Robotic architecture design to map an action language to a low level motion commands for skillful manipulation

Diseño de una arquitectura robótica para mapear un lenguaje de acción a comandos de movimiento de bajo nivel para manipulación hábil

Félix David Suárez Bonilla*. Federico Ruiz Ugalde.**

* Universidad Técnica Nacional (Tel: +506 8783 4957; e-mail: fsuarez@utn.ac.cr).
** Universidad de Costa Rica (Tel: +506 8309 6020; e-mail:fruiz@eie.ucr.ac.cr).

Resumen: Este artículo ofrece una visión general de una arquitectura robótica destinada a la manipulación hábil. Este diseño está destinado a cerrar la brecha entre la capa de alto nivel (capa de razonamiento y planificación) y el sistema de modelo de objetos (capa de control físico). Esta arquitectura propone una capa de interfaz que permite, de manera significativa, conectar tareas básicas con el controlador. En este artículo, discutimos cómo este sistema puede resolver tareas complejas específicas; analizamos el diseño de la unidad de accesibilidad y presentamos una visión general de los desafíos futuros en la implementación de todo el sistema.

Palabras clave: arquitectura robótica; manipulación hábil; capa de interfaz; tareas de complejas; modelos de asequibilidad; capa de control física; matriz de asequibilidad

Abstract: This paper gives an overview of a robotic architecture meant for skillful manipulation. This design is meant to close the gap between the high level layer (reasoning and planing layer) and the object model system (physical control layer). This architecture proposes an interface layer that allows, in a meaningful way, to connect atomic tasks with controller inputs. In this paper, we discuss how specific complex tasks can be resolved by this system; we analyze the affordance unit design and, we overview the future challenges in the implemenation of the whole system.

Keywords: robotic architecture; skillful manipulation; interface layer; complex tasks; affordance model; physical control layer; affordance matrix

1. INTRODUCTION

The future incorporation of robots in human activities will require them to comprehend commands expressed in natural language, recognize the environment and execute the required manipulation movements to acomplish the respective tasks [1]. In order to achieve this goal, we need robots capable of understanding high level tasks, so they can split them in compact and minimum tasks that can be somehow understood by the control layer in the robotic software arquitecture [2] [3].

Any complex task involves object physical manipulation [4] [5]. For example, if a human tells a robot to prepare a sandwhich by saying “Robot, make me a sandwich”, the robot should be able to split this complex task in many minimum tasks (also called atomic tasks); which are meant to manipulate the ingredients, the appliances, the utensils and the cookware. The software and the hardware must be prepared to accomplish these tasks.

In this paper, we start from a robotic architecture with three layers: a high level layer meant to process complex tasks, the interface layer meant to translate atomic tasks into inputs to physical object controllers and a low level layer meant to perform physical object control. The low level layer contains an object model system which consists in mathematical models of the objects so we can predict and control their behavior. This is a special case of the classical three tier (3T) architecture.

The object model must be able to perform a variety of actions, which are specified by the atomic tasks coming out from the high level layer. One action that has been implemented as an object model is *slide*, where a rigid planar body slides on a horizontal planar surface. This single action requires to find a solution for three double-integral equations over the surface of the object for every possible center of rotation [8]. The construction of object models is not a trivial problem.

The idea of having an object model that contains specific mathematical models for the actions is inspired in the analysis of the ventral premotor cortex in humans and monkeys. These studies showed that most of the F5 neurons in the monkey brain perform specific actions, rather than single movements that form them. F5 neurons are even divided into several actions classes like "grasping", "holding" or "tearing" neurons, in total parallelism with an object model system [9].

This research is meant to present a robotic architecture meant to close the gap between the high-level layer, that processes complex and long-term tasks; and the low-level layer. The translation between both layers is performed by the interface layer. We think that mapping an action description language to the input of object controllers can address this problem.

Some of the specific contributions from this research are:

- A cognitive architecture design that can allow the robots to integrate a high level task to a low level control commands.
- A system to obtain an affordance model from a training corpus of atomic tasks, which is used as part of the interface layer.
- An affordance matrix which indicates the conditional probability, but we think it can also be used to model, represent and compare different environments.

2. ROBOTIC ARCHITECTURE

The robotic architecture has been divided in three layers: a high level layer, an interface layer, and a low level layer (See figure 1). The high level layer processes “complex tasks” and splits it into more simple tasks which can be executed by the low level layer, these simple tasks are called “atomic tasks”. The low level layer contains the controllers which are in charge of controlling the objects [8] [10] [11] [12].

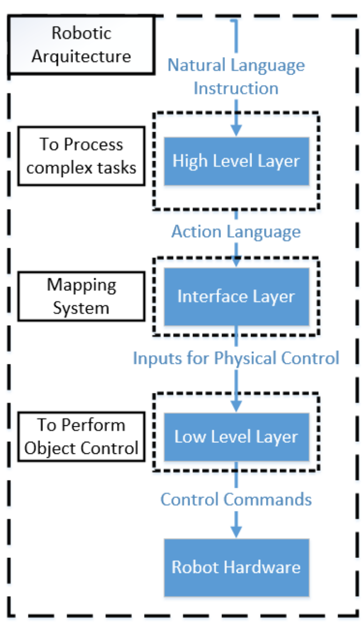

Figure 1. Robotic architecture which consist in three layers: a high level layer meant to process complex tasks, the interface layer meant to translate atomic tasks into inputs to physical object controllers and a low level layer meant to perform physical object control. [7].

The high level layer performs planning and reasoning; and the low level layer is in charge of prediction, perception and physical object control. We can find very different challenges in both layers; the high level layer is more related to: artificial intelligence, machine learning, natural processing language and knowledge representation. In the other hand, the low level layer is more related to: mechanics, automatic control, mathematical modeling and kinematics.

The interface layer in the middle maps the “atomic task” with the respective controller inputs in the low level layer. For example, taking the atomic task: “Push the glass of milk away”; the verb “push” would translate to an input domain region that uses a push controller, “glass of milk” specifies the object to be controlled, and “away” gives the desired outcome of the object [13] [14].

The high level layer performs the segmentation of “complex tasks” into more simple actions (atomic tasks). This layer has been addressed by Ruiken, Muller, Gorges and Ternorth [15] [16] [17] [18]. The implementations of this high level layer starts with observations of human activities and performs optimizations to find the atomic tasks which minimize energy and effort [12].

4. RELATED WORK

This paper presents a three-layer architecture to translate higher layer natural language commands into lower layer control parameters. The translation is quintessentially performed by five components of the interface layer. To this effect, the intermediate layer translates single sentences in structured Spanish to lower level parameters. As such, it is an extension of [8] with a dedicated interface layer. The structure of the interface layer system is fairly complex, taking into account its capabilities, so in this paper we will mostly focus on the design and the experiments performed over the affordance unit.

The notion of affordance has been discussed in many settings and different formalizations have been proposed. The concept was firstly defined as “what an object offers” by Gibson [38]. In robotics, affordances have been used with different purposes: tool learning usage [39], object manipulation [40], inferring surroundings [41], and grasping learning [42]. In this research we aimed to use affordances to calculate the feasibility to execute an instruction, with the final purpose of traslating an action language to low-level representation.

5. INTERFACE LAYER DESIGN

The interface layer design is shown in figure 2. It contains five modules: Affordance Unit (AU), Mapping Unit (MU), Determination of Reference (DR), Object Properties (OP) and Feasibility Analysis (FA).

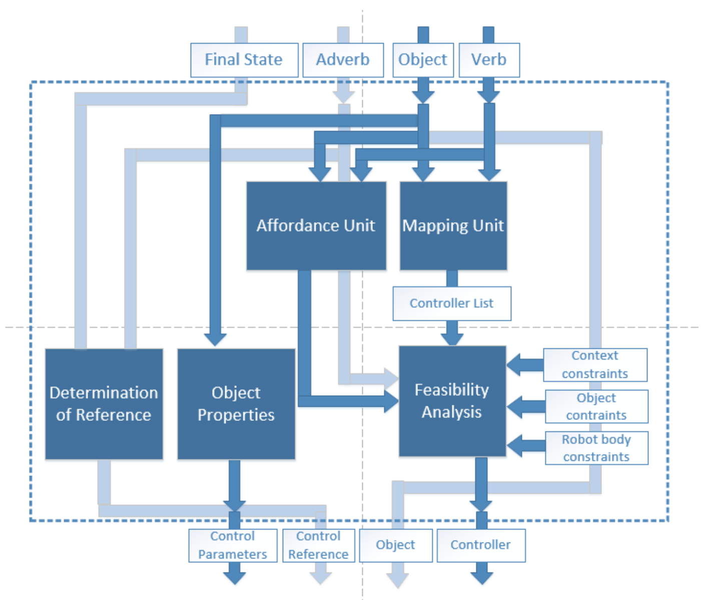

Figure 2. Interface layer design which contains five modules: Affordance Unit (AU), Mapping Unit (MU), Determination of Reference (DR), Object Properties (OP) and Feasibility Analysis (FA).

The AU takes the object and the verb to determine if it complies with the affordability. The MU, based on the object and the verb, determines the list of controllers that can used to execute the action. The DR module, based on the final state and the adverb, determines the reference value for the object controller. The OP is meant to provide the object properties that are needed to execute the low level control. The FA module, based on a list of contraints and weights, will determine which controllers are more suitable to execute the action.

5.1 Affordance Unit

The AU checks the verb and the object, and declares if both entities satisfy the affordability. For example, if the verb is "turn on" and the object is "oven", there is affordability, because the oven can be turned on. On the other hand, if the verb is "turn on" and the object is "glass", there is no affordability because the object "glass" cannot be "turned on".

To implement this module, we can use hardcoded tables that includes valid combinations of verbs and objects. The output of this module is "true" or "false". The output "true" means that the object and the verb comply with the affordability; and the output "false" means that both elements do not comply with affordability. This unit can be also addresed by calculating the conditional probability of having a given object after a manipulation action, based on a maximum likelihood estimation for the bigram Verb + Object:

$$P(object|verb) = \frac{count(verb\ ,object)}{count(verb)} \tag{1}$$

To do this, it is required to generate a probability affordance model based on a training dataset. This model can be obtained from a corpus of instructions, apply post tagging to obtain the verb and the object, lemmatize both elements, and then register the frecuencies in a file or database. In this research, we followed this approach instead of hardcoded tables.

5.2 Mapping Unit

The MU processes the object and verb contained in the atomic task to generate a list of controllers able to perform the actions with the specified object. The implementation of this module faces some challenges, such as the fact that very few object models have been developed as for now. To conduct the experiments, we will suppose that there is complete library of object models that can be used.

5.3 Determination of Reference

The DR module determines the reference value to be used by the physical object controllers. The value is calculated based on two elements: the adverb and the final state. For example: in the atomic task "move the glass near the oven", the adverb is "near" and the final state is "oven". Both elements are used to calculate where to place the object.

There are different types of adverbs expressing different meanings, such as: adverbs of time, adverbs of place and adverbs of manner. Translating the adverbs into values is a difficult problem because adverbs can be very subjective and very tied to the context.

5.4 Object Properties

The OP module constains the object properties required by the object models to perform the prediction and control. For example, a "glass" can have a lot of properties such as: weight, coefficient of friction, shape and hardness. These values are required by the low level layer to perform the physical object control.

5.5 Feasibility Analysis

This module calculate the best controller, based on: the affordability value, the list of controllers and the constraints. These limitation can be related to the context, the robot body and the objects. We consider this unit should be able compute a cost function for every single controller, given for the following equation:

$$f(a_v, w_i, ct_0, ct_1, ct_2, \dots) = a_v \cdot \left(\sum_{i=0}^{N} ct_i\right) + w_i \tag{2}$$

where the a_v is affordance value given by equation (1) or any estimation of this value, and ct_i are numerical values which might refer to three types of constraints: context, object, and robot body.

The cost function propossed in equation (2) will be used as a start point in the analysis of the interaction between the MU, the AU, and the FA. For now this function is mostly exploratory, but it is considered a decent start. The idea of having the affordance value in the equation is a key item, since the affordability determines the actual feasibility of the manipulation action.

All these units have challenges by themselves. For example, calculating the reference value for placing adverbs requires to know the context in which the actions are performed. Every single block in this layer requires a lot of research, so we can succesfully implement it.

6. EXPERIMENT SETUP

In this section, we explore the AU which is in charge of calculating a value meant to be used by the FA. The rest of blocks in the interface layer will be considered for future research.

6.1 Training of affordance unit

Instead of manually creating harcoded tables with the affordance information, we decided to create a training dataset of atomic tasks, for a specific scenario, and based on that, generate an affordance model. This method allows to automatically generate an affordance model for any scenario or environment. This approach is quicker a more flexible.

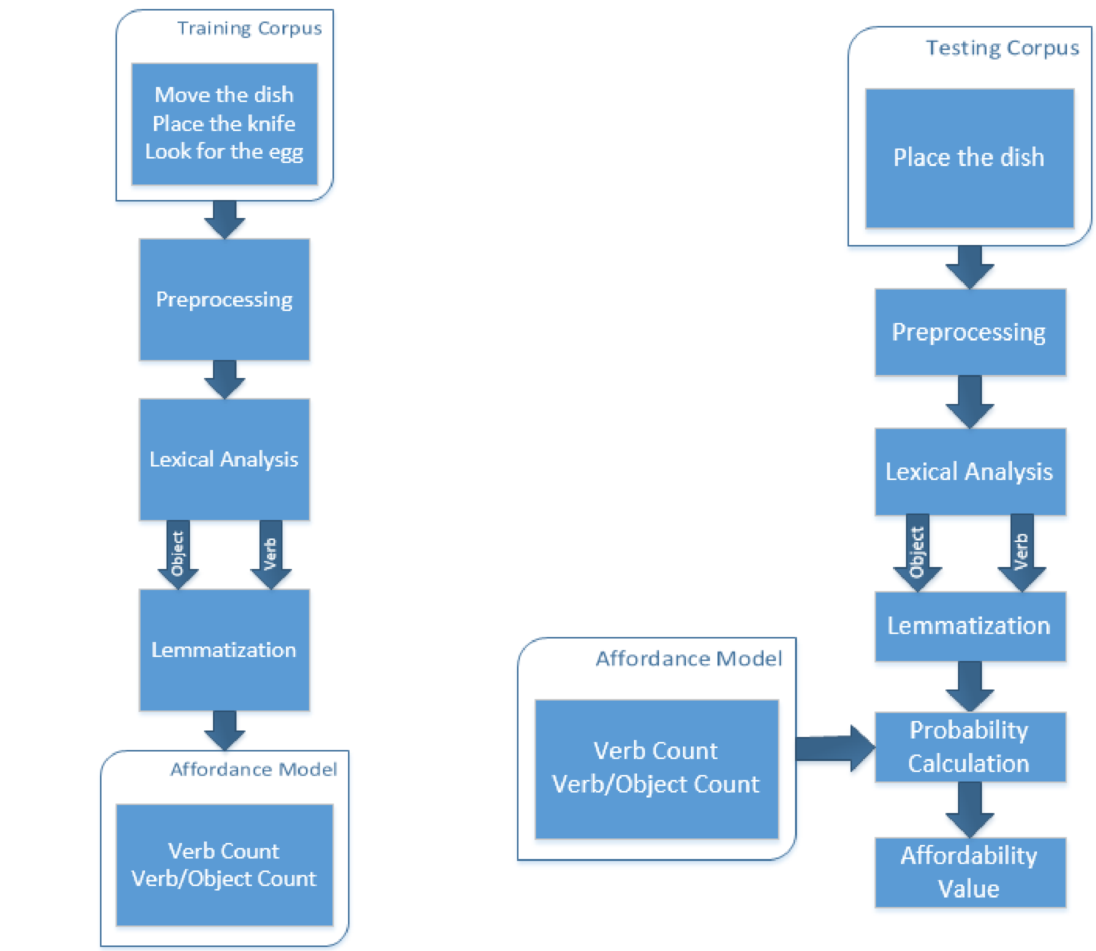

Figure 3. Training and testing of affordance unit. Training consists in three stages: preprocessing, lexical analysis, and lemmatization. Testing consists in four stages: preprocessing, lexical analysis, lemmatization, and probability calculation.

The training system consist in three stages: preprocessing, lexical analysis and lemmatization. The preprocessing stage removes comments and puntuation marks, it also converts the atomic commands to lowercase. The lexical analysis recognizes the object and the verb, it performs name entity recognition. Finally, the lemmatization stage identifies the verb's lemma (See figure 4).

The affordance model implementation consists in two text files: one to register the number of ocurrences of each verb (unigram), and another to register the number of ocurrences of Verb + Object (bigram). This information will be used to calculate the probability indicated by equation (1) for any posible combination of verbs and objects.

Once, the affordance model is generated, the conditional probability can be calculated. The affordability calculation process has four stages: preprocessing, lexical analysis, lemmatization, and probability calculation. The first three stages are identical to ones being used to generate the affordance model. The last stage is the new one, and it is meant to calculate the probability given by equation (1).

A zero probability means that the bigram did not have any ocurrency in the training corpus which leads to a no-affordability result. The zero probability result might also mean that the training corpus was not big enough to contain the ocurrency. To avoid this issue we must use a sufficient training corpus.

7. RESULTS

The conditional probability of the five most used verbs and the five most frequent actions, in a kitchen environment, is shown in table I. This table is called "affordance matrix", and it turns out to be very compact and simple way to describe and characterize an environment.

TABLE I. AFFORDANCE MATRIX

		Objects				
		Dish	Knife	Tomato	Bread	Egg
Verb	Move	0.12	0.14	0.10	0.07	0.03
	Look for	0.17	0.09	0.05	0.00	0.09
	Place	0.12	0.03	0.12	0.15	0.06
	Enter	0.00	0.00	0.10	0.00	0.00
	Pour	0.00	0.00	0.16	0.00	0.08

The affordance matrix shows the conditional probability of manipulating an object once the action has been specified. For example, the conditional probability given by *P(Dish/Move)* is *0.12*, which is the probability of manipulating a "dish" once we have already selected "move" as the manipulation action.

If the probability is zero, the obvious conclussion is that there is no affordability between the action and the object, however this could also be due to the size of the training corpus. Unlike many other areas of engineering, a corpus of atomic tasks for specific scenarios was not found, so a small training corpus of 236 atomic task had to be created.

The affordance matrix has some interesting properties that make us able to calculate the object and the action ocurreny based on the affordance matrix. If A is the affordance matrix, the object ocurrency $c(o_i)$ and the action ocurreny can be calculated as below:

$$\boldsymbol{c}(o_i) = \boldsymbol{A}^t \cdot \boldsymbol{c}(a_i) \quad (3)$$

$$\boldsymbol{c}(a_i) = (\boldsymbol{A}^{-1})^t \cdot \boldsymbol{c}(o_i) \quad (4)$$

The affordance matrix has the potential to be used as a more effective and generic way to model and compare dynamic environments. It might also provide a better and more compact description of the scene in comparison to a physical level description.

The affordance matrix enable us to reason with high-level knowledge about the environment, instead of the traditional physical level description, which have been mostly developed for static environments, such as: grid maps, line maps, elevation maps, and meshes.

8. CONCLUSSIONS

A robotic architecture design, meant to close the gap between the high-level layer and the low-level layer was presented. The key element is this architecture is the interface level layer design that consists in five modules: affordance unit, mapping unit, determination of reference, object properties, and feasibility analysis (See figure 2). The architecture was studied by showing how we can resolve a natural language command “make me a breakfast”, through the different layers of the robotic architecture, from the higher level to the lower level.

Every module in the interface layer is a challenge by itself and its implementation is not yet completely resolved. But one specific element was analyzed in this research: the affordance unit. This unit was addressed by creating a model of conditional probabilities and it was tested based on a kitchen environment. The conditional probability of the five most used verbs and the five most frequent actions, in a kitchen environment, is shown in table I, which is called “affordance matrix”.

In the affordance matrix, if the conditional probability for a given object and a given action is zero, the conclussion migh be that there is no affordability between the action and the object. A zero value could also be due to the size of the training corpus. In the future, a bigger dataset will be built and analyzed for different environments. We also consider that this matrix has the potential to be used as a more effective way to model and compare dynamic environments. It might also provide a better and more compact description of the scene in comparison to a physical level description.

Future work should be focused on how to design, study and implement remaining subsystems and all interactions between them. In regards to the affordance unit, a bigger corpus and the inclussion of more scenarios is needed.

8. ACKNOWLEDGMENTS

The present paper is a partial result of a research project entitled "Manipulation of everyday objects using an object model system. Creation of an object model library and a new object model, 322-B6-279" funded by the Research Agency of the University of Costa Rica (Vicerrectoría de Investigación). Additional support is provided by the National Technical University (UTN), University of Costa Rica Dean's office, Electrical Engineering department, Engineering Research Institute.

REFERENCES

[1] M. A. V. J. Muthugala and A. G. B. P. Jayasekara, "A Review of Service Robots Coping With Uncertain Information in Natural Language Instructions," in IEEE Access, vol. 6, pp. 12913-12928, 2018.

[2] F. Ruiz-Ugalde, G. Cheng and M. Beetz, "Fast adaptation for effect-aware pushing," 2011 11th IEEE-RAS International Conference on Humanoid Robots, Bled, 2011, pp. 614-621.

[3] S. Cousins, "Challenges of building personal robots" 2011 IEEE Hot Chips 23 Symposium (HCS), Stanford, CA, 2011, pp. 1-36.

[4] H. M. R. T. Bandara, M. M. S. N. Edirisighe, B. L. P. M. Balasooriya and A. G. B. P. Jayasekara, "Development of an interactive service robot arm for object manipulation," 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya, 2017, pp. 1-6.

[5] Ruiz-Ugalde, F., G. Cheng and M. Beetz: Prediction of action outcomes using an object model. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1708–1713, October 2010.

[6] Romay, A., S. Kohlbrecher, D. C. Conner y O. von Stryk: Achieving versatile manipulation tasks with unknown objects by supervised humanoid robots based on object templates. In 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pp. 249–255, November 2015.

[7] Ruiz-Ugalde, F., G. Cheng y M. Beetz: Fast adaptation for effect-aware pushing. In 2011 11th IEEE-RAS International Conference on Humanoid Robots, pp. 614–621, October 2011.

[8] Ruiz Ugalde, Federico: Compact Models of Objects for Skilled Manipulation, 2014. Thesis.

[9] Rizzolatti, G., Fogassi, L. and Gallese, V.: Motor and cognitive functions of the ventral premotor cortex. Current Opinion in Neurobiology. Current Opinion in Neurobiology. Volume 12, Issue 2, 2002, pp. 149-154.

[10] Quack, B., F. Worgotter y A. Agostini: Simultaneously learning at different levels of abstraction. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4600–4607, September 2015.

[11] Ferrein, Alexander, Christopher Maier, Clemens Muehlbacher, Tim Niemueller, Gerald Steinbauer and Stavros Vassos: Controlling Logistics Robots with the Action-based Language YAGI. In Proc. of 9th International Conference on Intelligent Robotics and Applications (ICIRA2016), Tokyo, Japan, 2016.

[12] Fang, J., J. Zhao, F. He and X. Lin: Design and research of three-layers open architecture model for industrial robot software system. In 2013 IEEE International Conference on Mechatronics and Automation, p´aginas 104–109, August 2013.

[13] Domingos, Pedro: What’s missing in AI: The interface layer. University of Washington, Washington, USA, 2007.

[14] Domingos, Pedro y Daniel Lowd: Markov Logic: An Interface Layer for Artificial Intelligence. Morgan and Claypool Publishers, 1st edition, 2009, ISBN 1598296922, 9781598296921.

[15] Ruiken, D., T. Q. Liu, T. Takahashi y R. A. Grupen: Reconfigurable tasks in belief-space planning. In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pp. 1257–1263, November 2016.

[16] Muller, Armin, Ra Kirsch and Michael Beetz: Transformational planning for everyday activity. In 17th international, pp. 248–255, 2007.

[17] Gorges, N., A. J. Schmid, D. Osswald y H. Worn: A framework for creating, coordinating, and executing skills on a humanoid robot. In 2007 7th IEEE-RAS International Conference on Humanoid Robots, pp. 385–391, November 2007.

[18] Tenorth, M. and M. Beetz: Priming transformational planning with observations of human activities. Universidad Técnica de Munich, Intelligent Autonomous Systems, Garching, Germany, 2010.

[19] Rodríguez Consuegra, Francisco: La filosofía del lenguaje: su naturaleza y su contexto. Dianoia, 48(50):41–68, 2003.

[20] Thenmozhi, D., R. Seshathiri, K. Revanth and B. Ruban: Robotic simulation using natural language commands. In 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), pp. 1–4, Jan 2017.

[21] Domingos, Pedro: What’s missing in AI: The interface layer. University of Washington, Washington, USA, 2007.

[22] Domingos, Pedro y Daniel Lowd: Markov Logic: An Interface Layer for Artificial Intelligence. Morgan and Claypool Publishers, 1st edition, 2009, ISBN 1598296922, 9781598296921.

[23] M. Tenorth, D. Nyga and M. Beetz, "Understanding and executing instructions for everyday manipulation tasks from the World Wide Web," 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, 2010, pp. 1486-1491.

[24] Yamazaki, K., Y. Watanabe, K. Nagahama, K. Okada and M. Inaba: Recognition and manipulation integration for a daily assistive robot working on kitchen environments. In 2010 IEEE International Conference on Robotics and Biomimetics, pp. 196–201, Dec 2010.

[25] Winkler, J. and M. Beetz: Robot action plans that form and maintain expectations. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5174–5180, Sept 2015.

[26] Lagrand, C., M. v. d. Meer and A. Visser: The Roasted Tomato Challenge for a Humanoid Robot. In 2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 341–346, May 2016.

[27] Blodow, N., L. C. Goron, Z. C. Marton, D. Pangercic, T. Ruhr, M. Tenorth and M. Beetz: Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4263–4270, Sept 2011.

[28] Feix, T., I. M. Bullock y A. M. Dollar: *Analysis of Human Grasping Behavior: Correlating Tasks, Objects and Grasps.* IEEE Transactions on Haptics, 7(4):430-441, Octubre 2014, ISSN 1939-1412.

[29] Krivic, S., E. Ugur y J. Piater: A robust pushing skill for object delivery between obstacles. In 2016 IEEE International Conference on Automation Science and Engineering (CASE), páginas 1184–1189, Agosto 2016.

[30] Zhang, C. and J. A. Shah: Co-optimizing task and motion planning. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4750–4756, Octuber 2016.

[31] Ferrein, Alexander, Christopher Maier, Clemens Muehlbacher, Tim Niemueller, Gerald Steinbauer and Stavros Vassos: Controlling Logistics Robots with the Action-based Language YAGI. In Proc. of 9th International Conference on Intelligent Robotics and Applications (ICIRA2016), Tokyo, Japan, 2016.

[32] Guerra-Filho, G. y Y. Aloimonos: A Language for Human Action. Computer, 40(5):42–51, Mayo 2007, ISSN 0018-9162.}

[33] Sugiura, Komei and Naoto Iwahashi: Learning Object-manipulation Verbs for Human-robot Communication. In Proceedings of the 2007 Workshop on Multimodal Interfaces in Semantic Interaction, WMISI ’07, pp. 32–38, New York, NY, USA, 2007. ACM, ISBN 978-1-59593-869-5.

[34] Calli, B., A. Walsman, A. Singh, S. Srinivasa, P. Abbeel and A.M. Dollar: Benchmarking in Manipulation Research: Using the Yale-CMUBerkeley Object and Model Set. IEEE Robotics and Automation Magazine, 22(3):36 – 52, 2015.

[35] Zhang, Shiqi, Fangkai Yang, Piyush Khandelwal and Peter Stone: Mobile Robot Planning Using Action Language, pp. 502–516. Springer International Publishing, Cham, 2015, ISBN 978-3-319-23264-5.

[36] Glenberg, Arthur M. and Michael P. Kaschak: Grounding language in action. Psychonomic Bulletin & Review, 9(3):558–565, Sep 2002, ISSN 1531-5320.

[37] I. Koštál, "A .NET application searching for data in a log file of the KUKA industrial welding robot". Proceedings of the 16th International Conference on Mechatronics - Mechatronika 2014, Brno, 2014, pp. 656-661.

[38] J. J. Gibson, The Theory of Affordances. Psychology Press, Sep. 1986, pp. 127-143.

[39] A. Stoytchev, "Behavior-grounded representation of tool affordances," in IEEE ICRA, April 2005, pp. 3060-3065.

[40] E. Ugur, M. R. Dogar, M. Cakmak, and E. Sahin, "Curiosity-driven learning of traversability affordance on a mobile robot," in IEEE ICDL, 2007, pp. 13-18.

[41] P. A. Ramírez, S. Ramamoorthy and K. S. Lohan, "Object Affordances by Inferring on the Surroundings*," 2018 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), Genova, Italy, 2018, pp. 69-70.

[42] H. O. Song, M. Fritz, D. Goehring and T. Darrell, "Learning to Detect Visual Grasp Affordance," in IEEE Transactions on Automation Science and Engineering, vol. 13, no. 2, pp. 798-809, April 2016.