *cues* on environment geometries that leverage user intuition, to stimulate further motion and grasp planning activity. The *'Theoretical Framework'* Section also describes a grasp and motion planning pipeline our user collaborates with in order to recover from manipulation stalls[3], discuss our empirical setups, and results of two independent user studies relevant to scenarios of assisting the Continuous Planner (Chapter 3, Figure 3.4).

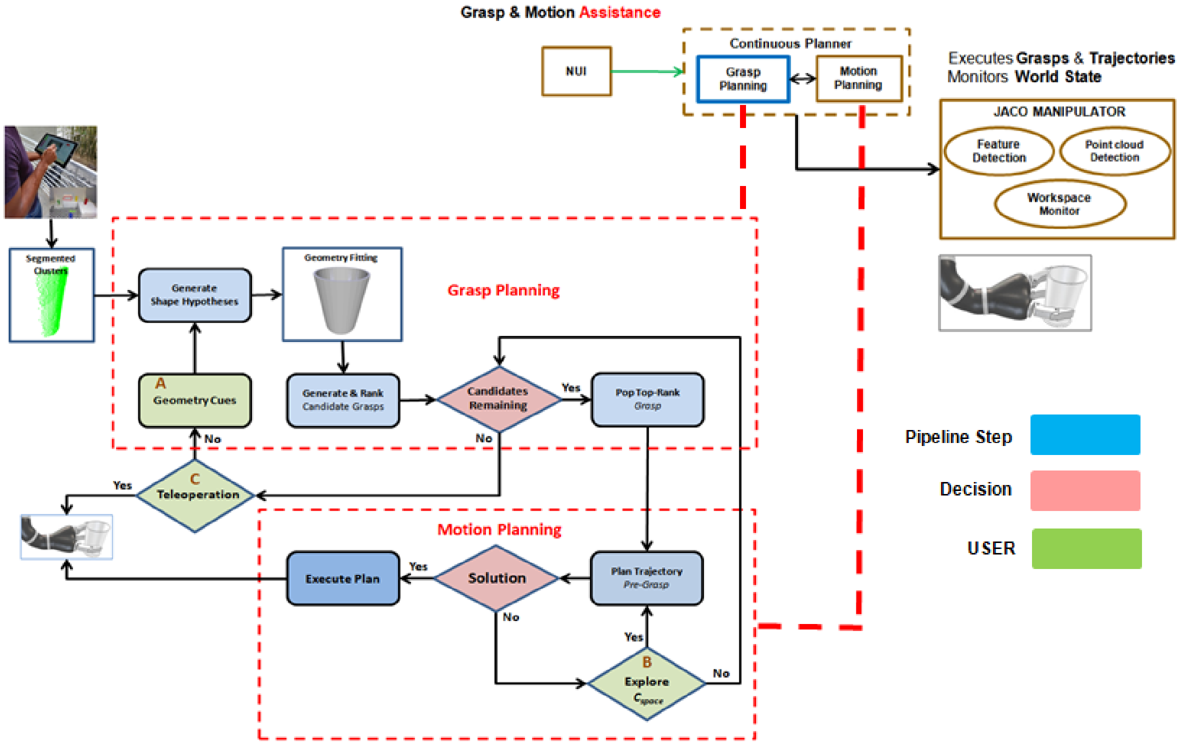


Figure 5.1: Manipulation Planning pipeline inside the Continuous Planner. Process and Decision blocks in green are performed by our 'human-in-the-loop'.
Process Block 'A' : User presents a Shape Hypothesis (Section 5).
Decision Block 'B' : User decides to either allow the pipeline to pop the next candidate grasp, or to select workspace region(s) for preferential C_{space} sampling.
Decision Block 'C' : User decides to teleoperate the grasp, or collaborate on a new shape hypothesis.

[3] Chapter 4, Section *'Interface Design'*, *'Collaboration & Assistance ('Assist')'*

Design Motivation

Continuous manipulation planning for robotics under incomplete observations of environment geometries are predominantly addressed with a fully autonomous, interconnected pipeline of manipulator grasp and motion plans [CCM13], [PGS17] (Figure 3.4). When a feasible grasp posture is found, the *pre-grasp* pose for the posture is offered as the goal for motion trajectory planning.

In **Grasp Planning**, object geometries are important parameters for finding feasible and stable grasp postures. Methods for autonomous geometry completion and fitting can be one or a combination of *extrapolation* [BJL11], [KAE12], [TZC09] and *matching* with 3-D shape features from model databases [FAJ16], [LSW17]. It becomes apparent that autonomous reconstitution destabilize when scanned geometries for grasping are novel, sparse, or ambiguous. Further, locating a stable grasp would be irrelevant when **Motion Planning** is unable to return a feasible manipulator trajectory to the pre-grasp pose. Our approach is for users to participate in the Manipulation Planning pipeline (Figure 5.1), providing *environment cues* that stimulate geometry extrapolation, and configuration space (C_{space}) exploration (Section *'Collaborating with C_{space} Exploration in Motion Planning'*).

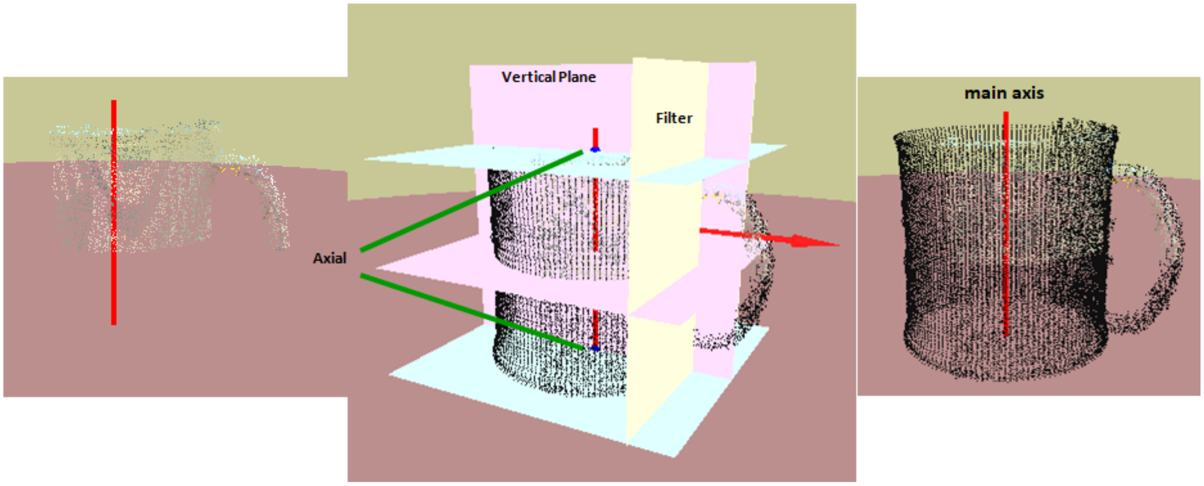


Figure 5.2: (***Left***): A simulation of a noisy, occluded, and sparse point cluster of the *'White-Cup'* model from the KIT object models database.
(***Center***): Example of a 4-tupled shape hypothesis; aside from the filter plane, the tuple expresses a vertical plane reflection, horizontal plane reflection, and a *bounded* axial symmetry along the main axis.
(***Right***): The result when the hypothesis is applied to the /textitleft point cloud.

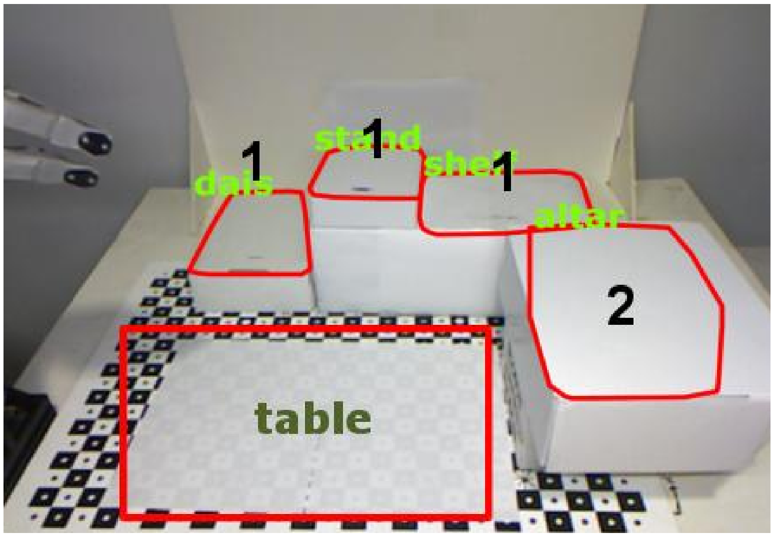


Figure 5.3: Artifacts assigned as *Locations*-Only. Numbers indicate available capacity for each location. Figure is an exact copy of Figure 4.2, reproduced here for convenience.

Theoretical Framework

Geometries obtained after interactive symbol grounding (Chapter 3, Section *'Interface Design'*) or through automated recognition can be novel, sparse, noisy, and ambiguous at the same time. Factors, to name a few, such as clutter occlusion [PSB11], sensor noise, sampling rates, robot mobility and time constraints [QAC16] can limit the quality and size of the scans. Without necessary prior information, reasonable shape hypotheses for geometry-fitting [TW05], [BJL11], [MTG18], [VKW18] are nearly impossible for Grasp Planning to predict. However, it may be possible for users to contribute estimated hypotheses that leverage their understanding of the object's appearance. The user hypotheses can be further refined and validated against surrounding scene geometries by the Grasp Planner before Geometry Fitting. When the Motion Planner (Figure 5.1) fails to locate a feasible trajectory to the pre-grasp or, in fact, any destination pose in a motion plan, teleoperation could be used to move the arm into more amenable configurations (Chapter 6) for a motion re-plan. But, it is possible to avoid teleoperation by having the user, instead, to predict highly-probable regions of configuration free (C_{free}) space to explore alternatives.

Continuous Planning

Our extended Manipulation Pipeline inspired from Quispe et.al. [QAC16] resides in the Continuous Planner (Figure 3.4), illustrated in Figure 5.1. Upon receipt of segmented points or depths, the following steps occur automatically if there were no stoppages requesting user intervention;

1: **Generate Shape Hypotheses**; Shape hypotheses, such as Composite Symmetries[4] [TW05], [BJL11], Extrusion [KAE12], [QMG15] and Medial Axes [TZC09], [VKW18] are defined from existing geometries to extrapolate further spatial primitives and normals. Filtering [WKZ16] and collision tests may be performed to validate and refine the cloud further.

2: **Geometry Fitting**; Mesh prediction methods, such as Poisson Surface Reconstruction [KBH06], Superquadrics (SQ) [QAC16], [MTG18], convex-hulls, or even primitive shapes (cylinders, spheres, cuboids) [MKC03] may be used to solve the optimal representation of the shape.

3: **Generate & Rank Candidate Grasps**; The Fitted Geometry and surrounding scene geometries are sent to a grasp planner. Candidate grasps for the fitted geometry with a *specific* manipulator arm are modeled, ranked, and enqueued according to a chosen stability metric, such as *force-closure* [SEB12]; combinations of metrics which can include rating the arm trajectory approach besides the grasp posture may also be used [QAC16].

4: **Pop Top-Rank Grasp**; The highest-ranked grasp is removed from the queue and sent to the Motion Planner.

[4] Combinations of two or more basic symmetries. For example, a cube has three orthogonal plane reflection symmetries.

5: **Plan Trajectory Pre-Grasp**; If a feasible manipulator trajectory to the pre-grasp is found, the entire grasp and motion plan becomes executable. Otherwise, the same evaluation is repeated with the next-highest ranked grasp from the queue.

The pipeline terminates when either a feasible manipulation plan was found, or when the candidate grasp queue is empty. If the latter condition occurs, the user can either opt to teleoperate the grasp (Chapter 6), or collaborate with grasp-planning further on the shape hypothesis (Section *'Collaborating with Geometry Cues in Grasp Planning'*) to provide another mesh prediction for the next cycle (Figure 5.1: Process Block **'A'**).

Collaborating with Geometry Cues in Grasp Planning

Human cognition has a definitive advantage over machines in mitigating ambiguities and missing data when perceiving the same geometries. Psychological studies suggest that humans are able to predict obscure geometries through *controlled scene continuation*, completing the hidden regions by visual propagation and generalization of the known geometries [BF05], [BJL11]. It is worth mentioning that methods for the automatic completion[5] of grasp geometries are advancing rapidly, projecting better results with each iteration. However, the literature have also reported the limitations that exacerbate a poor fit (and subsequent candidate grasps) are attributed mainly to inaccurate spatial predictions from the shape

[5] We established that grasp geometries are processed through a two-stage pipeline; (1) Generating Shape Hypotheses for further spatial information (Section *'Continuous Planning'*, Step 1:), (2) Fitting an Optimized Shape (Section *'Continuous Planning'*, Step 2:)

hypotheses [BJL11], [KAE12], [CZS13], [QMG15], [FAJ16], [LSW17]. Importantly, it suggests that cooperative modeling on the hypotheses alone is a sufficient condition for users to have significant influence in a grasp plan. We validate the use of *symmetry cues* in the collaborative model with the following design considerations:

1: **High Affordance**; Norman defines an *affordance* to be "fundamental properties that determine just how the thing could possibly be used" [Nor02]. In our context, user interaction should be *afforded* to express hypotheses about the shape when they encapsulate *common* and intuitive descriptors that require little to no formal training. The human cognitive ability of scene continuation allows a user to acquire understanding of an appearance through the object's local, *global*, and volume-based descriptors [BF05], [BJL11]. When presented evidence of similar descriptors in *different* but partially hidden objects, the invisible regions are predicted, or extrapolated, according to the visual evidence and *completion rules* gained through prior visual experience. A global descriptor, or *cue*, that is commonly expressed in many household objects is the concept of symmetry [TW05]; a user can express a shape hypothesis with cues that indicate symmetries of a household object.

2: **Exhibit Low Entropy**; The cue(s) should express the least-likely occurring hypotheses (of scanned geometries) that hold the most spatial information. Object symmetries and medial axes [Blu67] are examples of shape descriptors with low entropy that can extrapolate global information about the shape [TZC09], [CC12]. A user could point out occurrences of ambiguous, *composite* symmetries along the main axes[6] of a scan which seem implausible to robots. As illustration with Figure 5.2, the occurrence of two orthogonal plane reflection symmetries *and bounded* axial symmetry on a point cloud (*left*) can be expressed with three points and a normal from the main axis (*center*), resulting in the modified cluster (*right*). Extracting the main axes can either be automatic [TZC09], [KAE12], [QMG15] or with user modeling [CZS13], [HDT07]. If the main axes are indeterminate themselves, the 3-sweep technique [CZS13] can be offered to model a hypothesis for the main axis; two sweeps joining three spatial points that define the planar base for the axis, while the optional third sweep determines the extent. On a side-note, 3-sweep is constrained by the modeling range of generalized cylinders and cuboids, rendering it inadequate to offer hypotheses outside the range, such as composite symmetries.

The estimated hypotheses from a user are offered to the Grasp Planner *before* the generation of their own hypotheses (Figure 5.1: Block **'A'**). Grasp Planning may handle the hypotheses with one or more of the following options;

1: **'Seeding'**; The estimated hypotheses may be used to initialize the parameters of automatic methods that generate closely-similar hypotheses. Examples from past literature include searches for the dominant symmetry, extrusion axes, and medial axes [TZC09], [BJL11], [KAE12], [QMG15], [VKW18].

[6] It can be established that the main axes are *extensions* from their Medial Axis Transforms (MAT) [MGP10]

2: **Validation**; The estimated hypotheses may be used as similarity models [OMT03] to refine or validate the spatial predictions of the automated shape hypotheses.

3: **Shape Hypothesis**; The estimated hypotheses may be used either on its own, or included with the automated hypotheses, to extrapolate further spatial primitives for Geometry Fitting.

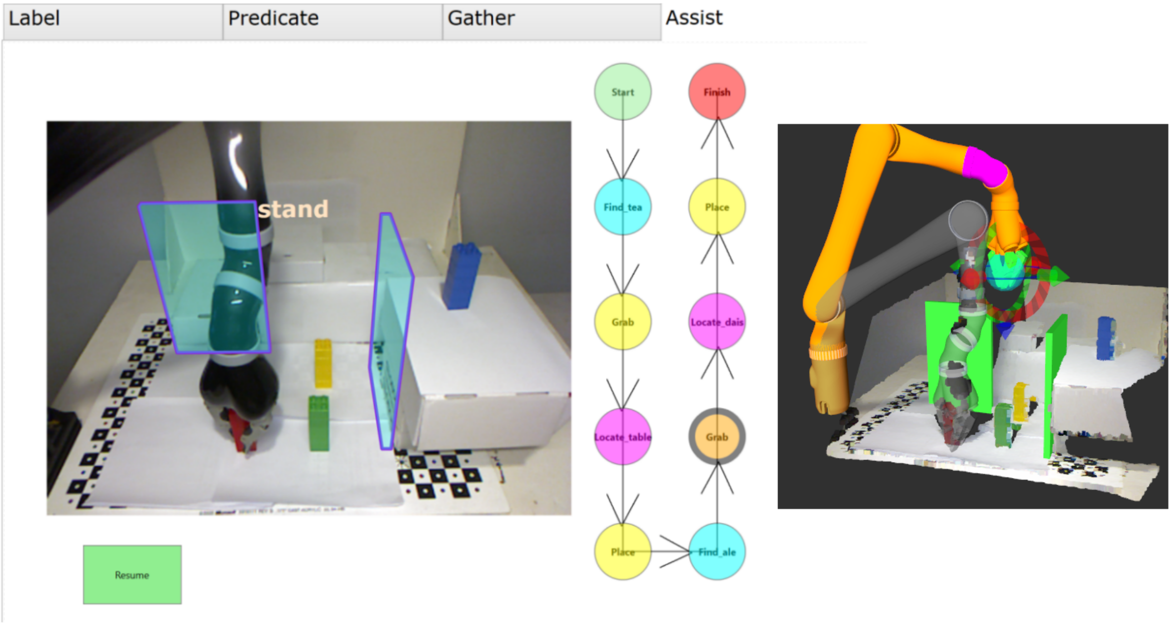


Figure 5.4: Example of a simulated stall: Blocking virtual obstacles prevent the motion plan from plotting a trajectory between experiment locations ‘table’ and ‘dais’ (Figure 5.3)
(***Left***): ‘stand’ selected as a waypoint; the space above ‘stand’ generates further C_{space} samples for the search-tree.
(***Center***): Highlighted node displays the current task status when the stall occurred.
(***Right***): Interactive markers are used to adjust the end-effector towards ‘dais’.

Motion Planning in $\boldsymbol{C_{space}}$. The pose for an n degrees-of-freedom (DOF) robot manipulator, or arm, can be represented as a *unique* point in an n-dimensional space called the **configuration space** (C_{space}) [LW79], [Car06], [ES14], [DCQ16]. It is possible for the *end-effector*, or hand, to maintain the *same* pose in an Euclidean workspace under *multiple* configurations. Using the JACO arm to illustrate, the JACO has six joints and links (excluding the fixed-base) with the last being a three-fingered hand (Figure 5.4). Assuming the base link stays in the same position with the arm never fully extended, a configuration specific to the JACO arm can be represented as a single **point** in a 6-dimensional C_{space} as a tuple of joint-angles[7]. It is possible for many points to appear in C_{space} that each maps to the *same* workspace orientation and position of the hand. In context with the Manipulation Pipeline (Section *'Continuous Planning'*), it is worth mentioning here that it is possible for the pre-grasp pose of each candidate grasp to have multiple feasible configurations in C_{space} as well. Therefore, we state the C_{space} of a JACO arm to be the set of *all feasible* configurations within joint angular limits. The subset of C_{space} that avoids collision with obstacles in the workspace is the configuration free space (C_{free}), while its complement becomes the obstacle space (C_{obst}). Hence, the general motion-planning problem becomes the search for a trajectory of configurations inside C_{free} that links between the current configuration of the arm to a goal configuration [DCQ16].

[7] The angle between two links on either side of a joint in a robot arm

Trajectory in $\boldsymbol{C_{free}}$. State-of-the-art motion planning occurs through a **sampling** approach inside C_{free}. For a more detailed review of Sampling Based Planning (SBP) research, interested readers are forwarded to Carpin [Car06], Elbanhawi [ES14], and Denny [DCQ16]. In general, a graph, or search-tree, that includes the start and goal configurations as connected nodes is grown from random samples of C_{free}. The graph becomes an implied *roadmap* of the real workspace which can be queried for trajectories expressed as a sequence of configurations. However, the probability of finding a successful path through the graph is dependent on the sampling strategy and its *expansiveness*; valid trajectories can be missed when insufficient samples are taken, or when too few connections are made due to poor visibility[8] between neighboring nodes in the graph [ES14], [DCQ16].

Cooperative $\boldsymbol{C_{space}}$ Exploration. When no motion plan is returned for a trajectory to the pre-grasp (Figure 5.1: Block **'B'**) or any goal destination, a common occurrence is attributed to under-sampled C_{free} at key workspace regions between the current configuration of the arm to the goal [DSA18]. Sampling is not efficient in cluttered, or obstacle, regions[9] of the workspace, resulting in poor visibility between C_{free} samples taken around the obstacle space. To get around the problem, a human-in-the-loop could teleoperate the hand directly to the goal (Chapter 6); a more amenable second approach that reduces teleoperation is to move the hand nearer to the obstacle region and resume planning. The second approach improves the sampling probability of C_{free} around C_{obst} [RTL06], growing further feasible

[8] A pair of configurations are defined as *visible* to each other when they can be connected in C_{free}

[9] The *narrow passage* problem [DCQ16]. Cluttered regions form narrow, but passable, passageways between two sparse regions.

configurations into the roadmap that improve the odds of finding a path. Inspired by the success of human-in-the-loop methods reported in Kent [KSC17] and Denny [DSA18], the third approach is for users to leverage labeled artifacts[10] in the workspace as *way-point* recommendations. Instead of teleoperating to the obstacle region as in the second approach, a user offers the artifacts that are close enough to the obstacle region, influencing the planner to concentrate C_{free} sampling around the artifacts.

As an illustrative example with our interface from Chapter 4 (Section *'Prototype & Experiment Design'*, *'Interface Setup'*) and Figure 5.4, a simulated manipulation stall prevents the JACO arm from placing 'ale' (red ego block) at 'dais' (Figures 5.3 & 5.4). The user has three options; (1) Teleoperate the place action (Chapter 6). (2) Teleoperate closer to 'dais' and resume motion planning, or (3) Offer the 'stand' artifact as a way-point (or region for C_{space} samples) and resume motion planning. For Option (3), the artifact(s) can be offered either by tapping the labeled artifact on the image or in the Model Pane (Figure 4.3). Teleoperation can alternatively be performed with interactive markers[11] (Figure 5.4).

[10] Chapter 3, Section *'Interface Design'*, *'Symbol & Concept Grounding ('Label')'*

[11] The interactive markers for teleoperating the JACO arm is a simple 6-DOF 3-D widget attached to the hand. It allows a user to control translation and rotation of the hand in the workspace by clicking and dragging along the colored gimbals (Figure 5.4).

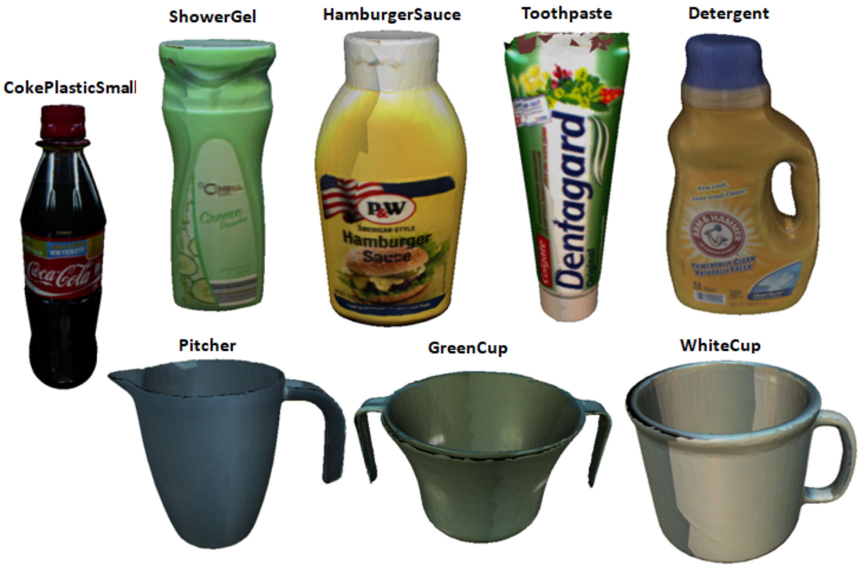


Figure 5.5: Selected household objects from the KIT object models dataset [KXD12] & BigBIRD dataset [SSN14]; *'CokePlasticSmall'* is used for a tutorial before the actual study commences.*'GreenCup'* was modified with an extra handle. Only 'Detergent' is retrieved from the BigBIRD dataset.

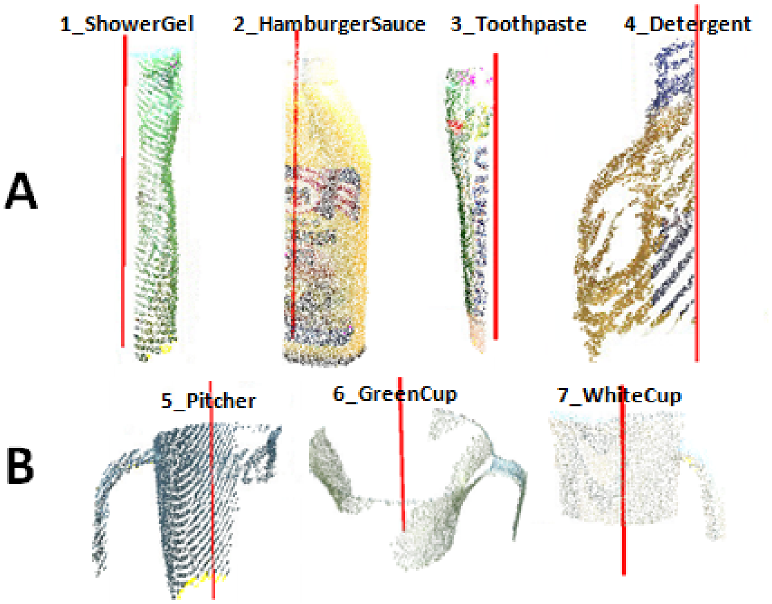


Figure 5.6: Single-view depth measurements [BRH14] with simulated noise and occlusion are presented with their main axes. Tasks are divided under groups 'A' & 'B'; the number prefix at each name indicates the order of point clouds presented to the user.

Study Design: GraspIt!

The goal of our study was to evaluate how non-experts would perceive using symmetries to describe 3-D shapes, and whether it can be effective as a collaborative model to a grasp planner. We posit that a user with little to no experience about 3-D modeling would favor describing object shapes as a hypothesis of symmetry descriptors. We begin with our assumption that the user provides recovery assistance through a pick-and-place manipulation of a common household object. Though it is possible for our 'human-in-the-loop' model to scale (Figure 5.1), the assumption helps to shape the remaining study design considerations. To answer the question whether it could be effective with automatic methods that complete grasp geometries, we consider how the user hypotheses shall be handled by the pipeline; i.e. whether they would be utilized for 'seeding', validation, or as a resource for spatial primitives. The last option was selected (i.e. as a resource) as it exhibits method independence. Thus, we would further evaluate *similarity* of extrapolated spatial primitives to a ground truth.

Shape Hypothesis

We state the following elemental and *dominant* symmetries to be common in many household objects [KXD12]; (1) Plane reflection, and (2) Axial [TW05], [BJL11], [KAE12], [CZS13], [QMG15]. We establish that the symmetries are dependent on the object main axis (Section *'Collaborating with Geometry Cues in Grasp Planning'*). Many household objects have at least one dominant symmetry, but it is also not uncommon to find composite symmetries; besides the three-orthogonal plane reflection symmetry (e.g. a cube), two other common composite symmetries are a dual-orthogonal plane reflection symmetry (e.g. Figure 5.2, 'WhiteCup' is modeled along one of the vertical plane reflections with a horizontal plane reflection), and the axial with single-orthogonal plane reflection symmetry [TW05]. Hence, a shape hypothesis could be expressed with a tuple that holds one or more of the following descriptors; (1) a set of vertical plane reflections, (2) a set of horizontal plane reflections, and (3) an axial symmetry. The main axis is implied in the hypothesis given that it is a *necessary* requirement. In the study, each point cloud will be provided the main axis to support the expressed symmetry descriptors (Figure 5.6).

From a user perspective, symmetry could be confusing as they apply *globally* on an object. For example, a user may indicate an axial symmetry to an incomplete 'WhiteCup' point cloud (Figure 5.2 (*Left*)) though it should apply only at the cylindrical cluster. A *filtering* definition can be included to the hypothesis that partitions out geometries where *certain* symmetries do not apply. In this user study, we shall simplify the expression to one or more of the following descriptors; (1) a vertical plane reflection, (2) horizontal plane reflection, (3) bounded axial symmetry, and (4) a vertical plane filter. The filter plane (Figure 5.2) separates, or *cuts*, away geometries that are exempt from the axial symmetry rule (e.g cup handles), and also from a vertical plane reflection if their planes are orthogonal.

A user is not restricted to express further hypotheses on resulting point clouds if they wish to extrapolate further.

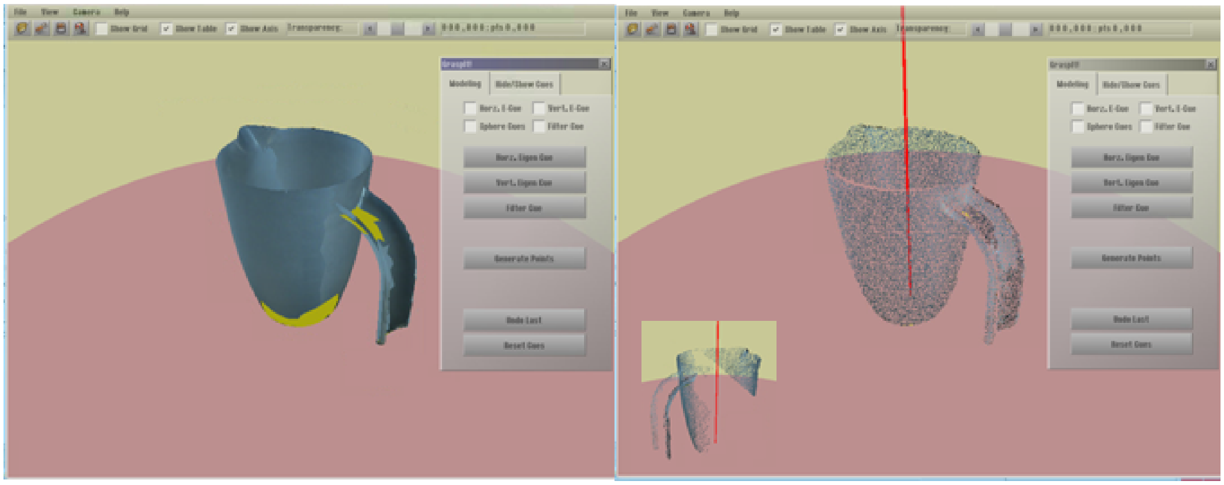


Figure 5.7: (***Left***) : The *'pitcher'* mesh (KIT) is loaded to the virtual platform and rests on table-top by default. The camera view is the implied parameter to the single-view depth measurement.
(***Right***) : The result point cloud after the simulated measurement completes.

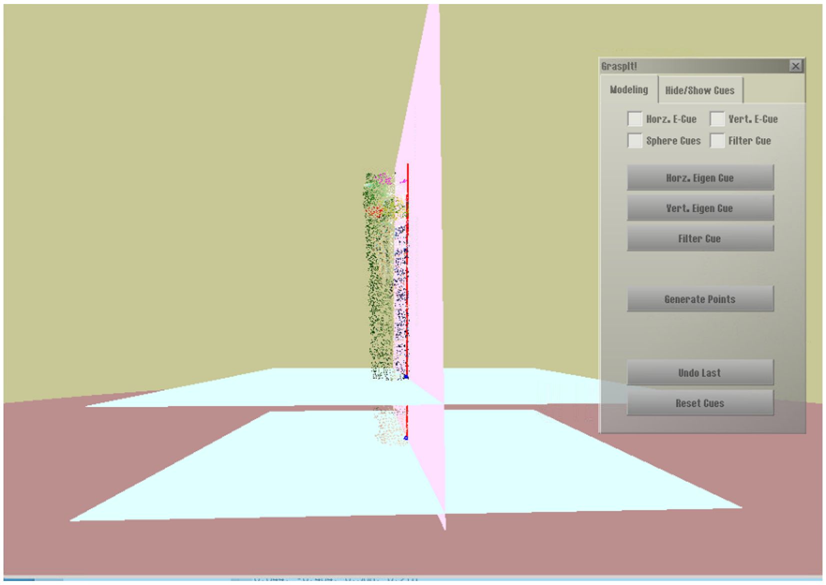


Figure 5.8: Modeling a shape hypothesis describing *'Toothpaste'*; a vertical plane reflection, and axial symmetry *bounded* between two points (planes) along the main axis.

Experiment Setup

A set of 7 point-clouds was prepared from the KIT object models database [KXD12] and BigBIRD dataset [SSN14] (Figure 5.5), each presenting an incomplete point cloud that a user presents hypotheses for extrapolation during the study. We developed a virtual platform for the dual-purpose of preparing the simulated depth measurements (Figure 5.7) and an interface a user leverages to explore and express hypotheses (Figure 5.8). After a depth measurement, Perlin noise is introduced and one-third (33%) of the points are randomly removed (Figure 5.6). To simulate simple occlusion, half the remaining cloud was removed along a major eigenvector [BJL11] (e.g. lower-half of 'WhiteCup', Figure 5.2); but sufficient precaution was taken to ensure the cloud maintained key textures that hold color cues for a user to locate where major symmetries could reside.

Subjects

To evaluate our shape hypothesis *abstraction* with non-experts, we recruited 12 participants from outside major fields in Science, Technology, Engineering, and Mathematics (STEM); the average age was 24.8 with a standard deviation of 3. There were 9 females and 3 males. On a 7-point Likert scale, the sample reported very low experience with 3-D computer graphics (M = 1.33, SD = 0.49), 3-D modeling or editing (M = 2.25, SD = 1.3), and 3-D printing (M = 1.08, SD = 0.25). Only 3 knew at least one programming language on a novice level, and the mean experience with the most familiar was 3.3 (SD 1.15). As for robot-related experience, the sample reported very low experience with manipulators (M = 1.33, SD = 0.65) and software (M = 1.5, SD = 1.17).

Experiment Tasks

After a tutorial, the user would be presented further point clouds in sequence illustrated at Figure 5.6. Excluding the tutorial, tasks can be divided into two groups, 'A' & 'B'; the first four make up 'A' and remaining 'B'.

Point clouds from 'A' are dual-orthogonal vertical reflection plane symmetries; repeated expressions of vertical plane symmetries should suffice under normal circumstances, including point cloud '4_Detergent' when the filter plane is used on the handle geometries. However, it would be interesting to observe whether a user could refine the hypothesis further with an axial symmetry; according to the entailment hierarchy [TW05], an axial symmetry imply the presence of plane symmetries, but unlike planar reflections, a circle of points around the main axis extrapolates instead of a single reflected point across the plane. For example, a bounded axial symmetry, illustrated in Figure 5.8, can be included to describe point clusters at '3_Toothpaste'; it can also be applied to the bottle-cap clusters at '2_HamburgerSauce', and '4_Detergent'.

Without handle geometries, point clouds from 'B' are essentially axial symmetries; and users are informed of the *mandatory* requirement to express them. Interesting tasks include ‘GreenCup’ modified with an extra handle, and ‘WhiteCup’ that require both vertical and horizontal reflection planes.

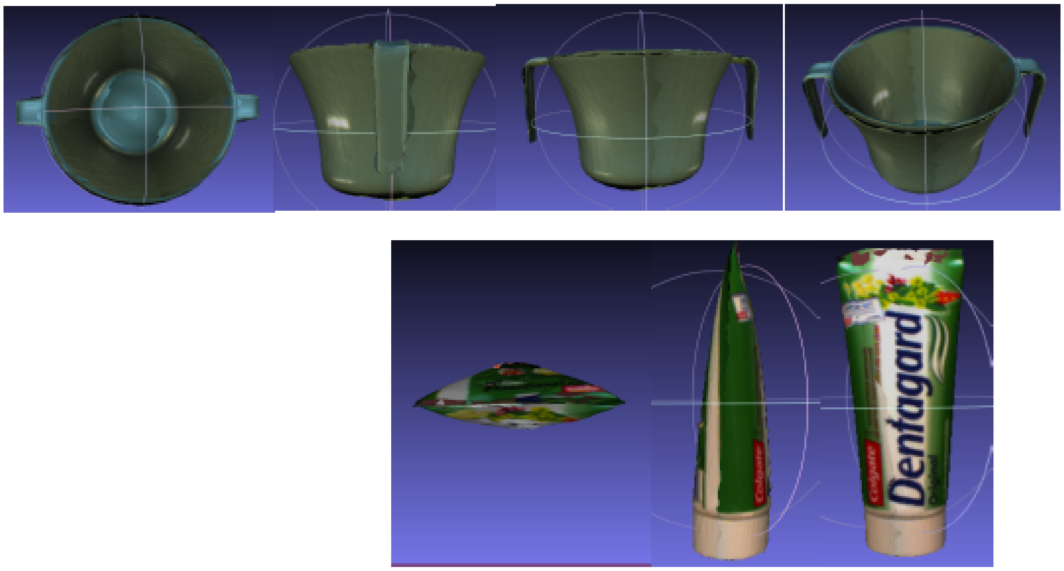


Figure 5.9: ***(First Row)***: Top, Side, Front, and Perspective projections of ‘GreenCup’ ***(Second Row)***: Top, Side, & Front projections of ‘Toothpaste’

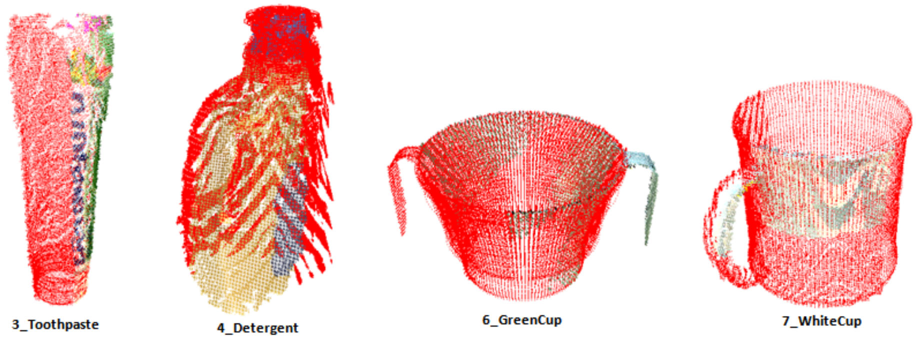


Figure 5.10: Sample user point clouds with extrapolated vertices (red).

Study Procedure

Our recruited participants reported to our laboratory and presented with the virtual platform (Figure 5.7). A demographics survey was given; after completion, instructions were given orally about completing the study. The user was given a short tutorial that introduces the platform, object scanning, and the interface in order to express a hypothesis; they were introduced to the concepts of vertical reflection, horizontal reflection, (bounded) axial symmetry, and a filter plane. They were shown how to express each descriptor, and extrapolate according to the expressed descriptor(s). They were also shown how to undo a previous extrapolation, if necessary, and modify the descriptor(s). The user is provided with 2-D snapshots of the objects' top, front, and side projections (Figure 5.9). The user is further informed there are no time limits to complete the task, multiple extrapolations are not restricted, and that the scanned object is resting on a table-top. After completing the tasks, a questionnaire was given to solicit feedback regarding the interface and using the descriptors to extrapolate shape. Each user was given $10 for their participation time. The study lasted approximately 40 minutes.

User Feedback

We wanted to evaluate if users favor expressing symmetries as a shape model for an object; our questionnaires were geared towards understanding user perception. We asked our users whether the descriptors were easy to learn, intuitive, useful, coherence with real objects, and likable.

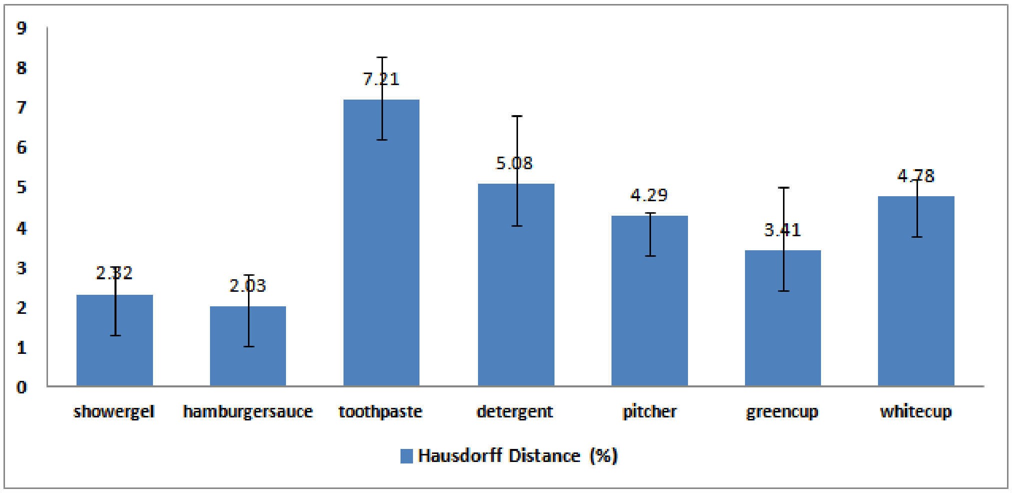


Figure 5.11: Normalized Hausdorff distances in percentage values (Section 5).

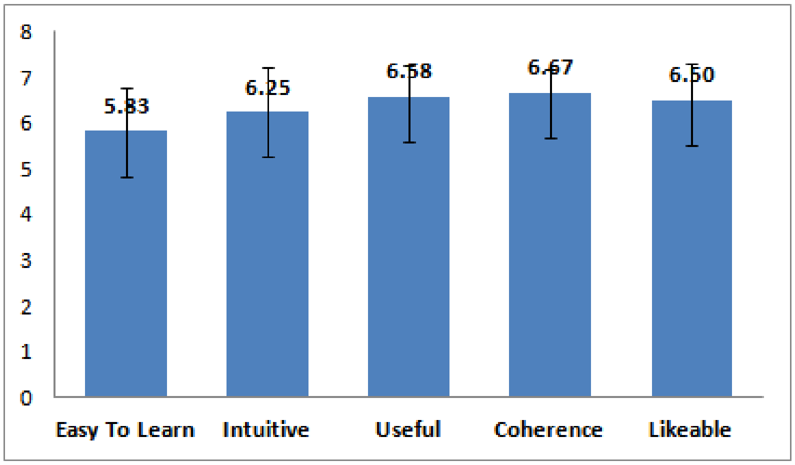


Figure 5.12: Participant responses to the use of symmetry descriptors as a shape hypothesis.

Before post-processing, point clouds were down-sampled and filtered to remove any duplicate and statistical outliers. To determine similarity of the point clouds with ground truth models in KIT, we use the Hausdorff distance metric to compare between their convex-hulls. As the depth measurements were simulated, the ground truth and corresponding point cloud share a common coordinate system and transformation. Hausdorff distance values between the two convex-hulls are normalized afterwards with the diagonal length of the combined models' oriented bounding box. The normalized minimum and maximum Hausdorff distance values are 2.03% and 7.21% respectively (Figure 5.11), with the mean normalized Hausdorff distance at 4.16%.

From our observations and user feedback, participants learned quickly and became more familiar with each task progression about the descriptors required for aesthetic results. Five of the twelve participants expressed the bounded axial symmetry for the bottle cap clusters at the second task ('2_HamburgerSauce'), eleven participants at '3_Toothpaste', and eight at '4_Detergent'. All twelve managed to complete the last three tasks with little prompting.

The interface was very well received (Figure 5.12); participants recognized the ease, speed, and utility of *global* shape descriptors; a participant remarked "Easy to use once you get the hang of it. Generates objects fast", with another summarizing "how easy it was to pick it up and use it. Was also fast to do, especially if you're very familiar with it." Participants found the descriptors fun and useful for people who are 3-D modeling and computer graphics novices; as one participant summarized, “I like that it created 3D objects for me in real time. As someone who struggles with technology, after some guidance most of the models were doable”, one participant remarked that “It quickly became sort of like a matching game and I got better quickly.” Some participants provided further comments about features they would prefer, for example, support for more than one filter plane for symmetry exclusions, and different-colored (layered) extrapolated vertices with each hypothesis iteration.

Discussion

Our shape hypothesis abstraction, leveraging human intuition directly into the expression of symmetries, provides a proof-of-concept model for non-experts to collaborate spatial information with a grasp planner. Though it was cast and tested as a spatial information resource in the user study, it can be re-applied either as validation or to refine automated hypotheses; as the shape was expressed with positioned *cues*, it can also be re-positioned by automated methods for seeding similar hypotheses. The main limitation of the model is its dependence on the object main axis; future iterations for the hypothesis would need to consider optimal interactions that incorporate an axis into the hypothesis should it become indeterminate through automatic means.

Study Design: MoveIt!

The goal of this study was to evaluate the collaborative model with motion-planning (Figure 5.1, Block 'B') by stimulating C_{space} exploration (Section *'Collaborating with C_{space} Exploration in Motion Planning'*). The 'Blocks World' empirical set-up (Chapter 4, Section *'Prototype & Experiment Design'*) is re-visited again to perform the study. Readers are referred to the user study performed at Chapter 4 for a review of the task domain, infrastructure, and our prototype.

A main consideration for the study design is to minimize *Grasp Planning* effect on user perception and preference to the model (Figure 5.1). Effectively, Grasp Planning effect is removed when the pipeline is ***shorted***; we shall state the following design specifications;

1: **Perfect Grasp Posture**; There is only one solution grasp posture available from the queue.

2: **Probabilistic Complete Motion Plan**; A trajectory solution to the pre-grasp can be found within *reasonable* bounds.

Both specifications provide (i) *Localized* interaction only at the Motion Planner (Block 'B'), and (ii) Avoiding grasp teleoperation (Figure 5.1, Block 'C'). The 'Blocks World' manipulation pipeline can simulate these requirements, facilitating a simple working grasp posture and motion planning (ROS *MoveIt!*) between artifacts.

Empirical Study & Tasks

Three simple 'Blocks World' manipulator tasks (Figure 5.13) are prepared by a proctor before the study commences. All necessary artifacts in the domain are present in the Model Pane (Figure 4.3). A total of six simulated manipulator stalls belonging either to a 'pick' or a 'place' action shall occur during the runs. We performed a within-subjects user study that compares the collaborative motion planning model with a teleoperative control condition; a user shall be instructed to either teleoperate closer to the destination with interactive markers (Figure 5.4) to alleviate motion planning (control condition), or use the Assist Panel (Chapter 3, Figure 3.6) to select a *Location* artifact (way-point) for preferred C_{free} sampling, in a counter-balanced design.

Subjects

Kent et al. [KSC17] posits the ring-and-arrow approach of interactive markers to be unproductive and mentally demanding in teleoperation; a user has to adjust both the 3-D camera view to the target and transformation of the 6-DOF markers in order to teleoperate with reasonable precision. However, the use of 6-DOF markers is commonplace for teleoperative and navigation tasks in open-source robotics [QCG09], [LHC12], [CHL12], [CCC13]. To mitigate a possible bias against the control, we decided to sample participants within STEM with little to no experience about robot manipulators. On a 7-point Likert scale, our users reported very low experience with robots (M = 1.3, SD = 1.0) and medium experience with pen/touch interfaces for leisure (M = 3.6, SD = 2.0) and for work (M = 3.1, SD = 2.1). However, 11 users knew at least one programming language, and the mean skill level with their most familiar language was 4.2 (SD 1.7).

Study Procedure

The user was given a quick tutorial about the task domain, control, and interface. Then, the user proceeds to execute three different 'Blocks World' manipulator tasks, in order, prepared by the proctor beforehand. As the robot arm performed the actions for each task, a simulated stall occurred in a total of six occasions across the tasks. Each time, the user was asked either to select a way-point through the Assist Panel (Figure 3.6) or teleoperate with interactive markers, in a counterbalanced design. After all the exceptions have been rectified, a questionnaire was given in order to help understand which technique was more preferable when resolving the stall. Each user was given $10 for their participation time. The study lasted approximately 45 minutes.

User Feedback

We wanted to evaluate if there were preference disparities between Interactive marker teleoperation and way-point techniques. We asked our users to rate each method in the factors of frustration, comfort, easy to learn, intuition, fun, likability, helpfulness, and easy to assist. Additionally, we asked the user to pick which method was the best for various factors - easier to assist, intuition, frustration, comfort, fun, and overall. We also recorded the total task completion time required to resolve the simulated stalls for each technique.

Qualitative Analysis

We performed Wilcoxon Signed Rank Tests to help understand if one technique was received better by our participants. We used Holm's Sequential Bonferroni Adjustment method to control Type I errors. We found significant difference ($p < .05$) between Interactive marker teleoperation and way-point technique for Frustration, Learning, Helpfulness, and Easiness (Table 5.1).

We also performed χ^2 tests on our final survey data points and again performed Holm's Sequential Bonferroni Adjustment. We found a significant statistical difference between selection frequencies for the "Less Frustrating" and "Comfort" questions, in which the way-point technique was selected more often. It also garnered all votes from our participants from the "Easier to Assist Robot" question. See Table 5.2 for statistical values.

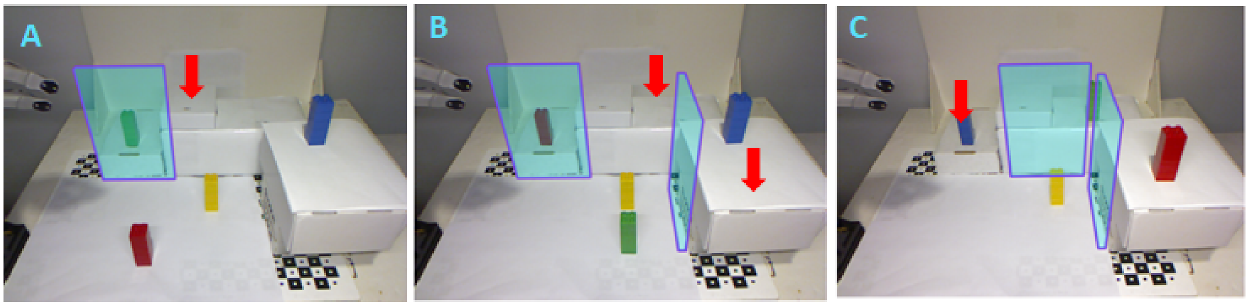


Figure 5.13: Virtual obstacle arrangement from Tasks 'A' (left) to Task 'C'. Red arrows indicate potential regions a user selects for further C_{free} sampling.
(Task A): User selects the artifact 'stand' in order for motion planning to plot a trajectory to 'tea' (green lego block)
(Task B): User could either select 'shelf' or 'altar' to plot a trajectory to 'beer' (blue lego block)
(Task C): User selects 'dais' to plot a place action trajectory to 'stand'

Table 5.1: *MoveIt!* 1st Post-Survey Statistical Tests

Question	Z	p-val
Frustration	-3.329	.001
Comfort	-2.252	.024
EasyToLearn	-3.114	.002
Intuition	-1.749	.080
Fun	-1.106	.269
Liked	-2.43	.015
EasyAssist	-2.816	.005
Helpful	-3.052	.002

Table 5.2: *MoveIt!* 2nd Post-Survey Statistical Tests. For the "Easier" question, all participants selected the way-point technique

Question	χ^2	df	p-val
Easier Completion	-	-	-
More Intuitive	1.143	1	.285
Less Frustrating	10.286	1	.001
More Comfortable	7.143	1	.008
More Fun	0.286	1	.593
Final Pick	4.571	1	.033

Table 5.3: User comments from the MoveIt! study

Waypoints is way easier to use, it is faster to understand
I liked waypoints because it doesn't need to move the robot manually
I liked the waypoints much better because it was easier for me and quicker
I liked the interactive method to operate the arm. Well, I am a gamer so I love to get subjected to new challenges
Though the interactive marker had more control over the arm movement, waypoints were more easy to use and without worrying about the arm
Conceptually I preferred the interactive mode. I liked having direct control of the arm. I just wish it was smoother
It is easier to use waypoints but interactive markers is more fun

Discussion

Our results expectedly show that users found the interface selecting waypoints for C_{free} sampling much easier to use than teleoperating the robot arm. Using way-points was found to be less frustrating and more comfortable, and it naturally reduced the task completion time significantly, compared to the control technique. An interesting point, however, is that users in this study found the control to be more fun to use. This can be attributed to demographics as well; as the sample is taken from students or graduates of Computer Science and Engineering, the users are relatively experienced with 3D Editing tools, and very experienced with mouse-keyboard Video Gaming. Table 5.3 shows sample comments we collected from our participants.

CHAPTER 6
GUIDING USERS IN SHARED PERCEPTUAL TELEOPERATION

Perceptual Shared Control in Human-Robot Teams

Chapters 2 to 5 present a branch of HRI research about robot-centricity; where intelligent agents are in control to decide the high-level actions needed for a task, while human agents behave as assistants or peers. Chapter 2 also reviewed prior literature and objectives of the dissertation to examine interface requirements, architecture framework, and roles of interactive computing for humans to convey perceptual knowledge and cooperative guidance to robots under a shared autonomy.

This chapter explores the requirements and design of autonomous agents under collaborative scenarios where the roles are reversed; a user teleoperates a humanoid robot for pick-and-place tasks, while autonomous agents assist user teleoperation discreetly from the background. Polar-opposite characteristics between human (cognition) and robot (compute) still apply under this scheme; the human agent teleoperating the robot is highly adept at navigating the robot at reasonably optimal paths to the target object for grasping [DS13], while simultaneously aware of avoiding the obstacles during teleoperation. However, manual teleoperating with high accuracy is difficult due to factors such as viewpoint, human error, and lack of informative tactile feedback, especially when the avateering metaphor is used. Artificial agents can enhance grasp accuracy by arbitrating the teleoperative input and assist without disrupting the user's overall experience and expectation of natural avateering [KPL14].

Robot Teleoperation With Intuitive Metaphors

There has been a significant amount of literature reported on the successful use of intuitive metaphors to teleoperate robots [PKL13], [GS08], [LFP12], [NS11]. Guo et al.[GS08] explored interaction with robots using different intuitive motion controllers such as wiimotes[1], while Lichtenstern et al. [LFP12] explored commanding multiple robots using 3D selection techniques and gestures that do not require any tangible devices. Similarly without any device encumberment, Ng et al. [NS11] explored a falconeering metaphor for interacting with an AR drone, while Pfeil [PKL13] explored five different 3D metaphors that teleoperate the same drone. These works have demonstrated that teleoperation and interaction with non-anthropomorphic robots, when applied through an appropriate metaphor using Natural User Interfaces, can be effective, intuitive, and comfortable. As for anthropomorphically-similar or Humanoid Robots, by inference, avateering could be an effective teleoperating strategy since it leverages on human-robot embodiment similarities.

[1] https://en.wikipedia.org/wiki/Wii_Remote

Humanoid or Manipulator Teleoperation With Avateering

Humanoid Robots possess enormous potential to perform as surrogates in telepresence or teleaction scenarios, especially in-lieu of real emergency personnel for disaster response under hazardous environments [KPL14]. Teleoperating humanoids with little or no device encumberment, such as with skeletal-joint tracking using Microsoft's Kinect[2] or Asus's Xtion depth sensors[3], offer an interesting and cost-effective approach to allow for a more natural, intuitive, and non-obtrusive interface. Additionally, service robots such as the PR2, or any robot manipulator like the JACO arm, are fairly anthropomorphic, providing opportunity for users to teleoperate these robots remotely via avateering.

There are essentially three research directions that address the use of human-motion for humanoid avateering; The first covers primarily human-motion mapping and prototyping [RMM05], [KWL05], [LME11], [PK12], [KB12], using the inexpensive Kinect motion sensor to perform direct markerless human-imitation or manipulator control without device encumberment [NL12], [SC11]. These teleoperation techniques can be stated as puppeteering or avateering [HKB08]. The second direction investigates design of usage metaphors from the Human-Robot Interaction (HRI) perspective [GS08], [NS11], [PKL13], leveraging on latent human intuition to control the robot correctly. Lastly, the final direction investigates the development of intelligent agents under the RAP system (Robots, Agents, and People) [SJP02] to enhance the teleoperator's control of the robot. Especially when teleoperating

[2] https://en.wikipedia.org/wiki/Kinect

[3] https://www.asus.com/3D-Sensor/Xtion

with avateering, user interfaces associated with them are often inadequate, rendering simple manipulation tasks often tedious and sometimes impossible. Assistive agents can help users by predicting their intent, and using the predictions to augment their input into the robot [DS13]. However, it is non-trivial to design a teleoperation strategy that suitably applies the state-of-the-art from all directions. It turns out that often, the principles each purportedly contradict one another. For instance, avateering with motion capture on a HR can be intuitive, but the technique becomes cumbersome when attempting to accomplish simple manipulation tasks, such as grasping [KAS13], [CTC12]. The assistive agents can be applied to assist in the hardest tasks, such as manipulator position and force augmentation [WNU09], [LME11], [DS13], but prediction and application of user intent to arbitrate user-input is like an art, and more often than not, it is better for the agent to take complete teleoperative control for arbitrary time intervals. In these instances, agent assistance breaks the avateering metaphor, adversely affecting the controlling user's perception and expectation.

It is hard to design an assistive agent that not only predicts what users may want, and how it may assist, but also arbitrate inconspicuously such that users may perceive they have full avateering control at all times using unencumbered motion capture. Section *'Modeling Agents that Assist Discreetly'* shall discuss the schematics of such an agent that activates and assists discreetly at suitable timings to allow users perceive avateering control, while teleoperating a Humanoid Robot to grasp, move, and release an object at a goal location. The discussion is constrained to manipulators that initiate non-prehensile grasping, and modeling axioms shall be stated which guide the development of such an agent. The Humanoid Robot used for the study in this chapter is a Webots[4] simulation of the DARWIN-OP (Darwin) robot[5].

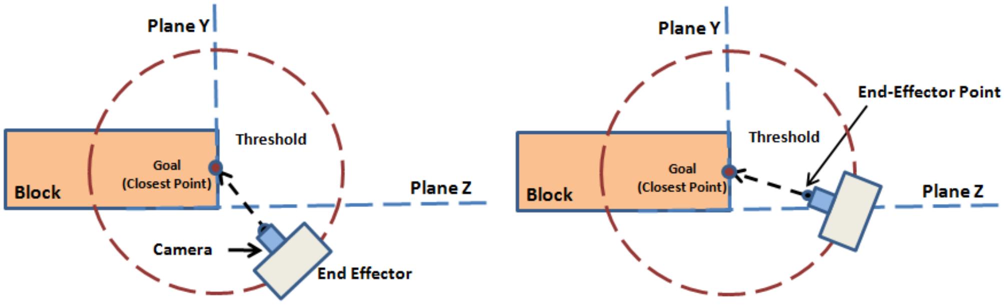


Figure 6.1: The manipulator shown on the left acquires a potential goal point, but the agent is not activated as it does not meet the locality requirement, since it is located behind Plane Z. At the next tick, the user moves the manipulator that meets the locality requirement (in front of Planes Y & Z, and within the threshold), which then activates the agent to take control and move the manipulator to the last-acquired goal point.

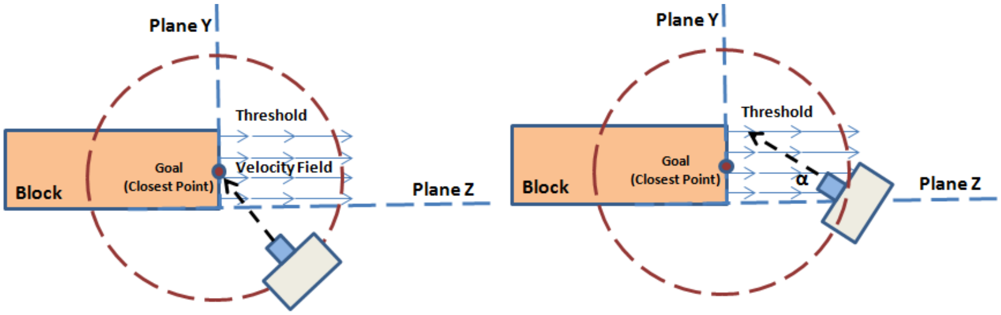


Figure 6.2: Similar to Figure 6.1, manipulator shown on the left acquires a potential goal point, but the agent is not activated. At the next tick, the user moves the manipulator that meets the locality requirement, but the agent will only activate and take control if it approaches the object plane the point resides on at an angle smaller or equal to a set α, and its velocity exceeds a set threshold.

[4] https://www.cyberbotics.com/webots.php

[5] www.robotis.com/xe/darwin_en

Modeling Agents that Assist Discreetly

The hypothetical agent should arbitrate at two phases, i) Before, and ii) After making contact with the target object. The agent shares its perceptual space with the user, and also possesses near-perfect spatial information about that space.

Before Grasp Contact

The main idea behind the agent letting a user maintain avateering control consists of determining its i) Timing of Activation, and ii) Type of Arbitration, based on predicting the users' expectation of the manipulator trajectories around objects before and after making contact.

Agent Activation: A straight-forward heuristic

Target proximity, if localized adequately, can be a simple and effective heuristic to determine where and when the agent should activate. Tracing a ray from the center of the eye-in-hand camera, the point it hits is usually the target (or the closest point). Figure 6.1 illustrates how we can take the ray-trace distance as an input to a Euclidean radial-distant activation threshold. The agent can activate whenever this threshold is breached. We state this heuristic with the following axiom:

Axiom (1) : The closest target point, based upon the pose of the manipulator (which directly affects the eye-in-hand camera pose), is usually, the intended target from the user.

Figure 6.1 also illustrates how the manipulator can be poorly located away from the region where the user would prefer the agent to be activated, which in this case is by the right side of the block. A better localization heuristic is to check further whether the manipulator is between planes Y and Z. Used together with a radial threshold, the agent can be activated where the manipulator is both properly localized and orientated towards a valid target point on the object.

Improving the base heuristic on activation timing

Axiom (1) addresses agent activation locality ('where') sufficiently, but not necessarily 'when'. Assuming that upon activation, the agent takes full control and moves the manipulators to their target placements, localization is not sufficient to hide the noticeable arbitration to a keen eye. It is further exacerbated when considering the operator's attentive field-of-view (FOV) would usually be upon the task-space regions where the agent activates. This observation postulates to two additional axioms as we model alternative agent arbitrative behavior:

Axiom (2) : The agent should blend its activation-arbitrative actions into the robot's current executing action or operating environment.

Axiom (3) : The agent should divert some or all activation-arbitrative actions away from *the user's main focal point of attention (Sleight-Of-Hand Principle).*

An application of Axiom (2) is to allow free avateering and delay activation till the agent detects an expected action triggered by the manipulator in order to make contact with the object; This criteria can be named Just-In-Time (JIT) Activation. One such instance exists is where the user sends the manipulator, at some reasonable velocity and angle to the normal of the object planes, towards their target points. The agent takes control only when the manipulator velocity with respect to the plane exceeds a stated threshold. Additionally, the corrective action it performs also blends and conceals within the momentum moving the manipulator towards the plane. The threshold is modeled as a velocity vector field, growing at increasing length from the surface using its plane normal (Figure 6.2). The changing field implicitly models how a user would approach the object surface, given that the manipulator velocity usually does reflect a user's level of intent about approaching the object; For instance, a user who feels confident enough would send the manipulators towards the object plane from far at greater speeds, and at a vector that is closely parallel to the plane normal (e.g smaller than α, Figure 6.2). At slow speeds, the agent can activate (if the dot-product threshold is met) since the velocity-field vector gets shorter with decreasing distance to the object contact surface. Linear and rotational velocities of the manipulators can be computed using the geometric Jacobian with their associated Denavit-Hartenburg (DH) coordinate frames, and the resulting dot product between the manipulator's linear velocity and velocity-field vector dependent upon its position in the activation space serves as input to decide activation. For instance using Figure 6.2, once the locality requirement is met by the manipulator, when the dot-product computation between the manipulator's linear velocity and velocity-field vector it intersects inside the field exceeds a set value, the agent will take full control at the next time-step.

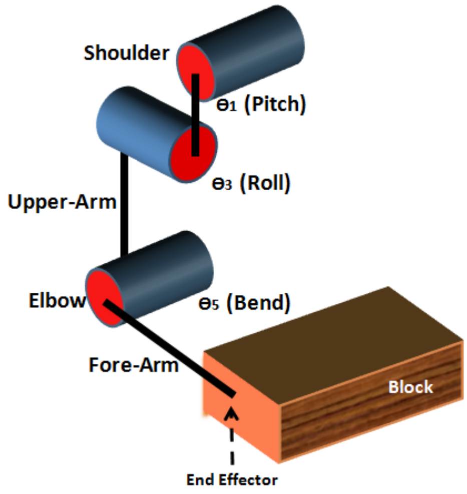


Figure 6.3: A 3 Degree-Of-Freedom Arm Manipulator applying non-prehensile grasping at the side of the block

Alternative Arbitration Strategy After Activation

Axiom (2) models the agent to surreptitiously arbitrate user input by quickly taking and relinquishing full control when the manipulator meets the activation requirement based on locality and approach. Axiom (3) models an alternative input mediation policy that hypothesizes i) whether the assistive agent, upon activation, can allow the user a measure of full control (e.g. user can voluntarily move the manipulator towards or away from the target points) but maintains its arbitration role through partial control, ii) the option of deactivating the agent, and iii) whether the agent can accomplish both i) and ii) without the user noticing. One idea is to leverage on the DOF redundancy of the manipulator, letting a user control the most significant degrees-of-freedom (DOF) that accomplishes the task from his point of view that dominates attention, and letting the agent control any remaining DOF to correct the trajectory error. Figure 6.3 illustrates such an example using a DARWIN-OP arm with 3DOF. Upon meeting the agent activation requirement, it is sufficient for the user to control only θ_3 (roll) in order to send the manipulator to make contact with the right side of the block, while the agent sets a manipulator pitch (θ_1) and bend (θ_5). Additionally, if the manipulator rolls away from the activation threshold, the agent deactivates and the user regains full DOF control. The manipulator pose, location, and task action also helps to conceal the arbitration performed, as the block constrains any available operating space the manipulator can move.

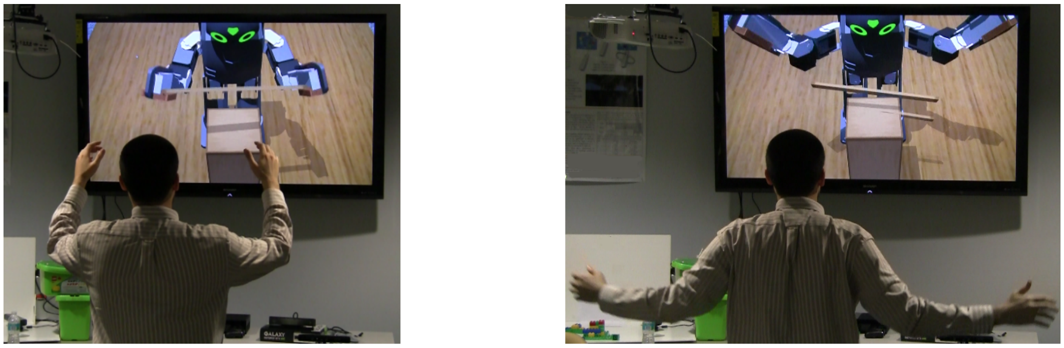


Figure 6.4: Left Image: User attempts to grasp a block by the sides and lift it away from the crate. Right Image: Agent deactivates with a 'release' gesture; Darwin's arms recalibrate back slowly to user-space

After Grasp Contact

User intent is hard to predict after the manipulator makes contact. Therefore:

Axiom (4) : The agent should assist in maintaining a suitable contact force between the manipulators and object as the user desires. Otherwise, the user regains full avateering control.

In many task scenarios, it is very useful for the agent to help the user maintain an arbitrary force contact on the object when applying a grasp. Force input augmentation, such as force-feedback, can be suitably applied to maintain a steady grasping force on the object if a user desires so. It is observed that users have difficulty holding on to the block with avateering, frequently moving the manipulators inadvertently away from the grasp points as they attempt to move it away from the crate (see Section: *'Agent Implementation & Trial Description'*). The main difficulty resides in determining a user's intent to either maintain or release grasp. A feedback system presenting a representation of the force contact on the target object, either visually or by tactile, can be helpful for the user. However, the lack of physical constraints in between the user's hands afforded by the real object prevents an adequate ascertainment of the grasp points (Figure 6.4). There are many heuristics which can evaluate how the agent can assist here upon contact, and in fact, the agent can opt to do nothing and de-activate with the user having full control. One strategy, however, is to take a holistic approach and consider both user and robot states, and predict the next action the teleoperating user could take. Gesture and pose recognition with a-posteriori prediction can be encoded into the agent, for instance using HMM [PKJ05] and POMDP models [PG07], and based upon results from the prediction an agent can choose whether to maintain contact or deactivate.

Agent Implementation & Trial Description

Four agents were developed using the modeling principles outlined from Axioms (1) to (4). Each agent corresponds to a distinctive technique, which we state as i) No-Degrees-Of-Freedom (No-DOF) ii) No-Degrees-Of-Freedom Just-In-Time (No-DOF-JIT) iii) One-Degree-Of-Freedom (1-DOF) and iv) Two-Degrees-Of-Freedom (2-DOF). A Webots simulation was used to devise a simple trial to test the main hypotheses that:

Hypothesis 1: User perception of Avateering can co-exist with the presence of an assistive agent in the background that arbitrates the user's input.

Hypothesis 2: User can be agnostic to the presence of an assistive agent when the corrective action it performs blends with users' expectation of the manipulators' behavior during avateering.

Each technique prescribes to a different heuristic in their approach on interacting with the object. However for the study, all techniques shall share the same strategy in deciding for the user whether to maintain the grasp for transport after contact. The trial requires users to avateer a Darwin to complete a set of actions in the following order; (i) Grasp a plank by the sides off the crate, (ii) Lift the plank, (iii) Move the plank away from the crate (iv)

Drop the plank off the side of the standing platform. Gesture recognition heuristics from both arms are used by the agent to decide whether to deactivate after contact (Figure 6.4).

No-DOF & No-DOF Just-In-Time (JIT) Techniques

No-DOF is designed with the 'naive' approach, applying the modeling principle of Axiom (1) alone. The agent takes full control of Darwin's arms after their manipulators meet the locality requirement, performing an arbitrative action by moving the arms to their grasp points, and relinquishes back control. The user will have no input control during this arbitration phase. Corrective action is applied to all entries of the vector [q] of the manipulator, where [q] represents the joint variables of Darwin's three degree-of-freedom right-arm. With exception to the 2-DOF technique, all techniques used the iterative Damped-Least-Squares (DLS) [BK05] algorithm to compute the manipulators' corrective paths to their grasp points. The set of Forward-Kinematics (FK) equations tightly couple between the joint-angle variables in [q] (θ_1,θ_3,θ_5), rendering their solutions highly non-linear. Without stating the associated DH parameters, the FK system of equations (position) for Darwin's right arm are:

$$\{X\} = \begin{pmatrix} L_1 c_1 - L_3 s_3 - L_H s_1 s_5 + c_1 c_3 (L_2 + L_H c_5) \\ L_3 c_1 + L_1 s_1 + L_H c_1 s_5 + c_3 s_1 (L_2 + L_H c_5) \\ L_2 s_3 + L_H c_5 s_3 \end{pmatrix}$$

where $\{X\}$ represents the manipulator point in 3D space, L_1 & L_3 represent upper-limb link offsets introduced to mitigate singularities, L_2 & L_H represent upper & lower arm lengths respectively, and the following abbreviations were used: $c_i = cos(\theta_i)$, $s_i = sin(\theta_i)$, $i = 1, 3, 5$ Upon agent activation, projected operation and task-space of Darwin's manipulators around the plank usually provide good seeding data, allowing the algorithm to converge to a valid Inverse Kinematics (IK) solution in a low-order of iterations. If no valid solution is obtained after a set amount of iterations, user avateering input is used instead. After contact is made, the agent maintains a locking grasp on the object, and deactivates (unlocks) upon recognition of a 'release' gesture made by the user (Figure 6.4). For the remaining techniques, the agent's arbitrative strategy after contact with the object remains constant.

No-DOF JIT is similar to No-DOF, but includes the modeling principles of Axiom (2). Besides locality, the agent activates if the manipulators exceed the velocity and angle-of-approach threshold with respect to the object plane where the goal point resides.

1-DOF & 2-DOF Techniques

1-DOF applies the alternative arbitration principle supported in Axiom (3). The user retains control of Darwin's upper-arm roll (Figure 6.3) when the agent activates (Axiom (1)), and can choose to deactivate the agent before contact by rolling the manipulator away from the locality threshold. Additionally, computing IK solutions can be less expensive (agent becomes more efficient during arbitration), as unique geometric solutions are possible if one of the elements in [q] is known.

2-DOF JIT is similar to 1-DOF, but the user retains control of the additional degree of elbow-bend (i.e. agent controls only the upper-arm pitch). Besides control, an additional difference involves the agent's arbitrative behavior. Once the 1-DOF agent acquires the grasp point, it can choose either to set the elements in [q] it controls immediately, or to interpolate values in-between to set a path the manipulator takes to make contact with the object plane. Instead, 2-DOF sets the path frame-by-frame after the goal point is acquired, using an iterative approach similar to the Cyclic-Coordinate Descent (CCD) algorithm [Ken12]. The object plane that the manipulator intends to make contact serves as a very useful constraint, assuring that user-set inputs of the upper-arm's roll and elbow-bend (θ_3,θ_5) will reach the goal point's z-coordinate. Based on Darwin's right-arm FK, the agent only needs to adjust the upper-arm pitch (θ_1) to move the manipulator closer to the grasp point in coordinates x-y with respect to the user's inputs for θ_3 and θ_5 in that current frame.

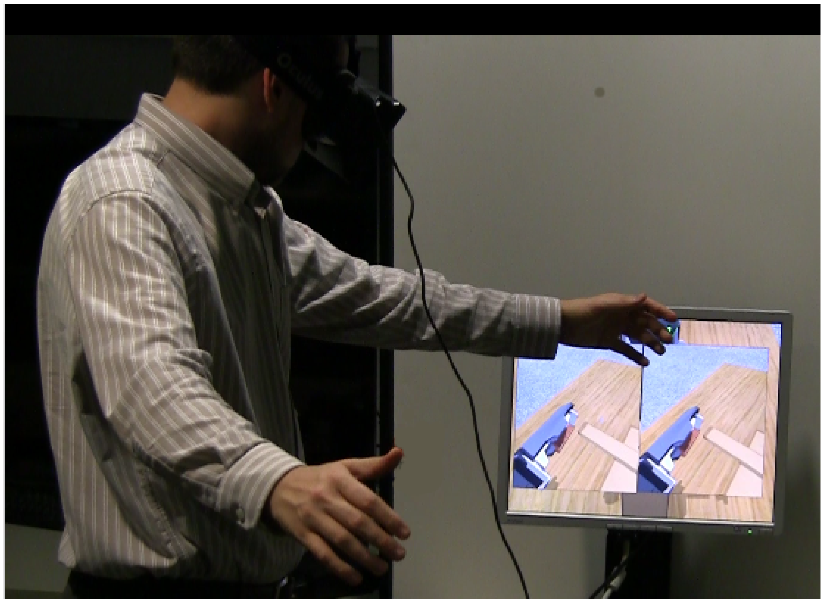


Figure 6.5: User attempting the trial in a telepresence setting

Table 6.1: In-between survey questions after completing a trial assisted by a technique of random choice

I quickly understood how to control the entire robot
It was easy to pickup the plank from the crate
The robot arms moved according to my expectation
As a whole, performing the task was easy
It was easy to make the robot walk
The robot walked according to my expectations.
Overall, the robot moved according to my expectations

Empirical Study

The following sections describe the virtual environment setup, task objectives, devices and software, and participant demographics.

Participants, Devices & Software

15 students from the University of Central Florida are recruited for the user study. Final pool consists of 12 males and 3 females, ages between 18 and 41. The trial is a simulation developed using Webots EDU installed on Ubuntu 12.04 LTS. Webots provides a realistic model of the Darwin-OP, since it is commonly used for Robocup [6]. The manipulators are modified to allow for easier grasping of virtual objects. Skeletal-joint data from the user

[6] http://www.robocup2013.org/472-2

is captured using the Kinect motion sensor and post-processed with OpenNI 2.2[7] and NiTE 2.2.0.11 middleware[8].

User Study Design

Darwin was placed on a platform along with a crate, upon which rested a plank (Figure 6.4). The participant was tasked with guiding Darwin's arms to grasp and lift the plank, navigate it to an edge of the platform, and release it. Each participant was allowed five attempts to complete the trial per technique which excludes the control technique. The control is essentially free avateering, with no agent assisting the user to complete any action during the trial. Prior to commencing the trial, the user was allowed a brief training session to become familiar with the simulation. Order of techniques executed at each trial are randomized but ensured to be unique. If Darwin successfully picked up the plank, the user was required to turn and walk Darwin via body gestures to transport the plank to either side of the platform and release it. Once both conditions were met, or if the plank was knocked off the crate or dropped too early, the scenario was reset.

[7] http://www.openni.org

[8] http://www.openni.org/files/nite

Table 6.2: Paired Samples T-Tests between control and assistive techniques

Technique	t_{14}	p-val
No-DOF	-9.886	.005
No-DOF JIT	-5.229	.0071
1-DOF	1.143	.0056
2-DOF	10.286	.00625

Quantitative & Qualitative Metrics

The main hypothesis of the user study is there will be superior techniques against the control ('Free' Avateering) if an agent did benefit the users. The number of successes and failures when lifting and releasing the plank (''Grasp" and ''Release" rates) were recorded. After each technique was used a total of five times to complete the task, each participant proceeds to fill out a small survey consisting of seven questions about the experience (Table 6.1). Each question was measured on a 7-point Likert scale. All participants were made unaware of which technique they were using at any given time so as to eliminate any user bias towards any technique. This way, the survey questions provide an opportunity to gain any insight whether the user still felt in-control of Darwin during the trial.

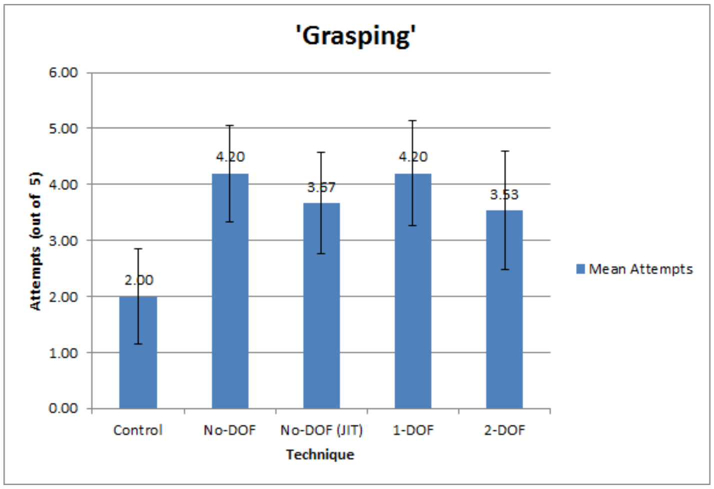


Figure 6.6: Pickup ('Grasping') Success Rate

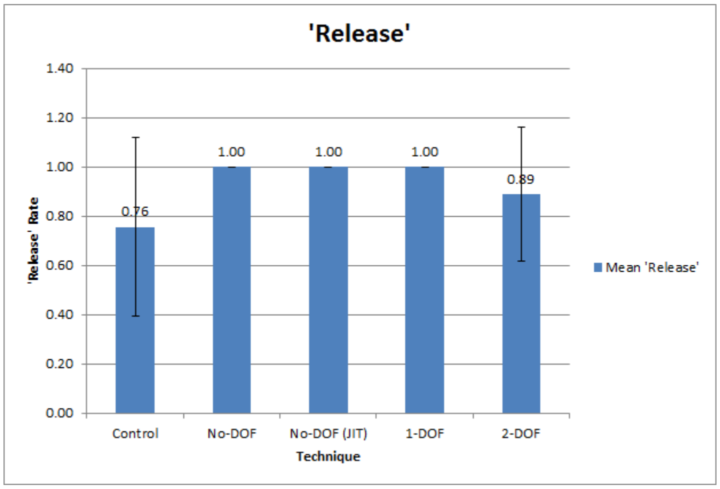


Figure 6.7: Let-Go ('Release') Success Rate

Results & Discussion

Quantitative Analysis

Using a repeated measures ANOVA test, significant difference was found between the techniques ($F_{4,14} = 15.852; p < .0001$), with the sphericity assumption maintained. With the control technique removed, no significant difference was found across the techniques ($F_{3,14} = 2.040; p = 0.123$). Figure 6.6 illustrates the task completion rate for picking up the plank. It can be observed very clearly that the avateering metaphor performed significantly better with agent assistance. T-tests were performed between all pairs of techniques for a total of 10 comparisons, and using Holm's Sequential Bonferroni's Adjustment [Hol79] to control Type I errors. Significant difference was found between the control and remaining techniques (Table 6.2). As expected, the avateering metaphor greatly benefited from an assistive agent when grasping the plank. The four agent-assisted techniques allowed the participant to pickup the plank twice as often as the control base metaphor, proving that free avateering is not sufficiently efficient by itself. Additionally, two of the techniques assisted in maintaining a grasp on the plank until end of trial. Figure 6.7 illustrates the success rate for maintaining the grip and dropping the plank when needed. This success rate is dependent upon the prior task of successfully lifting the plank off the crate. The large standard deviation of this success rate suggests that the presence of expert users who accustomed quickly to avateering benefit little from the agent's assistance. This also suggests that the agent is useful for non-expert or inattentive users, or to alleviate the inadequacy of the motion sensor interface either through noise, or by inadvertent arm movements during the transport. It is also observed that users who are more confident with their movements scored better with the 'No-DOF JIT' & '2-DOF' techniques, while users who are feeling more careful excelled with 'No-DOF' & '1-DOF'. This suggests that the threshold parameters across the techniques can be pre-calibrated according to a user's affect before undergoing the trials.

Qualitative Analysis

Friedman tests were used to determine statistical significance between the techniques. In the event of significance, we used Wilcoxon's signed rank tests between all techniques, for a total of 10 comparisons. There were no statistical significance between the control technique and others with regards to user perception. This aligns with the study's expectations; users were relatively unable to perceive that an agent was assisting with the teleoperation. However, the participants felt that it was easier to pickup the plank off the crate using 3 of the 4 techniques (Table 6.3). Second, the users responded that the robot arms moved more accordingly to their expectations, when using a technique involving the agent (Table 6.4). This positive outcome reveals that although the users did not have knowledge of an assistive agent arbitrating the manipulator actions, they still felt more in control than free avateering. Additionally, not only do the arms move according to their expectation, their perception of control during free avateering is similar to any other agent-assisted technique.

Table 6.3: Mean and median values of user perception towards ease of 'pickup'

Technique	Mean	Median
No-DOF	5.1	5
No-DOF JIT	4.8	5
1-DOF	5.0	6
2-DOF	3.9	3
Control	4.1	4

Table 6.4: Mean and median values of user perception towards expectation of robot arm movements while avateering

Technique	Mean	Median
No-DOF	5.6	6
No-DOF JIT	5.0	5
1-DOF	5.1	5
2-DOF	4.1	4
Control	4.7	5

Conclusion: Non-obtrusive User Assistance in HRTs'

Even though users were oblivious to the agent's corrective actions, grasp completion was achieved at significantly higher rates compared with free avateering. The study shows that under situational observation of past trajectories, it is possible to model agents or robots to assist users with teleoperation non-obtrusively without disrupting the teleoperative experience and expectation.

CHAPTER 7
CONCLUSION & FUTURE WORK

In this work, we devised a service architecture in Chapter 3 that describes our theoretical autonomous pipeline of high-level task planning inside a service robot. In Chapters 4 & 5, we used the architecture to *localize* human-in-the-loop collaborative models along the pipeline which allow non-experts to convey natural abstractions of high-level manipulation tasks, semantic revisions, geometries, and trajectories through stylus-based modalities of a *mobile* interface. By assuming perfect grasp postures and object recognition (with color-features) in a 'Blocks World' scenario, we evaluated cooperative task modeling and recovery through semantic grounding and goal-predicate coalescence given an action corpus (Chapter 4). We explored further interactions that could help a service robot recover the manipulative action through the Grasp and Motion Planning pipeline instead of teleoperating the action (Chapter 5). If teleoperative recovery becomes unavoidable, it can be performed through egocentric immersion and proprioception with arm and hand gestures (Chapter 6).

Positive results from the user studies can be attributed to one or a combination of three major factors as a result of robot-centric collaborative influence; (1) *Reusability*, (2) Speed, and (3) User Affordance.

First Study: PlanIt! With the task modeled as a logical conjunction of effects, the expression can be reused under any structural changes to the world; further, the model is abstracted away from the action corpus and analytical considerations for the conditions needed to perform each action. A user can model a task quickly by focusing only on its *objectives*, rather than considering *how* it should be performed; the course of actions is decided entirely through the robot's autonomous planning. Further, the user has intuitive and *implicit* control over the manipulative actions usable on any artifact through its *semantic* revisions *without* having to be cognizant of the available actions within the corpus. The modeled task can also be performed through a *partially observable* environment, and is possibly recoverable by relieving the *condition* that caused the exception; rather than teleoperating the action, or to signal a re-plan.

Second Study: GraspIt! By limiting the range to household objects where symmetry is commonly observed, participants are afforded to express reasonable hypotheses about the hidden shape *globally* through *controlled scene continuation* along symmetry descriptors. As the hypotheses are secondary resources of spatial primitives (points and normals) from the user, they could be reused for parameter initialization or validation of automated hypotheses that also predict missing geometries and shape.

Third Study: MoveIt! Instead of visualizing a robot's motion through 3-D space as sequences of timed configurations, it is more common and intuitive for a user to predicate a course between the current and goal locations whenever autonomous motion planning is unable to locate a feasible path for the robot. In our context of the user study, teleoperation is feasible whenever motion planning is unable to plot a trajectory either for a 'pick' or 'place' action for the JACO arm. But instead of predicating or teleoperating the whole trajectory, key waypoints which fall along or near the user-projected path can be offered to influence and stimulate motion planning. The waypoints are not direct inputs to the motion plan, but rather, function as region indicators where a motion plan should intensify C_{free} sampling. The interaction is further simplified in our interface, where recognized or labeled artifacts located closely to the projected path can be offered (through selection on the video stream or Model Pane) instead of prescribing specific points inside the workspace.

Fourth Study: TeleoperateIt! By leveraging on human-robot anthropomorphic similarities, the manipulators of a service robot can be teleoperated through egocentric immersion and proprioception with arm and hand gestures while wearing a HMD [KPL14]. Teleoperative precision and experience were maintained by an assistive agent adjusting the user's teleoperative input in the background; the levels of arbitration needed is determined by predicting the user's intentions observed from the manipulators' trajectories around the scene geometries.

PlanIt! Cooperative Modeling: Limitations & Future Directions

Aside from empirical control (Chapter 4, Section *'Prototype & Experiment Design'*), the 'Blocks World' domain was also used to mitigate user bias towards contemporary technical challenges encumbering automatic grasp-planning and object recognition. First, our prototype facilitated evaluation of user perception towards collaborative detection, grounding, and semantic revisions of *unknown* perceptual instances from the robot's point of view. Second, user performance and preference towards effect-based cooperative modeling with autonomous-planning can also be measured against visual programming that require full knowledge about the task artifacts. Though an eventual direction is to move the study outside toy examples, there are a few directions that can be explored further within the 'Blocks World' domain, or through virtual simulations when performing the tasks with an actual robot manipulator is not an important consideration;

Beyond *Object* and *Location* concept relations. It is actually possible to progress towards a deeper and richer ontology schema while remaining in 'Blocks World'; for example, the ontology schema from Figure 3.1 could be used, requiring a palette of more than two colors (six in this case) that currently suffice for grounding *'Object'* and *'Location'* concepts to the known artifacts. Besides color, block height and width can be varied within reasonable bounds in order to simulate further objects for the empirical design. With a richer ontology schema, it remains an interesting question how the Model Pane (Chapter 3, Figure 3.5) should present the information when each known artifact could be represented by more than two concept relations. Entailing usability questions, such as the increase of cognitive load that accompanies visual clutter, will follow with each iteration of the prototype design.

Sketch Strokes. Lasso strokes of different-color stylus ink were used for grounding and recognition of unknown perceptual instances with different concepts to the Ontology KB (Chapter 3, Section *'Interface Design'*). It would be interesting to explore additional ink strokes to apply on the geometries *within* the lasso that may provide further cues for autonomy to extract hidden features and properties. For example, a line stroke representing a plane-cut may indicate a separation of geometries with special characteristics from the main body (Chapter 1, Figure 1.2), such as handle geometries of a mug, or brittle material that require delicate grasping. Further usability questions, such as ideal stroke representations non-experts would expect for the cues, would be important in order to measure whether the strokes are intuitive representations.

Predicate Construction. It will be important to evaluate how predicate (or effect) construction (Chapter 3, Figure 3.5) would impact user performance and perception when a richer ontology schema is introduced together with a growing action corpus. The current prototype did not require users to select a predicate function, since there were only two possible effects (*at-location*, *at-obj-location*) inside the expanded corpus of three actions (Chapter 4, Section *'Prototype & Experiment Design'*); the name of the function can be implied through parameter selection of either an *Object-Location* or *Object-Object* concept pair. However, it can be observed that the selected concepts can perform a *filtering* operation through the list of predicate functions; the filter also applies through the corpus, eliminating actions whose effect do not match the parameter signature of the selected concepts. For example, selecting an *Object-Object* concept pair eliminates 'clear-top' and 'stack' (Chapter 4, Listing 4.4), since the respective effects are parameterized by an *Object-Location* concept pair. It would be interesting to see how user perception and performance changes when the predicate function name needs user specification after selection of the concept parameters from the Model Pane.

Predicate Coalescence. Similar to the Model Pane, excessive entries within the Predicate Pane (Chapter 3, Figure 3.5) can be a negative influence on user performance and perception towards modeling a task with predicate coalescence. It would be interesting to see how information within the Predicate Pane can be presented for user selection as more are added to the pane.

GraspIt! Collaborative Model: Limitations & Future Directions

The user study was designed to validate symmetry as an intuitive descriptor non-experts would favor to express reasonable hypotheses about a common household object's hidden geometries and 3-D shape. For each task, the object's main axis was provided to assist the evaluation of user intuition and understanding for ambiguous and composite symmetries present along the axis. It remains an interesting question how a user would perform when the main axis is not provided, further requiring inclusion to the expressed hypothesis. A possibility can be found through the 3-sweep technique [CZS13]; a user prescribes three points defining the centroid of a planar base where the axis passes through, with an optional fourth point that specifies the extent (or it is presumed to be infinite). Besides symmetry, it would also be interesting to explore alternative descriptors amenable to non-experts in expressing 3-D shape of common household objects.

MoveIt! Collaborative Model: Limitations & Future Directions

The user collaboration with motion planning was tested against *simulated* manipulation stalls; the user study sets up the motion planner to be "successful" in C_{free} sampling such that the samples ensure a path between the current and goal configurations. Thus, as long as the user selects a "correct" waypoint (i.e. labeled artifact) for that particular stall which were decided in advance by the proctor, the manipulator will proceed with the modified trajectory that includes passing through the selected waypoint. As state-of-the-art motion planning is a probabilistically-complete approach to trajectory finding, the user study was designed in this manner based on the assumption that motion planning is *guaranteed* to find a feasible trajectory after a user's *positive* influence on the sampling strategy. In reality, the *narrow passage* problem [DCQ16] that encumbers motion planning makes teleoperation a more common and straightforward option to recover the action. It would be interesting to evaluate the changes in user perception towards cooperative motion-planning through C_{space} exploration in real-world problems that may not guarantee a feasible solution within reasonable time.

Concluding Remarks

It is possible for non-experts to become efficient users of an autonomous service robot through *robot-centric* collaborative models. We envision that service robots can become commonplace at home when their utility is maximized by granting effective access to their controlling architectures of autonomous-planning. This becomes possible through novice-oriented mobile interfaces that enable non-technical users to convey natural abstractions of high-level tasks, recovery assistance, and teleoperative control.

APPENDIX A
IRB Approval Letters

University of Central Florida Institutional Review Board
Office of Research & Commercialization
12201 Research Parkway, Suite 501
Orlando, Florida 32826-3246
Telephone: 407-823-2901 or 407-882-2276
www.research.ucf.edu/compliance/irb.html

Approval of Human Research

From: **UCF Institutional Review Board #1**
FWA00000351, IRB00001138

To: **Senglee Koh**

Date: **April 29, 2014**

Dear Researcher:

On 4/29/2014 the IRB approved the following human participant research until 4/28/2015 inclusive:

Type of Review:	Submission Correction for UCF Initial Review Submission Form Expedited Review
Project Title:	Towards a Natural Human-Robot Team Multimodal Gestural Interface
Investigator:	Senglee Koh
IRB Number:	SBE-13-09552
Funding Agency:	
Grant Title:	
Research ID:	N/A

The scientific merit of the research was considered during the IRB review. The Continuing Review Application must be submitted 30days prior to the expiration date for studies that were previously expedited, and 60 days prior to the expiration date for research that was previously reviewed at a convened meeting. Do not make changes to the study (i.e., protocol, methodology, consent form, personnel, site, etc.) before obtaining IRB approval. A Modification Form **cannot** be used to extend the approval period of a study. All forms may be completed and submitted online at https://iris.research.ucf.edu .

If continuing review approval is not granted before the expiration date of 4/28/2015, approval of this research expires on that date. When you have completed your research, please submit a Study Closure request in iRIS so that IRB records will be accurate.

Use of the approved, stamped consent document(s) is required. The new form supersedes all previous versions, which are now invalid for further use. Only approved investigators (or other approved key study personnel) may solicit consent for research participation. Participants or their representatives must receive a copy of the consent form(s).

In the conduct of this research, you are responsible to follow the requirements of the Investigator Manual.

On behalf of Sophia Dziegielewski, Ph.D., L.C.S.W., UCF IRB Chair, this letter is signed by:

Signature applied by Patria Davis on 04/29/2014 12:54:40 PM EDT

IRB Coordinator

University of Central Florida Institutional Review Board
Office of Research & Commercialization
12201 Research Parkway, Suite 501
Orlando, Florida 32826-3246
Telephone: 407-823-2901 or 407-882-2276
www.research.ucf.edu/compliance/irb.html

Approval of Human Research

From: **UCF Institutional Review Board #1**
FWA00000351, IRB00001138

To: **Senglee Koh** and Co-PIs: **Kevin Pfeil**

Date: **April 12, 2018**

Dear Researcher:

On 04/12/2018 the IRB approved the following human participant research until 04/11/2019 inclusive:

Type of Review:	UCF Initial Review Submission Form Expedited Review
Project Title:	Controlling Robot Arm with different modes
Investigator:	Senglee Koh
IRB Number:	SBE-18-13921
Funding Agency:	
Grant Title:	
Research ID:	N/A

The scientific merit of the research was considered during the IRB review. The Continuing Review Application must be submitted 30days prior to the expiration date for studies that were previously expedited, and 60 days prior to the expiration date for research that was previously reviewed at a convened meeting. Do not make changes to the study (i.e., protocol, methodology, consent form, personnel, site, etc.) before obtaining IRB approval. A Modification Form **cannot** be used to extend the approval period of a study. All forms may be completed and submitted online at https://iris.research.ucf.edu .

If continuing review approval is not granted before the expiration date of 04/11/2019, approval of this research expires on that date. When you have completed your research, please submit a Study Closure request in iRIS so that IRB records will be accurate.

Use of the approved, stamped consent document(s) is required. The new form supersedes all previous versions, which are now invalid for further use. Only approved investigators (or other approved key study personnel) may solicit consent for research participation. Participants or their representatives must receive a copy of the consent form(s).

All data, including signed consent forms if applicable, must be retained and secured per protocol for a minimum of five years (six if HIPAA applies) past the completion of this research. Any links to the identification of participants should be maintained and secured per protocol. Additional requirements may be imposed by your funding agency, your department, or other entities. Access to data is limited to authorized individuals listed as key study personnel.

In the conduct of this research, you are responsible to follow the requirements of the Investigator Manual.

This letter is signed by:

Signature applied by Gillian Morien on 04/12/2018 01:39:20 PM EDT

Designated Reviewer

University of Central Florida Institutional Review Board
Office of Research & Commercialization
12201 Research Parkway, Suite 501
Orlando, Florida 32826-3246
Telephone: 407-823-2901 or 407-882-2276
www.research.ucf.edu/compliance/irb.html

Approval of Human Research

From: **UCF Institutional Review Board #1**
FWA00000351, IRB00001138

To: **Senglee Koh** and Co-PI: **Kevin Pfeil**

Date: **June 21, 2018**

Dear Researcher:

On 06/21/2018 the IRB approved the following human participant research until 06/20/2019 inclusive:

Type of Review:	UCF Initial Review Submission Form Expedited Review
Project Title:	Completing 3D Spatial Geometries with Symmetric Cues
Investigator:	Senglee Koh
IRB Number:	SBE-18-13920
Funding Agency:	
Grant Title:	
Research ID:	N/A

The scientific merit of the research was considered during the IRB review. The Continuing Review Application must be submitted 30days prior to the expiration date for studies that were previously expedited, and 60 days prior to the expiration date for research that was previously reviewed at a convened meeting. Do not make changes to the study (i.e., protocol, methodology, consent form, personnel, site, etc.) before obtaining IRB approval. A Modification Form **cannot** be used to extend the approval period of a study. All forms may be completed and submitted online at https://iris.research.ucf.edu .

If continuing review approval is not granted before the expiration date of 06/20/2019, approval of this research expires on that date. When you have completed your research, please submit a Study Closure request in iRIS so that IRB records will be accurate.

Use of the approved, stamped consent document(s) is required. The new form supersedes all previous versions, which are now invalid for further use. Only approved investigators (or other approved key study personnel) may solicit consent for research participation. Participants or their representatives must receive a copy of the consent form(s).

All data, including signed consent forms if applicable, must be retained and secured per protocol for a minimum of five years (six if HIPAA applies) past the completion of this research. Any links to the identification of participants should be maintained and secured per protocol. Additional requirements may be imposed by your funding agency, your department, or other entities. Access to data is limited to authorized individuals listed as key study personnel.

In the conduct of this research, you are responsible to follow the requirements of the Investigator Manual.

This letter is signed by:

Signature applied by Gillian Morien on 06/21/2018 04:29:31 PM EDT

Designated Reviewer

LIST OF REFERENCES

[BD06] T. Bailey and H. Durrant-Whyte. "Simultaneous localization and mapping (SLAM): part II." *IEEE Robotics Automation Magazine*, **13**(3):108–117, Sept 2006.

[BF05] Toby P. Breckon and Robert B. Fisher. "Amodal volume completion: 3D visual completion." *Computer Vision and Image Understanding*, **99**(3):499 – 526, 2005.

[BFB05] D. J. Bruemmer, D. A. Few, R. L. Boring, J. L. Marble, M. C. Walton, and C. W. Nielsen. "Shared Understanding for Collaborative Control." *Trans. Sys. Man Cyber. Part A*, **35**(4):494–504, July 2005.

[BJL11] J. Bohg, M. Johnson-Roberson, B. Len, J. Felip, X. Gratal, N. Bergstrm, D. Kragic, and A. Morales. "Mind the gap - robotic grasping under incomplete observation." In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 686–693, May 2011.

[BK05] Samuel R Buss and Jin-Su Kim. "Selectively damped least squares for inverse kinematics." *journal of graphics, gpu, and game tools*, **10**(3):37–49, 2005.

[BK10] Jeannette Bohg and Danica Kragic. "Learning Grasping Points with Shape Context." *Robot. Auton. Syst.*, **58**(4):362–377, April 2010.

[BKS12] S. Balakirsky, Z. Kootbally, C. Schlenoff, T. Kramer, and S. Gupta. "An industrial robotic knowledge representation for kit building applications." In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1365–1370, Oct 2012.

[Blu67] Harry Blum. "A Transformation for Extracting New Descriptors of Shape." In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pp. 362–380. MIT Press, Cambridge, 1967.

[BRH14] J. Bohg, J. Romero, A. Herzog, and S. Schaal. "Robot arm pose estimation through pixel-wise part classification." In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3143–3150, May 2014.

[BRJ11] J. Bohren, R. B. Rusu, E. Gil Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Msenlechner, W. Meeussen, and S. Holzer. "Towards autonomous robotic butlers: Lessons learned with the PR2." In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 5568–5575, May 2011.

[Car06] Stefano Carpin. "Randomized Motion Planning: A Tutorial." **21**, 01 2006.

[CBH11] J. Y. C. Chen, M. J. Barnes, and M. Harper-Sciarini. "Supervisory Control of Multiple Robots: Human-Performance Issues and User-Interface Design." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **41**(4):435–454, July 2011.

[CC12] Y. S. Chen and Ming-Te Chao. "Skeletonization based on the medial-axis and symmetry information." In *2012 International Conference on Machine Learning and Cybernetics*, volume 4, pp. 1627–1631, July 2012.

[CCC13] T. L. Chen, M. Ciocarlie, S. Cousins, P. M. Grice, K. Hawkins, K. Hsiao, C. C. Kemp, C. H. King, D. A. Lazewatsky, A. E. Leeper, H. Nguyen, A. Paepcke, C. Pantofaru, W. D. Smart, and L. Takayama. "Robots for humanity: using assistive robotics to empower people with disabilities." *IEEE Robotics Automation Magazine*, **20**(1):30–39, March 2013.

[CCM13] A. Cowley, B. Cohen, W. Marshall, C. J. Taylor, and M. Likhachev. "Perception and motion planning for pick-and-place of dynamic objects." In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 816–823, Nov 2013.

[CDS10] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard. "Learning and Reproduction of Gestures by Imitation." *IEEE Robotics Automation Magazine*, **17**(2):44–54, June 2010.

[CHL12] M. Ciocarlie, K. Hsiao, A. Leeper, and D. Gossow. "Mobile manipulation through an assistive home robot." In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5313–5320, Oct 2012.

[CTC12] Nutan Chen, Keng Peng Tee, Chee-Meng Chew, and Rui Yan. "Intuitive Interaction for Robotic Grasping." In *Proceedings of the Workshop at SIGGRAPH Asia*, WASA '12, pp. 105–111, New York, NY, USA, 2012. ACM.

[CWF10] Andrew Correa, Matthew R. Walter, Luke Fletcher, Jim Glass, Seth Teller, and Randall Davis. "Multimodal Interaction with an Autonomous Forklift." In *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction*, HRI '10, pp. 243–250, Piscataway, NJ, USA, 2010. IEEE Press.

[CZS13] Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. "3Sweep: Extracting Editable Objects from a Single Photo." *ACM Trans. Graph.*, **32**(6):195:1–195:10, nov 2013.

[DCQ16] J. Denny, J. Colbert, H. Qin, and N. M. Amato. "On the theory of user-guided planning." In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4794–4801, Oct 2016.

[DS13] Anca D Dragan and Siddhartha S Srinivasa. "A Policy-blending Formalism for Shared Control." *Int. J. Rob. Res.*, **32**(7):790–805, June 2013.

[DSA18] Jory Denny, Read Sandström, and Nancy M. Amato. *A General Region-Based Framework for Collaborative Planning*, pp. 563–579. Springer International, Cham, 2018.

[DSK12] P. Dames, M. Schwager, V. Kumar, and D. Rus. "A decentralized control policy for adaptive information gathering in hazardous environments." In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 2807–2813, Dec 2012.

[EKR13] L. Einig, D. Klimentjew, S. Rockel, L. Zhang, and J. Zhang. "Parallel plan execution and re-planning on a mobile robot using state machines with HTN planning systems." In *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, pp. 151–157, Dec 2013.

[ES14] M. Elbanhawi and M. Simic. "Sampling-Based Robot Motion Planning: A Review." *IEEE Access*, **2**:56–77, 2014.

[FAJ16] M. Firman, O. M. Aodha, S. Julier, and G. J. Brostow. "Structured Prediction of Unobserved Voxels from a Single Depth Image." In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5431–5440, June 2016.

[FM14] J. Fasola and M. J. Matari. "Interpreting instruction sequences in spatial language discourse with pragmatics towards natural human-robot interaction." In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2720–2727, May 2014.

[Fon01] Terrence W Fong. *Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, November 2001.

[FON12] H. C. Fang, S. K. Ong, and A. Y. C. Nee. "Interactive Robot Trajectory Planning and Simulation Using Augmented Reality." *Robot. Comput.-Integr. Manuf.*, **28**(2):227–237, April 2012.

[FWR10] Udo Frese, René Wagner, and Thomas Röfer. "A SLAM Overview from a User's Perspective." *KI - Künstliche Intelligenz*, **24**(3):191–198, 2010.

[FY07] Xiaocong Fan and John Yen. "Realistic Cognitive Load Modeling for Enhancing Shared Mental Models in Human-agent Collaboration." In *Proceedings of the 6th*

International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '07, pp. 60:1–60:8, New York, NY, USA, 2007. ACM.

[GN92] Naresh Gupta and Dana S. Nau. "On the complexity of blocks-world planning." *Artificial Intelligence*, **56**(2):223 – 254, 1992.

[GS08] Cheng Guo and Ehud Sharlin. "Exploring the use of tangible user interfaces for human-robot interaction: a comparative study." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pp. 121–130, New York, NY, USA, 2008. ACM.

[GSR14] A. Gaschler, M. Springer, M. Rickert, and A. Knoll. "Intuitive robot tasks with augmented reality and virtual obstacles." In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6026–6031, May 2014.

[HBP13] K. Hausman, F. Balint-Benczedi, D. Pangercic, Z. C. Marton, R. Ueda, K. Okada, and M. Beetz. "Tracking-based interactive segmentation of textureless objects." In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 1122–1129, May 2013.

[HCB13] H. B. Hmida, C. Cruz, F. Boochs, and Nicolle. "Knowledge base approach for 3d objects detection in point clouds using 3d processing and specialists knowledge." 2013.

[HDT07] Anton van den Hengel, Anthony Dick, Thorsten Thormählen, Ben Ward, and Philip H. S. Torr. "VideoTrace: Rapid Interactive Scene Modelling from Video." In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[HKB08] G. Hoffman, R. Kubat, and C. Breazeal. "A hybrid control system for puppeteering a live robotic stage actor." In *RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 354–359, Aug 2008.

[HM11] F. Hashikawa and K. Morioka. "Mobile robot navigation based on interactive SLAM with an intelligent space." In *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*, pp. 788–789, Nov 2011.

[Hol79] S. Holm. "A simple sequentially rejective multiple test procedure." *Scandinavian Journal of Statistics*, **6**:65–70, 1979.

[IPK03] S. Iba, C. J. J. Paredis, and P. K. Khosla. "Intention aware interactive multimodal robot programming." In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, pp. 3479–3484 vol.3, Oct 2003.

[IZI09] Kentaro Ishii, Shengdong Zhao, Masahiko Inami, Takeo Igarashi, and Michita Imai. *Human-Computer Interaction – INTERACT 2009: 12th IFIP TC 13 International Conference, Uppsala, Sweden, August 24-28, 2009, Proceedings, Part II*, chapter Designing Laser Gesture Interface for Robot Control, pp. 479–492. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[JQN12] Z. Ji, R. Qiu, A. Noyvirt, A. Soroka, M. Packianather, R. Setchi, D. Li, and S. Xu. "Towards automated task planning for service robots using semantic knowledge representation." In *IEEE 10th International Conference on Industrial Informatics*, pp. 1194–1201, July 2012.

[KAE12] O. Kroemer, H. Ben Amor, M. Ewerton, and J. Peters. "Point cloud completion using extrusions." In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pp. 680–685, Nov 2012.

[KAS13] K. H. Khokar, R. Alqasemi, S. Sarkar, and R. V. Dubey. "Human motion intention based scaled teleoperation for orientation assistance in preshaping for grasping." In *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, pp. 1–6, June 2013.

[KB12] J. Koenemann and M. Bennewitz. "Whole-body imitation of human motions with a Nao humanoid." In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, pp. 425–425, March 2012.

[KB17] G. Kazhoyan and M. Beetz. "Programming robotic agents with action descriptions." In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 103–108, Sept 2017.

[KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. "Poisson Surface Reconstruction." In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pp. 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[Ken12] Ben Kenwright. "Inverse Kinematics - Cyclic Coordinate Descent (CCD)." *J. Graphics Tools*, **16**(4):177–217, 2012.

[KF11] Sertac Karaman and Emilio Frazzoli. "Sampling-based Algorithms for Optimal Motion Planning." *Int. J. Rob. Res.*, **30**(7):846–894, June 2011.

[KLM96] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. "Reinforcement Learning: A Survey." *J. Artif. Int. Res.*, **4**(1):237–285, May 1996.

[KMD10] Tobias Kaupp, Alexei Makarenko, and Hugh Durrant-Whyte. "Human-robot Communication for Collaborative Decision Making - A Probabilistic Approach." *Robot. Auton. Syst.*, **58**(5):444–456, May 2010.

[KMM12] U. Klank, L. Msenlechner, A. Maldonado, and M. Beetz. “Robots that validate learned perceptual models.” In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 4456–4462, May 2012.

[KPL14] S. L. Koh, K. Pfeil, and J. J. LaViola. “Enhancing the robot avateering metaphor discreetly with an assistive agent and its effect on perception.” In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 1095–1102, Aug 2014.

[KSC17] David Kent, Carl Saldanha, and Sonia Chernova. “A Comparison of Remote Robot Teleoperation Interfaces for General Object Manipulation.” In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, HRI ’17, pp. 371–379, New York, NY, USA, 2017. ACM.

[KSL15] Z. Kootbally, C. Schlenoff, C. Lawler, T. Kramer, and S.K. Gupta. “Towards Robust Assembly with Knowledge Representation for the Planning Domain Definition Language (PDDL).” *Robot. Comput.-Integr. Manuf.*, **33**(C):42–55, June 2015.

[KWL05] J. Kofman, Xianghai Wu, T. J. Luu, and S. Verma. “Teleoperation of a robot manipulator using a vision-based human-robot interface.” *IEEE Transactions on Industrial Electronics*, **52**(5):1206–1219, Oct 2005.

[KXD12] Alexander Kasper, Zhixing Xue, and Rdiger Dillmann. “The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics.” *The International Journal of Robotics Research*, **31**(8):927–934, 2012.

[LFP12] M. Lichtenstern, M. Frassl, B. Perun, and M. Angermann. “A prototyping environment for interaction between a human and a robotic multi-agent system.” In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, pp. 185 –186, march 2012.

[LHC12] Adam Eric Leeper, Kaijen Hsiao, Matei Ciocarlie, Leila Takayama, and David Gossow. “Strategies for Human-in-the-loop Robotic Grasping.” In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI ’12, pp. 1–8, New York, NY, USA, 2012. ACM.

[LME11] M. Liu, A. Micaelli, P. Evrard, A. Escande, and C. Andriot. “Interactive dynamics and balance of a virtual character during manipulation tasks.” In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1676–1682, May 2011.

[LOM14] G. H. Lim, M. Oliveira, V. Mokhtari, S. Hamidreza Kasaei, A. Chauhan, L. Seabra Lopes, and A. M. Tom. "Interactive teaching and experience extraction for learning about objects and robot activities." In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 153–160, Aug 2014.

[LSI11] Kexi Liu, Daisuke Sakamoto, Masahiko Inami, and Takeo Igarashi. "Roboshop: Multi-layered Sketching Interface for Robot Housework Assignment and Management." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 647–656, New York, NY, USA, 2011. ACM.

[LSW17] D. Li, T. Shao, H. Wu, and K. Zhou. "Shape Completion from a Single RGBD Image." *IEEE Transactions on Visualization and Computer Graphics*, **23**(7):1809–1822, July 2017.

[LTS13] B. Lewis, B. Tastan, and G. Sukthankar. "An adjustable autonomy paradigm for adapting to expert-novice differences." In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1656–1662, Nov 2013.

[LW79] Tomás Lozano-Pérez and Michael A. Wesley. "An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles." *Commun. ACM*, **22**(10):560–570, October 1979.

[MAK12] H. Mikita, H. Azuma, Y. Kakiuchi, K. Okada, and M. Inaba. "Interactive symbol generation of task planning for daily assistive robot." In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pp. 698–703, Nov 2012.

[MCS10] M. Martinez, A. Collet, and S. S. Srinivasa. "MOPED: A scalable and low latency object recognition and pose estimation system." In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2043–2049, May 2010.

[MGP10] Balint Miklos, Joachim Giesen, and Mark Pauly. "Discrete Scale Axis Representations for 3D Geometry." In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, pp. 101:1–101:10, New York, NY, USA, 2010. ACM.

[Mil87] R.K. Miller. *Industrial robot handbook.* Fairmont Press, 1987.

[MKC03] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen. "Automatic grasp planning using shape primitives." In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pp. 1824–1829 vol.2, Sept 2003.

[MTA00] Benoit Moreau, Nicola Tomatis, and Kai O. Arras. "Multimodal Web Interface for Task Supervision and Specification." In *Proc. of SPIE, vol. 4195, Mobile Robots XV and Telemanipulator and Telepresence Technologies VII*, Boston, USA, 2000.

[MTG18] Abhijit Makhal, Federico Thomas, and Alba Perez Gracia. "Grasping Unknown Objects in Clutter by Superquadric Representation." *2018 Second IEEE International Conference on Robotic Computing (IRC)*, pp. 292–299, 2018.

[MVJ14] S. Morante, J. G. Victores, A. Jardn, and C. Balaguer. "On using guided motor primitives to execute Continuous Goal-Directed Actions." In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 613–618, Aug 2014.

[NCH13] H. Nguyen, M. Ciocarlie, K. Hsiao, and C. C. Kemp. "ROS commander (ROSCo): Behavior creation for home robots." In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 467–474, May 2013.

[NGP13] S. E. Navarro, J. Grter, H. Pauer, B. Hein, and H. Wrn. "Partial view geometric reconstruction of objects with rotational and planar symmetries for grasping tasks." In *Advanced Robotics (ICAR), 2013 16th International Conference on*, pp. 1–8, Nov 2013.

[NGT04] Dana Nau, Malik Ghallab, and Paolo Traverso. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[NL12] V. V. Nguyen and J. H. Lee. "Full-body imitation of human motions with kinect and heterogeneous kinematic structure of humanoid robot." In *System Integration (SII), 2012 IEEE/SICE International Symposium on*, pp. 93–98, Dec 2012.

[Nor02] Donald A. Norman. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA, 2002.

[NS11] Wai Shan Ng and E. Sharlin. "Collocated interaction with flying robots." In *RO-MAN, 2011 IEEE*, pp. 143 –149, 31 2011-aug. 3 2011.

[OMT03] R. Ohbuchi, T. Minamitani, and T. Takei. "Shape-similarity search of 3D models by using enhanced shape functions." In *Proceedings of Theory and Practice of Computer Graphics, 2003.*, pp. 97–104, June 2003.

[PG07] Joelle Pineau and Geoffrey J. Gordon. *Robotics Research: Results of the 12th International Symposium ISRR*, chapter POMDP Planning for Robust Robot Control, pp. 69–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[PGS17] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. "Grasp Pose Detection in Point Clouds." *I. J. Robotics Res.*, **36**:1455–1473, 2017.

[PHJ17] C. Paxton, A. Hundt, F. Jonathan, K. Guerin, and G. D. Hager. "CoSTAR: Instructing collaborative robots with behavior trees and vision." In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 564–571, May 2017.

[PK12] R. M. Pierce and K. J. Kuchenbecker. “A data-driven method for determining natural human-robot motion mappings in teleoperation.” In *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 169–176, June 2012.

[PKJ05] Hye Sun Park, Eun Yi Kim, Sang Su Jang, Se Hyun Park, Min Ho Park, and Hang Joon Kim. *Pattern Recognition and Image Analysis: Second Iberian Conference, IbPRIA 2005, Estoril, Portugal, June 7-9, 2005, Proceedings, Part I*, chapter HMM-Based Gesture Recognition for Robot Control, pp. 607–614. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[PKL13] Kevin Pfeil, Seng Lee Koh, and Joseph LaViola. “Exploring 3D Gesture Metaphors for Interaction with Unmanned Aerial Vehicles.” In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI ’13, pp. 257–266, New York, NY, USA, 2013. ACM.

[PPT12] D. Pangercic, B. Pitzer, M. Tenorth, and M. Beetz. “Semantic Object Maps for robotic housework - representation, acquisition and use.” In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4644–4651, Oct 2012.

[PSA01] Dennis Perzanowski, Alan C. Schultz, William Adams, Elaine Marsh, and Magda Bugajska. “Building a Multimodal Human-Robot Interface.” *IEEE Intelligent Systems*, **16**(1):16–21, January 2001.

[PSB11] B. Pitzer, M. Styer, C. Bersch, C. DuHadway, and J. Becker. “Towards perceptual shared autonomy for robotic mobile manipulation.” In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 6245–6251, May 2011.

[PSW00] R. Parasuraman, T. B. Sheridan, and C. D. Wickens. “A model for types and levels of human interaction with automation.” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, **30**(3):286–297, May 2000.

[PTP12] Dejan Pangercic, Moritz Tenorth, Benjamin Pitzer, and Michael Beetz. “Semantic Object Maps for Robotic Housework - Representation, Acquisition and Use.” In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[QAC16] A. H. Quispe, H. B. Amor, and H. I. Christensen. “Combining arm and hand metrics for sensible grasp selection.” In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1170–1176, Aug 2016.

[QCG09] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully B. Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. “ROS: an open-source Robot Operating System.” In *ICRA Workshop on Open Source Software*, 2009.

[QMG15] A. H. Quispe, B. Milville, M. A. Gutirrez, C. Erdogan, M. Stilman, H. Christensen, and H. B. Amor. "Exploiting symmetries and extrusions for grasping household objects." In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3702–3708, May 2015.

[RMB09] R. B. Rusu, Z. C. Marton, N. Blodow, A. Holzbach, and M. Beetz. "Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments." In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3601–3608, Oct 2009.

[RMM05] A. Rennuit, A. Micaelli, X. Merlhiot, C. Andriot, F. Guillaume, N. Chevassus, D. Chablat, and P. Chedmail. "Passive control architecture for virtual humans." In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1432–1437, Aug 2005.

[RTL06] Rodriguez, Xinyu Tang, Jyh-Ming Lien, and N. M. Amato. "An obstacle-based rapidly-exploring random tree." In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 895–900, May 2006.

[SAB07] Marjorie Skubic, Derek Anderson, Samuel Blisard, Dennis Perzanowski, and Alan Schultz. "Using a Hand-drawn Sketch to Control a Team of Robots." *Auton. Robots*, **22**(4):399–410, May 2007.

[SBB12] J. Stckler, N. Biresev, and S. Behnke. "Semantic mapping using object-class segmentation of RGB-D images." In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3005–3010, Oct 2012.

[SC11] H. B. Suay and S. Chernova. "Humanoid robot control using depth camera." In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 401–401, March 2011.

[SCC14] L. She, Y. Cheng, J. Y. Chai, Y. Jia, S. Yang, and N. Xi. "Teaching Robots New Actions through Natural Language Instructions." In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 868–873, Aug 2014.

[SDB13] C. Schou, J. S. Damgaard, S. Bgh, and O. Madsen. "Human-robot interface for instructing industrial tasks using kinesthetic teaching." In *IEEE ISR 2013*, pp. 1–6, Oct 2013.

[SEB12] A. Sahbani, S. El-Khoury, and P. Bidaud. "An overview of 3D object grasp synthesis algorithms." *Robotics and Autonomous Systems*, **60**(3):326 – 336, 2012. Autonomous Grasping.

[SES14a] D. Seidel, C. Emmerich, and J. J. Steil. “Model-free path planning for redundant robots using sparse data from kinesthetic teaching.” In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4381–4388, Sept 2014.

[SES14b] D. Seidel, C. Emmerich, and J. J. Steil. “Model-free path planning for redundant robots using sparse data from kinesthetic teaching.” In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4381–4388, Sept 2014.

[She92] Thomas B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control.* MIT Press, Cambridge, MA, USA, 1992.

[SHI09] Daisuke Sakamoto, Koichiro Honda, Masahiko Inami, and Takeo Igarashi. “Sketch and Run: A Stroke-based Interface for Home Robots.” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’09, pp. 197–200, New York, NY, USA, 2009. ACM.

[SJP02] Paul Scerri, Lewis Johnson, David V Pynadath, Paul Rosenbloom, Nathan Schurr, Mei Si, and Milind Tambe. “Getting Robots, Agents and People to Cooperate: an Initial Report.” In *Getting Robots, Agents and People to Cooperate: an Initial Report*, 2002.

[SM06] R. Schraft and C. Meyer. “The need for an intuitive teaching method for small and medium enterprises.” In *ISR 2006 - ROBOTIK 2006, ser. VDI-Berichte, vol. 1956. VDI-Wissensforum*, p. 95, 2006.

[SPO12] B. Sankaran, B. Pitzer, and S. Osentoski. “Failure recovery with shared autonomy.” In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 349–355, Oct 2012.

[SPT01] Paul Scerri, David Pynadath, and Milind Tambe. “Adjustable Autonomy in Real-world Multi-agent Environments.” In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS ’01, pp. 300–307, New York, NY, USA, 2001. ACM.

[SSC10] D. Shah, J. Schneider, and M. Campbell. “A robust sketch interface for natural robot control.” In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 4458–4463, Oct 2010.

[SSN14] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel. “BigBIRD: A large-scale 3D database of object instances.” In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 509–516, May 2014.

[SSW10] Yuta Sugiura, Daisuke Sakamoto, Anusha Withana, Masahiko Inami, and Takeo Igarashi. "Cooking with Robots: Designing a Household System Working in Open Environments." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 2427–2430, New York, NY, USA, 2010. ACM.

[SWW11] Julie Shah, James Wiken, Brian Williams, and Cynthia Breazeal. "Improved Human-robot Team Performance Using Chaski, a Human-inspired Plan Execution System." In *Proceedings of the 6th International Conference on Human-robot Interaction*, HRI '11, pp. 29–36, New York, NY, USA, 2011. ACM.

[SYC14] Lanbo She, Shaohua Yang, Yu Cheng, Yunyi Jia, Joyce Chai, and Ning Xi. "Back to the Blocks World: Learning New Actions through Situated Human-Robot Dialogue." In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pp. 89–97, Philadelphia, PA, U.S.A., June 2014. Association for Computational Linguistics.

[TB13] Moritz Tenorth and Michael Beetz. "KnowRob: A knowledge processing infrastructure for cognition-enabled robots." *The International Journal of Robotics Research*, **32**(5):566–590, 2013.

[TFC12] G. Taylor, R. Frederiksen, J. Crossman, M. Quist, and P. Theisen. "A multimodal intelligent user interface for supervisory control of unmanned platforms." In *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, pp. 117–124, May 2012.

[TS10] F. Tombari and L. Di Stefano. "Object Recognition in 3D Scenes with Occlusions and Clutter by Hough Voting." In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pp. 349–355, Nov 2010.

[TW05] S. Thrun and B. Wegbreit. "Shape from symmetry." In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pp. 1824–1831 Vol. 2, Oct 2005.

[TZC09] Andrea Tagliasacchi, Hao Zhang, and Daniel Cohen-Or. "Curve Skeleton Extraction from Incomplete Point Cloud." In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pp. 71:1–71:9, New York, NY, USA, 2009. ACM.

[VAD12] N. Vahrenkamp, T. Asfour, and R. Dillmann. "Simultaneous Grasp and Motion Planning: Humanoid Robot ARMAR-III." *IEEE Robotics Automation Magazine*, **19**(2):43–57, June 2012.

[VKW18] N. Vahrenkamp, E. Koch, M. Wchter, and T. Asfour. "Planning High-Quality Grasps Using Mean Curvature Object Skeletons." *IEEE Robotics and Automation Letters*, **3**(2):911–918, April 2018.

[WKZ16] K. Wolff, C. Kim, H. Zimmer, C. Schroers, M. Botsch, O. Sorkine-Hornung, and A. Sorkine-Hornung. "Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction." In *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 118–127, Oct 2016.

[WMO12] M. Wongphati, Y. Matsuda, H. Osawa, and M. Imai. "Where do you want to use a robotic arm? And what do you want from the robot?" In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 322–327, Sept 2012.

[WNU09] C. Weber, V. Nitsch, U. Unterhinninghofen, B. Farber, and M. Buss. "Position and force augmentation in a telepresence system and their effects on perceived realism." In *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, pp. 226–231, March 2009.

[WOZ10] M. Weser, D. Off, and J. Zhang. "HTN robot planning in partially observable dynamic environments." In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 1505–1510, May 2010.

[ZC14] Z. Zhao and X. Chen. "Semantic mapping for object category and structural class." In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 724–729, Sept 2014.

[ZZL11] J. Zhang, Y. Zeng, H. Liu, Y. Hu, and J. Zhang. "MBHP: A memory-based method on robot planning under uncertainties." In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pp. 366–371, Dec 2011.

[ZZL12] Y. Zeng, J. Zhang, W. Lu, Y. Hu, and J. Zhang. "Memory-based hierarchical task and motion planning." In *Information and Automation (ICIA), 2012 International Conference on*, pp. 412–417, June 2012.