Real-World Implementation of an Auction Behavior-Based Robotic Architecture (ABBRA)

Bradford A. Towle Jr.
Department of Computer Science and Engineering
University of Nevada, Reno
Reno, Nevada, USA
towle@cse.unr.edu

Monica Nicolescu
Department of Computer Science and Engineering
University of Nevada, Reno
Reno, Nevada, USA
monica@cse.unr.edu

***Abstract*— Robots for real world applications face multiple challenges due to the dynamic nature of the environment. These challenges require that robotic architectures can handle multiple goals, some of which may be conflicting, and can also handle time-constraints for completion of those goals. In addition, the architectures need to be flexible enough to change their priorities should a change in the environment require it. The Auction Behavior-Based Robotic (ABBRA) architecture proposed in this paper meets these requirements by allowing the robot to handle dynamic changes in the environment while making the most suited decision for the robot at that moment. This paper details two new features of ABBRA and proves that the architecture can function in a real environment. ABBRA supports mapping and the use of clock (real) time for enforcement of time-constraints of goals while handling multiple conflicting goals simultaneously, as well as dynamically adding a new goal during run time. The paper presents the results of five real world experiments using the Segway Robotic Mobile Platform (RMP) in which the proposed auction behavior-based system is deployed in a real and dynamic environment, with dynamically changing constraints.**

Keywords - Behavior Based Robotic; Auction Mechanics; Real-world Implentation

I. INTRODUCTION

Implementing a robot architecture for real world applications requires the capability of handling dynamic constraints. For example, certain goals that the robot needs to achieve may be time sensitive, some goals may have conflicting objectives with other goals, and finally the robot may receive new information from the environment and need to re-evaluate its goals priority. An Auction Behavior-Based Robotic Architecture (ABBRA) was previously developed to meet these requirements and allow multiple conflicting behaviors the ability to compete for control of any one robotic actuators [1]. To accomplish this, each behavior acts as a potential bidder for control over an actuator. Each behavior collects data from the environment, state or possibly the user and integrates the data into a value that represents its importance to run at that moment. This value is called the *behavior activation level*. The behavior will then bid against all of the other behaviors activation levels and the highest value will gain control of the actuator.

ABBRA was subsequently extended by adding the ability to handle two other dynamic constraints[2]. The first was allowing the robotic architecture to enforce time-constraints on the behaviors specifying when they had to begin, thus increasing a behavior's activation level if the remaining time became too short. However, this constraint was limited to using program cycles for time measurement and was unable to predict how long a behavior would take to run. The second dynamic constraint integrated into ABBRA allowed the architecture to add a new behavior dynamically during runtime without restarting the robot's controller. Each new behavior would simply register for the auction of the actuator it needed and simply begin bidding on the next round. This resulted in a seamless integration of its new behavior with the existing behaviors and the architecture.

Figure 1: Segway RMP used for tests.

In this paper, the addition of two new features is presented that expand the capabilities of the architecture and demonstrate its feasibility in the physical world. Moreover, this paper describes the results of a physical implementation of ABBRA on the Segway Robotic Mobile Platform (RMP), proving it can fulfill these requirements in the real world (Figure 1). The two new features introduced are i) mapping and localization, and ii) the ability to use real clock time to compute a task's time-constraints. The ability to use maps greatly improved the performance of ABBRA. Using a map built by a Simultaneous Localization and Mapping (SLAM) algorithm the robot can calculate a path to the goal and thus have a better assessment of how long the goal would take to complete. The robot did not actually use the pathfinder to navigate but rather used it to find the true distance to the goal. With this addition, the overall performance of the architecture was substantially increased when using the true distance as the only spatial metric. The second improvement on the architecture allows time-constraints enforcement by actual time instead of program cycles. This allows the user to specify the time-constraint in hours, minutes and seconds making it much more human understandable. The robot is therefore capable of calculating of how long it would take to reach a goal because of the true distance and real time implementation. Thus, the robot could take specific action for any goal that was unreachable in the allotted time-constraint (either abandon it or ask a human for additional help).

This paper is structured into the following sections: Section II provides the related work on this project. Section III explains the testing protocol and details the additional features of the architecture. Section IV provides the results from the tests and their evaluation. Section VI provides a brief plan for future work and makes concluding statements.

II. Related Works

There are several different approaches that can be used for robotic-decision making. Deliberative architectures plan and create a ideal solution for the known world [3, 4]. However, problems can arise when these systems encounter dynamic environments, when new solutions need to be recomputed. An alternative deliberative method is voting, which allows each robot behavior to choose the best action for itself. This works well when behavior outputs are similar and a subset of outputs can be chosen to vote on [5]. Arbitration is another option in which a robot has multiple behaviors, but must choose only one to execute. However, when multiple goals that conflict need control of the same actuator [6], the architecture must prioritize [7] or select which behavior is most applicable [8].

ABBRA's design is clossest to the winner-take-all methods. Here the robot must choose between a set of possible behaviors and choose the one it determines to be the most applicable. In real world, practical applications, the problems arise when a robot needs to handle conflicting goals and when the robot needs to switch tasks due to new requirements or changes in the environment. Activation Networks [6] solve conflicting goal by allowing behaviors to promote other behaviors by injecting "activation energy". The behavior that has the most activation energy will win control. ABBRA does not use inter-behavior communication to promote activation for a certain behaviors but instead, uses the environment to determine which behavior is most efficient to run. This philosophy follows the standard behavior based paradigm where data from the environment provides state information [9]. Once the environmental data has been collected these behavior based paradigms will use this information and either prioritize goals [7] or create/activate inhibition signals to prohibit conflicting goals [10]. ABBRA extends this concept, instead of simply prioritizing goals, it will allow them to compete and dynamically change their priority should the environment change.

Market-based (auctions) paradigms are widely used in multi-agent robotic systems. Since the seminal paper [11] the number of market based robotic papers has increased dramatically increased [12-21]. However, these papers focus on multi-agent (multi-robots) and solve a different problem than ABBRA. Here are some key differences: first, the first difference is multi-agents involve robots competing for a task where as ABBRA deals with multiple behaviors competing for control over the actuators of a single robot. Thus, the reward for the auction is different. Second, multi-agents will bid for tasks whenever they becomes available, whereas in ABBRA the bidding occurs continually thus allowing for dynamic changes in the environment. Third, multi-agent systems must monitor the robot who won the tasks to ensure that it is performing well [14]. In ABBRA if a task does not perform well another task will out-bid it on the next cycle. Fourth, multi-agent systems must worry about external conflicts between robots, where ABBRA resolves conflicting goals by allowing environmental and temporal metrics to influence which behavior has won [22]. Lastly, a lot of research focuses on allowing individual agents to cooperate with each other. ABBRA will automatically allow behaviors be run in parallel as long as they do not use the same actuator [15]. Because of these differences, the problem ABBRA solves and the domain of multi-agent systems are substantially different.

III. Architectural Changes and Methodologies

This project introduces two new changes to the architecture itself. The first modification was the ability to use a map to calculate the true distance between the robot and a known goal. With this addition, any behavior that requires the robot to go to a goal with a known location could determine if it had enough time to make it to the goal. If it did not, the robot would take specific action on that task: either abandon it or ask a human for help and clarification if more time is allowed. The second alternative is currently being implemented as a part of extending the robot's interactive capabilities. As a third alternative, the robot thought it had enough time to finish a task, but the time was below 10% remaining, the robot would increase is nominal speed slightly in order to achieve the goal.

The second improvement was to make all of the task time-constraints enforced by true clock time. Therefore, regardless of how long a program cycle took the clock will still enforce the time-constraint. This also give a more human understandable way to assign time constraints for tasks because it is now in hours, minutes and seconds no longer in program cycles. This update forced some changes with the times fitness function.

In order to decide what behaviors would gain control of the actuators, ABBRA sends specific metrics into a single fitness function for that particular metric type to acquire the contributions for the activation level[1, 23]. For example, Equation 1 details how to calculate the activation level contribution from a spatial metric. Once the behavior calculates all these values, they are then summed and averaged (Equation 2).

$$CTAL = 1 - \frac{BehaviorMetricValue}{MaxMetricValue} \quad (1)$$

CTAL = Contribution to Activation Level

$$ActivationLevel = \frac{\sum_{For\ All\ Metrics}(Behaviors\ Activation\ Level\ Contributions)}{Total\ Number\ Metrics} \quad (2)$$

Regarding time, the system clock enforces the time-constraint system the fitness function needs to be changed. The earliest starting time and the max time allowed of all of the behaviors is stored and integrated into the activation level. These values are used to find the percent time remaining instead of the internal variable that was independent for each behavior (Equation 3). This will allow a large diversity of time-constraints to compete simultaneously because they will be using the same max values for their calculations. Also note that equation 4 integrates the estimated time to finish into the calculations to ensure the robot also considered this when calculating it activation level contribution.

$$CTAL = e^{\frac{-1*(TC - Time - FinishTime)}{LongestTimeConstraint - EarliestStartTime}} \quad (3)$$

TC = Time-constraint for current behavior
Time = Current time of the system clock
FinishTime = the estimated time it will take to finish the goal based on the true distance.
CTAL = Contribution to Activation Level

IV. Results

The architecture was validated by five tests in the real world using a Segway RMP. Each test consists of three known goals, and one unknown goal - the green marker. The first test contains no time-constraints and no dynamic addition of goals during runtime. Therefore, the robot simply chose to go to the closest goal first. The second test gave goal 2 a time-constraint that could not be met therefore the robot abandoned the goal. The third test was the same as the second test except the time-constraint was achievable. This demonstrates that a goal with an achievable time-constraint will have priority over those goals that do not have a time-constraint. The fourth test dynamically adds a goal in runtime without any time-constraint proving the goal can be integrated without causing any change in the behaviors priority. The fifth test dynamically adds a goal with an achievable time-constraint causing an immediate priority change by the robot to attempt to achieve the new goal. This demonstrates the robot can dynamically adjust to changes in the environments.

The robot will be using these behaviors in the following tests:

- Go to specific point on the map (three instances)
- Center and go to green object (single instance)
- ABBRA Auction mechanism
- Several other miscellaneous behaviors to read ROS packages.

The “go to specific point” behavior would be given a known coordinates and go to that location. This was used for the known goals. This behavior was accurate to a quarter of a meter.

The center and go to green object would search out for a green object in the world and then move toward it. Once it was within a single meter it would consider the goal met. This behavior was used to simulate a goal that was not known.

The ABBRA auction mechanism allowed different instances to bid for control of the actuator and it would allow the architecture to adapt to a dynamic environment. This allows for opportunistic execution for that moment. For example, should the robot discover the green object and the current goal being executed does not have a time constraint the robot can choose to take the opportunity to finish the “center and move toward green object” goal before continuing on. These tests were run in a school building with relatively narrow corridors.

Test 1: The control test

The first test ran on the Segway RMP demonstrated the performance of ABBRA under static conditions. In this test, there were three known points in the world that the robot had to visit and an unknown goal the robot had to find. At the beginning of the run, the behaviors for each goal got initialized without any time-constraints. The small oscillation between goal 4 and goal 2 is because the blob tracker temporarily lost the green object (Figure 2).

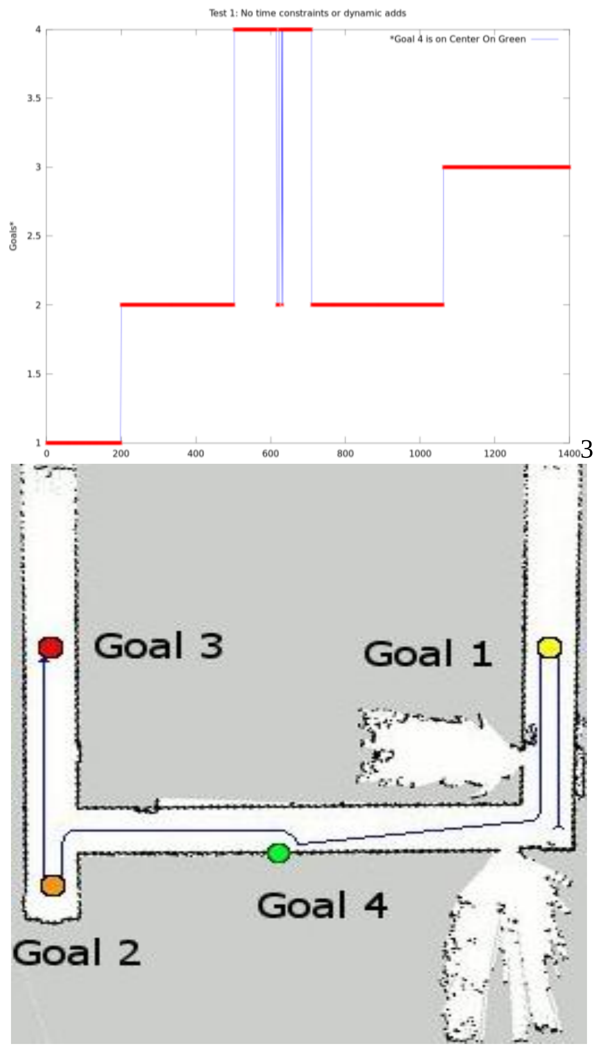

Figure 2: (Top) Order the tasks were completed (Bottom) The path the robot took.

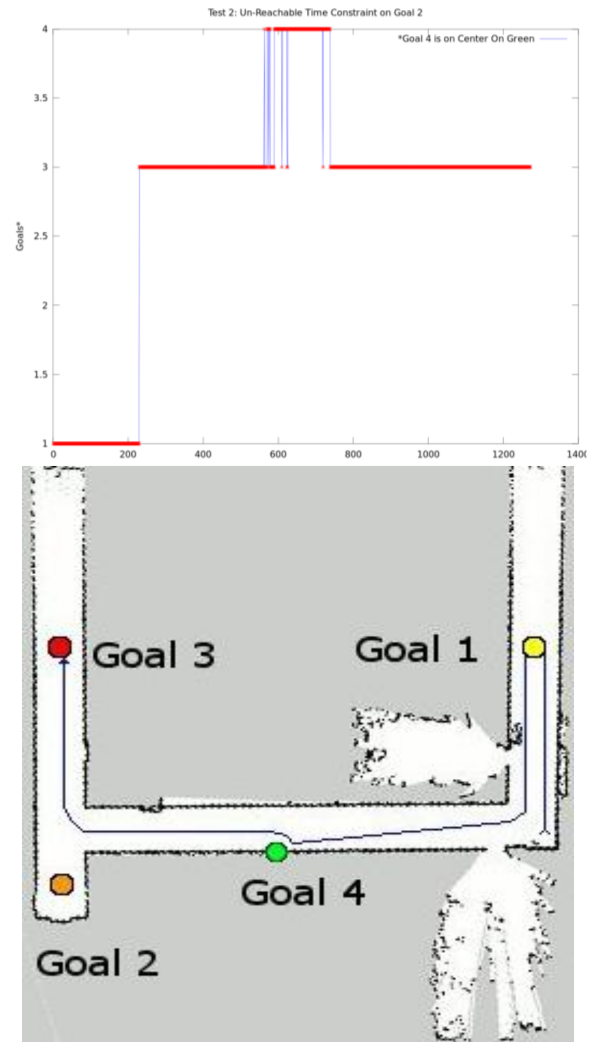

Figure 3: (Top) Order of tasks completed for Test 2 - Note: No goal 2 (Bottom) : The path the robot took to achieve the goal.

Test 2: Time-constraint that cannot be met

The second test demonstrated a new functionality in ABBRA. Since we are using real time to measure the time-constraint and the robot has a path to known goals it can estimate the time it will take to complete the tasks. This offered a new opportunity for ABBRA – if the robot cannot reach the goal in the time-constraint, it will abandon the tasks. This may not always be an ideal solution but this method was chosen to demonstrate the predictive ability of ABBRA. In this test goal 2 is given a time constraint that cannot be met, therefore, is never performed. Again, the blob tracker had a similar problem with losing sight of the green target, hence the oscillations with goal 3 and goal 4 (Figure 3).

Test 3: Time-constraint that is achievable

The third test repeats the last test except this time the time-constraint for goal 2 is achievable. Thus goal 2 is completed first. Since goal 3 is closer, it is finished next. On the return trip to goal 1 the center and move toward green tasks is completed. Note that it is not completed when the robot first sees it. The robot did not detect the green object until it was close to it, therefore goal 4 took less time to actually center and move toward it (Figure 4).

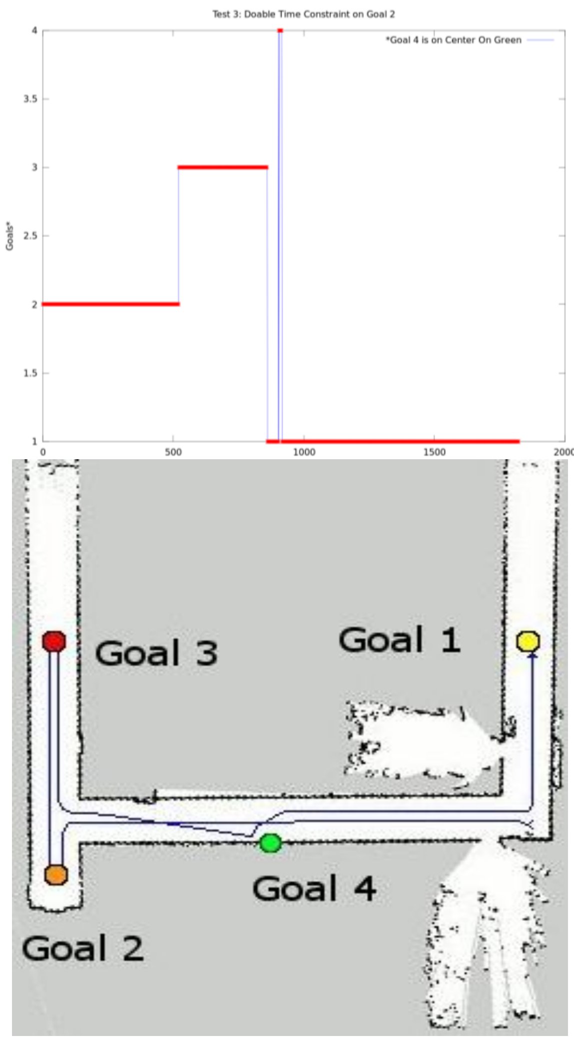

Figure 4: (Top) Order of tasks completed in test 3 (Bottom) Path taken by the robot in test 3

Test 4: Dynamically adding a task with no time-constraint

In this test getting to goal 1 is only requested four minutes after the beginning of the experiment. When it does start there is no time-constraint given to it therefore the other goals are finished before it. Notice the difference between this run and the last, the center on green object is completed before goal 2. Since there was not time constraint, center on green occurred before goal 2, proving the robot did make the opportunistic decision in this situation. Again, the oscillation with center on green object occurs when the blob tracker temporarily lost the green target (Figure 5).

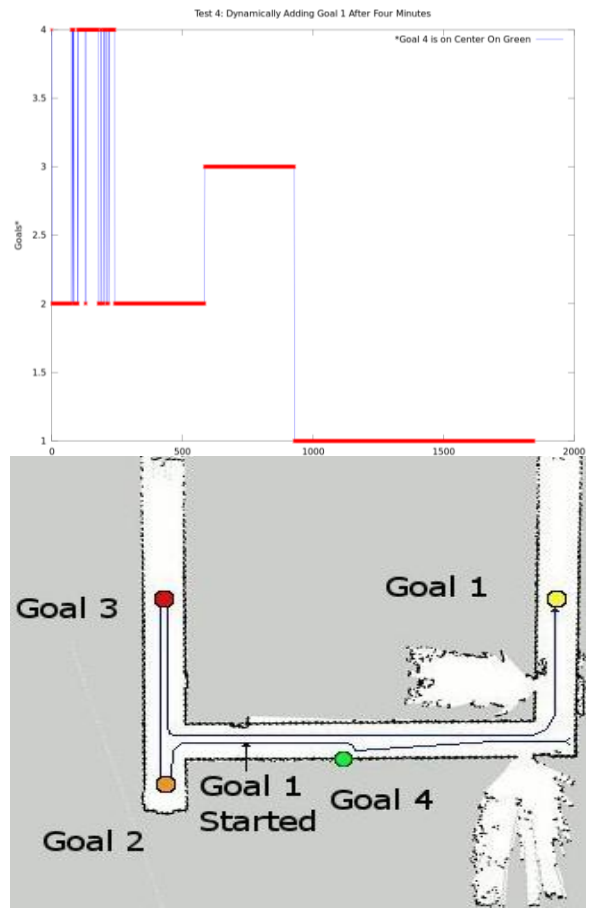

Figure 5: (Top) Order of tasks completed in test 4 (Bottom) Path taken by the robot in test 4

Test 5: Dynamically adding a task with a time-constraint

Test 5 will repeat test four except this time goal 1 is given a critical, but achievable, time-constraint. Notice the robot will have completed the center on color and be moving toward goal 2 when goal 1 activates. This demonstrates the ability for the ABBRA to adapt to dynamic changes in the environment (Figure 6).

V. FUTURE WORK AND CONCLUSION

Several interesting research topics need to be addressed for future work. The first is to add the capability for the robot to naturally interact with a human and receive requests for new tasks. This has always been one of the end goals because ABBRA lends itself to abstracting task to a level where a human communicates. This in turn allows the human the ability to give high-level commands without worrying about every low level task that is required in order to achieve the goal. Moreover, guidelines need to be setup to ensure that humans can safely interact with an auction behavior-based system. Since the robot can dynamically calculate priority, it is feasible to have a scenario where the robot can ask the human to wait while it finishes its current task(s). Obviously, this ability has intrinsic safety concerns however, it will be necessary for the robot to avoid human interaction in certain situations. For example, in a crowded building, the robot may receive un-authorized commands from by-standers or even if the commands are legitimate, there may be too many for the robot to process. Therefore, the robot needs to be able to avoid human interaction when there are critical tasks that require attention. Likewise, the user should be able to get the robots attention when there is an urgent task that needs attention.

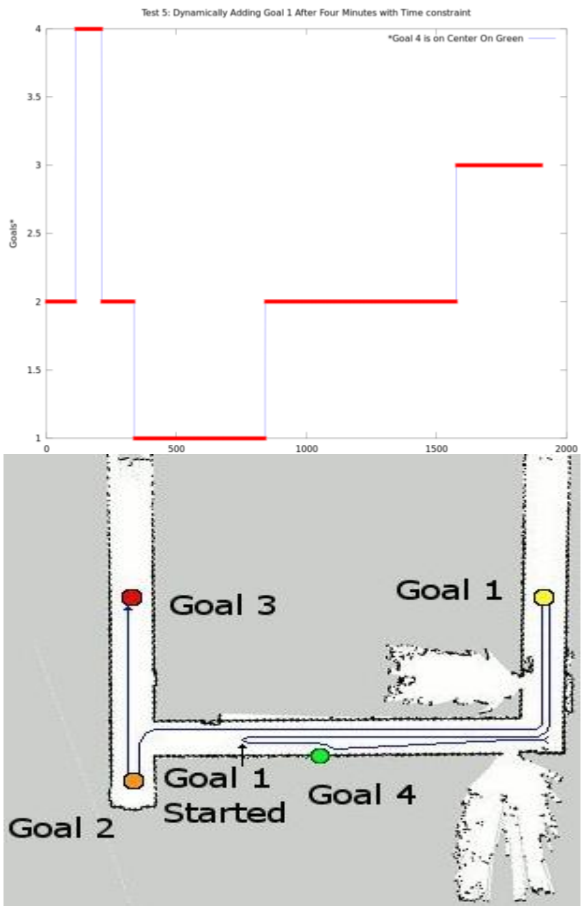

Figure 6: (Top) Order of tasks completed in test 5 (Bottom) Path taken by the robot in test 5

Finally, research will be done on the effects of having multiple users and multiple robots interacting simultaneously using ABBRA. This will determine if ABBRA can be expanded to human-robot teamwork applications [24-27]. The goal here is to study the interactions of heterogeneous teams (robots and humans) more than research task allocation to multi-agents. It is hypothesized that ABBRA will allow a single user to control several robots efficiently and be able to determine which messages require the human's attention and which can be simply logged without the humans involvement.

In conclusion, this paper has presented three new features of the ABBRA architecture. The robot will now use actual path distance instead of Euclidean estimation to determine the distance to a goal. This allows for a much more accurate response and allows the robot to determine whether a certain time-constraint can be met. The use of clock time now allows the robot to enforce time-constraints regardless of how fast or slow a program cycle takes. This system also allows the user request that tasks be performed in terms of hours, minutes and seconds thus making the interaction more natural for a human. Finally, this paper shows the successful deployment of ABBRA in the real world. This deployment also demonstrated that ABBRA could dynamically adapt to the addition of a new goal and handle multiple time-constraints. This was proven with the use of five scenarios mirroring those done on the previous work on ABBRA[2].

These results have shown that an Auction Behavior-Based Robotic Architecture is a viable solution for the real world, especially when the robot must quickly adapt to new information from the environment, such as the service robotics industry. This paper has shown, with ABBRA, robots can adapt to changes in the environment, make opportunistic decisions regarding new goals when they become available and handle multiple conflicting goals or time constraints simultaneously.

References

[1] B. A. Towle, and M. Nicolescu, "Fusing Multiple Sensors through Behaviors with the Distributed Architecture," in 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Salt Lake, Utah, 2010, pp. 115-120.

[2] B. A. Towle Jr, and M. Nicolescu, "Applying Dynamic Conditions to an Auction Behavior-Based Robotic Architecture," *Int'l Conf. Artificial Intelligence (ICAI'11),* vol. Volume 1, no. July 18-21, 2011, pp. 6, 2011.

[3] R. Volpe, I. Nesnas, T. Estlin *et al.*, "The CLARAty architecture for robotic autonomy," in Aerospace Conference, 2002, pp. 1.

[4] T. Estlin, R. Volpe, I. Nesnas *et al.*, "Decision-making in a robotic architecture for autonomy."

[5] J. Rosenblatt, and C. Thorpe, "Combining multiple goals in a behavior-based architecture." pp. 136-141.

[6] P. Maes, "How to do the right thing," *Connection Science*, vol. 1, no. 3, pp. 291-323, 1989.

[7] R. A. Brooks, "A ROBUST LAYERED CONTROL-SYSTEM FOR A MOBILE ROBOT," *Ieee Journal of Robotics and Automation,* vol. 2, no. 1, pp. 14-23, 1986.

[8] J. Koseck , and R. Bajcsy, "Discrete event systems for autonomous mobile agents," *Robotics and Autonomous Systems,* vol. 12, no. 3-4, pp. 187-198, 1994.

[9] R. Brooks, "Elephants don't play chess," *Robotics and autonomous systems,* vol. 6, no. 1-2, pp. 3-15, 1990.

[10] M. Proetzsch, T. Luksch, and K. Berns, "Development of complex robotic systems using the behavior-based control architecture iB2C," *Robotics and Autonomous Systems,* vol. 58, no. 1, pp. 46-67, Jan, 2010.

[11] R. Davis, and R. G. Smith, “Negotiation as a metaphor for distributed problem solving,” *Artificial intelligence*, vol. 20, no. 1, pp. 63-109, 1983.

[12] F. Brandt, W. Brauer, and G. Weiss, “Task assignment in multiagent systems based on vickrey-type auctioning and leveled commitment contracting,” *Cooperative Information Agents IV-The Future of Information Agents in Cyberspace*, pp. 11-44, 2004.

[13] P. Faratin, C. Sierra, and N. R. Jennings, “Negotiation decision functions for autonomous agents,” *Robotics and Autonomous Systems*, vol. 24, no. 3-4, pp. 159-182, 1998.

[14] B. P. Gerkey, and M. J. Mataric, “Sold!: Auction methods for multirobot coordination,” *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 758-768, 2002.

[15] B. Jennings, and Å. Arvidsson, “Co-operating market/ant based multi-agent systems for Intelligent Network load Control,” *Intelligent Agents for Telecommunication Applications*, pp. 71-71, 1999.

[16] H. Jung, M. Tambe, and S. Kulkarni, "Argumentation as distributed constraint satisfaction: Applications and results." pp. 324-331.

[17] R. Krovi, A. C. Graesser, and W. E. Pracht, “Agent behaviors in virtual negotiation environments,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 29, no. 1, pp. 15-25, 1999.

[18] M. J. Matari , G. S. Sukhatme, and E. H. Østergaard, “Multi-robot task allocation in uncertain environments,” *Autonomous Robots*, vol. 14, no. 2, pp. 255-263, 2003.

[19] R. G. Smith, “The contract net protocol: High-level communication and control in a distributed problem solver,” *Computers, IEEE Transactions on*, vol. 100, no. 12, pp. 1104-1113, 1980.

[20] K. Sycara, and D. Zeng, “Coordination of multiple intelligent software agents,” *International Journal of Cooperative Information Systems*, vol. 5, no. 2, pp. 181-212, 1996.

[21] M. P. Wellman, and P. R. Wurman, “Market-aware agents for a multiagent world,” *Robotics and Autonomous Systems*, vol. 24, no. 3-4, pp. 115-125, 1998.

[22] M. B. Dias, and A. Stentz, *Traderbots: A market-based approach for resource, role, and task allocation in multirobot coordination*: Citeseer, 2003.

[23] B. A. Towle Jr, and M. Nicolescu, “Applying Dynamic Conditions to an Auction Behavior-Based Robotic Architecture.”

[24] T. Fong, C. Thorpe, and C. Baur, “Robot, asker of questions,” *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 235-243, 2003.

[25] T. Fong, C. Thorpe, and C. Baur, “Multi-robot remote driving with collaborative control,” *Ieee Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 699-704, 2003.

[26] T. Fong, C. Thorpe, and C. Baur, *Collaborative control: a robot-centric model for vehicle teleoperation*: Carnegie Mellon University, The Robotics Institute, 2001.

[27] T. Fong, N. Cabrol, C. Thorpe *et al.*, "A personal user interface for collaborative human-robot exploration."