VICA: A Vicarious Cognitive Architecture Environment Model for Navigation Among Movable Obstacles

Halim Djerroud and Arab Ali-Chérif
Laboratoire Paragraphe, Université Paris VIII, 2 rue de la Liberté 93526 Saint-Denis, France

Keywords: Navigation Among Movable Obstacles, Cognitive Architecture, Multi-agent System.

Abstract: This article presents a new Cognitive Architecture Environment model for Navigation Among Movable Obstacles (NAMO). This model is the result of a novel approach based on the Theory of Mind and more particularly on the notion of 'vicariance' as an essential strategy of the robot's interaction with outside world. The implementation of our model follows the advances in AI and the Cognitive Robotics research area, where a cognitive architecture environment is represented as a Multi-Agent System (MAS). The MAS representation offers the robot the ability to produce a representation of its environment as well as the possibility to run all types of action simulations in order to anticipate the environment's reactions. The environment state values, both predictive and real as transcribed during simulation and real action movements, are compared to each other in order to keep the correct ones and avoid errors. This is a continuous learning and leads to the construction of a safe path of actions into a dynamic environment. The experiment results show the efficiency of our model, offering an intelligent guide to the robot in order to perform tasks among mobile agents, by avoiding a maximum number of obstacles.

1 INTRODUCTION

In recent years, a lot of research has dealt with the goal of how making a robot able to efficiently maneuver in a crowded space (Renault et al., 2019), such as a domestic environment, filled with different types of agents such as human, robot, or other. We propose to reach the same goal from a different point of view: how the environment, which can not be specified in advance, could offer to the robot the knowledge gained through interactions with it and the other agents, and their experience of acting in this environment. The robot has to manage the possible unpredictability of the objects in the environment and its actual behavior when it executes its plan. Our approach is based on the theory of mind (Theory of Mind - ToM), and more particularly the concept of vicariance as defined by Berthoz (BERTHOZ Alain, 2015). Vicariance is a polysemous term but fundamentally the concept is equivalent to the idea of the potential substitution of one solution or function for another. According to this idea, it is possible to perform the same tasks with different systems, solutions or behaviours which constitutes the basis for diversity.

In this article we present the concept and implementation of a vicarious cognitive architecture model environment for Navigation Among Movable Obstacles (NAMO), named VICA. The contributions are the following: 1) The robot has the ability to produce a mental image of its environment; 2) the robot is able to run simulations in order to anticipate the environment's reactions; 3) The robot interacts with the environment and learns from all the environment state values during the training states.

In the domain of cognitive architecture and more specifically in cognitive robotics (Lemaignan et al., 2011), most architectures are generic and few of them can truly manage the complexity of interactions between objects being in the same environment. Numerous applications (Mueggler et al., 2014) are currently developed for robots that move in an environment with movable obstacles, but most research papers dealing with this problem (Moghaddam and Masehian, 2016)(Mirabel and Lamiraux, 2016) do not always make explicit the underlying cognitive architecture.

Our work is part of the theory of mind, which is also a branch of the philosophy of mind. We are strongly inspired by the work of the physiologist Alain Berthoz (Berthoz, 2008). He describes the brain as a predictor and action simulator. The brain's function is to anticipate future environmental events and simulate the adequate movement to fulfill a need, this is the principle of Vicariance. We represent the cogni-

Djerroud, H. and Ali-Chérif, A.
VICA: A Vicarious Cognitive Architecture Environment Model for Navigation Among Movable Obstacles.
DOI: 10.5220/0010269602980305
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 298-305
ISBN: 978-989-758-484-8

tive architecture environment model as a MAS system intelligently reinforced by the continuous simulations experience acquisition.

2 RELATED WORKS

2.1 Navigation Among Movable Obstacles (NAMO)

Current work in the domain of NAMO can be divided into two main categories: offline planning and online planning.

Offline planning assumes that all information about the space, in which the robot will move, is known in advance. Among the research on offline NAMO, Chen et al. (Chen and Hwang, 1990) first proposed a grid-based method to represent the environment, using both a global and local planner, in order to find a path that leads to a goal. The algorithm is not optimal because it does not solve a wide variety of problems. The work of (Okada et al., 2004) presents a humanoid robot with three sub-groups and planners, one for each of the following elements: environment, movement and manipulation. The algorithm does not take into account mobile obstacles that indirectly block the path to the goal(Stilman and Kuffner, 2008). Most results on offline NAMO research show that this approach isn't effective, it lacks flexibility.

In online planning the robot has only partial knowledge of its environment, and forces the robot to modify its original plan based on new information acquired during its journey. The research of Wu et al. (Wu et al., 2010) is one of the first to bring the subject of NAMO in an unknown environment. It presents an algorithm based on simple assumptions. The algorithm gathers new information during the robot's movement, and identifies new data that do not affect the calculations of the previously established path. However the local solutions resulting from new information on the immediate environment do not solve all cases. In similar work for NAMO, in an unknown environment (Wu et al., 2010), the proposed algorithm turns out to be optimal under certain conditions, but numerous cases still remain insolvable. Today's trend of online NAMO seems more promising and better-suited. The work of (Levihn et al., 2013) presents a method providing a theoretical decision solution for action selection for NAMO applied to a continuous-control robot. The algorithm combines Markov decision-based planners (MDP) as well as Monte-Carlo simulation. The presented planners solve some of the problems encountered in specific NAMO cases, but they never present a generic solution applicable in all cases regard of a vicarience mind.

2.2 Cognitive Architectures

Research in cognitive architectures for NAMO, show that these approaches offer flexibility in their way to generalizing problems. Work in the domain of cognitive architecture is longstanding, and state-of-the-art examples can be found in (Ye et al., 2018)(Kotseruba et al., 2016). It falls into three main families: 1) Bio-inspired cognitive architecture. 2) Cognitive architecture for the solving of artificial intelligence problems. 3) Cognitive architecture based on psychological and philosophical theories.

2.2.1 Bio-inspired Cognitive Architecture

Has the objective of modelling human behavior, and has been under continuous development since the late 1970s (Martin et al., 2020) (Remmelzwaal et al., 2020). Among the best-known, we quote: ACT-R (Adaptive Control of Thought-Rational) (Anderson, 2019) is organized in a set of modules, each dealing with a different type of information corresponding to an equivalent in humans (visual, perception, memory, manual, etc.). Each module has its own version of the three memories: Working memory (WM), Declarative memory (DM) and Procedural memory (PM). Coordination is ensured by a central system. CLARION (Connectionist Learning with Adaptive Rule Induction On-line) (Sun, 2006) is a hybrid architecture that combines symbolic and connectionist representations, while developing artificial agents.

2.2.2 Cognitive Architectures for Solving Artificial Intelligence Problems

These architectures are based on logic programming and machine learning algorithms. We quote SOAR (State, Operator And Result) (Laird, 2012) which is a purely "symbolic AI" architecture, as well as iCub (Vernon et al., 2011). ICARUS (Choi and Langley, 2018) is more recent, storing two distinct forms of concepts. They both imply relations between the objects and need hierarchical organization of long-term memory. To the best of our knowledge, there is no implementation of these architectures in robots that act in real environments.

2.2.3 Cognitive Architectures based on Philosophical Theory

They are based on philosophical and psychological theories and deal with problems such as action, perception, reasoning and intentionality. In BDI architecture (Belief, Desire, Intention) (Rao and Georgeff, 1991) we find the theory where beliefs and desires are the cause of the intention to act, like in PRS (Procedural Reasoning System) (Wooldridge, 2009) for rational agents. LIDA (Learning Intelligent Distribution Agent) (Friedlander and Franklin, 2008) is a cognitive architecture based on Bernard Baars's Global Workspace psychological theory (Baars, 2005). It has a cognitive cycle divided into three phases: comprehension, attention and selection of action, and learning. These phases are repeated indefinitely. CARMEL (in French - Compréhension Automatique de Récits, Apprentissage et Modélisation des Échanges Langagiers) is an architecture developed by Grard Sabah (Sabah and Briffault, 1993). In this system the agent makes itself a symbolic representation of the one it will interact with.

Currently, researchers also deal with the Theory of Mind (ToM) which is itself a branch of the philosophy of mind. Our work follows this vein and is inspired by the works of physiologists. (Berthoz and Debru, 2015) describes the brain as a predictor and action simulator and the main functions are: anticipation of future events and simulation of the appropriate movements in order to respond accordingly. The author calls this principle Vicariance.

3 PRELIMINARY

Neuroscience has inspired many researches in robotics, the goal of which is to achieve efficiency like natural systems such as the brain. We give some examples like the role of mental simulation of the road in navigation (Trullier et al., 1997), or how the brain simulates Newtonian laws and determines the trajectory of objects in space (McIntyre et al., 2001), or how it simulates the rotation of an object (Wexler et al., 1998).

(Berthoz, 2017)(Berthoz, 2012)(Berthoz, 2000) provides a general theory of the brain functioning. In the principle of Vicariance, the anticipation and simulation of the appropriate movements to fulfill a need. The simulation of imagining a movement (Wexler et al., 1998), matches to mentally simulating the body's movement in the computational space of the brain. This mechanism is essential for quick movements; the brain takes all or a part of sensory information and process it in order to act. The brain simulates internally possible actions before choosing and engaging in one, knowing that in many cases it is not possible to test multiple actions.

Let us consider a robot using such a system, moving in a crowded space with movable obstacles. The robot, facing a dynamic obstacle on a path, must avoid the obstacle while advancing itself. It considers the movement of the obstacle to avoid collision when crossing the obstacle's path. The function of simulation and prediction is fundamental. The robot plans the action, while planing the movement at the same time. Therefore, the system selects what is important or pertinent sensory information for this movement. In other words, at every phase of movement, the brain will pre-select sensory input considered as important. On the other hand, the system is not limited to select important sensors only, because it can predict the state in which the objects should be if the movement is accomplished as they should be. In this article we propose an implementation based on this hypothesis. To represent the environment (a mental image of the environment) in terms of data structure, the work of (Djerroud and Cherif, 2019)(Djerroud and Cherif, 2018) show that MAS are perfectly capable of representing the environment with its lows and rules. We seek to reproduce observable environments, in MAS form, able to modify simulation's parameters, to understand its functions, and finally predict the future state of the system. The cognitive architecture presented in this article uses MAS, which offers an environmental engine with the possibility to create agents and integrate rules on the fly. MAS is defined as follows:

$$MAS =< Agents, Environment, Coupling >$$

$$Agents = Agent_1, ..., Agent_n$$
$$Agent_i =< State_i, Input_i, Output_i, Process_i >$$
$$Environment =< State_e, Process_e >$$

A MAS is composed of a set of agents, an environment and the coupling between them. An agent is defined as a set of *states*, *inputs*, *outputs* and *process*. A *state* is the set of attributes which define an agent. *Inputs* and *outputs* are sub-sets of *states*, whose variables are coupled with the environment. The inputs and outputs can represent the sensors and performers of an agent. The *process* is an autonomous process executed internally by the agent. The *coupling* is a mechanism enabling the linking of agent attributes to the environment. The environment is defined as follows:

$$Environment =< States, Rules, Process >$$

$$States =< SharedAttributes, InternalAttributes >$$

The environment is considered an active entity. It has its own process, so it is able to change its state independently the actions of the agents that evolve within. It is composed of *states*, *rules* and *process*. *States* are the attributes representing the state of the environment at a given moment. We can distinguish two types of attributes: *shared attributes* and *internal attributes*. The former is shared with the agents (position, for example), while the latter is not. *Internal attributes* are the internal properties of the environment, such as size, coefficient of friction in a certain space, etc.

Rules are the rules that the environment must respect. For example, two agents must not be in the same place at the same time. Rules are defined as follows:

$$rule =< expression \; ? \; action1 : action2 >$$

A rule is represented as an expression. It corresponds to the law that the environment must check. The actions correspond to the actions to be performed, depending on whether the rule is respected or not.

4 VICA ARCHITECTURE: GLOBAL VIEW

VICA architecture (cf. Fig1) works in a loop. Each cycle allows: 1) to create an internal representation of the observed environment, 2) to simulate and observe the result of these actions, 3) to choose a plan of actions (among simulated plans of action) and apply it to the environment. 4) to observe the environment and evaluate its behavior according to the simulation (learning).

VICA functions in a deliberative loop, this allows action simulation plans (an ordered set of actions) and finally produces only one. VICA has a specificity that enables independent modules to run in a loop. Each cycle of execution of each module ends with a deliberation and the result is delivered in the form of a message to other modules. Each loop takes as input perceptions from the environment, which improve internal representation of the environment. During the deliberative loop, the cognitive part of VICA, called *Abstract Space* (AS) reproduces the perceptions of the environment in a MAS (Djerroud and Cherif, 2019) and represents each perceived entity in the environment as an agent. Subsequently, each agent is enriched with all knowledge already known on this type of agent; this knowledge concerns particularly the possible actions on this agent. In the case of the detection of a box, the system already knows the possible actions, for example push. In the case of the detection of a person, the knowing actions are for example asking to move. For now, the associated actions are integrated in a database. We plan in the short term to integrate a module allowing to deduce automatically possible actions on an object, commonly called *affordance* (Sun et al., 2014). If the object is not referenced, the possible actions are not known. The system will search for the actions of the nearest object; in VICA this aspect is called *vicariance*.

VICA is composed of four modules. Each module functions independently and in a loop. Modules communicate with each other with ACL messages. The first module *Detection and Agentification* enables the observation of the environment and the extraction of useful information, such as detecting an obstacle. The second module *Abstract Space (AS)* can be considered as a physics engine[1]. It is able to represent entities in the form of agents and apply laws in order to observe and predict the evolution of entities in the environment. The sub-module *Simulation* enables the construction of a MAS in terms of observation. The module *Evaluate Outcome* indicates to MAS the action to be simulated, evaluates the results of different simulations to make a decision, and then applies the best action. The final module *Select Action* indicates how to execute a plan of action in the real environment.

4.1 Principal Modules of VICA Architecture

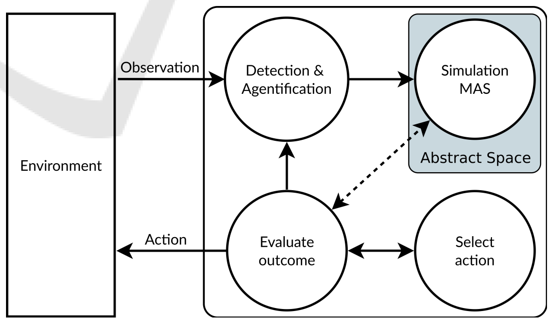

Figure 1: Conception scheme of VICA architecture.

4.1.1 Detection and Agentification

The role of this module is to collect and merge the information coming from the sensors of the robot and to express them as agents in the MAS integrated in the *AS*. More precisely, the mobile robot is equipped with several sensors (RGB camera, depth camera, LI-

[1] physics engine: software providing an approximate simulation of real physical systems.

DAR, pressure sensor, etc.). Sensor data is analyzed to obtain as much information as possible about the observed scene. An image recognition system is applied to the RGB camera to determine the type of object and its shape (box, table, robot, human, etc.), then the depth camera indicates the dimensions of the detected objects and finally the LIDAR indicates the distance between the moving robot and the objects, as well as between the objects themselves. The second role of this module is Agentification which goal is to present data to *AS*. When an entity is detected, it collects as much information as possible about that entity, sending it to the *AS* via messages using the ACL language. Before sending the entities detected in the scene to the *AS* module, an object authentication system adds more information about the possible actions on these entities. Information is stored automatically to a database, some of it is deduced by a machine learning algorithm (for example the force to apply to move an object according to its shape and dimensions) or by simple calculations (for example the volume of a geometric object).

4.1.2 Abstract Space

It enables the construction of an internal representation of the observed environment. It represents the current state of the environment in a MAS. Each element of the environment is represented as an agent. An agent is therefore considered as an internal representation of an object. The first part of this sub-system is responsible for creating agents. When an object is detected, the system represents it as an agent. Therefore, we can identify two cases, either the detected object already exists in the *AS* or it is new, in this case an agent is created or updated. The MAS created in *AS* represents a static image of the environment at a given moment. The *Simulation MAS* module offers a set of services that allow the module *Evaluate outcome* to accomplish some tasks. Among the offered services : a) *Simulation*: enables launching the simulation of an action (chosen by the *Select Action* module) b) *Result simulation*: enables consulting the attributes (results) after simulation. c) *Commit / RollBack*: will allow the *Evaluate outcome* module to launch multiple simulations and then choose a single action to perform. Between each simulation, the *Evaluate outcome* subsystem must reinitialize the MAS to its initial state to be ready to perform a new simulation; this is the main role of the *Rollback* service. After choosing the action, the sub-system applies a *Commit* to validate this action. The commit re-initializes the MAS to its last state that corresponds to the environment (before the simulation) and informs *AS* of the chosen plan of action in order to compare the real results obtained during the last cycle with the simulation ones.

4.1.3 Select Action

This module includes a set of procedures describing simple actions. Each action represents a robot's movement. It is generally used to perform a simple task, such as moving forward, turning left or right, pushing an object, and so on. These actions are hard coded, i.e. the system can not enrich the possible actions. This module serves as a knowledge base of robot's actions; for example the module *Evaluate Outcome* wants execute an action such as "go forward", then it does not need to know how to do this action. It need only call a "go forward" routine, the details of the implementation of this routine is described in this module.

4.1.4 Evaluate Outcome

The objective of this module is to provide a plan of action. It constructs an oriented and weighted graph between the current position of the robot and the goal, the graph is obtained using information from *AS*. In VICA implementation, heuristics on the first-level of the graph correspond to effort (distance + effort if the robot must push an obstacle). Heuristics on distance are obtained through the distances provided by LIDAR and the depth camera, heuristics concerning the effort needed is obtained by simulation. Heuristics on the other levels correspond to the linear distance between the node and the goal. Whenever the robot advances the heuristics on the first level of the tree are recalculated, and the distances are readjusted if necessary. The choice of a plan of action is obtained by the $A*$ algorithm [2]. The system chooses a branch to explore and performs simulation to update the heuristics. The process is repeated until all first-level branches have been explored. This phase produces a new graph that considers the effort needed to move the obstacle. The new graph will be used to determine the path to be taken by $A*$. Of course it is possible to use other path search algorithms, for now we have only experimented $A*$.

[2] In our case, we use two heuristics, the estimated cost to overcome the closest obstacle that hinders the passage, which is represented by the first level of the graph, and the distance that separates the robot from the goal, in the rest of the levels. When the robot advances towards to the goal, the graph is reduced, so the second level becomes the first and so on.

5 IMPLEMENTATION AND EXPERIMENTS

The goal of VICA is to enable the robot to make a plan and adapt it, if necessary, during its route, to reach the goal. The plan may involve to a) move obstacles in order to pass, and b) ask another agent (for example, another person or robot blocking its path) to let it pass. A first version of VICA was implemented for a virtual robot in order to evaluate the behaviour produced. It was tested in a scenario that presented a cluttered environment including different obstacles of varying size.

Below, we present an example of simple scenario describing the robot confronted with an environmental configuration in which the goal is difficult to reach without moving the objects in its path (cf Fig 2). (1) The module responsible for perception observes the scene and produces a representation in a multi-agent system (MAS). (2) The module tasked with actions produces different plans of action in the form of a graph. (3) All action plans are simulated: push the cube, push the cylinder or pass between the two. The robot is equipped with a force sensor. During the simulation, the force required to move each obstacle is recorded. After simulation, the heuristics (the force necessary to move an obstacle) in the graph are updated, and a path is calculated with help from algorithm A*. In the simulation Fig.2, the action plan between the two objects is identified as the path with the least resistance. (4) Finally, the chosen plan of action is applied to the environment.

The graph (cf. Fig.3) shows the possibilities of actions in the previous environment configuration presented in Fig.2. Each node can represent an obstacle that the robot considers able to move, or a path between two obstacles to reach a position closer to the goal (G). The robot uses heuristics (H) that obtain via simulations that will be used to run the A* algorithm.

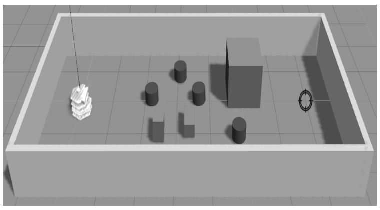

Figure 2: Example of random configuration.

The Table 1 compares our VICA model with two other well-known RRT and D* Lite planning methods. VICA is able to move obstacles, the number of movements is indicated in the column (Movements). The execution time of the movement is indicated in seconds in the (Time) column. The columns (distances) indicate the length traveled in centimeters. The distance between the starting position and the goal is fixed (350 cm) for all the experiments. Our method is able to do better than D *, because it is able to move obstacles and find a better path instead of avoiding obstacles. The experiments shown here only involve two types of obstacles (fixed and movable), interactive obstacles (robots, humans, etc.) are not tested.

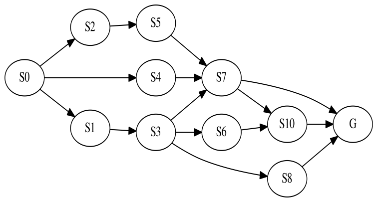

Figure 3: Generated graph according to the observed environment.

VICA models human-like navigation behaviors when facing obstacles. Often the human brain when seeking solutions, chooses solutions that require the least effort. For example, if we wish to pass to the other side either going around the table, or moving the chair that hinders the passage, the second solution is often chosen. The solution proposed by VICA gives results close to human behavior. The implemented learning process forces the robot to interact with others to learn and increase its ability to respond to similar situations. After failing to move unmovable objects, the robot considers other actions. This process is similar to natural human behavior.

6 CONCLUSION AND PERSPECTIVES

In this article, we described VICA, a VIcarious Cognitive Architecture applied to a mobile robot operating in a crowded environment (NAMO). We produced a model with MAS representation of the environment, where the robot is able to inform and understands the evolution of it, while acting and changing its behavior appropriately. The assumption of how the environment could offer to the robot the knowledge gained through interactions, was confirmed by the results that show the model's efficiency. VICA offers an intelligent guide to the robot to perform tasks among mobile agents, by avoiding a maximum number of obstacles while reducing the computation time. This architecture is further validated by interactions with more complex objects (eg men, other robots, etc.) and complex scene configurations to verify that the robot is able to evolve in a truly complex and natural envi-

Table 1: The results of simulation.

	VICA			RRT	D* Lite
Nbr of obstacles	Nbr of movements	Travel time (sec)	Distance traveled (cm)	Distance	Distance
5	0	112	370	418	365
10	2	118	388	436	387
20	4	152	385	444	380
30	7	142	358	465	485
40	9	145	390	487	498
50	10	190	378	395	395
60	12	180	397	409	404
70	13	145	395	415	411
80	14	170	412	-	-
90	15	186	460	-	-
100	15	178	489	-	-

ronment. Further work on knowledge representation is necessary to be compatible with different types of entities evolving in the environment. The multimodal perception module will be completed for the extraction of all the possibilities of actions on an object (Prospects). Improving the learning module with a gradually dynamic knowledge would ensure the best configuration for the goal in any environment.

REFERENCES

Anderson, J. R. (2019). Cognitive architectures including act-r. *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society*, 26(3):295–296.

Baars, B. J. (2005). Global workspace theory of consciousness: toward a cognitive neuroscience of human experience. *Progress in brain research*, 150:45–53.

Berthoz, A. (2000). *The brain's sense of movement*, volume 10. Harvard University Press.

Berthoz, A. (2008). *Neurobiology of" Umwelt": How Living Beings Perceive the World*. Springer Science & Business Media.

Berthoz, A. (2012). Simplexity: Simplifying principles for a complex world (g. weiss, trans.) cambridge.

Berthoz, A. (2017). *The vicarious brain, creator of worlds*. Harvard University Press.

Berthoz, A. and Debru, C. (2015). *Anticipation et prédiction: du geste au voyage mental*. Odile Jacob.

BERTHOZ Alain, T. M.-H. (2015). Towards creative vicariance. *Presses Universitaires de Vincennes, Revue Hybrid*, 2:1–6.

Chen, P. C. and Hwang, Y. K. (1990). Practical path planning among movable obstacles. Technical report, Sandia National Labs., Albuquerque, NM (USA).

Choi, D. and Langley, P. (2018). Evolution of the icarus cognitive architecture. *Cognitive Systems Research*, 48:25–38.

Djerroud, H. and Cherif, A. A. (2018). Visualization tool for jade platform (jex). In *Proceedings of the Future Technologies Conference*, pages 481–489. Springer.

Djerroud, H. and Cherif, A. A. (2019). Environment engine for situated mas. In *ICAART (1)*, pages 129–137.

Friedlander, D. and Franklin, S. (2008). Lida and a theory of mind. *Frontiers in Artificial Intelligence and Applications*, 171:137.

Kotseruba, I., Gonzalez, O. J. A., and Tsotsos, J. K. (2016). A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications. *arXiv preprint arXiv:1610.08602*, pages 1–74.

Laird, J. E. (2012). *The Soar cognitive architecture*. MIT press.

Lemaignan, S., Ros, R., Alami, R., and Beetz, M. (2011). What are you talking about? grounding dialogue in a perspective-aware robotic architecture. In *2011 RO-MAN*, pages 107–112. IEEE.

Levihn, M., Scholz, J., and Stilman, M. (2013). Planning with movable obstacles in continuous environments with uncertain dynamics. In *2013 IEEE International Conference on Robotics and Automation*, pages 3832–3838. IEEE.

Martin, L., Jaime, K., Ramos, F., and Robles, F. (2020). Declarative working memory: A bio-inspired cognitive architecture proposal. *Cognitive Systems Research*.

McIntyre, J., Zago, M., Berthoz, A., and Lacquaniti, F. (2001). Does the brain model newton's laws? *Nature neuroscience*, 4(7):693–694.

Mirabel, J. and Lamiraux, F. (2016). Constraint graphs: Unifying task and motion planning for navigation and manipulation among movable obstacles.

Moghaddam, S. K. and Masehian, E. (2016). Planning robot navigation among movable obstacles (namo) through a recursive approach. *Journal of Intelligent & Robotic Systems*, 83(3-4):603–634.

Mueggler, E., Faessler, M., Fontana, F., and Scaramuzza, D. (2014). Aerial-guided navigation of a ground robot among movable obstacles. In *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*, pages 1–8. IEEE.

Okada, K., Haneda, A., Nakai, H., Inaba, M., and Inoue, H. (2004). Environment manipulation planner for humanoid robots using task graph that generates action sequence. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 2, pages 1174–1179. IEEE.

Rao, A. S. and Georgeff, M. P. (1991). Modeling rational agents within a bdi-architecture. *KR*, 91:473–484.

Remmelzwaal, L. A., Mishra, A. K., and Ellis, G. F. (2020). Brain-inspired distributed cognitive architecture. *arXiv preprint arXiv:2005.08603*.

Renault, B., Saraydaryan, J., and Simonin, O. (2019). Towards s-namo: socially-aware navigation among movable obstacles. In *Robot World Cup*, pages 241–254. Springer.

Sabah, G. and Briffault, X. (1993). Caramel: A step towards reflection in natural language understanding systems. In *Proceedings of 1993 IEEE Conference on Tools with Al (TAI-93)*, pages 258–265. IEEE.

Stilman, M. and Kuffner, J. (2008). Planning among movable obstacles with artificial constraints. *The International Journal of Robotics Research*, 27(11-12):1295–1307.

Sun, R. (2006). From cognitive modeling to social simulation. *Cognition and multi-agent interaction: From cognitive modeling to social simulation*, page 79.

Sun, Y., Ren, S., and Lin, Y. (2014). Object–object interaction affordance learning. *Robotics and Autonomous Systems*, 62(4):487–496.

Trullier, O., Wiener, S. I., Berthoz, A., and Meyer, J.-A. (1997). Biologically based artificial navigation systems: Review and prospects. *Progress in neurobiology*, 51(5):483–544.

Vernon, D., Von Hofsten, C., and Fadiga, L. (2011). *A roadmap for cognitive development in humanoid robots*, volume 11. Springer Science & Business Media.

Wexler, M., Kosslyn, S. M., and Berthoz, A. (1998). Motor processes in mental rotation. *Cognition*, 68(1):77–94.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.

Wu, H.-n., Levihn, M., and Stilman, M. (2010). Navigation among movable obstacles in unknown environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1433–1438. IEEE.

Ye, P., Wang, T., and Wang, F.-Y. (2018). A survey of cognitive architectures in the past 20 years. *IEEE transactions on cybernetics*, 48(12):3280–3290.