RoboTurk

Robocup 2011 Search and Rescue League

Team Description Paper

Hatice Kose Bagci, Ibrahim Onuralp Yigit, Semih Kavak, Yasin Kadakal

Artificial Intelligence and Robotics Laboratory
Istanbul Technical University
Computer Engineering Department
Istanbul, TURKEY

http://air.cs.itu.edu.tr/
{hatice.kose, yigiti, semih.kavak, kadakal}@itu.edu.tr

1. Introduction

In 2011, RoboTurk team will be contributing the Search and Rescue League for the first time. Our team was founded at September, 2010, and consists of undergraduate students. We are a part of the Artificial Intelligence and Robotics Laboratory (AIR Lab) at the Istanbul Technical University [1], and our team is the only team contributing to the Search and Rescue League from our university so far. We have improved our agents over sample package in Robocup Rescue Simulation Software 2010. Our group has ongoing research on dynamic planning and market based dynamic task allocation, therefore we want to use this competition as a research and test bed for our studies on these topics. We have been attending the Standard Platform League since 2001, using the robot soccer as a test bed in our research activities, and published our research in several conferences and journals [2-15].

2. Team members & Their Contributions

Our group consists of 3 undergraduate students and an Assistant professor as our team leader.

- Hatice Kose Bagci (Team Leader)

  Assist. Prof. Dr. Hatice Kose-Bagci is the team leader of RoboTurk team. She had experience in the Robocup Standard Platform League as a part of the Bogazici University's Cerberus Team (2001-2005). Her research focused on single/multi-agent localization, dynamical planning and market based task allocation. She is currently working on Human-robot interaction, mainly on assistive robots, non-verbal communication and interaction games.

- Ibrahim Onuralp Yigit (Developer)
- Semih Kavak (Developer)
- Yasin Kadakal (Developer)

3. Software Architecture

RoboTurk software architecture which is based on rescuecore2 library consists of artificial intelligent methods. The software architecture has these components:

a. Perception: Filtering environmental inputs and decrypting communication messages on voice and radio channels.

b. Agent State: Including all information about RoboTurk agent.

- ***World Model:*** Providing information of map objects and RoboTurk world model is constructed as an undirected connected graph which consists of nodes and weighted edges.
- ***Knowledge Base:*** Implementing a collection of data supports reasoning, learning and decision mechanism.
- ***Goals:*** Describing agent's primary purpose and task.

c. Reasoning: Implementing probabilistic reasoning engine based on Bayesian Network.

d. Searching: Finding shortest path using A* search algorithm.

e. Planning: Containing team strategies based on marketing algorithm approach which contributes the maximization of the overall team profit.

f. Learning: Building learning mechanism which allows the agent to operate in initially unknown environments and situations.

g. Decision: Evaluating the results coming from all RoboTurk decision supporting mechanisms and deciding what actions will be taken.

h. **Communication:** Providing communication protocol which is used by agents when sending or receiving messages.

RoboTurk Software Architecture

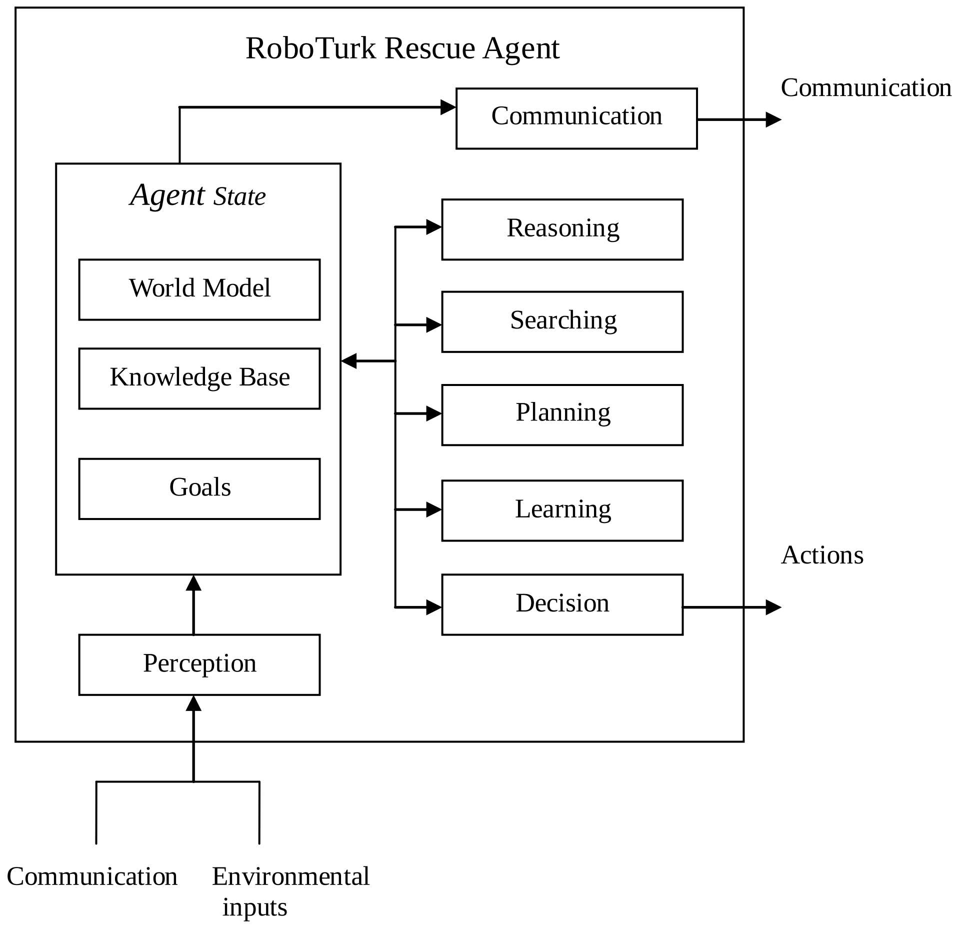

Figure 1. RoboTurk Software Architecture

4. Perception

RoboTurk agents take two inputs; one of them is environmental inputs and the other one is the data achieved by communication with other agents like police force, ambulance center etc. First of all the environmental inputs are the data which seen or heard by the agent itself. However, since an agent can achieve all kind of information, RoboTurk agent filters and processes the data coming from environmental inputs. The aim of the filtering is not achieving or storing the data which is irrelevant to the agent's goals and duties. After filtering, the data is processed to decide if agent takes an action immediately about the data or label and store the data to use in next cycles. The other kind of perception is achieved from communication between agents. In the RoboTurk communication model, the message or the data is encrypted since the channels have their bandwidth limit and noise. Because the message is first encrypted and then sent to the receiver, the receiving agent must decrypt it. A RoboTurk agent can decrypt the message received by communication using the initially arranged methods. After decrypting, the context of the message is processed to taken an action or not.

5. Agent State

In the RoboTurk Software architecture, Agent state is one of the most crucial structure. RoboTurk agent's states include a dynamic world model which provides agent's information about the current state of the map. This information consists of the data including buildings, roads and their specific attributes. RoboTurk world model is designed as a special graph to have maximum performance when searching and planning path to a target. The other part of the agent state is knowledge base. The knowledge base is the collection of the data achieved or produced by the agent to be used in reasoning, learning and decision mechanism. And final part of agent state is goals. Goals describe and bound the tasks of the agent. RoboTurk agents have certain goals but since we use market approach and reasoning, the agent can specify a goal for itself.

5.1 World Model

Every element in the simulation has its object instance. We created our own world model based on a connection graph. Our world model is constructed as an undirected connected graph consists of nodes and weighted edges. All map objects are represented by nodes of the graph. Edges of the graph have weight depending on two main factors which are the distance and the blockage between nodes. Moreover, it provides specific information of map objects. When it receives updated information from the kernel, the connection graph is reconstructed by using updated information.

5.2 Knowledge Base

In this part, RoboTurk agents' knowledge capacities are mentioned. The data resources of the agents' knowledge base are taken from the perception module as filtered and learning mechanism. The knowledge base creates a foundation for learning. Moreover, knowledge base includes information of previous actions, disasters etc. And this information is used by reasoning to calculate the probability of states may occur in next cycles. Actually knowledge base provides a rule system for agents to be considered.

5.3 Goals

RoboTurk agents have both static and dynamic goals. These goals describe agent's primary purpose and actions. Static goals of the agents are, such as saving civilian, extinguishing building and opening blocked roads. Dynamic goals of agents are produced by agent itself to maximize total profit. To achieve these goals, agents use decision supporting mechanisms such as reasoning, planning, learning etc.

6. Reasoning

RoboTurk agents use reasoning mechanism under uncertainty. RoboTurk network model is based on Bayesian network to reason under uncertainty. The network model represents the probabilistic relationships between states in the agent's world model. Moreover, the network model is a directed acyclic graph of which nodes represent state and the edges symbolize conditional dependences. Queries coming from RoboTurk agents can be answered using this network by computing posterior probability distribution. Therefore, the reasoning mechanism can not only estimate certainties, but also help agents to make a decision.

7. Searching

RoboTurk agents find the least-cost path by using A* search algorithm. The algorithm uses a best-first search and finds the best path from start to one of the goals. A* runs better performance by using heuristics. It finds solution by combining g(n) and h(n),

$$f(n) = g(n) + h(n) \tag{1}$$

g(n) gives the path cost from the start node to node n, and h(n) is the estimated cost of the cheapest path from n to the goal [16]. Therefore, we estimates f(n) that is the least-cost solution through n.

8. Planning

RoboTurk team is working on a market-based dynamic task allocation and planning approach for this year's competition. There are static goals as saving civilian, extinguishing building and opening blocked roads. But the efficiency and success of the goal changes according to the agent implementing it. Sending any agent or all agents to a problem area might cause problems since the cost and success of all agents in a situation might not be same. Therefore using a market based dynamic task/goal allocation based on the current status and cost of each agent for that task is more efficient than sending all/any agent to a problematic area. The cost function plays a very important role in the efficiency and success of the task allocation. As more features are included in the cost function, the cost function will cover the current status better. The agents can send their costs of implementing certain tasks in the current cycle via communication and then the agent with the smallest cost wins the auction and gets the opportunity to fulfill the task. Unlike the other applications of market based algorithms [12-15], here the roles are static, an “ambulance” agent is always an ambulance it cannot change its role due to the current situation and cost. This simplifies the problem to the case that the auctions can be applied to each role separately (a police agent cannot attend an ambulance auction). Also there is no need to use a central unit to allocate the tasks/goals, since each agent can listen to the channels to get the relevant costs and decide to get or leave the tasks by comparing own cost with the other similar agents. This decreases communication cost and avoid the central unit failures or single agent failures to affect the whole system. This becomes beneficial especially when dealing with physical agents in real-time systems.

9. Learning

Learning module in our software architecture uses feedback from the knowledge base and determines how decision mechanism will be modified to make a better decision in the next cycles. Reinforcement Learning can be used when the agent takes information about consequences of a sequence of its actions [12]. RoboTurk agents use Reinforcement Learning to take action in the dynamic world model. Reinforcement Learning helps the controlling the agents with experience and rewards in a framework [17]. Therefore, we expect that the agents will be enlightened how to act in the dynamic states.

10. Decision

In the RoboTurk software architecture decision layer is the last layer before an agent takes action. Decision layer is executed after the other layers because it uses the results coming from all RoboTurk decision supporting mechanisms such as searching, planning etc. While decision layer is executed, the layer evaluates the results and produces alternative actions. Then decision layer chooses the most effective one that will provide team maximum profit. After deciding the action that will be applied, decision layer sends the action to the kernel.

11. Communication

Communication is the best way of gathering information about the world for an agent. Because, without the communication an agent can only have the knowledge of its area around. However, to take the most efficient action in a situation, agent can also scan the knowledge of alternative data may lead to another action. RoboTurk agents communicate each other using random channels. This method contributes agents not to overload a channel. If all agents use same channels, the bandwidth of the channel may be passed and the some message can’t be achieved. Moreover if we assign each agent to a channel statically this may cause some messages to be lost when there is noise on that channel. Random way also contributes noise handling because since the number of the messages decreased on a channel a message can be sent more than once. To not to overpass the bandwidth limit we also designed a specific encrypted message protocol. Although the encrypted message includes all the info like sender agent type, sender agent id, etc., it needs less byte to store the message. Since the size of messages is decreased, RoboTurk agents are able to communicate efficiently.

12. Conclusion

This report summarizes the RoboTurk team efforts and strategies for the 2011 Robocup Search and Rescue League. This work is a part of an ongoing research, and our aim is to use this competition as a test bed to improve our researches on market based dynamic task allocation strategies. Also Search and Rescue League is a great opportunity for the researchers, especially undergraduate students to get use to work with artificial intelligence and robotics, and improve their ability to write good programs and be a part of a team. We would like to use this opportunity as a first and important step for the students who want to do research in robotics, and continue to take part in the Search and Rescue competitions in the following years.

References

1. AIR Lab page, 2011, http://air.cs.itu.edu.tr/
2. KOSE, H. and H. L. Akin, “A Collaborative Multi-robot Localization Method for a Team of Autonomous Mobile Robots”, ICAR 2007,13th International Conference on Advanced Robotics, Jeju, Korea, August 21–24, 2007, pp.641–646, 2007.
3. KOSE, H. and H. L. Akin, “The Reverse Monte Carlo Localization Algorithm,” Robotics and Autonomous Systems, Vol. 55, No. 6, pp. 480-489, 2007.
4. KOSE, H., , B. Celik and H. L. Akin,, “Comparison of Localization Methods for a Robot Soccer Team”, International Journal of Advanced Robotic Systems, Vol. 3, No. 4, pp.295-302, 2006.
5. KOSE, H and H. L. Akin, “A fuzzy touch to R-MCL localization algorithm”, RoboCup 2005: Robot Soccer World Cup IX, A. Bredenfeld, A. Jacoff, I. Noda, Y. Takahashi (Eds.), LNCS Vol. 4020, pp. 420 - 427, 2006.
6. KOSE, H and H. L. Akin, , “Robots From Nowhere,” RoboCup 2004: Robot Soccer World Cup VIII, Daniele Nardi, Martin Riedmiller, Claude Sammut, et al. (Eds.), LNCS, Vol. 3276, pp.594-601, 2005.
7. KOSE, H and H. L. Akin, “Experimental Analysis and Comparison of Reverse-Monte Carlo Self-Localization Method”, CLAWAR/EURON Workshop on Robots in Entertainment, Leisure and Hobby, December 2 – 4, 2004, Vienna, Austria, pp, 85–90.
8. KOSE, H. , S. Bayhan and H. L. Akin, “A Fuzzy Approach to Global Localization in the Robot Soccer Domain”, IJCI Proceedings of International XII Turkish Symposium on Artificial Intelligence and Neural Networks TAINN 2003 ISSN 1304-2386 Vol:1, No:1 pp.1-7, July 2003.
9. KOSE, H. and H. L. Akin, “SALOME: Context Dependent Blending BaSed Controller

for FootbAll PLaying AutOnomous MobilE Robots”, 2002 FIRA Robot World Congress, pp.620–625.
10. KOSE, H. and H. L. Akin, “ORCO: Bayesian Multinet Inference based Object Recognition in Robot Football Using a One-Dimensional Image,” ISCIS XVI, Sixteenth International On Computer And Information Sciences, November 5–7, 2001, Antalya, Turkey, pp.193–200.
11. KOSE, H. and H. L. Akin, “Object Recognition in Robot Fotball Using a One-Dimensional Image”, The Tenth Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN 2001), pp.291-300, 2001.
12. KOSE, H., K. Kaplan, C. Mericli, U. Tatlidede, and L. Akin, “Market-Driven Multi-Agent Collaboration in Robot Soccer Domain”, in V. Kordic, A. Lazinica and M. Merdan (Eds.), Cutting Edge Robotics, pp.407-416, pIV pro literatur Verlag, 2005.
13. KOSE, H , Tatlidede, C. Mericli, K. Kaplan and H. L. Akin, “Q-Learning based Market-Driven Multi-Agent Collaboration in Robot Soccer,” Proceedings, TAINN 2004, Turkish Symposium On Artificial Intelligence and Neural Networks, June 10-11, 2004, Izmir, Turkey, pp.219-228.
14. KOSE, H., Ç. Meriçli, K. Kaplan and H. Levent Akin, “All Bids for One and One Does for All: Market-Driven Multi-Agent Collaboration in Robot Soccer Domain”, Computer and Information Sciences-ISCIS 2003, 18th International Sympsium Antalya, Turkey, November 2003 Proceedings, LNCS 2869, pp. 529-536.
15. KOSE, H., K. Kaplan, C. Mericli and H. L. Akin, “Genetic Algorithms Based Market-Driven Multi-Agent Collaboration in the Robot-Soccer Domain”, FIRA Robot World Congress 2003, October 1 - 3, 2003, Vienna, Austria.
16. Russell, S. & Norvig, S., Artificial Intelligence: A Modern Approach (2nd ed.). pp. 97-99. New Jersey: Prentice Hall (2003)
17. Kwok, C. & Fox, D., “Reinforcement Learning for Sensing Strategies”, Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference, 28 Sept.-2 Oct. 2004.