

2020-08-10

Role and task allocation framework for Multi-Robot Collaboration with latent knowledge estimation

Gianni, Mario

<http://hdl.handle.net/10026.1/15751>

10.1002/eng2.12225

Engineering Reports

Wiley

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

RESEARCH ARTICLE

Role and task allocation framework for Multi-Robot Collaboration with latent knowledge estimation [†]

Mario Gianni¹ | Mohammad Salah Uddin²

¹University of Plymouth
School of Engineering, Computing and
Mathematics, Plymouth, Devon, U.K.
Email: mario.gianni@plymouth.ac.uk

²East West University
Computer Science & Engineering
Department, Dhaka, Bangladesh

Correspondence

Mohammad Salah Uddin
East West University
Computer Science & Engineering
Department, Dhaka, Bangladesh
Email: uddin@ewubd.edu

Summary

In this work a novel framework for modeling role and task allocation in Cooperative Heterogeneous Multi-Robot Systems (CHMRSs) is presented. This framework encodes a CHMRS as a set of multi-dimensional relational structures (MDRSs). This set of structure defines collaborative tasks through both temporal and spatial relations between processes of heterogeneous robots. These relations are enriched with tensors which allow for geometrical reasoning about collaborative tasks. A learning schema is also proposed in order to derive the components of each MDRS. According to this schema, the components are learnt from data reporting the situated history of the processes executed by the team of robots. Data are organized as a Multi-Robot Collaboration Treebank (MRCT) in order to support learning. Moreover, a generative approach, based on a probabilistic model, is combined together with Non-Negative Tensor Decomposition (NTD) for both building the tensors and estimating latent knowledge. Preliminary evaluation of the performance of this framework is performed in simulation with three heterogeneous robots, namely, two Unmanned Ground Vehicles (UGVs) and one Unmanned Aerial Vehicle (UAV).

KEYWORDS:

Multi-robot collaboration, Task Allocation, Knowledge Factorization, Knowledge Discovery, Knowledge Prediction

1 | INTRODUCTION

Multi-robot deployment is better suited than single robot deployment in many domain applications. For instance, in safety-critical applications like Urban Search and Rescue, the deployment of multiple robots speeds up area coverage thus increasing the chance to find all potential survivors¹. In rescue scenarios (considering the rescue environments after natural disaster such as earthquake, fire, building collapse etc.), it is very difficult for rescue workers or teams to access all the region of rescue environments due to the possible presence of radiation or extreme temperatures, dust, asbestos, hazardous substances². A promising solution offered by rescue robots to assist rescue teams in terms of:

1. minimizing personal injuries to rescue workers and rescue dogs by accessing unsettled structures,
2. raising speed of response by penetrating ordinarily inaccessible voids,

[†]Both authors were with Department of Computer Control and Management “A. Ruberti”, Sapienza University of Rome, Italy when the work was carried out. M. Gianni is now with School of Engineering, Computing and Mathematics, University of Plymouth, UK, and M.S. Uddin is now with Department of Computer Science and Engineering, East West University, Bangladesh

3. through the use of sensor fusion and multiple cameras in order to enhance the reach of rescue workers to regions that are otherwise inaccessible.

Rescue robots have many advantages compared to rescue workers and trained rescue dogs:

1. unlike humans, a rescue robot will not become stressed or fatigued³;
2. rescue robots can be developed in large quantities, while experienced rescue professionals and trained rescue dogs are sparse resources²;
3. robots are expendable but humans and rescue dogs are not: if a rescue robot is damaged, it can be easily repaired or replaced, but the loss of rescue workers or dogs could be very difficult due to their relationship within society².

Thus, the robot or multi-robot deployment minimizes human exposure to danger. In planetary exploration homogeneous robot deployment introduces redundancy making the overall system more robust and reliable⁴. In agricultural as well as in manufacturing the use of heterogeneous robots can cope with hardware limitations of individual robot payload for complex task accomplishment⁵. Still, in assembly multiple robots deployment reduces human workload.

However, making multi-robot deployment effectively operational has as a fundamental prerequisite the capability of the robots to collaborate among them^{6,7}. Collaboration strongly depends on the capability of the robots to communicate, that is, to share as well as to exchange information about context-dependent tasks^{8,9}. Information sharing can not take place if it does not exist a communication platform, common to all the robots, that is, a knowledge management structure, in which a language for representing knowledge, a mechanism for knowledge association, a protocol for information exchange and, finally, a memory system are well established^{10,11,12,13}.

Languages for representing knowledge in multi-robot systems are commonly based on beliefs intentions¹⁴, semantic networks¹⁵, frame languages¹⁶ and resource description frameworks^{17,18}. Knowledge association is responsible of the bidirectional information flow, where low-level data is passed upwards and the high-level information is returned downwards using logical inference¹⁹, bayesian inference¹⁴, semantic relationships and hierarchies²⁰ or computational learning methods²¹. FIPA²² together with KIF²³ are the standard protocols for communication. Finally, memory is usually deployed on either centralized²⁴ or on distributed²⁵ database systems.

Apart from providing a common ground for knowledge sharing supporting efficient collaboration among robots, knowledge management systems allow for information reuse¹¹.

In this regard, imagine a team of robots which operate into an environment such as a rescue scenario or a manufacturing building. Let us consider the variety of data, ranging from measurements, sensory information, robot descriptions, states and commands to environmental data, such as positions, maps and spatial relations, stored by the robots into the database of the knowledge management system. Now, imagine the same team of robots entering again in the same scenario, possibly changed due to some event. The team of robots might access to the information, gathered during previous operating activities and stored into the database, and filter out the information irrelevant for the current activities. Then, it might integrate this information with the incoming sensor measurements and reuse the fused information for reducing the computational overhead in the current task assignment process while taking into account the environmental changes.

Recently, several research efforts in robotics as well as in Artificial Intelligence (AI) have been made for developing robotic systems based on knowledge management structures incorporating information reuse. Tenorth *et al.* in²⁶ introduced KnowRob, a knowledge processing system that combines knowledge representation and reasoning methods with techniques for acquiring knowledge and for grounding the knowledge in a physical system thus serving as a common semantic framework for integrating information from different sources. KnowRob combines static encyclopedic knowledge, common-sense knowledge, task descriptions, environment models, object information and information about observed actions that has been acquired from various sources. An extension of KnowRob framework, named RoboEarth, has been developed in¹². On top of the KnowRob robot knowledge base, Beetz *et al.* developed Open-Ease²⁷, a remote knowledge representation and processing service for robots. Open-Ease allows to retrieve the memorized experiences of task episodes and to ask queries regarding what robots saw, reasoned and did as well as how robots did it, why and what effects it caused. Until a few years ago, knowledge management structures for modeling multi-robot systems were fully integrated a single standalone architecture, thus limiting the knowledge of the robots to that stored into the memory over the past task episodes¹⁶. Recently, with the advent of the emerging fields of the Internet of Things and of the Web of Things, robots can rely on cloud-computing infrastructures to have access to a vast amounts of processing power and data in order to improve capabilities such as speech recognition, language translation, path planning and 3D

mapping^{28,29,30,5,31}. The possibility to have access to big data repositories, such as the Web, has made even more effective the use of methods and techniques of learning in the context of multi-robot systems^{32,33}.

In robotics, assigning tasks to multiple robots is a challenging problem. Although, it has numerous practical applications. In the past, several strategies for task assignment were proposed for different application domains such as surveillance (aerial/underwater)^{34,35}, search and rescue³⁶, multi-robot patrolling³⁷, feeding operation³⁸ and health care³⁹. For example, multi-robot search and rescue missions aim to rescue survivors by exploring the surrounding environment. Each sub-task explores a small area of the environment and seeks survival. The successful completion of each of these sub-tasks results in the completion of the entire mission. As survival (available tasks) and the size of the robot team increase, the task allocation process becomes more difficult⁴⁰. Techniques for allocating tasks in multi-robot systems are commonly based on clustering^{41,42}, swarm intelligence⁴³ and resources⁴⁴. Some other task assignment methods for multi-robot systems are listed in the reference^{45,46,47}.

In this regard, we propose a framework for learning multi-dimensional relational structures (MDRSs) regulating role and task assignment in Cooperative Heterogeneous Multi-Robot Systems (CHMRSs). These structures define collaborative tasks as both temporal and spatial relations between the processes and tasks of the single robots in a logical as well as in a geometrical fashion. Through learning we extract from memorized experiences of task episodes, stored in the form of a suitably defined Multi-Robot Collaboration Treebank (MRCT), the syntaxes, the semantics and the geometrical spaces in which the MDRSs lie. Moreover, we propose a decomposition technique for dealing with both uncertainty of the data and the missing information in the treebank. Such a decomposition is also employed as a technique for knowledge discovery, based on link prediction. This framework allows for reasoning about task assignment based on both logical and geometrical inference. This work has been extended from the deliverable report of Tradr project⁴⁸.

The remainder of this work is organized as follows. Section 2 introduces the multi-dimensional relational structures. In Section 3 we describe the Multi-Robot Collaboration Treebank. Section 4 illustrates the main components of the proposed schema for learning MDRSs together with the probabilistic model for building the geometrical spaces on top of which collaboration lies. In Section 5 we describe the approach based on Non-Negative Tensor Decomposition (NTD) for latent knowledge estimation. Section 6 concludes the work with preliminary results in simulation and discussion.

2 | MODELING MULTI-ROBOT COLLABORATION THROUGH MULTI-DIMENSIONAL RELATIONAL STRUCTURES

Let us consider a CHMRS composed of two heterogeneous robots, that is, an UGV, named UGV1, and an UAV, named UAV2. Suppose that UGV1 has to explore an area of the environment (see Figure 1(a)). In order to navigate, this robot needs to have a representation of the area specifying what regions are traversable. The analysis of traversability builds upon a 3D metric representation of the area. However, it might happen that, due to the high degree of harshness of the terrain, the metric map built by UGV1 is quite sparse, thus making traversability analysis very inaccurate (see missing points on the left side of the robot in Figure 1(c)). Under this situation, UGV1 sends the request to UAV2 to fly over the area to build a more dense metric map (see Figure 1(b) and (d)). Upon the completion with success of this process, UGV1 requests the map of the area to UAV2, it integrates its own map with the map provided by UAV2 and, finally, it computes a more accurate estimate of the traversability of the surrounding. This form of collaboration can be represented by a MDRS encoding a temporal relation *Before* between four entities, two of type *Robot* and two of type *Process*.

Under this perspective, we model a CHMRS as a set $\mathbf{U}=\{S_1, \dots, S_n\}$ of multi-dimensional relational structures. Each MDRS is defined by a pair $\langle \mathbf{S}_i, \mathbf{Y}_i \rangle$. Where \mathbf{S}_i is the signature of MDRS. It comprises a relational symbol R_i of arity $K \in \mathbb{N}$, a finite set $\Sigma_i = \{\sigma_1, \dots, \sigma_{n_i}\}$ of types, also called sorts, with $n_i \leq K$, a finite set C_{i, σ_k} of constant symbols c_{i, σ_k} , for each sort $\sigma_k \in \Sigma_i$ and, finally, a countable set V_{i, σ_k} of variable symbols v_{i, σ_k} , for each sort $\sigma_k \in \Sigma_i$. Furthermore $\mathbf{Y}_i \in \mathbb{R}_+^{M_1 \times \dots \times M_K}$ is a non-negative multi-dimensional matrix, also called K -order tensor, where each M_k is equal to the cardinality of the set C_{i, σ_k} with $\sigma_k \in \Sigma_i$ sort of the k -th input term of R_i , for $k=1, \dots, K$.

In other words, \mathbf{Y}_i has a number of dimensions equal to the arity K of R_i . Along the k -th direction, \mathbf{Y}_i has a number of elements equal to the cardinality of the set C_{i, σ_k} , where $\sigma_k \in \Sigma_i$ is the sort of the k -th input argument of R_i .

Note that, according to the definition of \mathbf{Y}_i , each tuple of indices (i_1, \dots, i_K) , with $i_k=1, \dots, M_k$, corresponds to a tuple $(c_{i, \sigma_1}, \dots, c_{i, \sigma_K})$ of constant symbols, with $c_{i, \sigma_k} \in C_{i, \sigma_k}$ and $\sigma_k \in \Sigma_i$ sort of the k -th input term of R_i , for $k=1, \dots, K$.

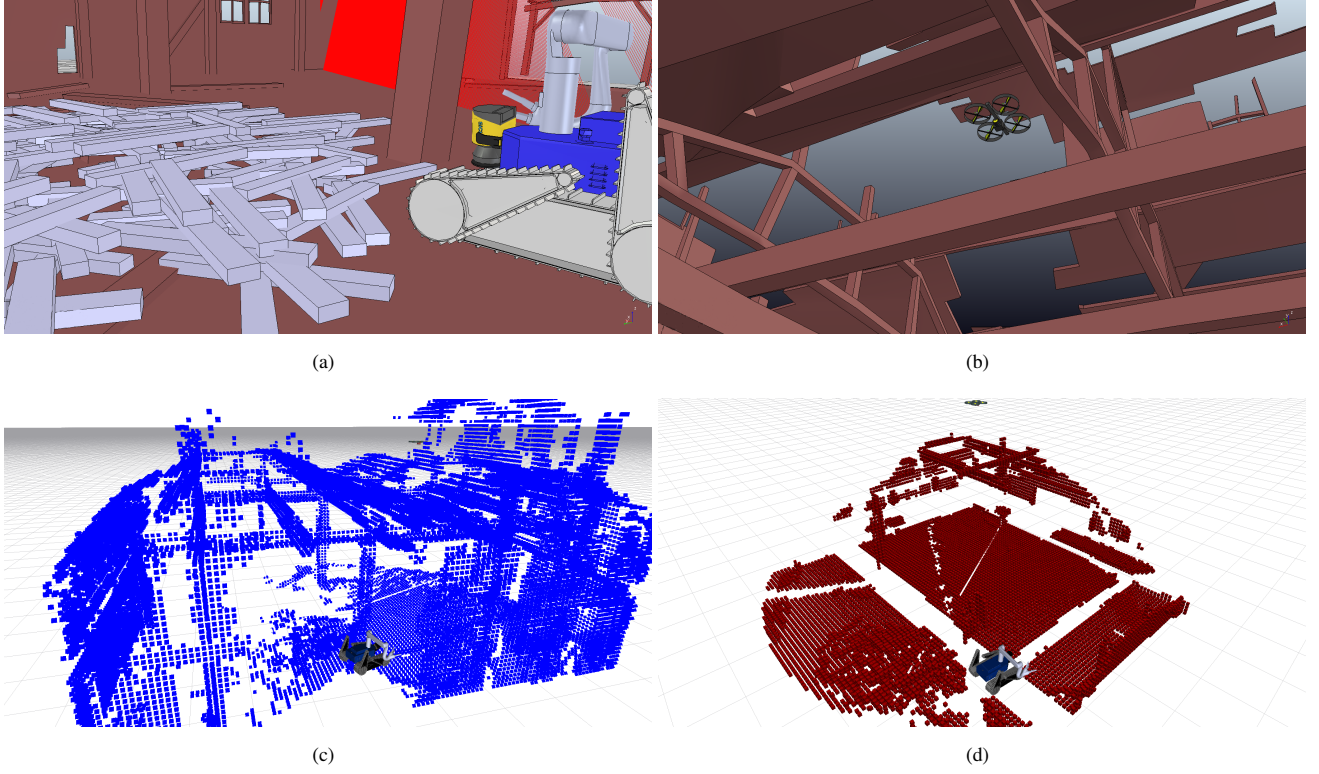


FIGURE 1 UGV1 and UAV2 in both the virtual simulated environment⁴⁹ and in ROS RVIZ⁵⁰. (a) UGV1 exploring the area; (b) UAV2 flying over the roof of the simulated collapsed building; (c) 3D metric map of the area built by UGV1 (blue point cloud); (d) 3D metric map of the area built by UAV2 (red point cloud).

The temporal relation representing the collaboration between UGV1 and UAV2, in the above example, is encoded by the a four-dimensional structure $S_i \in \mathbf{U}$, whose signature \mathbf{S}_i is defined as follows

$$K = 4, \quad \Sigma_i = \{\text{Robot}, \text{Process}\}, \quad R_i(\cdot, \cdot, \cdot, \cdot) \stackrel{\text{def}}{=} \text{Before}(\cdot, \cdot, \cdot, \cdot), \\ C_{i, \text{Robot}} = \{\text{UAV2}, \text{UGV1}, \dots\} \quad \text{and} \quad C_{i, \text{Process}} = \{\text{mapping}, \text{exploring}, \dots\}.$$

Finally, for the tuple (UAV2, mapping, UGV1, exploring) of constant symbols, there exists a tuple of indices (i_1, i_2, i_3, i_4) such that $y_{i_1, i_2, i_3, i_4}^i \in \mathbb{R}_+$ is the element of the four-tensor $\underline{\mathbf{Y}}_i$ associated with this tuple.

MDRSs are well suited for supporting two different types of inference. The first type is the standard logical entailment. The second type of inference is based on the extraction of fragments (e.g., tensor fibers and slices) of the tensors through mode- n operations. The strength of combining together these two types of inference for role and task assignment in multi-robot collaboration is demonstrated in the following example.

Example 2.1. Let us consider a CHMRS composed of three UGVs, named UGV1, UGV2 and UGV3, respectively. Let us suppose that \mathbf{U} includes a MDRS S_1 , having the following signature \mathbf{S}_1

$$K = 4, \quad \Sigma_1 = \{\text{Robot}, \text{Process}\}, \quad R_1(\cdot, \cdot, \cdot, \cdot) \stackrel{\text{def}}{=} \text{Equal}(\cdot, \cdot, \cdot, \cdot), \\ C_{1, \text{Robot}} = \{\text{UGV1}, \text{UGV2}, \text{UGV3}, \dots\} \quad \text{and} \quad C_{1, \text{Process}} = \{\text{grasping}, \dots\}.$$

S_1 represents a collaborative pick and place task. This task requires that two UGVs simultaneously hold and lift an object. According to this specification, this form of collaboration has been encoded by a temporal relation Equal between four entities, two of type Robot and two of type Process. Moreover, let us assume to have another structure S_2 representing the processes

which can be performed by each robot. The signature \mathbf{S}_2 of S_2 is defined as follows

$$K = 2, \quad \Sigma_2 = \{\text{Robot}, \text{Process}\}, \quad R_2(\cdot, \cdot) \stackrel{\text{def}}{=} \text{Process}(\cdot, \cdot), \\ C_{2, \text{Robot}} = \{\text{UGV1}, \text{UGV2}, \text{UGV3}, \dots\} \quad \text{and} \quad C_{2, \text{Process}} = \{\text{grasping}, \dots\}.$$

Now, suppose that UGV1 has to grasp an object for which it is required the collaboration of another UGV. Then, we want to know to which robot to assign this collaborative task. According to logical entailment, we know that, in order to answer to this query, we have to find, as usual, an interpretation $\mathfrak{I} = (\mathcal{I}, \nu)$, with \mathcal{I} first-order structure and ν variable assignment function mapping sorted variable symbols to elements of the domain of the right sort such that

$$\mathfrak{I} \models \exists v \text{Process}(v, \text{grasping}) \wedge \text{Equal}(\text{UGV1}, \text{grasping}, v, \text{grasping})$$

Let us assume that the interpretation maps the constant symbols into themselves in the domain. Moreover, according to \mathcal{I} , let us assume that

$$\text{Process}^{\mathcal{I}} = \{\langle \text{UGV1}, \text{grasping} \rangle, \langle \text{UGV2}, \text{grasping} \rangle, \langle \text{UGV3}, \text{grasping} \rangle\} \\ \text{Equal}^{\mathcal{I}} = \{\langle \text{UGV1}, \text{grasping}, \text{UGV2}, \text{grasping} \rangle \\ \langle \text{UGV1}, \text{grasping}, \text{UGV3}, \text{grasping} \rangle\}$$

Under the mappings $\nu[\text{UGV2}/v]$ and $\nu[\text{UGV3}/v]$, we know that both UGV2 and UGV3 can execute this task.

Now, let $y_{i_1, i_2, \tilde{i}_3, i_4}^1 \in \mathbb{R}_+$ and $y_{i_1, i_2, \hat{i}_3, i_4}^1 \in \mathbb{R}_+$, at positions $(i_1, i_2, \tilde{i}_3, i_4)$ and $(i_1, i_2, \hat{i}_3, i_4)$, be the elements in the four-tensor $\underline{\mathbf{Y}}_1$ associated to the tuples in $\text{Equal}^{\mathcal{I}}$. Let us also assume that these elements represent the number of times that UGV1 performed this task in collaboration with the two UGVs under consideration. According to the meaning that we are given to these tensor elements, the choice of the UGV which has to collaborate with UGV1 might be dictated by comparing the value of $y_{i_1, i_2, \tilde{i}_3, i_4}^1$ with the value of $y_{i_1, i_2, \hat{i}_3, i_4}^1$. Suppose that $y_{i_1, i_2, \hat{i}_3, i_4}^1 > y_{i_1, i_2, \tilde{i}_3, i_4}^1$. Then the choice falls onto UGV3.

As above, let (\tilde{i}_1, i_2) , (\hat{i}_1, i_2) and (\tilde{i}_1, i_2) be the three tuples of indices such that $y_{i_1, i_2}^2 \in \mathbb{R}_+$, $y_{\hat{i}_1, i_2}^2 \in \mathbb{R}_+$ and $y_{\tilde{i}_1, i_2}^2 \in \mathbb{R}_+$ are the elements in the matrix \mathbf{Y}_2 associated with the terms in the interpretation $\text{Process}^{\mathcal{I}}$. Now, suppose that these elements y_{i_1, i_2}^2 and $y_{\tilde{i}_1, i_2}^2$ represent the failure rate of the process grasping for UGV2 and UGV3, respectively. A comparison among the values of these two tensor elements should be also considered before assigning the task to UGV3. If the value of $y_{\tilde{k}_1, k_2}^2$ is greater than the value of y_{k_1, k_2}^2 then a better choice would be to assign the collaborative task to UGV2, rather than UGV3.

Example 2.1 demonstrates the ability of a system, modeled by a set of MDRSs, of flexibly reasoning on multiple choices by combining both logical inference and reasoning on tensor fields. However, this joint reasoning mechanism, as highlighted in Example 2.1, strictly depends by the meaning that we are given to the tensor elements. Moreover, it is still unclear where these multi-dimensional relational structures come from, how these structures are built and where tensor elements come from.

To this end, we developed a learning schema through which both signatures and tensors of each MRDS are learnt from data reporting the situated history of the activities performed by a group of cooperative heterogeneous robots. Here, data and, in particular, their linguistic structure, play a crucial role in making learning more tractable, as described in the next section.

3 | MULTI-ROBOT COLLABORATION TREEBANK

Documents describing reports of missions executed by a team of robots are organized as a treebank^{51,52,53,54}. A treebank is a collection of pairs $\langle \mathbf{s}_i, \mathcal{T}_i \rangle$, where each \mathbf{s}_i is a statement and each \mathcal{T}_i is a syntactic tree. Each statement is a sequence $s_{i,1}, \dots, s_{i,n}$ of words. Here, we assume that statements do not contain anaphoric as well as elliptical references. Each \mathcal{T}_i is composed of a set \mathcal{N}_i of nodes and a set \mathcal{E}_i of edges. \mathcal{N}_i is composed of a set $\mathcal{N}_{i,int}$ of intermediate nodes and a set $\mathcal{N}_{i,leaf}$ of leaf nodes. Each node $n_u \in \mathcal{N}_{i,int}$ is labeled with a non-terminal symbol NT of a formal system FS . On the other hand, each node $n_v \in \mathcal{N}_{i,leaf}$ is labeled with a terminal symbol T of FS . There exists an edge $\langle n_u, n_v \rangle \in \mathcal{E}_i$, with $n_u, n_v \in \mathcal{N}_i$ if there exists a production rule of the form $\gamma \rightarrow \delta$ in FS such that the label of n_u is equal to γ , the label of n_v is equal to δ , $\gamma \in NT$ and $\delta \in NT \cup T$. Intuitively, each syntactic tree is a derivation of the string of words composing a statement. A derivation is a sequence of rule expansions defined by the formal system.

In linguistic, a formal language is commonly used as a generative grammar. Instead, in natural language processing, it is used for semantic parsing of natural language sentences, possibly unconstrained and including complex compositional expressions.

In this context, the use of a formal language has a different purpose. It provides a priori a well-defined syntax of the relations encoding both individual and collaborative tasks. It has the effect to filter noise in the documents. Indeed, if a statement is aligned by a syntactic tree which does not correspond to any sequence of rule expansions of the formal system, then it can be simply disregarded and not considered in the learning phase. Finally, the syntactic trees associated with the statements together with the production rules of FS suggest a set of strategies to speed up learning of the MDRSs modeling a multi-robot system.

In the following we provide a partial description of the formal system together with examples of syntactic trees annotating admissible statements describing the activities of heterogeneous robots cooperating in the execution of both individual and collaborative tasks. The full version of the formal system underlying the treebank is provided in Appendix A.1.

The formal system consists of a set of domain-specific constituents. A constituent is either a single word or a group of words, acting as a unit. For the multi-robot collaboration domain application we specify atomic process phrase (APP), temporal task phrase (TTP), shifting-inhibition phrase (SIP) and atomic process execution state phrase (APESP) constituents. According to these domain-specific constituents, a statement (S) can consist of either an atomic process phrase or of a temporal task phrase, or of a shifting-inhibition phrase or of an atomic process execution state phrase, as specified by the following production rule

$$S \rightarrow \text{APP} \mid \text{TTP} \mid \text{SIP} \mid \text{APESP} \quad (1)$$

APP constituents are introduced to annotate statements describing single processes executed by individual robots. These processes comprise simple operations like acquiring an image, mapping an area of the environment, holding an object, but also spatial relations between objects, spatial relations between robots and relations of dispatching of information among robots. Syntactic trees associated to statements describing these activities and generated from the expansion of an APP constituent are illustrated in Figure 2.

TTP constituents syntactically specify a wide class of complex collaborative behaviors of a group of robots. According to the definition of TTP, each behavior is a composition of individual robot processes linked by a temporal constraint. An examples of a syntactic tree annotating a statement represented by the expansion of an TTP is shown in Figure 3

Statements in the treebank describing both shifting and inhibition behaviors of individual robots as well as reporting the status of the execution of a robot process are annotated by expanding SIP and APESP constituents, respectively. Syntactic trees associated with these kind of statements are illustrated in Figure 4(a) and Figure 4(b).

Note that, the formal system differs from the semantic grammars used in case theory due to the choice of word-alignment⁵⁵. In fact it is not a completely word-alignment grammar. Constituents may correspond to multi-word expressions, such as the spatial relation on top of or the process picked up. The main motivation is to facilitate learning of the MDRSs, as described in the next section.

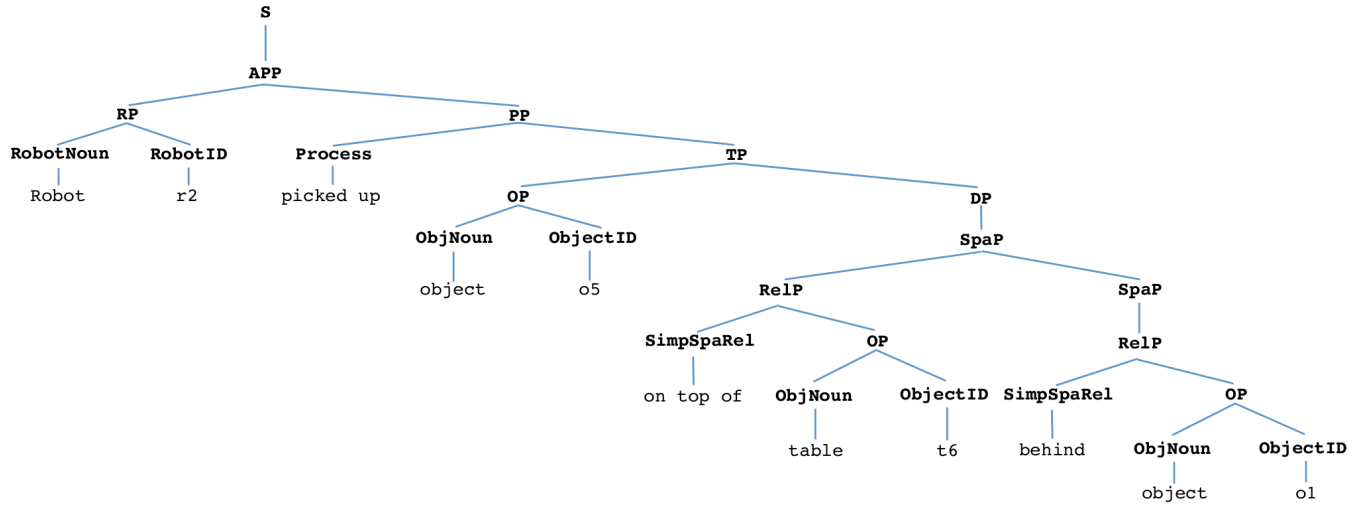
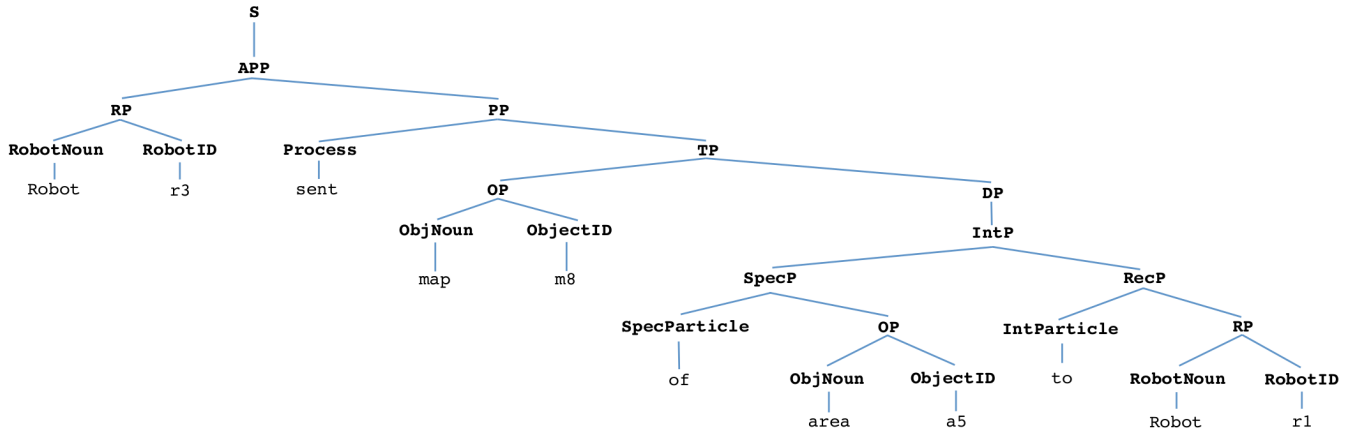
4 | LEARNING OF THE MULTI-DIMENSIONAL RELATIONAL STRUCTURES

Let $D = \{\langle s_i, \mathcal{T}_i \rangle\}_{i=1}^N$ be a collection of statement-syntactic-tree pairs where each syntactic tree annotates the corresponding statement on the basis of the formal system FS , introduced in Section 3. Learning of MDRS from the data-set D involves three main steps: (1) the definition of the signatures \mathbf{S}_i ; (2) building of the tensors \mathbf{Y}_i and, finally, (3) the estimation of the values of the elements of each tensor.

Signatures \mathbf{S}_i are directly derived from the specification of both constituents and production rules of the formal system and from the syntactic trees \mathcal{T}_i annotating the statements s_i . For example, let us consider the syntactic tree in Figure 5 annotating the following statement

S : Robot r5 inhibited wifi connection lost
while moving from area a37 to area a51

The sub-tree of the root node S is labeled with the non-terminal symbol SIP of FS . As introduced in Section 3, this symbol denotes a shifting-inhibition phrase constituent. This constituent encodes the behavior of a robot to either switch between processes or inhibit a stimulus by focusing on the process at hands. These two behaviors can be discriminated by looking at the the label of the right child of the node labeled with SIP. More precisely, if the label of the right child is labeled with the non-terminal symbol SwitchP in FS then the statement encodes a switching behavior. In other words, the production rule $\text{SIP} \rightarrow \text{RP SwitchP}$ has been applied to annotate the right branch of this sub-tree. Conversely, if the node is labeled with the non-terminal symbol InhibitP then the statement represents an inhibition behavior. This means that the node labeled with SIP

(a) S : Robot r2 picked up object o5 on top of table t6 behind object o1(b) S : Robot r3 sent map m8 of area a5 to robot r1**FIGURE 2** Syntactic trees of statements on the basis of production rules (A1)-(A16) in Appendix A.1.

has been expanded by applying the production rule $SIP \rightarrow RP \text{ InhibitP}$ of FS (more details about the meaning of these rules can be found in Appendix A.1). The label of the right child of the root in the syntactic tree in Figure 5 verifies the latter case. Therefore, from both the specification of both constituents and production rules of FS we know that the statement encodes a relation of inhibition *Inhibition* between three types of entities, the first of type *Robot*, the second of type *Stimulus* and the third of type *Process*. We also know that the set of constants $C_{i,Robot}$ of sort *Robot* contains a constant symbol *r5*, the set of constants $C_{i,Stimulus}$ of sort *Stimulus* includes a constant symbol *wifi connection lost* and, finally, the set of constants $C_{i,Process}$ of sort *Process* contains the symbol *moving*. By applying these simple heuristics to every pair $\langle s_i, \mathcal{T}_i \rangle$ the signature \mathfrak{S}_i of the MDRS representing inhibition behaviors is extracted from the treebank D (see left side of Figure 6).

Now, let \mathfrak{S}_i be the signature of S_i , extracted from a treebank, according to the heuristics mentioned above. The tensor \underline{Y}_i associated with S_i , with signature \mathfrak{S}_i , is build as follows. The number of dimensions of \underline{Y}_i is fixed to be equal to the arity K of $R_i \in S_i$. If the k -th input term of R_i is of sort $\sigma_k \in \Sigma_i$ then, along k -th direction the number of elements is fixed to be equal to the cardinality M_k of the constant set C_{i,σ_k} . Moreover, each index $i_k \in \{1, \dots, M_k\}$ is linked to one and only one constant symbol $c_{i,\sigma_k} \in C_{i,\sigma_k}$. The 3-order tensor associated with the three-dimensional relational structure representing inhibition behaviors of a robot, on the basis of these rules, is illustrated on the right side of Figure 6.

Finally, the last step of learning of MDRS is to estimate the values of the entries of the tensors. To this end, we propose a generative approach based on a probabilistic model, similar to those applied for document classification and information retrieval^{56,57,58}.

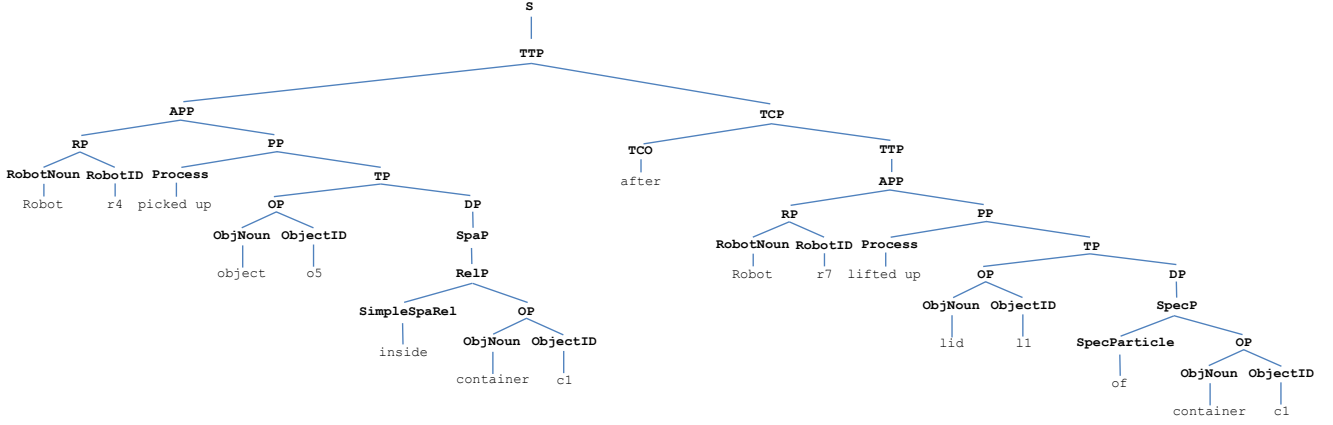


FIGURE 3 S : Robot r4 picked up object o5 inside container c1 after robot r7 lifted up lid l1 of container c1.

Let \mathcal{L}_i be the language defined by the signature \mathfrak{S}_i of the MDRS S_i . Let \mathcal{U}_i be the Herbrand Universe of \mathcal{L}_i , namely, the set of all ground terms of \mathcal{L}_i . Let \mathcal{B}_i be the Herbrand Base of the signature \mathfrak{S}_i , namely, the set of all possible ground atoms that can be formed from the relation symbol $R_i \in \mathfrak{S}_i$ and terms in \mathcal{U}_i . We denote with $\mathcal{B}_i^D \subseteq \mathcal{B}_i$ the sub-set of those ground atoms in \mathcal{B}_i whose argument terms label the leaf nodes of the syntactic trees \mathcal{T}_i annotating the statements s_i in the treebank D encoded by relation $R_i \in \mathfrak{S}_i$. Let \mathcal{H}_i be the Herbrand interpretation of $R_i \in \mathfrak{S}_i$ whose domain is \mathcal{U}_i , where every constant is interpreted as itself and where $R_i^{\mathcal{H}_i} \equiv \mathcal{B}_i^D$, that is, \mathcal{B}_i^D specifies which ground atoms are true in the interpretation.

Now, we introduce a binary vector $\mathbf{x} \in \{0, 1\}^W$ where W is the cardinality of \mathcal{B}_i^D and each component x_w is equal to 1 if the labels of the leaf nodes of the syntactic trees \mathcal{T}_i annotating the statements s_i encoded by relation $R_i \in \mathfrak{S}_i$ are the argument terms of the ground atoms which are true, that is, that are in $R_i^{\mathcal{H}_i}$. x_w is equal to 0 otherwise. According to this definition, $\sum_{w=1}^W x_w = 1$. We denote with $\mu_w \in [0, 1]$ the probability that $x_w = 1$. Then the probability of the vector \mathbf{x} is given by

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{w=1}^W \mu_w^{x_w} \quad \text{with} \quad \sum_{w=1}^W \mu_w = 1 \quad (2)$$

Let us assume that the pairs $\langle s_i, \mathcal{T}_i \rangle \in D$ are independent. Let us denote with m_w the number of times that the component $x_w = 1$ in D . According to this definition a treebank can be represented by a vector $\mathbf{m} = (m_1 \dots m_W)^\top$ of counts whose probability distribution follows a Multinomial with parameter vector $\boldsymbol{\mu}$

$$p(\mathbf{m}|\boldsymbol{\mu}) = \frac{N!}{\prod_{w=1}^W m_w!} \prod_{w=1}^W \mu_w^{m_w} \quad (3)$$

Here $N = \sum_{w=1}^W m_w$. Now, knowing that

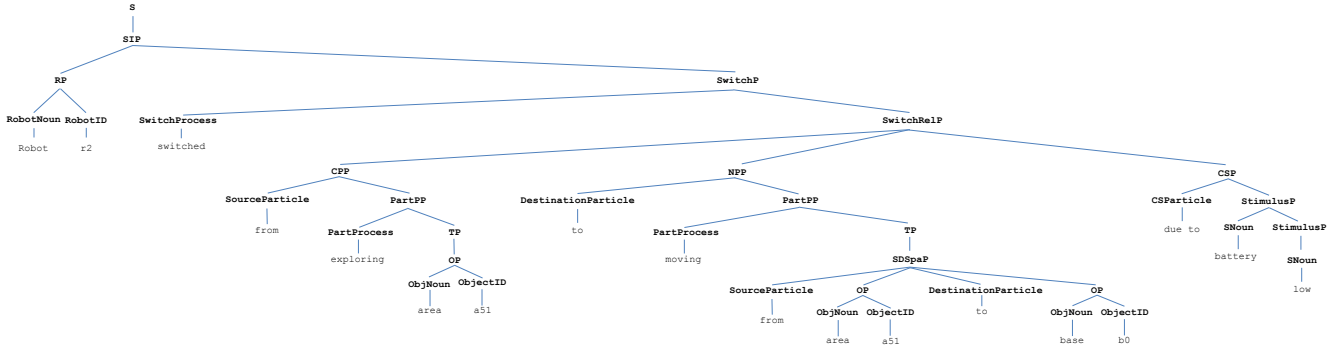
- from the definition of $\underline{\mathbf{Y}}_i$, each tuple of indices (i_1, \dots, i_K) , with $i_k = 1, \dots, M_k$, corresponds to a tuple $(c_{i, \sigma_1}, \dots, c_{i, \sigma_K})$ of constant symbols, with $c_{i, \sigma_k} \in C_{i, \sigma_k}$ and $\sigma_k \in \Sigma_i$ sort of the k -th input term of R_i ;
- the leaf nodes of the syntactic trees \mathcal{T}_i might or might not be labeled with symbols in the tuples $(c_{i, \sigma_1}, \dots, c_{i, \sigma_K})$;
- the terms $c_{i, \sigma_1}^{\mathcal{H}_i}, \dots, c_{i, \sigma_K}^{\mathcal{H}_i}$ might or might not belong to $R_i^{\mathcal{H}_i}$;

under these isomorphisms, the entries of the tensors $\underline{\mathbf{Y}}_i$ are filled according to the following rule

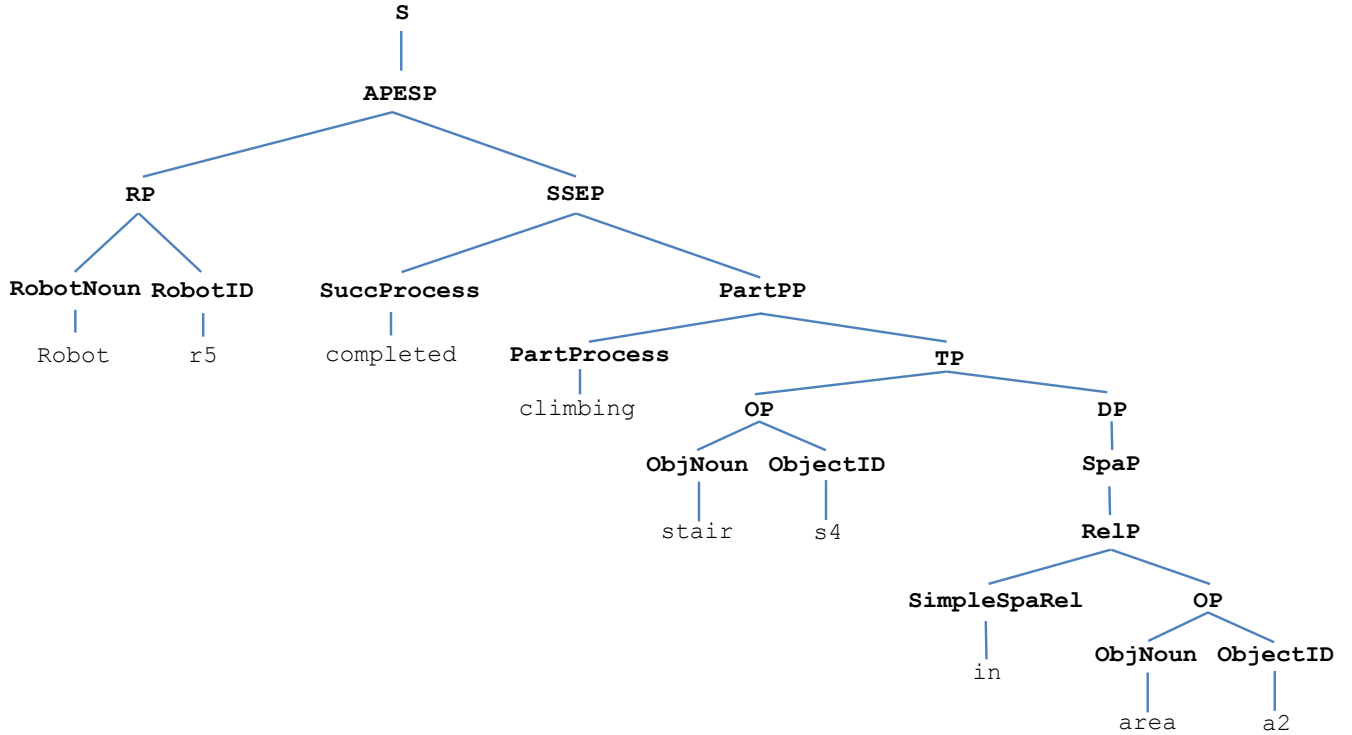
$$y_{i_1, \dots, i_K}^i = \begin{cases} p(x_w = 1 | \mathbf{m}) & \text{if } c_{i, \sigma_1}^{\mathcal{H}_i}, \dots, c_{i, \sigma_K}^{\mathcal{H}_i} \in R_i^{\mathcal{H}_i} \\ \text{null} & \text{otherwise} \end{cases}$$

Here null is introduced as a place-holder to denote the missing entries of the tensor. $p(x_w = 1 | \mathbf{m})$ is the posterior predictive probability of the term $c_{i, \sigma_1}^{\mathcal{H}_i}, \dots, c_{i, \sigma_K}^{\mathcal{H}_i}$ associated with the tuple of indices (i_1, \dots, i_K) . This probability is computed according to the following distribution

$$p(x_w = 1 | \mathbf{m}) = \int p(x_w = 1 | \boldsymbol{\mu}) p(\boldsymbol{\mu} | \mathbf{m}) d\boldsymbol{\mu} \quad (4)$$



(a) S : Robot r2 switched from exploring area a51 to moving from area a51 to base b0 due to battery low



(b) S : Robot r5 completed climbing stair s4 in area a2

FIGURE 4 Syntactic trees of statements according to production rules (A20)-(A35) in Appendix A.1.

Here $p(\mu|\mathbf{m})$ is the posterior distribution of μ . This distribution is obtained by multiplying the likelihood in eq. (2) with the prior distribution of μ . Since the distribution over the space of the parameter vector μ is confined to a simplex of dimensionality $W-1$, as a consequence of the constraints $0 \leq \mu_w \leq 1$ and $\sum_{w=1}^W \mu_w = 1$, a Dirichlet prior with concentration parameters $\alpha_1, \dots, \alpha_W$ is chosen. Knowing that the Dirichlet is a conjugate prior of the Multinomial and that the posterior distribution is also Dirichlet, the posterior predictive distribution in eq. (4) takes the following form

$$\begin{aligned}
 p(x_w=1|\mathbf{m}) &= \int p(x_w=1|\mu_w) \left[\int p(\mu_{-w}, \mu_w|\mathbf{m}) d\mu_{-w} \right] d\mu_w \\
 &= \int \mu_w p(\mu_w|\mathbf{m}) d\mu_w = \mathbb{E}[\mu_w|\mathbf{m}] = \frac{m_w + \alpha_w}{\sum_{w=1}^W (m_w + \alpha_w)}
 \end{aligned} \tag{5}$$

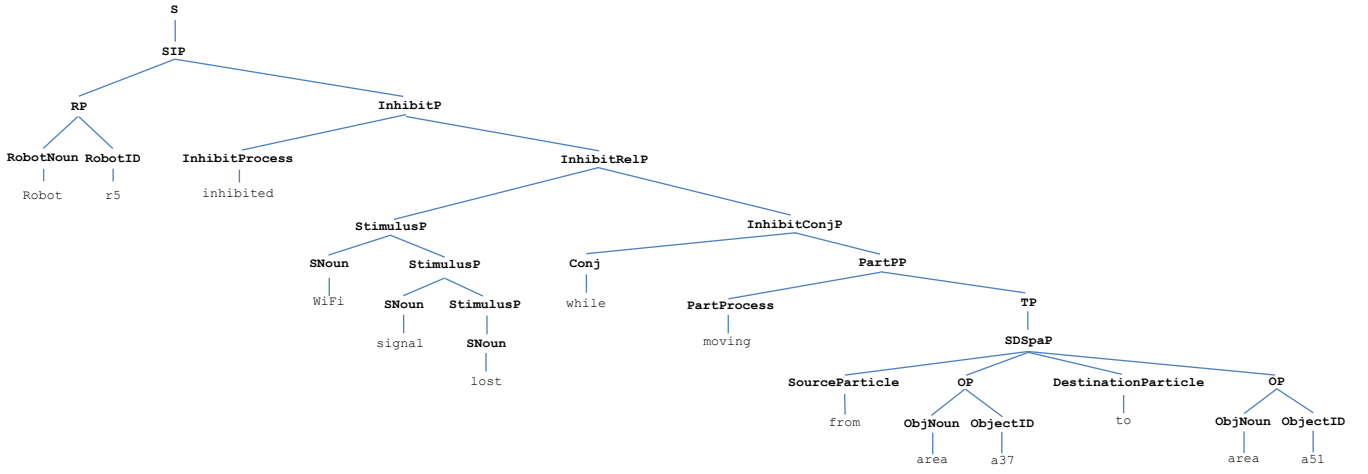


FIGURE 5 Syntactic tree of the statement S : Robot r5 inhibited wifi connection lost while moving from area a37 to area a51.

$$\Sigma_i = \{\text{Robot}, \text{Stimulus}, \text{Process}\}$$

$$R_i(\cdot, \cdot, \cdot) \stackrel{\text{def}}{=} \text{Inhibition}(\cdot, \cdot, \cdot)$$

$$C_{i,\text{Robot}} = \{\text{r5}, \dots\}$$

$$C_{i,\text{Stimulus}} = \{\text{wifi connection lost}, \dots\}$$

$$C_{i,\text{Process}} = \{\text{moving}, \dots\}$$

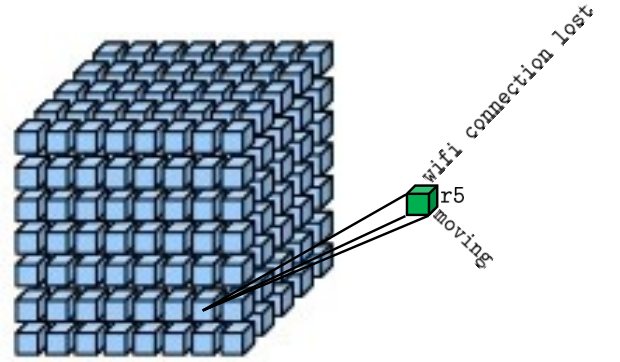


FIGURE 6 Signature and the 3-order tensor associated with the three-dimensional relational structure representing inhibition behaviors of a robot, with respect to both processes and stimulus occurrences

that is, it is equal to the expected value $\mathbb{E}[\mu_w | \mathbf{m}]$ of the posterior. Therefore, when the rule in (4) is satisfied, the values of the elements y_{i_1, \dots, i_K}^j of a tensor \mathbf{Y}_i are fixed to be equal to the probabilities computed according to eq. (5).

However, after this step, the tensor turns out to be sparse due to the lack of ground atoms in the interpretation $R_i^{\mathcal{H}_i}$. This depends upon the lack of statements in the treebank. In order to fill the missing entries of the tensors we resort to Non-Negative Tensor Decomposition (NTD)^{59,60,61}. Apart from completing the tensors, there are other reasons which motivate its application in the context of multi-robot collaboration. NTD reduces the dimensionality of the tensor through factorization. The components resulting from this factorization require less computational resources for both storage and information retrieval. NTD also filters the data thus reducing noise. Finally, by applying NTD, new knowledge can be discovered through link prediction^{62,63}.

5 | LATENT KNOWLEDGE ESTIMATION THROUGH NON-NEGATIVE TENSOR DECOMPOSITION (NTD) WITH MISSING DATA

Let $\mathbf{Y}_i \in \mathbb{R}_+^{M_1 \times \dots \times M_K}$ be the K -order tensor associated with a structure S_i . Given a positive index J , with $J \leq \min(M_1, \dots, M_K)$, NTD factorizes \mathbf{Y}_i into a set of K non-negative component matrices $\mathbf{A}^{(k)} = [\mathbf{a}_1^{(k)}, \mathbf{a}_2^{(k)}, \dots, \mathbf{a}_J^{(k)}] \in \mathbb{R}_+^{M_k \times J}$, with $k=1, \dots, K$, called

common loading factors, such that

$$\begin{aligned}\underline{\mathbf{Y}}_i &= \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(K)} + \underline{\mathbf{E}}_i \\ &= \underline{\mathbf{I}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \dots \times_K \mathbf{A}^{(K)} + \underline{\mathbf{E}}_i \\ &= \sum_{j=1}^J \mathbf{a}_j^{(1)} \circ \mathbf{a}_j^{(2)} \circ \dots \circ \mathbf{a}_j^{(K)} + \underline{\mathbf{E}}_i\end{aligned}\quad (6)$$

Here, \times_k denotes the mode- k tensor-matrix product, $\underline{\mathbf{I}}$ is the identity tensor and $\underline{\mathbf{E}}_i$ is the residual error. \circ denotes the outer product. The model in eq. (6) is often referred to as the Parallel Factor Analysis (PARAFAC) model with non-negativity constraints^{61,64}. Our objective here is to estimate the non-negative component matrices $\mathbf{A}^{(k)}$ or, equivalently, the set of vectors $\mathbf{a}_j^{(k)}$, with $k=1, \dots, K$ and $j=1, \dots, J$ given the number of factors J . Approaches based on Alternating Least Squares (ALS) minimization of the squared Euclidean distance are commonly employed for estimating the component matrices in NTD⁵⁹. However, these approaches do not deal with missing entries in $\underline{\mathbf{Y}}_i$ due to the incompleteness of the observations (see Section 4). In order to cope with this issue, we propose an approach which embeds a variant of ALS, named Fast Hierarchical Alternating Least Squares (F-HALS)⁶⁵, into an imputation-alternation schema⁶⁶. In this approach the missing values of $\underline{\mathbf{Y}}_i$ are imputed using the interim model $\hat{\underline{\mathbf{Y}}}_i = \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(K)}$, computed at the n -th iteration, as follows

$$\tilde{\underline{\mathbf{Y}}}_i = \underline{\mathbf{W}}_i \otimes \underline{\mathbf{Y}}_i + (\underline{\mathbf{1}} - \underline{\mathbf{W}}_i) \otimes \hat{\underline{\mathbf{Y}}}_i \quad (7)$$

213 Here $\underline{\mathbf{W}}_i \in \{0, 1\}^{M_1 \times \dots \times M_K}$ is the indicator tensor specifying which entries are missing in $\underline{\mathbf{Y}}_i$, $\underline{\mathbf{1}}$ is a tensor of the same dimension
214 of $\underline{\mathbf{W}}_i$ whose elements are all equal to 1 and \otimes is the Hadamard product. Once $\tilde{\underline{\mathbf{Y}}}_i$ is generated, the factor matrices are then
215 updated via the sequential minimization of a set of local cost functions with the same global minima (e.g., squared Euclidean
216 distances) performed by F-HALS⁶⁵. $\hat{\underline{\mathbf{Y}}}_i$ is then updated at every iteration according to (6) as well as $\tilde{\underline{\mathbf{Y}}}_i$, on the basis of eq. (7).

nonNegativeRandomInitnonNegativeRandomInit diagdiag More precisely, let us consider the following set of local functions

Algorithm 1 Imputation-Alternation Fast HALS

$\underline{\mathbf{Y}}_i$: K -order tensor; J : number of common loading factors $\underline{\mathbf{W}}_i$: indicator tensor.

K factors $\mathbf{A}^{(k)} \in \mathbb{R}_+^{M_k \times J}$ such that the cost functions in eq. (8) are minimized.

for (**do***[f]initialization) $k = 1 \ K \ \mathbf{A}^{(k)} \leftarrow ()$

for (**do***[f]normalization) $j = 1 \ J \ \mathbf{a}_j^{(k)} \leftarrow \mathbf{a}_j^{(k)} / \|\mathbf{a}_j^{(k)}\|_2 \quad \mathbf{T}^{(1)} \leftarrow ((\mathbf{A}^{(1)})^\top \mathbf{A}^{(1)}) \otimes \dots \otimes ((\mathbf{A}^{(K)})^\top \mathbf{A}^{(K)}) \quad \hat{\underline{\mathbf{Y}}}_i \leftarrow \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(K)}$
 $\tilde{\underline{\mathbf{Y}}}_i \leftarrow \underline{\mathbf{W}}_i \otimes \underline{\mathbf{Y}}_i + (\underline{\mathbf{1}} - \underline{\mathbf{W}}_i) \otimes \hat{\underline{\mathbf{Y}}}_i$; *[f]imputation step

repeat a stopping criteria is met $\gamma \leftarrow ((\mathbf{A}^{(K)})^\top \mathbf{A}^{(K)})$

for (**do***[f]alternation step) $k = 1 \ K \ k = K \ \gamma \leftarrow \mathbf{T}^{(2)} \leftarrow \tilde{\underline{\mathbf{Y}}}_{i,(k)} \mathbf{A}^{\ominus-k} \quad \mathbf{T}^{(3)} \leftarrow \mathbf{T}^{(1)} \oslash ((\mathbf{A}^{(k)})^\top \mathbf{A}^{(k)})$

for (**do***[f]update rule in eq. (12)) $j = 1 \ J \ \mathbf{a}_j^{(k)} \leftarrow \left[\gamma_j \mathbf{a}_j^{(k)} + [\mathbf{T}^{(2)}]_j - \mathbf{A}^{(k)} [\mathbf{T}^{(3)}]_j \right]_+ \quad k \neq K \ \mathbf{a}_j^{(k)} \leftarrow \mathbf{a}_j^{(k)} / \|\mathbf{a}_j^{(k)}\|_2$
 $\mathbf{T}^{(1)} \leftarrow \mathbf{T}^{(3)} \otimes ((\mathbf{A}^{(K)})^\top \mathbf{A}^{(K)}) \quad \hat{\underline{\mathbf{Y}}}_i \leftarrow \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(K)} \quad \tilde{\underline{\mathbf{Y}}}_i \leftarrow \underline{\mathbf{W}}_i \otimes \underline{\mathbf{Y}}_i + (\underline{\mathbf{1}} - \underline{\mathbf{W}}_i) \otimes \hat{\underline{\mathbf{Y}}}_i$; *[f]imputation step

$$D_F^{(j)}(\mathbf{a}_j^{(1)}, \mathbf{a}_j^{(2)} \dots \mathbf{a}_j^{(K)}) = \frac{1}{2} \|\tilde{\underline{\mathbf{Y}}}_i^{(j)} - \mathbf{a}_j^{(1)} \circ \mathbf{a}_j^{(2)} \circ \dots \circ \mathbf{a}_j^{(K)}\|_F^2 \quad (8)$$

for $j=1, \dots, J$, subject to the non-negativity constraints. Here $\mathbf{a}_j^{(k)} \in \mathbb{R}_+^{M_k}$ are the j -th column vectors of the loading matrices $\mathbf{A}^{(k)}$, with $k=1, \dots, K$. $\|\cdot\|_F$ is the Frobenius norm. $\tilde{\underline{\mathbf{Y}}}_i^{(j)}$ is the j -th sub-tensor of $\tilde{\underline{\mathbf{Y}}}_i$, defined as follows

$$\begin{aligned}\tilde{\underline{\mathbf{Y}}}_i^{(j)} &= \tilde{\underline{\mathbf{Y}}}_i - \sum_{p \neq j} \mathbf{a}_p^{(1)} \circ \mathbf{a}_p^{(2)} \circ \dots \circ \mathbf{a}_p^{(K)} \\ &= \tilde{\underline{\mathbf{Y}}}_i - \left[\sum_{p=1}^J \mathbf{a}_p^{(1)} \circ \mathbf{a}_p^{(2)} \circ \dots \circ \mathbf{a}_p^{(K)} - \left(\mathbf{a}_j^{(1)} \circ \mathbf{a}_j^{(2)} \circ \dots \circ \mathbf{a}_j^{(K)} \right) \right]\end{aligned}\quad (9)$$

Based on the mode- k unfolding representation of a tensor, the cost function in eq. (8) can be re-written as follows

$$D_F^{(j)}(\mathbf{a}_j^{(1)}, \mathbf{a}_j^{(2)} \dots \mathbf{a}_j^{(K)}) = \frac{1}{2} \|\tilde{\underline{\mathbf{Y}}}_{i,(k)}^{(j)} - \mathbf{a}_j^{(k)} \{\mathbf{a}_j\}^{\ominus-k \top}\|_F^2 \quad (10)$$

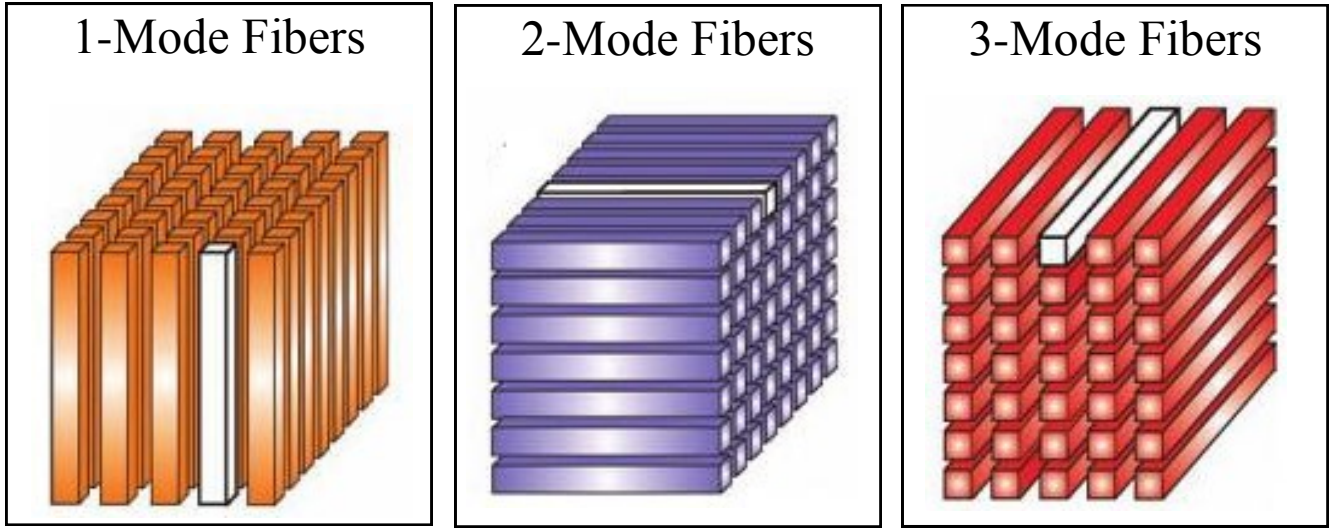


FIGURE 7 From left to right, 1-mode, 2-mode and 3-mode fibers of a third-order tensor.

for $j=1, \dots, J$ and $k=1, \dots, K$, where

$$\begin{aligned} \{\mathbf{a}_j\}^{\odot_{-k}\top} &= [\mathbf{a}_j^{(K)}]^\top \odot \dots \odot [\mathbf{a}_j^{(k+1)}]^\top \odot [\mathbf{a}_j^{(k-1)}]^\top \odot \dots \odot [\mathbf{a}_j^{(1)}]^\top \\ &= [\mathbf{A}^{\odot_{-k}\top}]_j \quad (\text{Kharti-Rao property}) \end{aligned} \quad (11)$$

Here \odot denotes the Kharti-Rao product. By setting the gradient of the cost functions in eq. (10) to zero, by replacing $\tilde{\mathbf{Y}}_{i,(k)}^{(j)}$ terms by those in eq. (9) and by exploiting the property of the Kharti-Rao and Kronecker product in eq. (11) we arrive at the update rules referred to as the Fast HALS NTD algorithm

$$\mathbf{a}_j^{(k)} \leftarrow \left[\gamma_j^{(k)} \mathbf{a}_j^{(k)} + [\tilde{\mathbf{Y}}_{i,(k)} \mathbf{A}^{\odot_{-k}}]_j - \mathbf{A}^{(k)} \left[\{\mathbf{A}^\top \mathbf{A}\}^{\otimes} \oslash (\mathbf{A}^{(k)})^\top \mathbf{A}^{(k)} \right]_j \right]_+ \quad (12)$$

for $j=1, \dots, J$ and $k=1, \dots, K$. Here $\gamma_j^{(k)}$ are the scaling coefficients. $\tilde{\mathbf{Y}}_{i,(k)}$ is the mode- k unfolding representation of $\tilde{\mathbf{Y}}_i$.

$$\{\mathbf{A}^\top \mathbf{A}\}^{\otimes} = ((\mathbf{A}^{(1)})^\top \mathbf{A}^{(1)}) \otimes \dots \otimes ((\mathbf{A}^{(K)})^\top \mathbf{A}^{(K)}) \quad (13)$$

\oslash denotes the element-wise division. $[\cdot]_+$ is the non-linear half-wave rectifying projection replacing negative values of the argument by zero or by a small positive value ϵ . $[\cdot]_j$ selects the j -th column vector of the matrix argument. At the n -th iteration the common loading factors first are updated according to the rules in eq. (12) and then are fed into eq. (7) to update both $\hat{\mathbf{Y}}_i$ and $\tilde{\mathbf{Y}}_i$ for the next iteration. This imputation-alternation schema is illustrated in Algorithm 1.

Now, let $\mathbf{A}^{(k)} = [\mathbf{a}_1^{(k)}, \mathbf{a}_2^{(k)}, \dots, \mathbf{a}_J^{(k)}] \in \mathbb{R}_+^{M_k \times J}$, for $k=1, \dots, K$, be the set of common loading factors, estimated according to the imputation-alternation Fast HALS in Algorithm 1. Let (i_1, i_2, \dots, i_K) be a fixed tuple of indices, with $i_k=1, \dots, M_k$. Then, the estimation of the value of the tensor at position (i_1, i_2, \dots, i_K) is given by

$$\hat{y}_{i_1, i_2, \dots, i_K} = \sum_{j=1}^J a_{i_1, j}^{(1)} a_{i_2, j}^{(2)} \dots a_{i_K, j}^{(K)} \quad (14)$$

However, reasoning about multiple choices for decision-making in a multi-robot system requires that entire one-dimensional fragments are extracted from the tensors. To this end, we resort to mode- k fiber operations on the tensors. Fibers can be commonly obtained from tensors by fixing all indices except one, as illustrated in Figure 7. In Example 5.1 we illustrate how these operations support reasoning as well as how link prediction handles knowledge discovery.

Example 5.1. Let us consider the scenario described in Example 2.1. UGV1 has to grasp an object for which it is required the collaboration of another UGV. Therefore, we have to take a decision regarding which UGV has to take at hands this collaborative task. Now, let us assume that the set $C_{1, \text{Robot}}$ of the constant symbols of sort Robot includes another constant symbol referred to another UGV, that is UGV4. This means that in the treebank under consideration, there was a pair $\langle s_i, \mathcal{T}_i \rangle$ which encoded a

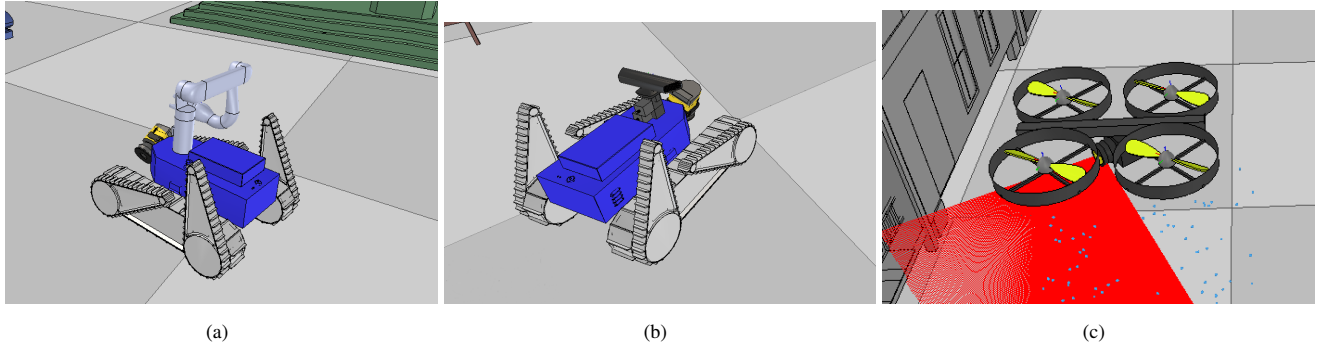


FIGURE 8 (a) First UGV endowed with an arm for manipulating objects; (b) second UGV equipped with a Pan-Tilt Unit with on top RGB-D camera sensor for dense 3D reconstruction; (c) UAV endowed with a 2D laser range finder, mounted on a servomotor, on the bottom of the chassis for 3D scanning of the surrounding

temporal relation `Equal` between four entities, two of type `Robot` and two of type `Process` and such that `UGV4` appeared as the label of a leaf of the right sub-tree of \mathcal{T}_i . Moreover, let us assume that the tuple $\langle \text{UGV1}, \text{grasping}, \text{UGV4}, \text{grasping} \rangle \notin \text{Equal}^{\mathcal{H}_1}$, namely, it does not belong to the Herbrand interpretation \mathcal{H}_1 of `Equal`. This means that there was no pair $\langle s_i, \mathcal{T}_i \rangle$ in the treebank that, as above, encoded a temporal relation `Equal` and it was such that `UGV1` appeared as the label of a leaf of the left sub-tree of \mathcal{T}_i . Consequently, the ground atom $\text{Equal}(\text{UGV1}, \text{grasping}, \text{UGV4}, \text{grasping}) \notin \mathcal{B}_1^D$ (see Section 4). Therefore, after the completion of the procedure of building of the tensor $\underline{\mathbf{Y}}_1$, the element in position $(i_1, i_2, \tilde{i}_3, i_4)$ of $\underline{\mathbf{Y}}_1$, corresponding to the term $(\text{UGV1}, \text{grasping}, \text{UGV4}, \text{grasping})$ results to be empty.

However, after the factorization of $\underline{\mathbf{Y}}_1$, on the basis of Algorithm 1, we have found a set $\mathbf{A}^{(k)} = [\mathbf{a}_1^{(k)}, \mathbf{a}_2^{(k)}, \dots, \mathbf{a}_J^{(k)}]$ of factor matrices, with $k=1, \dots, 4$, such that

$$\hat{y}_{i_1, i_2, \tilde{i}_3, i_4} = \sum_{j=1}^J a_{i_1, j}^{(1)} a_{i_2, j}^{(2)} a_{\tilde{i}_3, j}^{(3)} a_{i_4, j}^{(4)} \neq \text{null} \quad (15)$$

that is, we have an estimate of the value of tensor element associated with the term $\langle \text{UGV1}, \text{grasping}, \text{UGV4}, \text{grasping} \rangle$, even if this term was not present in the treebank. In other words, through the decomposition, we have discovered new knowledge. Now, by suitably performing a fiber operation on the tensor we obtain a one-dimensional fragment whose entries are an estimation of the collaboration of `UGV1` with all the other UGVs (or at least with all included in the domain of discourse). By interpreting this fragment as a recommendation vector, we can take a decision about which UGV has to be in charge of supporting `UGV1` in the task of grasping the object⁶⁷.

6 | EXPERIMENTS

Through this paper we have shown, with several examples, the main features of the framework. In particular, the use of a many-sorted first order logic, even if restricted to conjunctions of ground atoms, makes the framework expressive enough for modeling a heterogeneous team of robots. The use of temporal relations makes the framework expressive enough for encoding a wide class of collaborative tasks among the robots. Operations on tensors allow us to reason about multiple choices in role and task assignment. Tensor decomposition endows the framework with the capability of both dealing with missing information and discovering new knowledge through link prediction. In this section we aim to show another key attribute of the framework, namely, the capability of supporting inference for task assignment in communication denied situations.

In this regard, we built a multi-robot system, composed of two UGVs and one UAV, within a Virtual Simulated Environment. This environment comprises two main software modules. First module is responsible of modeling the dynamics of the system through a physics engine. This engine is based on the well-known Bullet Physics Library⁶⁸. First module also implements a layer of bi-directional interfaces which allows for the integration with the Robot Operating System (ROS)⁵⁰. The second software module is responsible of interconnecting the main functionalities of the robots under consideration (e.g., mapping, planning and control), developed in ROS, with the physics engine, through these bi-directional interfaces. First module has been developed

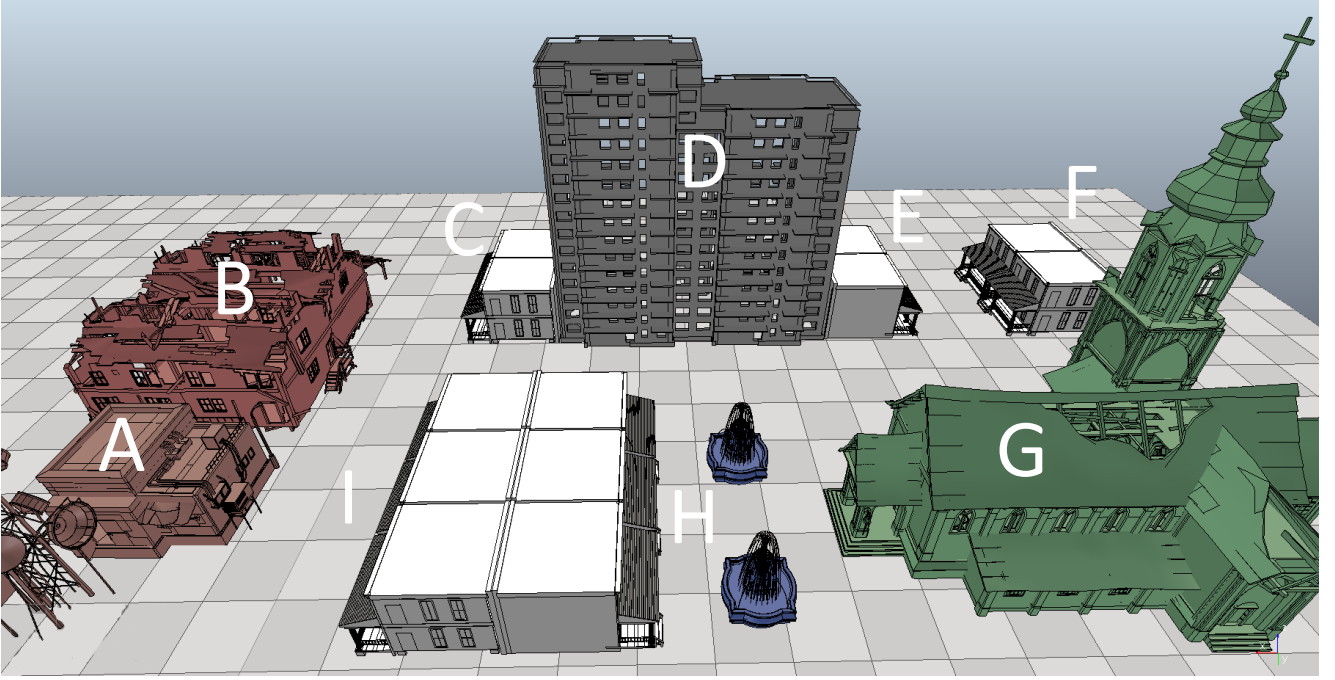


FIGURE 9 Simulated rescue scenario. Buildings are accessible from either staircases, or from windows, or from holes in the ceilings.

within the cross-platform V-REP⁶⁹. First UGV is endowed with an arm for manipulating objects (see Figure 8(a)). Second UGV is equipped with a Pan-Tilt Unit with on top RGB-D camera sensor for dense 3D reconstruction (see Figure 8(b)). The UAV is endowed with a 2D laser range finder, mounted on a servomotor, on the bottom of the chassis for 3D scanning of the surrounding (see Figure 8(c)). These robots are deployed in a urban virtual environment simulating a disaster scenario. The scenario is composed of 9 buildings (see Figure 9). Some of them are only accessible from a staircase. Others are accessible only from windows since we suitably simulated doors blocked at the ground floors. In others, it can be possible to enter only from holes in the roof caused by the collapse. Referred to the scenario in Figure 9, buildings A, C, E, F, H and I are only accessible through staircases leading to the free doors of the ground floors. Building B is accessible through both the hole in the roof and a door at the first floor. This door has not been blocked. Similarly, building G, namely the church on the right of Figure 9), is accessible through both the holes in the roof and a door. However, this door has been blocked. Building D is accessible only via its windows. The goal of the mission is to ensure that each building has been visited at least once by at least a robot of the team. A graph-based topological representation of the virtual environment is provided to all the robots^{70,71}. We simulate the presence of a communication channel through which robots can exchange information about the buildings yet to be visited as well as those already explored. A building is considered explored if a robot visited the node of the topological graph associated with that building. We also simulate events which resemble in the virtual environment, the deterioration of the channel, up to deny the communication between a robot and the rest of the team.

For the purpose of this experiment, we modeled this team of robots with the following MDRS

$$\begin{aligned}
 S_1 : \\
 K_1 = 4, \quad \Sigma_1 = \{\text{Robot}, \text{Process}, \text{Stimulus}\}, \\
 R_1(\cdot, \cdot, \cdot, \cdot) \stackrel{\text{def}}{=} \text{Switch}(\cdot, \cdot, \cdot, \cdot), \\
 C_{1,\text{Robot}} = \{\text{UGV1}, \text{UGV2}, \text{UAV1}\}, \quad C_{1,\text{Process}} = \{\text{grasping}, \text{moving}, \dots\} \quad \text{and} \\
 C_{1,\text{Stimulus}} = \{\text{elapsed time}, \text{battery low}, \text{missing information}, \dots\}
 \end{aligned}$$

$$\begin{aligned}
S_2 : \\
K_2 = 4, \quad \Sigma_2 = \{\text{Robot}, \text{Stimulus}, \text{Process}\}, \\
R_2(\cdot, \cdot, \cdot) \stackrel{\text{def}}{=} \text{Inhibition}(\cdot, \cdot, \cdot), \\
C_{2,\text{Robot}} = \{\text{UGV1}, \text{UGV2}, \text{UAV1}\}, \quad C_{2,\text{Process}} = \{\text{climbing}, \text{flying}, \dots\} \quad \text{and} \\
C_{1,\text{Stimulus}} = \{\text{wifi connecton lost}, \text{subtrack failure}, \dots\}
\end{aligned}$$

$$\begin{aligned}
S_3 : \\
K_3 = 2, \quad \Sigma_3 = \{\text{Robot}, \text{Process}\}, \quad R_3(\cdot, \cdot) \stackrel{\text{def}}{=} \text{Process}(\cdot, \cdot), \\
C_{3,\text{Robot}} = \{\text{UGV1}, \text{UGV2}, \text{UAV1}\} \quad \text{and} \quad C_{3,\text{Process}} = \{\text{track}, \text{explore}, \dots\}.
\end{aligned}$$

$$\begin{aligned}
S_4 : \\
K_4 = 2, \quad \Sigma_4 = \{\text{Robot}, \text{Process}\}, \quad R_4(\cdot, \cdot) \stackrel{\text{def}}{=} \text{FailProcess}(\cdot, \cdot), \\
C_{4,\text{Robot}} = \{\text{UGV1}, \text{UGV2}, \text{UAV1}\} \quad \text{and} \quad C_{4,\text{Process}} = \{\text{adapt}, \text{pick up}, \dots\}.
\end{aligned}$$

$$\begin{aligned}
S_5 : \\
K_5 = 2, \quad \Sigma_5 = \{\text{Robot}, \text{Process}\}, \quad R_5(\cdot, \cdot) \stackrel{\text{def}}{=} \text{SuccProcess}(\cdot, \cdot), \\
C_{5,\text{Robot}} = \{\text{UGV1}, \text{UGV2}, \text{UAV1}\} \quad \text{and} \quad C_{5,\text{Process}} = \{\text{fly}, \text{climb}, \text{compute}, \text{tracking}, \dots\}
\end{aligned}$$

$$\begin{aligned}
S_6 : \\
K_6 = 4, \quad \Sigma_6 = \{\text{Robot}, \text{Object}\}, \\
R_6(\cdot, \cdot, \cdot, \cdot) \stackrel{\text{def}}{=} \text{SendProcess}(\cdot, \cdot, \cdot, \cdot), \\
C_{6,\text{Robot}} = \{\text{UGV1}, \text{UGV2}, \text{UAV1}\}, \\
C_{6,\text{Object}} = \{\text{base0}, \text{buildingA}, \text{buildingB}, \text{map0}, \text{map1}, \text{door2}, \text{stair4}, \dots\}
\end{aligned}$$

$$\begin{aligned}
S_7 : \\
K_7 = 4, \quad \Sigma_7 = \{\text{Robot}, \text{Object}\}, \\
R_7(\cdot, \cdot, \cdot, \cdot) \stackrel{\text{def}}{=} \text{RequestProcess}(\cdot, \cdot, \cdot, \cdot), \\
C_{7,\text{Robot}} = \{\text{UGV1}, \text{UGV2}, \text{UAV1}\}, \\
C_{7,\text{Object}} = \{\text{base0}, \text{buildingA}, \text{buildingB}, \text{buildingC}, \text{buildingD}, \dots\}
\end{aligned}$$

$$\begin{aligned}
S_8 : \\
K_8 = 4, \quad \Sigma_8 = \{\text{Robot}, \text{Process}\}, \quad R_8(\cdot, \cdot, \cdot, \cdot) \stackrel{\text{def}}{=} \text{Before}(\cdot, \cdot, \cdot, \cdot), \\
C_{8,\text{Robot}} = \{\text{UGV1}, \text{UGV2}, \text{UAV1}\} \quad \text{and} \quad C_{8,\text{Process}} = \{\text{map}, \text{explore}, \text{acquire}, \dots\}.
\end{aligned}$$

272 At the beginning of the mission, we manually instructed the UGV endowed with the arm, named in the above MDRS UGV1,
 273 to explore the ground floor of building B. We instructed the UAV, encoded in the MDRS with UAV1 to enter into building D
 274 through a window in order to internally inspect the building. Finally, we manually instructed the UGV equipped with a Pan-Tilt
 275 RGB-D camera, represented in the framework by UGV2, to move inside building H. After this initial phase of bootstrapping,
 276 UGV1 successfully computed a path from its current position, represented by term base0 in the MDRS, to building B (encoded
 277 in the MDRS by the symbol buildingB) and started to execute it⁷². Similarly, both UGV2 and UAV1 started to move toward their
 278 own destination targets^{73,74}. During the phase of climbing the stairs at the entrance of building B, trajectory tracking process⁷⁵
 279 of UGV1 failed, causing the interruption of the task. Here, UGV1 has two possible choices: (1) recompute the path from its actual
 280 position to the target or (2) send a request of support to another robot. In order to show how the proposed framework handles
 281 collaborative tasks we forced UGV1 to select the second choice, namely ask for support. From the treebank we learned that UAV1
 282 built the map of a certain area before UGV1 explored that area. Moreover, we learnt that UGV1 requested the map of that area to
 283 UAV1 and that UAV1 sent this map to UGV1. On the other hand, from tensor decomposition we also have an estimate of this form

of collaboration between UGV1 and UGV2. Therefore, UGV1 knows that before exploring building B both UAV1 and UGV2 have to build the map of building B. In this regard, UGV1 has to take a decision concerning which robot has to be contacted in order to receive from it the map of building B. According to the estimates of this form of collaboration UGV1 chose UAV1. On the basis of this choice, UGV1 sent a request to UAV1, UAV1 flew over building B. Then UAV1 sent the map of the building to UGV1 which first integrated it with its own map of the area, then performed traversability analysis⁷⁶ and, successively, re-started to climb the stairs.

In the meanwhile, UGV2 finished to explore building H and it was moving toward another node of the topological graph which has not been visited yet, namely that corresponding to building F. At this point of the mission we simulated an event which interrupted the communication between UGV2 and both UGV1 and UAV1. UGV1 and UAV1 only know that UGV2 was moving toward building H, but they do not know the reason why UGV2 is not reachable on the communication channel. Moreover, they do not know whether UGV2 is still executing the task at hands or it has switched to another task. From the treebank we learnt that during a motion task UGV2 might be elicited by several stimuli, such as `wifi connecton lost`, `subtrack failure`, `battery low`, `tip over`, etc. We also learnt the behavior of UGV2 in the case in which one of these stimuli occur. However, both UGV1 and UAV1 know neither which event occurred, causing the interruption of the communications with UGV2 nor which task is currently carrying on. That is the situation in which both the result of the tensors factorization and the mode- k operations on the tensors can be employed for estimating the current status of UGV2. Indeed, by suitably extracting the one-dimensional fragments of the tensors $\underline{\mathbf{Y}}_1$ and $\underline{\mathbf{Y}}_2$, associated with the multi-dimensional structures S_1 and S_2 , respectively, we can have an estimate of the kind of stimuli which occurred as well as the response behavior of UGV2 to these stimuli. In fact, these estimate tell us whether UGV2 shifted from the current task to another task and which task has been chosen due to the occurrence of a stimulus (and which stimulus occurred) or if the robot inhibited the stimulus (and which stimulus has been inhibited) to focus on the task at hand. After reasoning on these estimates we discover that the stimulus `wifi connecton lost` might be occurred and that UGV2 probably inhibited this stimulus to continue its current task, that is, `explore building F`. Note that if UGV2 had shifted to another task such as `go back to the base station base0` then building F resulted to be not visited and a re-allocation of the task of either UGV1 or of UAV1 would had been needed in order to accomplish the goal of the mission.

7 | CONCLUSIONS

In this work we proposed a framework capable of learning, from data reporting the situated history of the activities performed by a team of robots, a model of collaboration. From the data we extracted information about the relations linking the robots, the tasks and the content-dependent features of the collaborative tasks. This information is used to define both the set of symbols and terms of the many-sorted first-order language encoding multi-robot collaboration. This information is further used to learn the geometrical spaces underlying both individual and group task execution. The spaces are decomposed for obtaining the latent factors regulating the tasks of the robots. The decomposition is also used for dealing with the lack of information in the data as well as for discovering new forms of collaborations between robots, never encountered in the memorized task episodes. We demonstrated performance in simulation experiments by using a team of robots. In the future, we plan to demonstrate the usefulness of such a framework in different robotic domains or scenarios. In this context, more simulated experiments with different environments are needed. Finally, the impact of this framework will be measured.

ACKNOWLEDGMENTS

This research is supported by EU-FP7-ICT-Project TRADR 609763. It also partially supported by the UK EPSRC funded Programme Grant Project, National Centre for Nuclear Robotics (NCNR) EP/R02572X/1.

Author contributions

Mario Gianni wrote the article, implemented the framework. Mohammad Salah Uddin performed the experiments, summarized the results and revised the article.

Conflict of interest

The authors declare no potential conflict of interests.

References

1. Tadokoro S. Earthquake Disaster and Expectation for Robotics. In: Tadokoro S., ed. *Rescue Robotics* Springer London. 2009 (pp. 1–16)
2. Casper J, Murphy RR. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 2003; 33(3): 367–385.
3. Burke JL, Murphy RR, Coover MD, Riddle DL. Moonlight in Miami: Field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise. *Human-Computer Interaction* 2004; 19(1-2): 85–116.
4. Parker LE. Multiple Mobile Robot Systems. In: Siciliano B, Khatib O., eds. *Springer Handbook of Robotics* Springer Berlin Heidelberg. 2008 (pp. 921–941)
5. Stenmark M, Malec J, Nilsson K, Robertsson A. On Distributed Knowledge Bases for Robotized Small-Batch Assembly. *IEEE Transactions on Automation Science and Engineering* 2015; 12(2): 519–528. doi: 10.1109/TASE.2015.2408264
6. Zhang F, Chen W, Xi Y. Improving Collaboration through Fusion of Bid Information for Market-based Multi-robot Exploration. In: ; 2005: 1157–1162
7. Lim C, Mamat R, Braunl T. Market-based approach for multi-team robot cooperation. In: ; 2009: 62–67
8. Grocholsky B, Swaminathan R, Keller J, Kumar V, Pappas G. Information Driven Coordinated Air-Ground Proactive Sensing. In: ; 2005: 2211–2216
9. Blumenthal S, Hochgeschwender N, Prassler E, Voos H, Bruyninckx H. An approach for a distributed world model with QoS-based perception algorithm adaptation. In: ; 2015: 1806–1811
10. Cipolleschi R, Giusto M, Li AQ, Amigoni F. Semantically-informed coordinated multirobot exploration of relevant areas in search and rescue settings. In: ; 2013: 216–221
11. Dietrich A, Zug S, Mohammad S, Kaiser J. Distributed management and representation of data and context in robotic applications. In: ; 2014: 1133–1140
12. Tenorth M, Perzylo AC, Lafrenz R, Beetz M. Representation and exchange of knowledge about actions, objects, and environments in the robearth framework. *IEEE Transactions on Automation Science and Engineering* 2013; 10(3): 643–651.
13. De Nicola R, Di Stefano L, Inverso O. Toward formal models and languages for verifiable multi-robot systems. *Frontiers in Robotics and AI* 2018; 5: 94.
14. Schwertfeger J, Jenkins O. Multi-robot belief propagation for distributed robot allocation. In: ; 2007: 193–198
15. MacGregor R. Retrospective on Loom. tech. rep., University of Southern California, Information Sciences Institute; Los Angeles: 1999.
16. Zhang T, Ueno H. A Frame-Based Knowledge Model for Heterogeneous Multi-Robot System. *IEEE Transactions on Electronics, Information and Systems* 2005; 125(6): 846–855. doi: 10.1541/ieejeiss.125.846
17. Gayathri R, Uma V. Ontology based knowledge representation technique, domain modeling languages and planners for robotic path planning: A survey. *ICT Express* 2018; 4(2): 69–74.
18. Olszewska JI, Barreto M, Bermejo-Alonso J, et al. Ontology for autonomous robotics. In: ; 2017: 189-194.
19. Lim GH, Suh IH, Suh H. Ontology-Based Unified Robot Knowledge for Service Robots in Indoor Environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 2011; 41(3): 492-509. doi: 10.1109/TSMCA.2010.2076404
20. Bateman J, Beetz M, Beßler D, Bozcuoglu AK, Pomarlan M. Heterogeneous Ontologies and Hybrid Reasoning for Service Robotics: The EASE Framework. In: ROBOT '17. ; 2017; Sevilla, Spain.

21. Pirri F. The well-designed logical robot: Learning and experience from observations to the Situation Calculus. *Artificial Intelligence* 2011; 175(1): 378 - 415. John McCarthy's Legacydoi: <http://dx.doi.org/10.1016/j.artint.2010.04.016>
22. O'Brien P, Nicol R. FIPA: Towards a standard for intelligent agents. *BT Technical Journal* 1998; 16(3).
23. Genesereth M, Fikes R. Knowledge Interchange Format Version 3.0 Reference Manual. tech. rep., Stanford Logic Group Report; Stanford University: 1992.
24. Union S. The Knowledge Graph Platform for the Enterprise. <http://stardog.com>; 2018.
25. Padalia N. *Apache Cassandra Essentials*. Packt Publishing Ltd . 2015.
26. Tenorth M, Beetz M. KnowRob – Knowledge Processing for Autonomous Personal Robots. In: ; 2009: 4261–4266.
27. Beetz M, Tenorth M, Winkler J. Open-EASE – A Knowledge Processing Service for Robots and Robotics/AI Researchers. In: ; 2015.
28. Beksi WJ, Spruth J, Papanikolopoulos N. Core: A cloud-based object recognition engine for robotics. In: IEEE. ; 2015: 4512–4517.
29. Koubaa A, Bennaceur H, Chaari I, et al. Robot Path Planning Using Cloud Computing for Large Grid Maps. In: Springer. 2018 (pp. 103–126).
30. Mohanarajah G, Usenko V, Singh M, D'Andrea R, Waibel M. Cloud-Based Collaborative 3D Mapping in Real-Time With Low-Cost Robots. *Automation Science and Engineering, IEEE Transactions on* 2015; 12(2): 423–431. doi: 10.1109/TASE.2015.2408456
31. Riazuelo L, Tenorth M, Di Marco D, et al. RoboEarth Semantic Mapping: A Cloud Enabled Knowledge-Based Approach. *IEEE Transactions on Automation Science and Engineering* 2015; 12(2): 432–443. doi: 10.1109/TASE.2014.2377791
32. Tenorth M, Winkler J, Beßler D, Beetz M. Open-ease: a cloud-based knowledge service for autonomous learning. *KI-Künstliche Intelligenz* 2015; 29(4): 407–411.
33. Bozcuoğlu AK, Kazhoyan G, Furuta Y, et al. The exchange of knowledge using cloud robotics. *IEEE Robotics and Automation Letters* 2018; 3(2): 1072–1079.
34. Ferri G, Munafo A, Tesei A, LePage K. A market-based task allocation framework for autonomous underwater surveillance networks. In: IEEE. ; 2017: 1–10.
35. Silva dLCB, Bernardo RM, Oliveira dHA, Rosa PFF. Unmanned aircraft system coordination for persistent surveillance with different priorities. In: IEEE. ; 2017: 1153–1158.
36. Shi J, Yang Z, Zhu J. An auction-based rescue task allocation approach for heterogeneous multi-robot system. *Multimedia Tools and Applications* 2018: 1–10.
37. Freda L, Gianni M, Pirri F, et al. 3D multi-robot patrolling with a two-level coordination strategy. *Autonomous Robots* 2019; 43(7): 1747–1779.
38. Nikitenko A, Lavendelis E, Ekmanis M, Rumba R. Task Allocation Methods for Homogeneous Multi-Robot Systems: Feed Pushing Case Study. *Automatic Control and Computer Sciences* 2018; 52(5): 371–381.
39. Kumar B, Sharma L, Wu SL. Job Allocation schemes for Mobile Service Robots in Hospitals. In: IEEE. ; 2018: 1323–1326.
40. Spenko M, Buerger S, Iagnemma K. *The DARPA robotics challenge finals: humanoid robots to the rescue*. 121. Springer . 2018.
41. Shoaib M. Multi-robot Task Allocation using Agglomerative Clustering. 2016.
42. Janati F, Abdollahi F, Ghidary SS, Jannatifar M, Baltes J, Sadeghnejad S. Multi-robot task allocation using clustering method. In: Springer. 2017 (pp. 233–247).

43. Schwarzrock J, Zacarias I, Bazzan AL, Araujo Fernandes dRQ, Moreira LH, Freitas dEP. Solving task allocation problem in multi unmanned aerial vehicles systems using swarm intelligence. *Engineering Applications of Artificial Intelligence* 2018; 72: 10–20.
44. Lee DH. Resource-based task allocation for multi-robot systems. *Robotics and Autonomous Systems* 2018; 103: 151–161.
45. Msala Y, Hamlich M, Mouchtachi A. A new Robust Heterogeneous Multi-Robot Approach Based on Cloud for Task Allocation. In: IEEE. ; 2019: 1–4.
46. Skarzynski K, Stepniak M, Bartyna W, Ambroszkiewicz S. SO-MRS: a multi-robot system architecture based on the SOA paradigm and ontology. In: Springer. ; 2018: 330–342.
47. Sarkar C, Paul HS, Pal A. A scalable multi-robot task allocation algorithm. In: IEEE. ; 2018: 1–9.
48. Mario Gianni MA, Garcia R, Cantucci F, Nardi F, Pushparaj K, Pirri F. Tradr-DR 4.2: Models for roles and decisions in collaboration. http://www.tradr-project.eu/wp-content/uploads/dr.4.2.main_public.pdf; 2016.
49. Gianni M, Ferri F, Pirri F. ARE: Augmented reality environment for mobile robots. In: Springer. ; 2013: 470–483.
50. Quigley M, Conley K, Gerkey B, et al. ROS: an open-source Robot Operating System. In: . 3. Kobe, Japan. ; 2009: 5.
51. Dukes K. Semantic annotation of robotic spatial commands. In: ; 2013.
52. Kuhlmann G, Stone P, Mooney RJ, Shavlik JW. Guiding a Reinforcement Learner with Natural Language Advice: Initial Results in RoboCup Soccer. In: ; 2004.
53. Marneffe M, Maccartney B, Manning C. Generating Typed Dependency Parses from Phrase Structure Parses. In: ; 2006.
54. Tellex S, Kollar T, Dickerson S, et al. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In: ; 2011: 1507–1514.
55. Fillmore CJ. The Case for Case. In: Bach E, Harms RT., eds. *Universals in Linguistic Theory* New York: Holt, Rinehart and Winston. 1968 (pp. 0–88).
56. Rennie JDM, Shih L, Teevan J, Karger DR. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In: ; 2003: 616–623.
57. Sebastiani F. Machine Learning in Automated Text Categorization. *ACM Comput. Surv.* 2002; 34(1): 1–47. doi: 10.1145/505282.505283
58. Teevan J, Karger DR. Empirical Development of an Exponential Probabilistic Model for Text Retrieval: Using Textual Analysis to Build a Better Model. In: ACM; 2003; New York, NY, USA: 18–25
59. Acar E, Yener B. Unsupervised Multiway Data Analysis: A Literature Survey. *IEEE Transactions on Knowledge and Data Engineering* 2009; 21(1): 6–20. doi: 10.1109/TKDE.2008.112
60. Liu Y, Shang F. An Efficient Matrix Factorization Method for Tensor Completion. *IEEE Signal Processing Letters* 2013; 20(4): 307–310. doi: 10.1109/LSP.2013.2245416
61. Chi EC, Kolda TG. On Tensors, Sparsity, and Nonnegative Factorizations. *SIAM Journal on Matrix Analysis and Applications* 2012; 33(4): 1272–1299. doi: 10.1137/110859063
62. Farias R, Comon P, Redon R. Data mining by nonnegative tensor approximation. In: ; 2014: 1–6
63. Symeonidis P, Nanopoulos A, Manolopoulos Y. A Unified Framework for Providing Recommendations in Social Tagging Systems Based on Ternary Semantic Analysis. *IEEE Transactions on Knowledge and Data Engineering* 2010; 22(2): 179–192. doi: 10.1109/TKDE.2009.85
64. Kolda TG, Bader BW. Tensor decompositions and applications. *SIAM review* 2009; 51(3): 455–500.

65. Phan A, Cichocki A. Multi-way nonnegative tensor factorization using fast hierarchical alternating least squares algorithm HALS. In: ; 2008.
66. Nathan Srebro Nati NS, Jaakkola T. Weighted Low-Rank Approximations. In: AAAI Press; 2003: 720–727.
67. Adomavicius G, Tuzhilin A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. on Knowl. and Data Eng.* 2005; 17(6): 734–749. doi: 10.1109/TKDE.2005.99
68. Coumans E. Bullet Physics SDK. <http://bulletphysics.org/wordpress/>; 2015.
69. Robotics C. Virtual Robot Experimentation Platform V-REP. <http://www.coppeliarobotics.com/index.html>; 2015.
70. Papadakis P, Gianni M, Pizzoli M, Pirri F. Constraint-free Topological Mapping and Path Planning by Maxima Detection of the Kernel Spatial Clearance Density. In: ; 2012: 71–79
71. Kruijff G, Kruijff-Korbayová I, Keshavdas S, et al. Designing, developing, and deploying systems to support human–robot teams in disaster response. *Advanced Robotics* 2014; 28(23): 1547–1570. doi: 10.1080/01691864.2014.985335
72. Menna M, Gianni M, Ferri F, Pirri F. Real-time autonomous 3D navigation for tracked vehicles in rescue environments. In: ; 2014: 696–702.
73. Wanderlingh F. VRep-Ros Quadcopter Swarm. <https://github.com/merosss/VRepRosQuadSwarm>; 2014.
74. Gianni M, Kruijff GJM, Pirri F. A stimulus-response framework for robot control. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2015; 4(4): 1–41.
75. Gianni M, Gonnelli G, Sinha A, Menna M, Pirri F. An Augmented Reality approach for trajectory planning and control of tracked vehicles in rescue environments. In: ; 2013: 1–6
76. Ferri F, Gianni M, Menna M, Pirri F. Point Cloud Segmentation and 3D Path Planning for Tracked Vehicles in Cluttered and Dynamic Environments. In: ; 2014.

How to cite this article: M. Gianni, and M.S. Uddin (2020), Role and task allocation framework for Multi-Robot Collaboration with latent knowledge estimation, *Engineering Reports*, 2020;00:1–6.

APPENDIX

A APPENDICES

A.1 The formal system underlying the Multi-Robot collaboration Treebank

The formal system FS used for annotating the statements in the Multi-robot collaboration treebank, introduced in Section 3 is composed of

- a finite set NT of non-terminal symbols;
- a finite set T of terminal symbols;
- a finite set P of production rules of the form $\gamma \rightarrow \delta$, with $\gamma \in NT$ and $\delta \in NT \cup T$;
- the symbol $S \in NT$ used to represent the whole statement;

Both Table A1 and Table A2 report the description of each non-terminal symbol of NT . Given the definition of all possible non-terminal symbols of the formal system FS , the production rules of P have been constructed as follows. A statement (S) can consist of either an atomic process phrase or of a temporal task phrase, or of a shifting-inhibition phrase or of an atomic process execution state phrase. An atomic process phrase is composed of a robot phrase (RP) and a process phrase (PP). A robot

TABLE A1 Description of the set *NT* of non-terminal symbols

Symbol	Description
APP	atomic process phrase
TTP	temporal task phrase
RP	robot phrase
PP	process phrase
RobotNoun	robot noun
RobotID	robot identifier
Process	process
TP	target phrase
OP	object phrase
SDSpaP	source-destination spatial phrase
DP	destination phrase
ObjNoun	object noun
ObjectID	object identifier
SourceParticle	source particle
DestinationParticle	destination particle
SpecP	specification phrase
SpaP	spatial phrase
IntP	interaction phrase
SpecParticle	specification particle
RelP	relational phrase
SimSpaRel	simple spatial relation
RecP	receiver phrase
IntParticle	interaction particle
TCP	temporal constraint phrase

phrase is composed of a robot noun (RobotNoun) followed by a robot identifier (RobotID). A process phrase is composed of a process (Process) followed by a target phrase (TP). A target phrase can simply comprise either an object phrase (OP) or a source-destination spatial phrase (SDSpaP). It can also be composed of an object phrase (OP) followed by a destination phrase (DP).

An object phrase can be specified by either a robot phrase or by an object noun (ObjNoun) followed by an object identifier (ObjectID). A source-destination spatial phrase comprises a source particle (SourceParticle) followed by an object phrase, followed by a destination particle (DestinationParticle), followed by another object phrase.

A destination phrase can be composed of either a specification phrase (SpecP) or of a spatial phrase (SpaP) or of an interaction phrase (IntP). A specification phrase is composed of a specification particle (SpecParticle) followed by an object phrase.

TABLE A2 Description of the set *NT* of non-terminal symbols

Symbol	Description
TCO	temporal constrain operator
SIP	shifting-inhibition phrase
APESP	atomic process execution state phrase
SwitchP	switching phrase
InhibitP	inhibition phrase
SwitchProcess	switching process
SwitchRelP	switching relation process
CPP	current process phase
NPP	new process phrase
CSP	causal stimulus phrase
PartPP	participle process phrase
PartProcess	participle process
CSParticle	causal stimulus particle
StimulusP	stimulus phrase
SNoun	stimulus noun
InhibitProcess	inhibition process
InhibitRelP	inhibition relation process
InhibitConjP	inhibition conjunction phrase
Conj	conjunction
SSEP	success state execution phrase
FSEP	failure state execution phrase
SuccessProcess	success process
FailProcess	failure process

According to this initial specification, the following production rules are included into the formal system

$$S \rightarrow APP \mid TTP \mid SIP \mid APESP \quad (A1)$$

$$APP \rightarrow RP \ PP \quad (A2)$$

$$RP \rightarrow RobotNoun \ RobotID \quad (A3)$$

$$PP \rightarrow Process \ TP \quad (A4)$$

$$TP \rightarrow OP \quad (A5)$$

$$TP \rightarrow SDSpaP \quad (A6)$$

$$TP \rightarrow OP \ DP \quad (A7)$$

$$OP \rightarrow RP \quad (A8)$$

$$OP \rightarrow ObjNoun \ ObjectID \quad (A9)$$

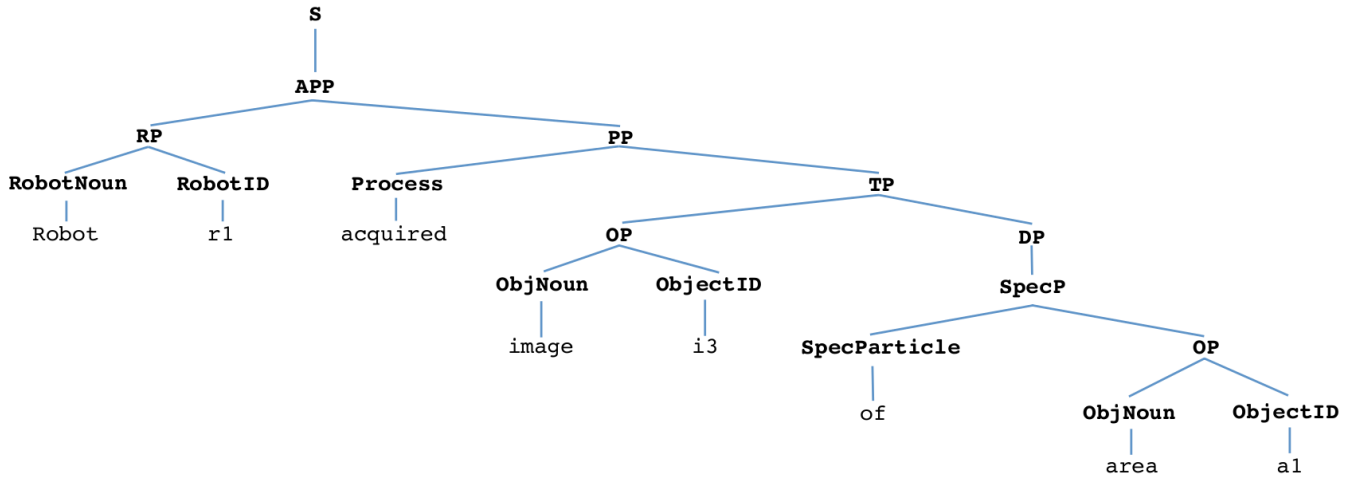
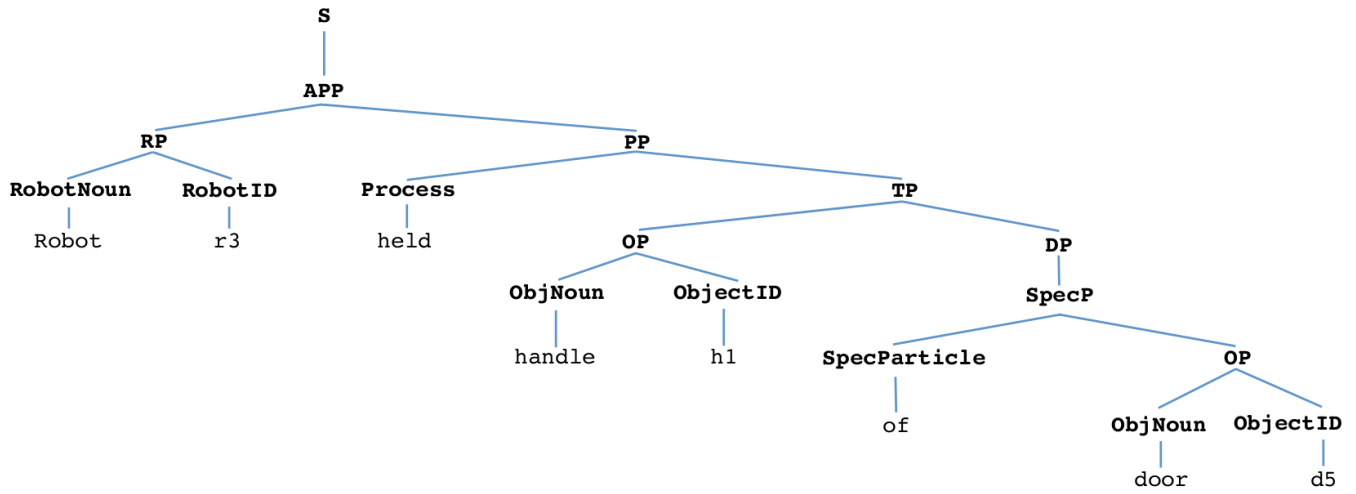
$$SDSpaP \rightarrow SourceParticle \ OP \ DestinationParticle \ OP \quad (A10)$$

$$DP \rightarrow SpecP \mid SpaP \mid IntP \quad (A11)$$

$$SpecP \rightarrow SpecParticle \ OP \quad (A12)$$

484 Given the set of production rules in (A1)-(A12), we can build syntactic trees of statements admitted in the treebank, as those
 485 illustrated in Figure A1.

A spatial phrase can be composed of either a relational phrase (RelP) or of relational phrase followed by a spatial phrase or of a source-destination spatial phrase. A relational phrase is composed of a simple spatial relation (SimSpaRel) followed by an

(a) S : Robot r1 acquired image i3 of area a1(b) S : Robot r3 held handle h1 of door d5**FIGURE A1** Syntactic trees of statements on the basis of production rules in (A1)-(A12).

object phrase. The corresponding production rules are as follows

$$\text{SpaP} \rightarrow \text{RelP} \quad (\text{A13})$$

$$\text{SpaP} \rightarrow \text{RelP SpaP} \quad (\text{A14})$$

$$\text{RelP} \rightarrow \text{SimSpaRel OP} \quad (\text{A15})$$

$$\text{SpaP} \rightarrow \text{SDSpaP} \quad (\text{A16})$$

Rules (A13)-(A16) have been suitably defined for aligning statements describing both robot manipulation tasks and robot motions from a starting to an ending position. Syntactic trees of such statements are illustrated in Figure 2(a), Section 3, and in Figure A2.

An interaction phrase is composed of either a receiver phrase (RecP) or of a receiver phrase preceded by a specification phrase. A receiver phrase is defined by an interaction particle (IntParticle) and a robot phrase. Therefore, the following rules are

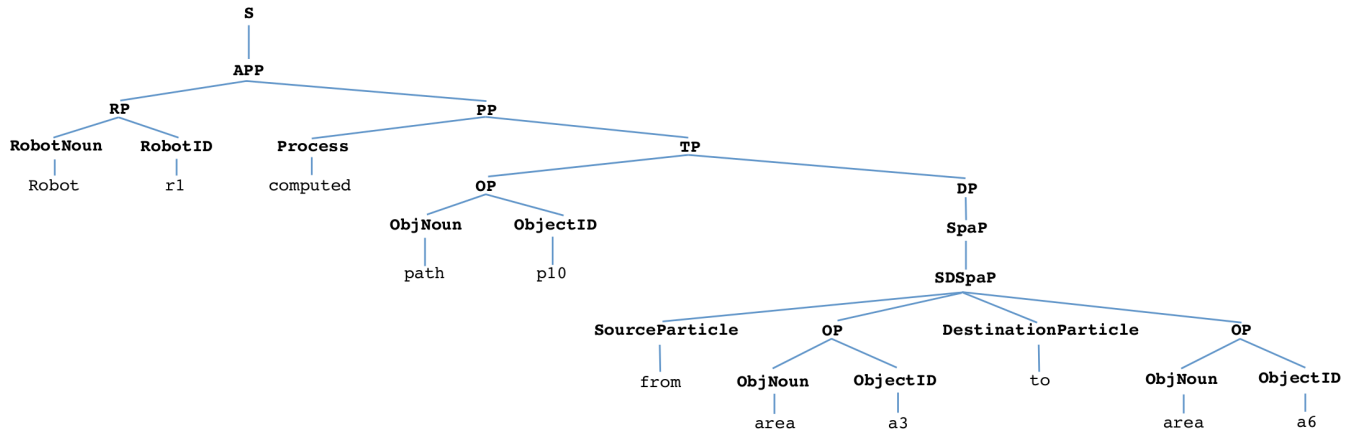


FIGURE A2 Syntactic tree of the statement S : Robot r1 computed path p10 from area a3 to area a6, aligned with the production rules in (A13)-(A16).

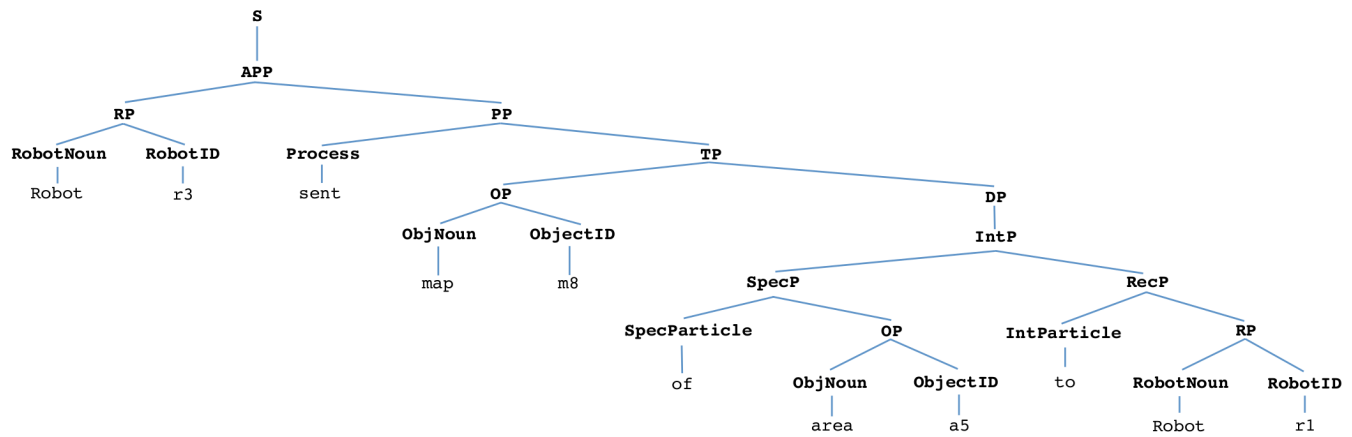


FIGURE A3 Syntactic tree of statement S : Robot r3 sent map m8 of area a5 to robot r1, on the basis of production rules in (A17)-(A19).

added into the system

$$\text{IntP} \rightarrow \text{RecP} \quad (\text{A17})$$

$$\text{IntP} \rightarrow \text{SpecP RecP} \quad (\text{A18})$$

$$\text{RecP} \rightarrow \text{IntParticle RP} \quad (\text{A19})$$

Rules (A17)-(A19) model statements specifying a preliminary form of collaboration such as information dispatching with or without prepositions. An example of these forms of collaboration is shown in Figure A3.

Note that, rules (A13)-(A16) can be also recursively expanded to align statements specifying forms of coordination among robots, in particular, in motion tasks, as illustrated in Figure A4.

Finally, a temporal task phrase (TTP) is composed of either an atomic process phrase or of an atomic process phrase followed by a temporal constraint phrase (TCP). A temporal constraint phrase consists of a temporal constraint operator (TCO) and a temporal task phrase. According to this definition, the following set of production rules is added to the system

$$\text{TTP} \rightarrow \text{APP} \quad (\text{A20})$$

$$\text{TTP} \rightarrow \text{APP TCP} \quad (\text{A21})$$

$$\text{TCP} \rightarrow \text{TCO TTP} \quad (\text{A22})$$

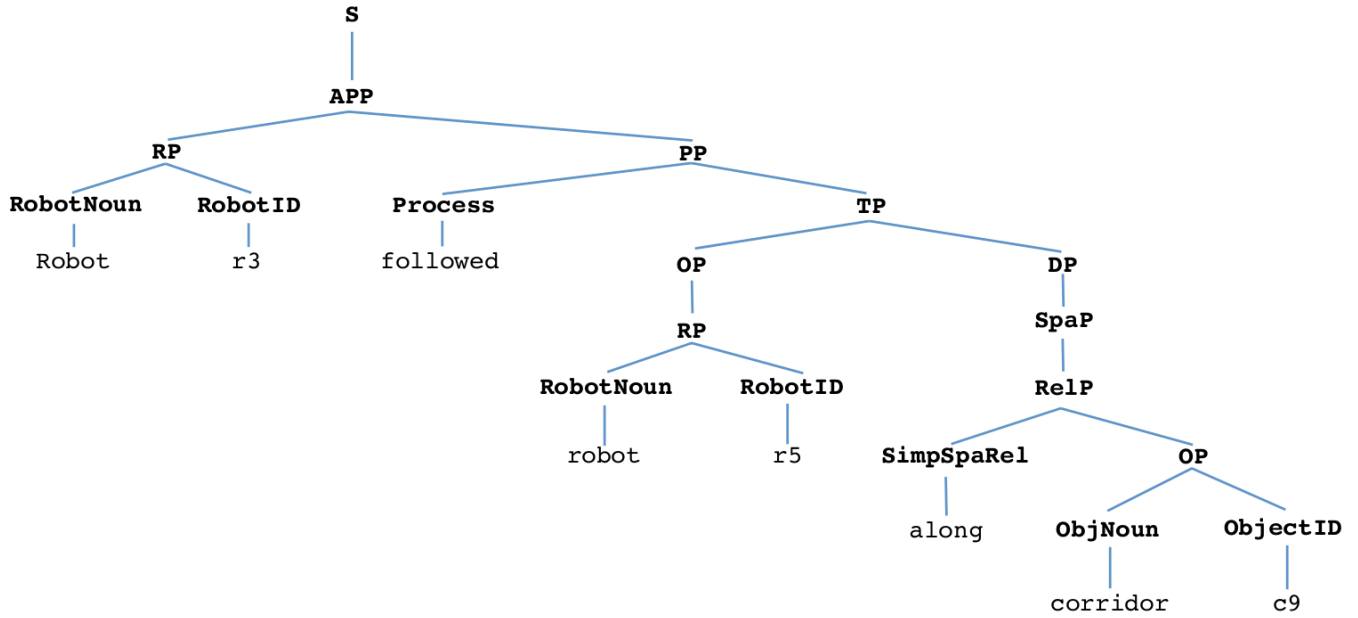


FIGURE A4 Syntactic trees of statement S : Robot r3 followed robot r5 along corridor c9, on the basis of production rules in (A13)-(A16).

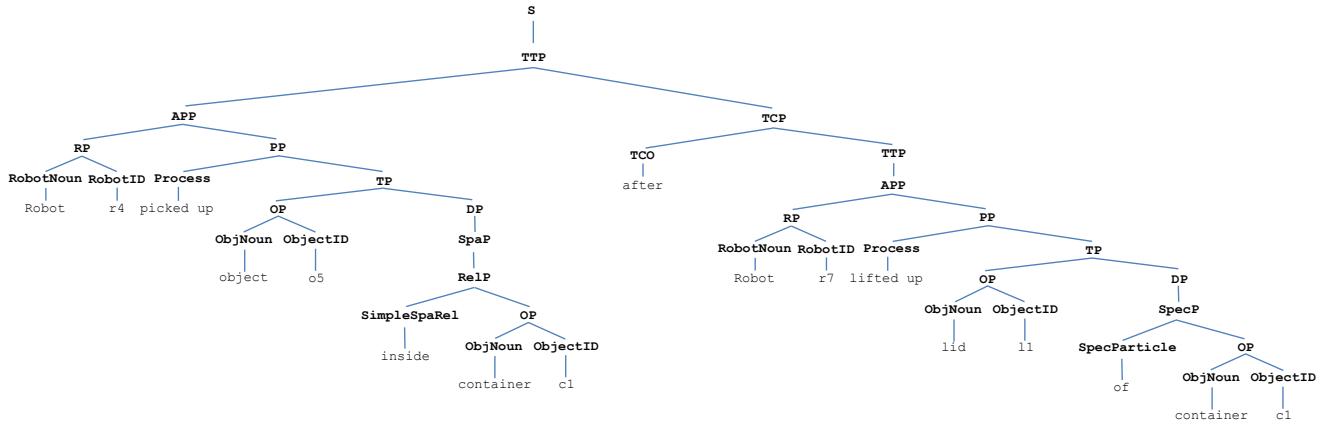


FIGURE A5 Syntactic tree of the statement S : Robot r3 requested map m22 of area a7 to robot r2 before robot r3 computed path p61 from area a11 to area a7, aligned according to production rules in (A20)-(A22).

Recursive rules (A20)-(A22) align statements describing a wide class of complex tasks, performed by a group of robots under collaboration. Each collaborative task is a composition of individual robot tasks linked by a temporal constraint. Examples of syntactic trees of statements expressing complex collaborative tasks are shown in Figure 3, Section 3, and in Figure A5.

A shifting-inhibition phrase (SIP) is composed of either a robot phrase followed by a switching phrase (SwitchP) or of a robot phrase followed by an inhibition phrase (InhibitP). A switching phrase is composed of a switching process (SwitchProcess) followed by a switching relation process (SwitchRelP). A switching relation process is composed of a current process phase (CPP) followed by a new process phase (NPP) followed by a causal stimulus phrase (CSP). A current process phase is composed of a source particle followed by a participle process phase (PartPP). A participle process phase is specified by a participle process (PartProcess) followed by a target phrase. A new process phase is composed of a destination particle followed by a participle process phase. Finally, a causal stimulus phrase is defined by a causal stimulus particle (CSParticle) followed by a stimulus phrase (StimulusP). A stimulus phrase can be composed of either a stimulus noun (SNoun) or of a stimulus

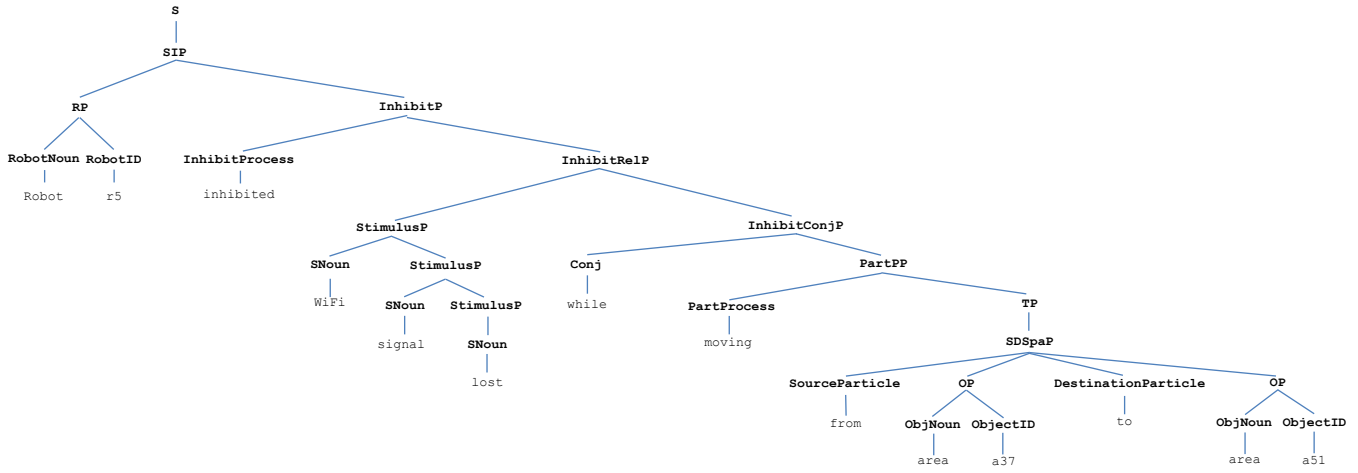


FIGURE A6 Syntactic tree of the statement S : Robot r5 inhibited wifi connection lost while moving from area a37 to area a51 according to production rules in (A23)-(A35).

noun followed by a stimulus phrase. An inhibition phrase is composed of an inhibition process (*InhibitProcess*) followed by an inhibition relation process (*InhibitRelP*). An inhibition relation process is composed of a stimulus phrase followed by an inhibition conjunction phrase (*InhibitConjP*). This is composed of a conjunction (*Conj*) followed by a participle process phase. On the basis of the above specification the formal system includes the following production rules

$$SIP \rightarrow RP \text{ SwitchP} \quad (A23)$$

$$SIP \rightarrow RP \text{ InhibitP} \quad (A24)$$

$$\text{SwitchP} \rightarrow \text{SwitchProcess} \text{ SwitchRelP} \quad (A25)$$

$$\text{SwitchRelP} \rightarrow \text{CPP} \text{ NPP} \text{ CSP} \quad (A26)$$

$$\text{CPP} \rightarrow \text{SourceParticle} \text{ PartPP} \quad (A27)$$

$$\text{PartPP} \rightarrow \text{PartProcess} \text{ TP} \quad (A28)$$

$$\text{NPP} \rightarrow \text{DestinationParticle} \text{ PartPP} \quad (A29)$$

$$\text{CSP} \rightarrow \text{CSParticle} \text{ StimulusP} \quad (A30)$$

$$\text{StimulusP} \rightarrow \text{SNoun} \quad (A31)$$

$$\text{StimulusP} \rightarrow \text{SNoun} \text{ StimulusP} \quad (A32)$$

$$\text{InhibitP} \rightarrow \text{InhibitProcess} \text{ InhibitRelP} \quad (A33)$$

$$\text{InhibitRelP} \rightarrow \text{StimulusP} \text{ InhibitConjP} \quad (A34)$$

$$\text{InhibitConjP} \rightarrow \text{Conj} \text{ PartPP} \quad (A35)$$

496 An example of syntactic trees of statements for task switching behaviors is illustrated in Figure 4(a), Section 3, and in Figure A6.

An atomic process execution state phrase (APESP) is composed of either a robot phase followed by a success state execution phase (SSEP) or of a robot phase followed by a failure state execution phase (FSEP). A success state execution phase is composed of a success process (*SuccProcess*) followed by a participle process phase whilst a failure state execution phase is specified by a failure process (*FailProcess*) followed by a participle process. The set of production rules annotating execution status statements is defined as follows

$$\text{APESP} \rightarrow \text{RP} \text{ SSEP} \quad (A36)$$

$$\text{APESP} \rightarrow \text{RP} \text{ FSEP} \quad (A37)$$

$$\text{SSEP} \rightarrow \text{SuccProcess} \text{ PartPP} \quad (A38)$$

$$\text{FSEP} \rightarrow \text{FailProcess} \text{ PartPP} \quad (A39)$$

497 Examples of syntactic trees of statements produced by rules (A36)-(A39) are shown in Figure 4(b), Section 3, and in Figure A7.

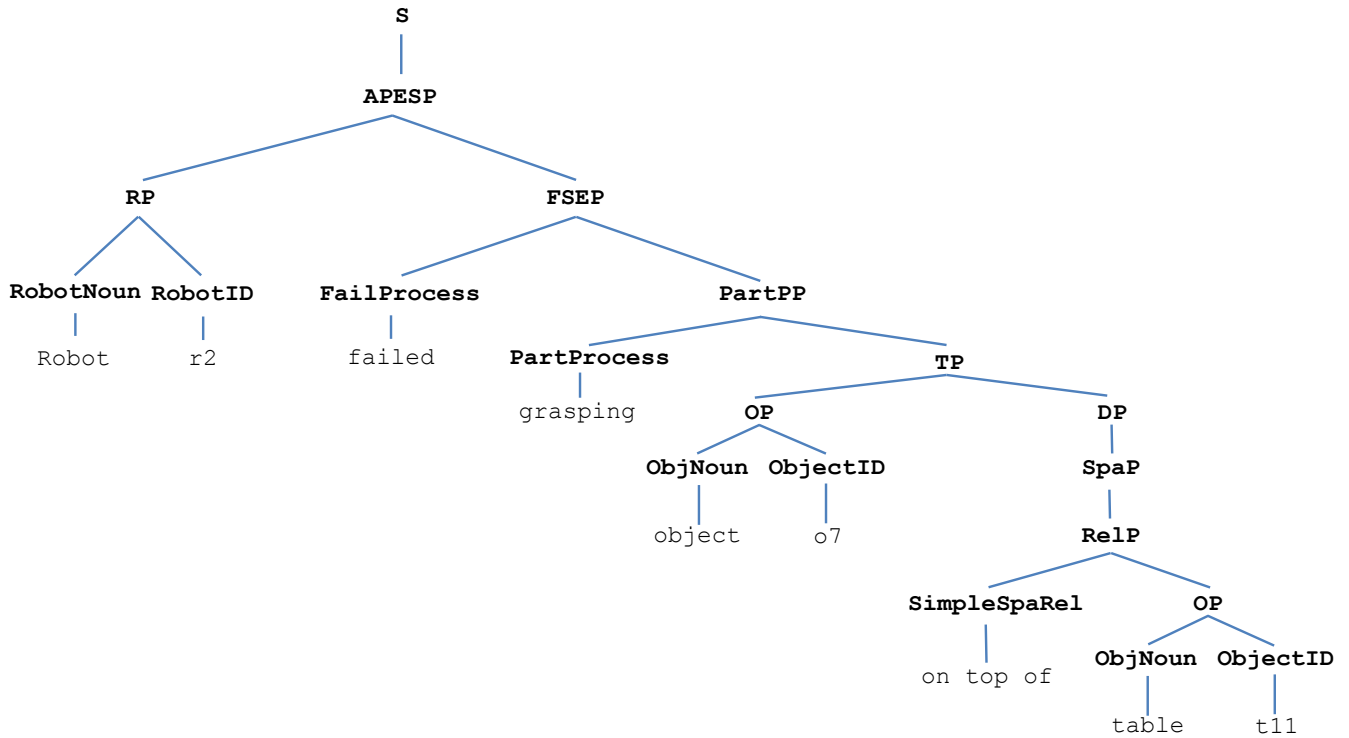


FIGURE A7 Syntactic tree of the statement S : Robot r2 failed grasping object o7 on table t11 according to production rules in (A36)-(A39)

To conclude, the formal system is completed with a set of production rules generating the leafs of the syntactic trees. An excerpt of these rules is described in the following

$$\text{ObjNoun} \rightarrow \text{image} \mid \text{area} \mid \text{door} \mid \text{handle} \mid \text{table} \mid \text{path} \mid \text{map} \quad (\text{A40})$$

$$\text{Process} \rightarrow \text{acquired} \mid \text{requested} \mid \text{picked up} \mid \text{computed} \mid \text{sent} \quad (\text{A41})$$

$$\text{SimpSpaRel} \rightarrow \text{on top of} \mid \text{behind} \mid \text{along} \mid \text{in front of} \mid \text{inside} \quad (\text{A42})$$

$$\text{TCO} \rightarrow \text{before} \mid \text{during} \mid \text{equal} \mid \text{start} \mid \text{overlap} \mid \text{meet} \quad (\text{A43})$$

$$\text{SwitchProcess} \rightarrow \text{switched} \quad (\text{A44})$$

$$\text{InhibitProcess} \rightarrow \text{inhibited} \quad (\text{A45})$$

$$\text{PartProcess} \rightarrow \text{exploring} \mid \text{moving} \mid \text{climbing} \mid \text{grasping} \quad (\text{A46})$$

$$\text{SuccProcess} \rightarrow \text{completed} \quad (\text{A47})$$

$$\text{FailProcess} \rightarrow \text{failed} \quad (\text{A48})$$

498 The head of the production rules in (A40)-(A48) are in the set T of terminal symbols of FS .

AUTHOR BIOGRAPHY



Mario Gianni is a lecturer in Computer Science and Robotics within the School of Engineering, Computing and Mathematics (SECaM), at the University of Plymouth, UK. From April 2015 to September 2017, he was an assistant professor at DIAG, Sapienza University of Rome and a member of the ALCOR Lab. From 2010 to 2013 he worked for the EU-FP7-ICT-247870 project NIFTI, aimed to provide natural collaboration between humans and teams of robots in urban search and rescue scenarios. At NIFTI, he designed an all-terrain 3D autonomous navigation framework for an articulated tracked robot. In 2012, He deployed the robotic system to assess damage to historical buildings and cultural artifacts, in Mirandola, Italy, hit by an earthquake. From 2014 until 2017 he worked for the EU-FP7-ICT-609763 project TRADR, developing new technology for long-term human-robot teaming for robot assisted disaster response. Where, he developed a framework for autonomous exploration, patrolling and coverage of rescue scenarios for a team of heterogeneous robots. In 2016, he deployed a team of ground and aerial robots to collect data for 3D textured models of the interior and exterior of two churches located in Amatrice, Italy, which were badly damaged by an earthquake. Recently, He was awarded by the EPSRC National Centre for Nuclear Robotics (NCNR) to undertake research activities in robot-teaming for collaborative terrain traversability assessment of nuclear environments. He established the Autonomous Mobile Robotics in Extreme Environments (AMREE) Lab in the University of Plymouth, UK. The AMREE is a new and emerging laboratory of SECaM pursuing cutting edge research activities in the field of Computer Vision, Perception, Artificial Intelligence and Robot Learning applied to autonomous robot-assisted response in hazards, prevention and safety-critical operations. Currently, He is collaborating with a company in Dorset to automate remotely operated vehicles for Explosive Ordinance Disposal domain applications.



Mohammad Salah Uddin is an assistant professor in the department of computer science and engineering at East West University, Bangladesh, where he has been since 2018. He received his doctoral degree in engineering in computer science (*ROBOTICS*) from the University of Rome (Sapienza) in 2017. His research interest span both robotics and sensory systems. Most of his work mainly focused on intelligent systems or robotics systems for solving social or economical problems. In 2018, He delivered a speech in Maker Faire Shenzhen 2018, as a forum speaker. He is a member of *IEEE* robotics and automation society as well as hold a *IEEE* professional membership. He has also some others national and international membership.