Modeling Dynamic Spatial Relations with Global Properties for Natural Language-Based Human-Robot Interaction*

Juan Fasola and Maja J Matarić, *Fellow, IEEE*

Abstract— **We present a methodology for the representation of dynamic spatial relations (DSRs) with global properties as part of an approach for enabling robots to follow natural language commands from non-expert users, with particular focus on the development of spatial language primitives. Our approach to modeling DSRs is based on related research in the fields of linguistics, cognitive science, and neuroscience, and contributes novel extensions to the semantic field model of spatial prepositions. We describe novel representations of the DSRs for "to", "through", and "around", discuss their applicability in path classification scenarios, and provide implementation details of path generation routines instantiating these DSRs for use in robot task planning. The paper concludes with an evaluation of our robot architecture implemented on a simulated mobile robot in a 2D home environment.**

I. Introduction

For autonomous service robots to become ubiquitous in household environments, they will need to be capable of interacting with and learning from non-expert users in a manner that is both natural and practical for the users. In particular, these robots will need to be capable of understanding natural language instructions in order to learn new tasks and receive guidance and feedback on task execution. This necessity is especially evident in assistive contexts, where the robots are interacting with people with disabilities, age-related (e.g., reduced mobility, limited eyesight) or otherwise (e.g., individuals post-stroke), as the users may not be able to teach the robot new tasks and/or provide feedback by demonstration.

Spatial language plays an important role in instruction-based natural language communication [18]. Spatial relations, both dynamic and static, expressed in language are often expressed by prepositions [1]. Therefore, the ability for robots to understand and differentiate between spatial prepositions in spoken language is crucial for their interaction with the user to be successful. Prepositions in English, as well as in many other languages, are identified as a closed class: there are only 80-100 prepositions (approximate count, as many are polysemous) and new words are not being added. The relatively small number of prepositions, combined with their extensive use in spatially oriented natural language communication across domains, makes the construction of spatial primitives *a priori* based on prepositions for autonomous service robots not only feasible, but also intuitive and beneficial.

Our previous work presented a framework for modeling DSRs with local properties (e.g., along), and demonstrated its applicability in robot execution of single commands with and without user-provided constraints [5, 6].

In this paper, we extend upon our previous work and present a general methodology for the representation of dynamic spatial relations (DSRs) with global properties, based on the semantic field model of prepositions [3], for use as spatial primitives in a robot architecture that enables autonomous service robots to follow natural language commands from non-expert users. We present novel representations of the DSRs for "to", "through", and "around", discuss their applicability in path classification scenarios, and provide implementation details of path generation routines instantiating these DSRs for use in robot task planning. Furthermore, we describe extensions to the planning process that allows for seamless execution of multi-step command sequences. We conclude the paper with an evaluation of our robot architecture implemented on a simulated mobile robot in a 2D home environment.

II. Related Work

Previous work that has investigated the use of spatial prepositions, and spatial language in general, includes the work of Skubic et al. [10], who demonstrated a robot capable of understanding static spatial relations in natural language instruction. Sandamirskaya et al. [9] investigated the use of Dynamic Neural Fields theory in a static spatial language architecture for use in human-robot cooperation tasks on a tabletop workspace. Similarly, the use of computational fields for static relations was implemented in a visually situated dialogue system by Kelleher and Costello [15]. These works all implemented pre-defined notions of spatial relations, however, researchers have also investigated learning these types of static spatial relations automatically from training data both on- and offline (e.g., [11, 12, 16, 17]). Our work aims to extend upon this related work by encoding not only static spatial relations for natural language instruction understanding, but also DSRs involving paths, as discussed in the next section.

In the context of natural language robot instruction, however, the use of DSRs has in fact been explored by recent work. Tellex et al. [7] developed a probabilistic graphical model to infer task/actions for execution by a forklift robot from natural language commands. Kollar et al. [8] developed a Bayesian framework for interpreting route directions on a mobile robot. In both of these works there

* Research supported by National Science Foundation grants IIS-0713697, CNS-0709296, and IIS-1117279.

J. Fasola is with the University of Southern California, Los Angeles, CA 90089 USA (e-mail: fasola@usc.edu).

M. J. Matarić is with the University of Southern California, Los Angeles, CA 90089 USA (e-mail: mataric@usc.edu).

was no explicit definition of the spatial relations used, static or otherwise, and instead they were learned from labeled training data. However, these approaches typically require the programmer to provide an extensive training data set of natural language input for each new application context, without taking advantage of the domain-independent nature of spatial prepositions. Our proposed approach develops novel, pre-defined templates for dynamic spatial relations, that facilitate use and understanding across domains, and whose computational representations enable guided agent execution planning.

Researchers have also explored mapping natural language instructions into a formal agent control language using a variety of types of parsers, including those that were constructed manually [13, 14], learned from training data [19, 26], and learned iteratively through interaction [20]. Of these, the work of Rybski et al. [13], Matuszek et al. [19], and Atrash et al. [26] rely on pre-defined agent behaviors as primitives, as opposed to spatial relations, which hinders, if not prohibits, the ability for a user to provide feedback modifications and/or constraints regarding agent execution of a specific primitive behavior. The work of Kress-Gazit et al. [14] and Cantrell et al. [20] leave the specification of primitives, from which new behaviors are learned, largely as an open research problem. However, the parsers used in their systems map words to meanings based on dictionary-based rules. Our methodology employs domain-generalizable spatial relations as primitives, and probabilistic reasoning for the grounding and semantic interpretation of phrases, thereby allowing for context-based instruction understanding and user-feedback modifiable agent execution paths.

III. Modeling Dynamic Spatial Relations

Our methodology for representing DSRs for use in natural language human-robot communication domains focuses specifically on the modeling of path prepositions, as spatial relations characterized in language are often expressed by prepositions [1]. Furthermore, spatial verbs that incorporate spatial relations can generally be substituted by a simpler verb and a preposition (e.g. "Leave" → "Go *out of*") [2]. Path prepositions include, among others: to, from, along, across, through, toward, around, into, onto, out of, and via. The discussion in this section focuses on the representation of the paths for "to", "through", and "around"; nevertheless, the methodology discussed is general and can readily be applied to represent additional DSRs.

Fundamentally, our approach encodes spatial language within the robot *a priori* as primitives. Static spatial relation primitives are represented using the *semantic field* model proposed by O'Keefe [3], where the semantic fields of static prepositions, parameterized by figure and reference objects, assign weight values to points in the environment depending on how accurately they capture the meaning of the preposition (e.g., for the static spatial preposition 'near', points closer to an object have higher weight). Our prior work has shown that this semantic field model can be extended to represent simple DSRs that encode optimal path direction at a local level (e.g., along, toward) [5, 6]. However, DSRs with meanings that enforce path characteristics at a global level require a more complex semantic representation [1, 21].

In the following subsections we will discuss: (1) a general framework for representing DSR primitives that encode global properties, and (2) example representations for the DSRs of "to", "through", and "around" to illustrate our approach.

A. General Representation for DSRs with Global Properties

To represent DSRs with global constraints, our approach identifies four classical AI conditions that each DSR may subscribe to, they are: 1) *pre-condition*, 2) *post-condition*, 3) *continuing-condition*, and 4) *intermediate-condition*. This type of condition-based approach to modeling path prepositions is based on findings in linguistics and cognitive science research on the constraint-based meanings of path prepositions [1, 21]. The methodology is akin to methods developed in the learning by imitation community for task and verb modeling (e.g., [23, 24, 25]). A unique characteristic of our approach, however, is that each condition is represented (typically) by either a semantic field, or by another DSR (which is in turn represented by semantic fields). In the representation, each DSR may have none, one, or multiple of each of the four conditions.

The four conditions enumerated were developed to operate over paths. Formally, a *path* is defined as an ordered set of points (i.e. path $P = \{p_0, p_1, p_2\ldots, p_n\}$), connected by (implicit) direction vectors (from p_i to p_{i+1}). Pre- and post-conditions must be satisfied for the start (p_0) and end (p_n) of the path, respectively. Intermediate-conditions must be satisfied for at least one point in the path, and continuing-conditions must be satisfied for all points in the path. Following this methodology, our condition-based DSR representation may be used both for path classification (e.g., during task learning by demonstration), discussed below, and for path generation (e.g., during robot task execution planning), discussed in Section V.

B. "To" Representation

To illustrate our approach for modeling DSRs, consider the path preposition "to". From linguistics literature, we understand that the path specified by "to" terminates *at* the reference region [1]. As a result, the DSR representation for "to" in our approach has a single post-condition containing the semantic field for the static spatial relation *at*:

$$to(x) = \begin{cases} \text{pre-condition:} & - \\ \text{cont-condition:} & - \\ \text{int-condition:} & - \\ \text{post-condition:} & at(x) \end{cases} \quad (1)$$

Note that because *at* is represented by a semantic field, it does not return a truth value. Instead, $at(x)$ is a function from points to weight values ($\mathbb{R}[0,1]$). Following is an example semantic field equation for *at*:

$$at(x)(p) = \exp[-(dist(x,p)^2)/2\sigma^2] \quad (2)$$

Where $dist(x,p)$ returns the minimum distance between the reference object x and point p; σ is the width of the field

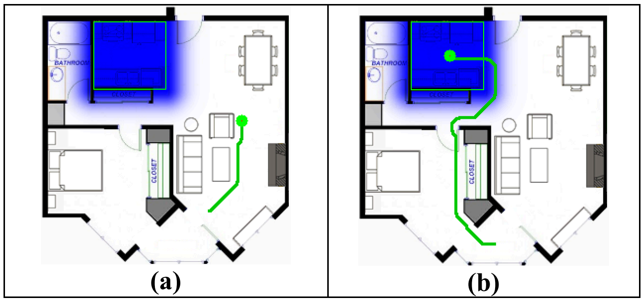

Figure 1. Two example paths for "to the kitchen". (a) Path value = 1.8×10^{-10}; (b) Path value = 1.0. Note: σ = robot width × 2.5

(dropoff parameter) which is context-dependent. By representing conditions as semantic fields, our approach facilitates probabilistic reasoning over paths; an essential quality for path classification, grounding, and generation. As an example, Fig. 1 shows two sample paths with classification results for the phrase "to the kitchen", while also displaying the field for *at*(*the kitchen*). The path values reported correspond to the *at* semantic field values for the path end points (i.e. the post-condition for *to*). As is evident by the results, one path is more acceptable than the other in capturing the meaning of the stated prepositional phrase.

DSRs closely related to that of "to" also have similar representations. For example, the representation for "from" is the reverse of that for "to", with the *at* field instead being set as a pre-condition. Additionally, the DSR representations for "into", "onto", and "out of" are all special cases of *to*, with the *at* field post-condition being replaced by the semantic fields for *in*, *on*, and *out*, respectively.

It is important to note that the DSR representation for "to" is versatile: although *at* is listed as the default post-condition, this determination may change based on context. As an example, consider the phrase "Stand beside the bed". Here, the (implicit) path relation is *to* and the static relation is *beside*. Hence, the post-condition for *to* would instead be set to the semantic field for *beside*. This substitution is appropriately handled in our methodology by the semantic interpretation module (discussed in Section IV), which infers path and static relations probabilistically given the natural language input.

C. "Through" Representation

The path preposition "through" has a few different semantic interpretations according to the linguistics literature. Therefore, in our approach we developed separate DSR representations for each of them. The most general interpretation asserts that "through" specifies a path with at least one point *in* the reference object [1]. This definition for "through" can most aptly be characterized by a global DSR representation with a single intermediate-condition containing the semantic field for *in* (see $through_1$ in Table I).

The remaining two interpretations considered by our methodology are both special cases of the first, more general, definition. The second interpretation depends on the topology of the reference object in that it requires that the start and end points, respectively, be coincident with boundaries at separate ends of the reference object [4]. This definition imposes a path traversing the inside of the reference object, end-to-end. Example uses include "Go through the doorway" and "Walk through the tunnel". To correctly model this interpretation, the topology (i.e. boundary connectivity) of the reference object must first be determined. While the implementation may vary according to the domain, determining the discrete entrance boundaries for a particular reference object is fairly straightforward in 2D/3D by evaluating edge connectivity. As an example, Fig. 2 shows the extracted topology of a hallway reference object in a simulated 2D home environment, displaying three separate entrance boundaries. Using the extracted topology, this second definition for "through" can be represented by a DSR with pre- and post-conditions each specifying points at different entrance boundaries, and with a single continuing-condition set to the semantic field for *in*. Table I presents this representation as $through_2$, with $B_i(x)$ representing the set of all points at entrance boundary i of reference object x.

The third semantic interpretation for "through" is similar to the second except that the path traverses an unbounded segment of the reference object (i.e. does not terminate at object boundaries) [4]. Thus, this definition simply imposes a path along the inside of the reference object. The DSR representation for this third definition of "through" contains two continuing-conditions with the semantic fields for *in* and *along*, respectively (see $through_3$ in Table I). The *along* semantic field is used in this representation to promote paths that travel parallel to the major axis of the reference object [5] such as to avoid boundedness. Paths that instead travel parallel to the minor axis would be more appropriate for the DSR for "across" [1, 4], whose representation in our approach is very similar to that for "through" albeit with the aforementioned distinction.

D. "Around" Representation

The path preposition "around", much like "through", is polysemous. According to Talmy, "around" denotes a circumcentric path (i.e. curved about a center) that can be either revolutional or rotational [4]. Both path types are

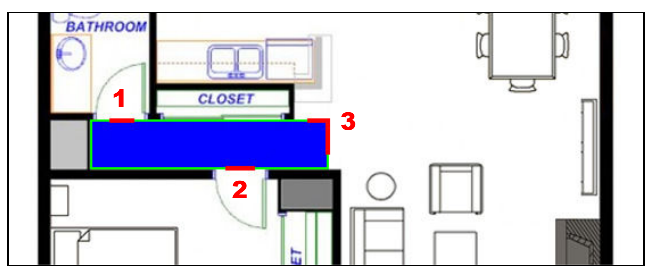

Figure 2. Topology of hallway in 2D home environment showing three (configuration space) entrance boundaries.

TABLE I. DSR REPRESENTATIONS FOR "THROUGH"

Conditions	$through_1(x)$	$through_2(x)$	$through_3(x)$
Pre-	-	$at(B_i(x))$	-
Continuing-	-	$in(x)$	$in(x)$, $along(x)$
Intermediate-	$in(x)$	-	-
Post-	-	$at(B_{j\neq i}(x))$	-

similar, except that revolutional paths refer to curved figure paths about a central reference object (e.g., "The boat sailed around the island"), whereas rotational paths denote a change in orientation of the figure itself (e.g., "John spun around") [1]. In the latter case, the figure can also be thought of as the movement path of a point (or points) within the reference object itself during its rotation, thereby illustrating the similarity between the two path types.

In order to represent the DSR for "around", our approach makes use of a novel semantic field that was developed to quantify the circumcentric nature of a given path. Specifically, the *circumcentric* semantic field maps paths to weight values; where weights are assigned according to the degree to which the orientation of the path changes relative to the center of the specified reference object, from start point to end point. The computation of this field is outlined in Algorithm I, and Fig. 3 shows an example path with its corresponding *circumcentric* field value for illustration.

In representing the two types of circumcentric paths for "around", it is important to consider that there are also two termination conditions per path type that are commonly expressed in language: half circle (180°), and full circle (360°). Examples include, as noted by Landau and Jackendoff: "Go all the way around …" (360°) vs. "Detour around…" (180°) [1].

Under these specifications, there are a total of four DSR representations for "around": two revolutional (half/full circle), and two rotational (half/full circle). Table II presents all four representations, sequentially labeled as $around_{1\text{-}4}$. The continuing-condition for each representation utilizes the semantic field for *in* to express whether or not the figure path is within (i.e. part of) the reference object. Additionally, the post-condition for each representation contains the *circumcentric* semantic field, whose arguments depend on whether the ideal path is a half or full circle.

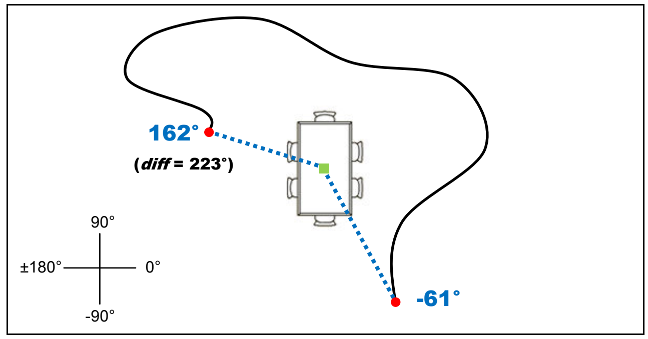

Figure 3. Path traveling around a dining table reference object, showing start and end orientations, with resulting *circumcentric* semantic field value = |223°|/360° = 0.61944.

ALGORITHM I. CIRCUMCENTRIC SEMANTIC FIELD COMPUTATION

circumcentric(x, P, $diff_{ideal}$)

1: $diff \leftarrow 0$
2: **for each** $i \in \{1,..,n\}$
3: $\quad diff_{new} \leftarrow \theta_{rel}(x,p_i) - \theta_{rel}(x,p_{i-1})$
4: $\quad diff \leftarrow diff + diff_{new}$
5: **return** $(\,|\,diff\,|\,/\,diff_{ideal}\,)$

Note. x is the reference object, P the path, $p_i \in P$, $\theta_{rel}(x,p)$ the orientation of point p relative to x, and $diff_{ideal}$ the ideal path orientation change relative to x from the start of the path to the end.

TABLE II. DSR REPRESENTATIONS FOR "AROUND"

Conditions	$around_{1,2}(x)$	$around_{3,4}(x)$
Pre-	-	-
Continuing-	$\neg in(x)$	$in(x)$
Intermediate-	-	-
Post-	$circumcentric(x, P, [180°_{1,3} \mid 360°_{2,4}])$	

IV. ROBOT ARCHITECTURE AND SYSTEM MODULES

We have developed a robot software architecture that incorporates our methodology for representing spatial language, and DSRs in particular, to enable natural language-based human-robot interaction with non-expert users. The architecture contains five system modules that enable the interpretation of natural language instructions, from speech or text-based input, and translation into robot action execution. They are: the syntactic parser, noun phrase (NP) grounding, semantic interpretation, planning, and action modules. In this section we will provide a brief overview of the system design and module functionality. For a complete description of the architecture and system modules, we refer the reader to [5].

The syntactic parser represents the entry point of our robot architecture, as it responsible for parsing the user-given natural language instruction into a format that the remaining modules can interpret. The instructions are provided as text strings, either by a speech recognizer (e.g., [22]) or keyboard-based input. Our system does not attempt to provide a solution for natural language processing in the general case, but instead focuses on well-formed English directives involving spatial language, for which we utilize a specialized grammar. After the syntax of the instruction has been determined, the parse tree is passed on to the grounding module which attempts to associate parsed NPs with known objects in the world. If it is successful, all observations are then passed on to the semantic interpreter for final instruction meaning association.

The semantic interpretation module utilizes a Bayesian approach and infers the semantics of the given instruction probabilistically using a database of learned mappings from input observations to instruction meanings. The four observation inputs to the module, and the output instruction semantics, are as follows:

semanticInterpreter(*verb*, *preposition* , *figure*, *ref. object*)
$\Rightarrow$ **(*command*, *DSR*, *static*)**

The input includes: the *verb* and *preposition* used in the instruction sentence, and the associated groundings for the specified *figure* and *reference objects* as determined by the NP grounding module. The resulting semantic output of the module includes: the *command* type, the *DSR* type, and the *static spatial relation* (if available). The command type is domain-specific, and may include commands such as: robot movement, speech output, learned tasks, etc. While the output specification was designed to represent the instruction of spatial tasks, the inference procedure utilized is general and can easily be modified or expanded to accommodate the requirements of the specific application domain, including the inference of non-spatial tasks.

Once the semantic interpreter has inferred the instruction semantics, the planning module attempts to find a solution for the robot given these command specifications, after which the solution is passed on to the action module for robot task execution.

V. Generating Paths for Dynamic Spatial Relations

In searching for robot action solutions for the interpreted command semantics, the planning module must consider not only the inferred command type and spatial relations, but also the *pragmatics* of the natural language instruction. These consist in the unvoiced constraints/specifications that accompany the spoken instructions and further specify the meaning that the speaker intends to convey; which can come from context, prior knowledge, norms, and other factors. Incorporation of specific pragmatic constraints during the planning process is a design decision that depends largely on the domain requirements.

In this section we present the implementations details of the DSR path generation procedures for *to*, *through*, and *around* used in our robot architecture for natural language instruction following. The procedures presented focus on robot movement commands, and illustrate how the representations of DSRs with global properties may be used (combined with the pragmatics of the specific instruction) for the purposes of path generation in robot task planning.

The A* search algorithm is the primary method used for both path planning and topology determination (discussed below) in the planning module; hence, the planner described operates over a discretized representation of the world space.

A. "To" Path Generation

The DSR representation for "to", as described in Section III, contains a single post-condition with the semantic field for *at*. Therefore, according to this representation, paths that satisfy the relation $to(x)$ are those whose endpoints satisfy the static spatial relation $at(x)$ (defined in (2)).

In the context of our robot architecture, x is the reference object identified during the grounding procedure for the given instruction. In our path generation procedure for *to*, the planner searches for the point p in the free space that maximizes the weight value $at(x)(p)$, which is a real-valued number in the range [0,1], and returns the shortest path to that location from the robot's current position as a solution.

The pragmatics in our procedure for *to* dictate that if there are multiple points in the free space with maximal weight values (i.e. equal to 1) that are inside the reference object (e.g. in a room), the planner should elect the point furthest from the object edges (i.e. most centrally located) as the end point of the solution path. Fig. 1(b) shows an example path for *to* generated under these conditions.

Finally, if the instruction given is part of a command sequence, the pragmatics indicate that expediency in the command solution is favored over optimality. Here, the procedure instead runs A* from the robot's current location to find the nearest point whose weight value exceeds a certain threshold (e.g., 80% of the maximum weight value in the free space), or whose distance within the reference object exceeds a minimum entry distance (e.g., 1 robot width).

B. "Through" Path Generation

Paths that satisfy the definition of $through_1$ (see Section III) are those with at least one point *in* the reference object x. To generate these types of paths, the planner simply generates paths $into(x)$ using the procedure described above for $to(x)$ and setting the post-condition to $in(x)$. However, use of the path preposition "through" in directives generally implies the DSR for $through_2$ or $through_3$.

The DSR representation $through_2$ has pre- and post-conditions that require points *at* separate entrance boundaries of x. In planning a solution for $through_2$, the planner first generates a path $into(x)$ to accomplish the pre-condition, and then generates a path $outof(x)$ with the added A* goal constraint that the exit boundary be different than the entrance boundary (determined using the extracted topology of x) to accomplish the post-condition. If at the start of planning the robot is already $in(x)$, the pre-condition is assumed to have been satisfied previously, and the planner subsequently generates a path $outof(x)$ without exit constraints. If there is only one entry boundary for x, the pragmatics dictate the path generation procedure change to that of $through_3$ before accomplishing the post-condition. In addition, if the instruction is part of a command sequence, once a path is generated to within some minimum distance inside x, the pragmatics change the path requirements to $through_1$ and planning for the next command is commenced.

The DSR representation $through_3$ specifies a path $in(x)$ and $along(x)$. To achieve such a path, the planner first generates a path $into(x)$ if the robot is not already $in(x)$, and then runs A* to find the furthest point away from the robot's location, that is still $in(x)$. The planner then generates a path to this point (staying $in(x)$) to accomplish the $along(x)$ continuing-condition (by default). Alternatively, a path $along(x)$ could be generated using the Voronoi graph of x.

C. "Around" Path Generation

Considering only the more complex revolutional cases of "around", the specific DSR implied depends largely on context, including for example, the topology of the region of space surrounding the reference object. For instance, a lack of 360° connectivity in the region could result in favoring a half circle (180°) interpretation for the DSR. In addition, determination of this topology is required in order to generate appropriate paths for the DSRs of "around".

To determine if the region of space surrounding reference object x contains 360° connectivity, the planner executes a breadth-first search starting at the robot's location. As the

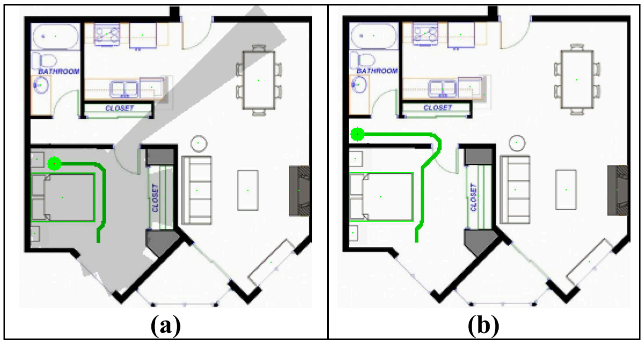

Figure 4. Two paths for "Go around the bed" with/without enforcing visibility region. (a) Path value = 0.857; (b) Path value = 0.895.

search progresses, the *circumcentric* semantic field value of each point in the free space is recorded (modified slightly from Algorithm I with the absolute value of *diff* removed to preserve signed path direction). If meet points are detected from the expanding wavefront of two paths from opposite directions with a combined path orientation difference of 360°, then the topology possesses 360° connectivity.

In the case of $around_2$ (full circle), once connectivity is determined the planner first generates a path to the nearest meet point. After which, the planner runs the breath-first search once again to find and plan a path to the nearest meet point on the opposite side of x, thus completing the 360° loop. If 360° connectivity is not available, the planner instead generates a path to the point with the maximum *circumcentric* field value (to maximize the post-condition weight). Similarly, in the case of $around_1$ (half circle), a path is generated to the point of maximum *circumcentric* field value (stopping at 180°).

Regarding the pragmatics of *around*, consider the instruction "Go around the bed" given by the user to a service robot co-located within the same room. Here, a likely unvoiced constraint is "Stay inside the room". To incorporate this constraint our planner enforces a global visibility constraint on the free space surrounding the reference object during search. Fig. 4 highlights the difference between paths generated with and without the visibility constraint, and illustrates its usefulness in practice with end-users. The path values reported correspond to the *circumcentric* semantic field values computed for the paths, with $diff_{ideal}$ = 180° (i.e. the post-condition for $around_1$).

VI. Evaluation

To evaluate the ability of our robot architecture to follow natural language directives, we conducted two separate test runs of the system testing the robot's ability to respond to multiple movement commands involving DSRs, provided as a sequence of instructions, both with and without user-specified constraints. The test runs served to evaluate the effectiveness of the semantic interpretation module in inferring the correct command specifications (*command*, *DSR*, *static relation*) given the natural language input, and to demonstrate the DSR path generation capabilities of the system. Our testing domain consisted of a simulated mobile robot operating within a 2D map of a home environment.

A dataset of 128 labeled training examples (each containing a list of observations with correct command specifications), was utilized for the probabilistic inference procedure of the semantic interpretation module. This dataset included the use of 8 different DSRs, 10 separate static spatial relations, 2 commands, and 22 different verbs, each appearing multiple times (and in novel combinations) among the examples.

The instruction sequence provided to the robot in both test runs, including the natural language constraints that were specified for the individual instructions, is listed in Table III. The sequence of instructions was identical for both runs, with the exception that the constraints listed were specified to the robot for Test Run #2 only. Hence, in Test Run #1 the robot did not operate under any user-specified constraints for the individual instructions. Constraints were provided to the robot in Test Run #2 to illustrate the flexibility of the path generation procedure to operate under user-specified constraints while also accomplishing the goals of the DSR path specification. The planning module accounts for user-specified constraints by introducing modifications to the A* cost function (using the semantic fields of the inferred static relations) during task planning, as detailed in [5].

Results of the inference procedure of the semantic interpretation module, with accompanied pragmatics, for the five instructions given in the instruction sequence for both test runs are provided in Table IV. As evidenced by the results, our robot architecture was able to successfully interpret the semantics of the natural language instructions provided by the user during both test runs of the system.

The DSR path generation results for the entire instruction sequence of Test Runs #1 and #2 are provided in Fig. 5 (a) and (b), respectively. The differences between the paths

TABLE III. Instruction Sequence given in Test Runs

Type	Natural Language Instruction
Instruction[1]:	Go around the bed
Constraint:	*Stay close to the bed*
Instruction[2]:	Travel through the hallway
Instruction[3]:	Go around the dinner table
Constraint:	*Keep away from the kitchen*
Instruction[4]:	Stand between the tv and the bookcase
Constraint:	*Travel between the couch and the coffee table*
Instruction[5]:	Walk through the kitchen
Constraint:	*Walk along the wall*

TABLE IV. Results of Semantic Inference and Pragmatics for Test Run Instructions

#	Semantics	Pragmatics
1	(RM, *around*, -)	$around_1$
2	(RM, *through*, -)	$through_2 \rightarrow through_1$
3	(RM, *around*, -)	$around_2$
4	(RM, *to*, *between*)	*to*
5	(RM, *through*, -)	$through_2 \rightarrow through_3 \rightarrow through_2$

Note. RM = robot movement command

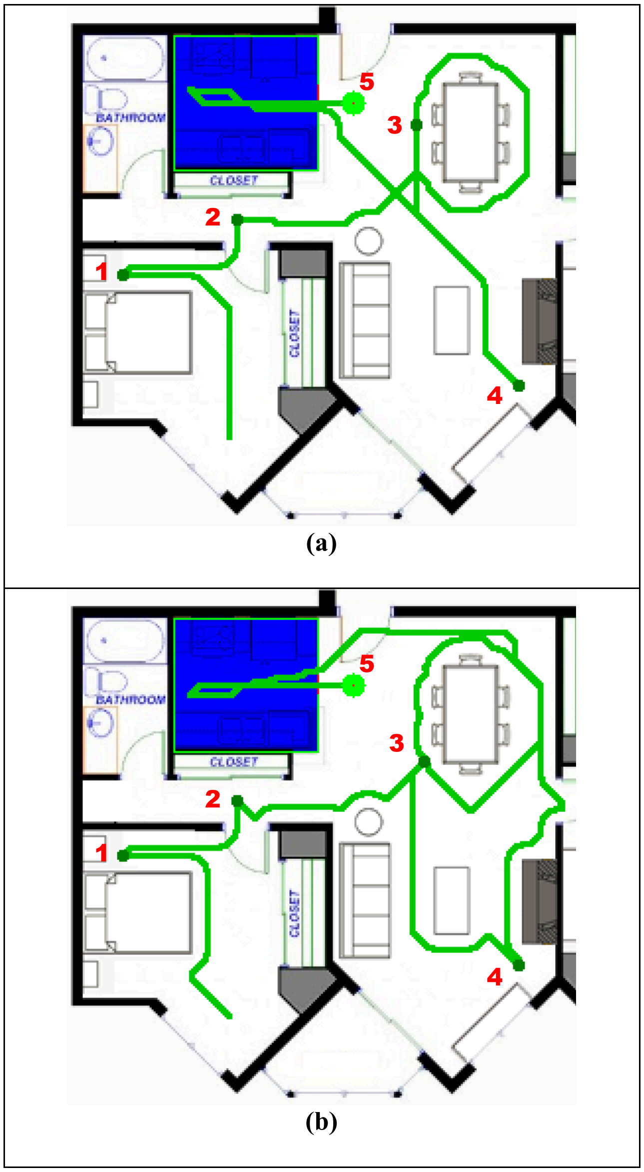

Figure 5. DSR path generation results for entire instruction sequence with and without user-specified constraints. (a) Test Run #1 (no constraints); (b) Test Run #2 (constraints). Note: path endpoints for each instruction are labeled with the instruction number.

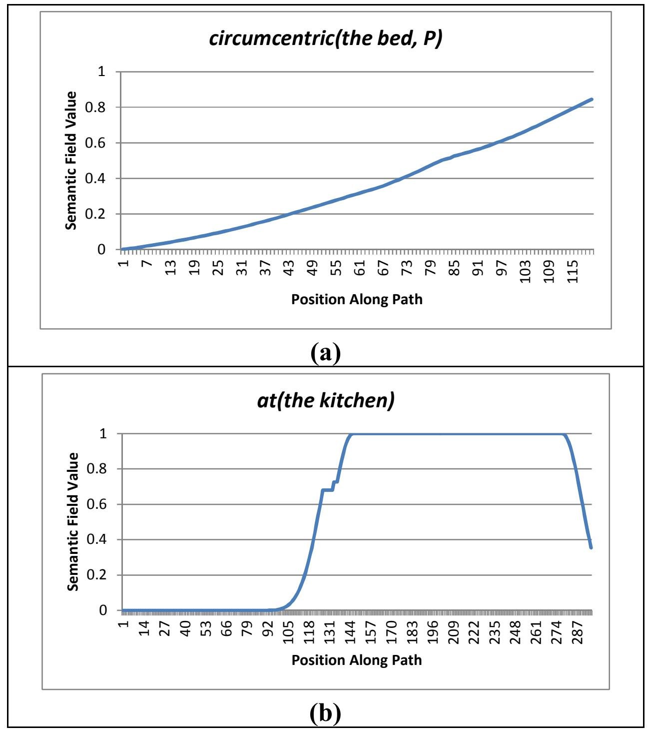

Figure 6. Semantic field values along execution paths in Test Run #1. (a) *circumcentric* field value along solution path for instruction 1; (b) *at* field value along solution path for instruction 5.

generated in both test runs highlight the impact of user-specified constraints on the resulting robot execution path. For example, in Test Run #1 the robot satisfies the DSR of the first instruction (*around*) by generating and executing the shortest path to the point within the visible region that possesses the maximum *circumcentric* field value among all points considered. In Test Run #2, the robot also generates a path to this point, but due to the user-specified constraint “Stay close to the bed”, the execution path runs along the border of the bed, resulting in a slightly longer path by comparison. This difference in path generation results is also observed for the last instruction in the sequence (“Go through the kitchen”), where in Test Run #2, the robot generates a comparably longer path to the inside of the kitchen by staying close to the edge of the rooms in consideration of the specified constraint “Walk along the wall”.

To illustrate the usefulness of the semantic field model towards representing static and dynamic spatial relation primitives for use in DSR path generation and classification, Fig. 6 shows the progression of the *circumcentric* and *at* field values along the execution paths generated for instructions 1 and 5, respectively. As demonstrated by the results, the values returned by the semantic fields are highly correlated with the progress made during path execution towards accomplishing the goals of the DSR inferred from the specified natural language instructions.

As evidenced by the semantic inference results shown in Table IV, and all robot execution paths for the DSRs of the instruction sequence displayed in Fig. 5, the robot architecture was able to demonstrate its potential by successfully following the natural language directives, with and without constraints, during each of the test runs performed for the purposes of system evaluation. In addition, the differences observed in the generated DSR paths for both test runs illustrate the capability of our approach to modeling DSRs with global properties in accomplishing natural language instructions in human-robot interaction scenarios under both user-specified constraints as well as unvoiced pragmatic constraints.

VII. Conclusion

We have described the need for enabling autonomous service robots with spatial language understanding to facilitate natural communication with non-expert users for task instruction, and have presented a general methodology we have developed towards addressing this research challenge. The contributions of this paper include the presentation of: a novel representation for DSRs with global properties that facilitates probabilistic reasoning over paths that can be applied for both path classification and path generation scenarios; example representations for the DSRs of "to", "through", and "around"; implementation details of the path generation procedures utilized by our system for these three DSRs; and discussion of relevant pragmatic constraints along with planning methods developed to address these constraints in multi-step robot execution planning of instruction sequences.

The results obtained from our evaluation testing demonstrate the potential of our methodology for representing dynamic spatial relations, interpreting the semantics of natural language instructions probabilistically, and generating appropriate agent execution plans under user-specified natural language constraints as well as unvoiced pragmatic constraints.

Our ongoing work will focus on the evaluation of our methodology implemented on real robots and interacting with end-users, thus addressing the limitations of our current evaluation obtained from 2D simulation with text-based input. Specifically, we plan to test our approach under environmental uncertainty (i.e., sensor noise), and with spoken natural language commands from target users (e.g., older adults). Towards this end, a corpus of spoken command utterances from target users will be collected prior to testing as additional data for our semantic interpreter and to further inform our grammar. In expanding the evaluation procedure, performance measures will also be captured from user ratings of robot task completion relative to the spoken commands. Lastly, our approach will be tested under more general path generation scenarios, including object manipulation/movement tasks (e.g., put the cup on the table in the dining room).

REFERENCES

[1] B. Landau and R. Jackendoff. What and where in spatial language and spatial cognition. *Behavioral and Brain Sciences*, 16(2):217–265, 1993.

[2] R. S. Jackendoff. *Semantics and Cognition*. MIT Press. 1983.

[3] J. O'Keefe. Vector grammar, places, and the functional role of the spatial prepositions in English, E. van der Zee and J. Slack, Eds. Oxford: Oxford University Press, 2003.

[4] L. Talmy. The fundamental system of spatial schemas in language. In Hamp, B., ed., *From Perception to Meaning: Image Schemas in Cognitive Linguistics*. Mouton de Gruyter, 2005.

[5] J. Fasola and M. J. Matarić. Using Semantic Fields to Model Dynamic Spatial Relations in a Robot Architecture for Natural Language Instruction of Service Robots. University of Southern California Center for Robotics and Embedded Systems Technical Report, CRES-13-001, 2013.

[6] J. Fasola and M. J. Matarić. Using Spatial Language to Guide and Instruct Robots in Household Environments. In 2012 AAAI Fall Symposium on Robots Learning Interactively from Human Teachers (RLIHT), Nov. 2012.

[7] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Approaching the Symbol Grounding Problem with Probabilistic Graphical Models. *AI Magazine*. 32(4): 64-76, 2011.

[8] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward Understanding Natural Language Directions. In Proc. ACM/IEEE Int'l Conf. on Human-Robot Interaction (HRI), 259–266, 2010.

[9] Y. Sandamirskaya, J. Lipinski, I. Iossifidis, and G. Schöner. Natural human-robot interaction through spatial language: a dynamic neural field approach. 19th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 600-607, 2010.

[10] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock. Spatial Language for Human-Robot Dialogs. *IEEE Transactions on SMC Part C*, Special Issue on Human-Robot Interaction, 34(2):154-167, 2004.

[11] C. Chao, M. Cakmak, and A.L. Thomaz. Towards grounding concepts for transfer in goal learning from demonstration. In Proceedings of the Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob), 2011.

[12] D.K. Roy. Learning visually grounded words and syntax for a scene description task. In *Computer Speech & Language*. 16(3-4): 353-385, 2002.

[13] P.E. Rybski, J. Stolarz, K. Yoon, and M. Veloso. Using dialog and human observations to dictate tasks to a learning robot assistant. *Journal of Intelligent Service Robots*, 1:159-167, 2008.

[14] H. Kress-Gazit, G.E. Fainekos, and G.J. Pappas. Translating structured English to robot controllers. *Advanced Robotics*, 22, 1343–1359, 2008.

[15] J.D. Kelleher, F.J. Costello. Applying Computational Models of Spatial Prepositions to Visually Situated Dialog. *Computational Linguistics*, 35(2):271-306, 2009.

[16] S. Mohan, A. Mininger, J. Kirk, and J.E. Laird. Learning Grounded Language through Situated Interactive Instruction. In AAAI Fall Symposium on Robots Learning Interactively from Human Teachers (RLIHT), 2012.

[17] N. Hawes, M. Klenk, K. Lockwood, G.S. Horn, J.D. Kelleher. Towards a cognitive system that can recognize spatial regions based on context. In Proceedings of the Twenty-Sixth Conference on Artificial Intelligence (AAAI), 2012.

[18] L.A. Carlson, and P.L. Hill. Formulating spatial descriptions across various dialogue contexts. In K. Coventry, T. Tenbrink, & J. Bateman (Eds.), Spatial Language and Dialogue, 88-103. New York, NY: Oxford University Press Inc, 2009.

[19] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to Parse Natural Language Commands to a Robot Control System. In Proc. of the International Symposium on Experimental Robotics (ISER), Québec City, Canada, 2012.

[20] R. Cantrell, P. Schermerhorn, M. Scheutz. Learning actions from human-robot dialogues. In Proc. IEEE RO-MAN, pp.125-130, 2011.

[21] J. Bohnemeyer. The Unique Vector Constraint: The impact of direction changes on the linguistic segmentation of motion events, E. van der Zee and J. Slack, Eds. Oxford: Oxford University Press, 2003.

[22] Nuance. 2013. Dragon NaturallySpeaking. www.nuance.com

[23] M. Nicolescu and M. J. Matarić. Task Learning Through Imitation and Human-Robot Interaction. In Models and Mechanisms of Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions, Kerstin Dautenhahn and Chrystopher Nehaniv (eds.), Cambridge University Press, 2005.

[24] D. Hewlett, W. Kerr, T. J. Walsh, P. Cohen. A Framework for Recognizing and Executing Verb Phrases. In 2011 Robotics: Science and Systems Workshop: HRI Workshop on Grounding Human-Robot Dialog for Spatial Tasks, Los Angeles, CA, 2011.

[25] M. Pardowitz, S. Knoop, R. Dillmann, R. D. Zollner. Incremental Learning of Tasks From User Demonstrations, Past Experiences, and Vocal Comments, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(2): 322-332, April 2007.

[26] A. Atrash, R. Kaplow, J. Villemure, R. West, H. Yamani, J. Pineau. Development and Validation of a Robust Interface for Improved Human-Robot Interaction, *International Journal of Social Robotics*, 1(4): 345-356, 2009.