

A FRAMEWORK FOR THE REALIZATION OF NETWORKED ROBOTS AND HUMAN BEINGS

P. Soni , Ram Kumar Rathore, Shivashankar B. Nair

Dept of Computer Science and Engineering
Indian Institute of Technology Guwahati, Assam
INDIA, PIN – 781039
{soni, r.ram, sbnair}@iitg.ernet.in

ABSTRACT

With a recent surge in robotics research, the need to develop a platform for robots and human beings to interact and co-exist has become imperative. Though advances in Internet technologies have enabled devices including robots to be plugged onto the network, a lot of hurdles still remain. In this paper we present an architecture of a social system of both robots and human beings wherein the entities co-operate over networks by sharing intelligence and building faith on each other. Faith makes the entities to interact more intensely thereby forming a social system.

KEYWORDS

Networked Robots, Information Sharing, Faith, Social System

1. INTRODUCTION

Robotics is a field that has many existing potential applications. Most often the expectations from the robots do not match current realities. The crux of the problem however, is to achieve this with real robots in real environments and in real time. In the early stages of development of networked robots, many robots were teleoperated [1, 2, 3]. The advances in Internet technology has thrown open a new era of Internet Robots [4, 5]. Internet based robots have been designed to perform some predefined tasks. The user does not have the flexibility in controlling the robot to cater to his needs. Further these robots lack facilities for reprogramming and sharing of intelligence. Data or knowledge acquired locally by the above methods cannot be used by or shared with other robots or devices on the Internet. *iRobIN* [6] provides a framework for sharing intelligence on Robin-II [5]. MANER [7] provides a routing protocol for an Ad-hoc Network of Robots. The work described in this paper is an extension *iRobIN* [6] and MANER [7] and envisages to provide intelligence more efficiently for an open architecture where all possible ways of

communication viz. human-human, human-robot and robot-robot co-exist. *iRobIN* is mainly a centralized system and a failure of its central intelligence server, designated the Intelligent Access Point (IAP), even for a small duration could result in the failure of the entire system. By introducing the concept of *faith* amongst robots connected to the network we aim to reduce the burden on the IAP by allowing robots to directly communicate with one another. The concept also makes the entire system more fault tolerant in the sense that even if the IAP is down the robots will still thrive using information shared amongst their peers. It is envisaged that the architecture explained herein may be used in diverse situations including construction scenarios. Computerized numerical controls in construction equipment could be controlled over the network and made to communicate and interact with each other more effectively to improve construction oriented performance. Robots deployed in construction scenarios can also monitor and interact with these machines as also humans to increase overall efficiency and safety.

2. COMMUNICATION ARCHITECTURE

In this paper we describe a new architecture to provide an easy way of communication between Networked robots and human Beings. The architecture as mentioned is an extension of *iRobIN* [6] framework with the concept of *faith* introduced to enhance sharing of knowledge and experience gained by individual robots. Thus doing so, it helps to provide an easy way for the creation of a co-operative human-robot society over a network. The proposed *faith* network layer leads to a faster method of sharing the learned information amongst the robots. Figure 1 depicts the overall architecture of the system.

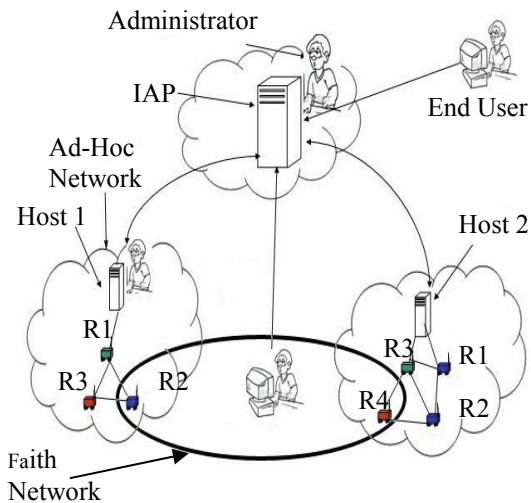


Figure 1 The Communication Architecture

The main components of the system include Robots, Human beings, Hosts (A server which act as a link between the Mobile Ad-hoc network of robots and the rest of the system.) and the IAP. Each robot can be heterogeneous in their functionalities. All of them share information hosted at the local host that controls it. The word information could mean, for instance, the various situations that the robots faced and the corresponding moves taken by the robot to overcome the situation. Knowledge sharing between robot-robot and human-robot enables a better control on the society of robots and provides for better utilization of the robots. When a robot is to be connected to the network, it has to have an ID. The host that caters to robot initially subscribes to the central server viz. the IAP. In doing so, the

IAP confers a Global Network Identifier (GNID) that identifies the host. Robots in turn have to subscribe to the host and the latter allocates a Robot Identifier (RID). Thus the robots, which form the final nodes, are identified by a GNID:RID pair within the network. Likewise human beings can also join the network by subscribing to the IAP and be identified by a Human Identifier (HID). Robots under the same host (H1) such as R1, R2, and R3 may constitute an Ad-hoc network as in MANER [7]. Communication amongst robots under a different host, for example R1 in H1 and R2 in H2, initially occurs via the IAP. We have used methodologies similar to *iRobIN* [6] to achieve this. Robots communicate with other entities in the network on-demand. Scenarios where such communication is essential include those when a robot is stranded and is unable to recover from the situation. Under such conditions it sends a request query for help via the host to the IAP. The IAP in its turn retrieves the robot configuration information (make, sensors mounted, etc.) stored at the time of registration to search for other similar robots in the network. It then contacts each of them to find possible solutions obtained from them and sends the best solution back to the requesting robot. The requesting robot selects and effects a solution. If it is able to recover from the problem at hand it builds *faith* in the robot/human being from which it obtained the solution. Faith of one robot on the other depends on the solution provided by the robot, the number of interactions with that particular entity and the frequency of communication.

2.1. IAP (Intelligent Access Point)

The Intelligent Access Point (IAP) is a remnant of the *iRobIN* [6] architecture. It acts as a central server for the entire system. The IAP stores the information about all the robots, devices and human beings. This includes configurations of the robot, information of its environment, query log for keeping track of the queries under processing and a Global Network Identifier (GNID) of each host. Equivalent information on human beings viz., name, Human Identifier (HID), area of interest, etc. are also hosted on the IAP. The IAP facilitates searching for the solutions to queries from the robots in the network. The querying robot attaches its GNID, RID pair in the query packet. The IAP

performs an exhaustive search on the network and finds all robots that have the same configuration. The search facilitates the finding of that robot which is very similar to the querying robot. The best match based on the robot configurations is found using equation 1.

$$M = \sum_{i=1}^N F_i, \text{ where } F_i = \begin{cases} 1, & \text{if } S_i = Q_i \\ 0, & \text{if } S_i \neq Q_i \end{cases} \quad (1)$$

where F_i is the match flag corresponding to i^{th} sensor on the querying robot, while M indicates the degree of match between two robots, S_i is the i^{th} sensor of the solution-providing-robot (the port, position and type), Q_i is the i^{th} sensor of the querying robot (port, position and type). The environment information is not currently included in the implementation but it may also be taken into account. Using the above search criteria, it was found that the robots forming faith have similar configurations and also imbibe similar behavior. Once the best match is found the IAP forwards the query to the corresponding hosts. The hosts in turn search in their respective local databases of the concerned robot(s) for sensor conditions identical to that posed by the querying robot. The hosts retrieve the corresponding actions taken and return the same to the IAP. The IAP then forwards the solution, GNID:RID pair and the IP address of the solution-providing robot to the host which controls the querying robot. This host stores the GNID, IP address from the solution packet and removes the IP information and forwards it to the querying robot as the solution. The IAP maintains a list of the best matches in order to enable the querying robot to ask for information from the remaining set of robots in case the solution from the first one fails. During the initial stage, the IAP acts as the central server for the whole network since all queries are forwarded to it. This could result in increase in server load. However as the system strengthens in faith building amongst the entities, the load in IAP is bound to reduce.

The IAP contains two servers as shown in the Figure 2 - a web server which is set up for catering to the human clients and another communication server for communicating with the robots.

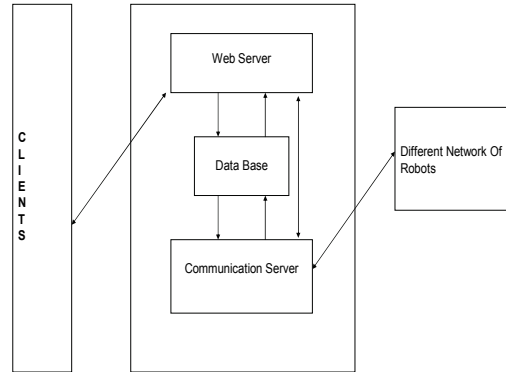


Figure 2 Internals of IAP

2.1.1. Web server

The web server provides a graphical interface to a registered human member so that they can obtain all related information about the whole system of robots and human beings. Other facilities provided by the IAP to the user via an interface include -

- 1) *Human Centric Information:* This interface shows all the members of system who are online and also helps to send messages to them. A new user can also register to the system by submitting his details to the interface.
- 2) *Robot information:* This helps the user to view all the robots, which are presently attached to the system. It also provides a simple way to program the robot remotely. The user can send a minimal number of high-level instructions to the robot. These are then interpreted at the robot host.
- 3) *Inbox:* The registered members also have an inbox facility where the user can receive messages from other entities (human beings and robots).

2.1.2. Communication server

It provides an easy way of robot-robot and human-robot interaction. The various modules within the communication server are as follows.

- 1) *Host Registration:* When host registers to the system, this function procures the IP address from the packet and issues a Global Network Identifier (GNID) to the host. The IAP stores this GNID, IP pair for future mapping of GNID to IP address.

- 2) *Procuring Robot configuration from host*: The job of this function is to procure the robot configuration and its Robot Identifier (RID) from the packet. The information is stored in a database for future use (for example searching for a particular robot with some specific property).
- 3) *Searching for solution*: This function provides a way to search for the best matched robot according to the querying robot. The input to the function is the RID, GNID pair of the querying robot and the output is a set of GNID, RID pair of robots that best match the configuration of the querying robot.

2.1.3. Database

Both servers (Web server and Communication server) use common databases. For example some of the databases used at the IAP include -

The Host table: This contains the GNID and IP address required for mapping of GNID to IP address. The entry in this table is stored when the host registers at the IAP and receives a GNID from the IAP.

The Human table: It contains all relevant information on the human beings. This includes personal information (Like name, age, sex), contact information, information on whether the human being is logged in or not, an alternative device number (for example PDA) and whether or not the human being is a deployer of a robot, etc.

The Query Log table: This contains the information of a particular global query; for instance the RID: GNID pair of the querying robot and the status information of the query. It is used to trace a robot in the network and find whether or not it is replying to a request for a solution.

2.2. Host

The host acts as a link between the robots and the IAP. If a deployer wishes to setup an ad-hoc network then its host has to go through the process of registration to get a global network identifier (GNID). The host stores information about the robots under its jurisdiction (eg. robot identifier issued to them at the time of registration process), solutions to the problems faced by the robot(s), and the faith built up by the robots on others.

2.3. Robots

The robots form the nodes of the network and are under the jurisdiction of a fixed host. It stores its Robot Identifier (RID) issued during the registration process. The robot also holds the GNID of the host to which it remains attached. Thus the RID: GNID pair makes the robot uniquely identifiable in the social system. It uses this pair as an identifier for further communication to other entities in the social system. The robot also stores the information on the possible moves taken to overcome an unexpected situation and the faith information on other robots. As the size of the memory on board a robot is limited, such information is periodically transferred to the respective host.

The robot can generate three types of queries Local Query (LQ), Global Query (GQ) and Faith Query (FQ). All these queries have dynamically varying response times. The Host, which controls the robot, keeps track of the average response time. Based on this the robot autonomously decides which of the queries should be used to procure assistance and in what order.

The various functions that are hosted on robot are given below:

- 1) *Generate Local Query*: The function takes the sensor values as the input parameters and generates a local Query for the robot. The local Query is forwarded among all the robots in the ad-hoc network to which querying robot belongs.
- 2) *Generate Global Query*: The function generates a global query for the robot by taking its sensor values. This Query is forwarded to its Host, which in turn forwards it to the IAP.
- 3) *Generate Faith Query*: This function also takes sensor values as the input parameters and generates a faith query for the robot. The function finds the best robot (one on which it has highest faith) from the faith table and forwards the query to its host with the robot's GNID: RID pair. The host in turn forwards it to the corresponding host under which the highest-faith robot is attached.
- 4) *Decide the Order of Query*: Robots maintain the average faith query, global query and

faith query response times as three state variables. With each query, a robot also generates an expected time indicating the amount of time it can wait to procure a solution. This function takes the parameter - *Expected time* - as input and finds the possible queries that a robot can ask. If more than one query is possible, the function can decide the order in which the query should be asked so as to minimize the response time.

3. COMMUNICATION SCENARIOS

Communications amongst the entities of the system could occur in various ways some of which have been highlighted below.

3.1. Robot to Robot

To illustrate the need for communication we use a simple problem of robots, R3 and R4 under Host 1, (as in figure 1) trying to find green spots on the floor. Another robot, R2, under a different host (Host 2) requires to be stationed on green spots to gather energy. If the robot R2 under Host 2 is unable to find the same after repeated trials in its environment it generates a query in one of the following ways:

Case 1 Global Broadcast Query

If the robot is not able to get a solution from local query (LQ), R2 generates a Global query and sends it over to the IAP. On receiving it, the IAP which hosts information on all robots, identifies all those robots which closely match/resemble the configuration and environment information of the querying robot and sends the query to each of them. The robots in their turn respond if they have a solution to the problem. The IAP consolidates, ranks all solutions and sends the best solution to the robot via its host. The querying robot tries to execute the solution and in doing so may or may not recover from the problem at hand. If it does, it informs its host which builds on the faith of this robot on the one that supplied the solution. The robot tries other solutions, if it does not succeed. If R3 and R4 under Host 1 have already discovered ways to find the green spots, it is highly possible that R2 of Host 2 will get a solution from either of them. In this way R2 manages to accomplish its task (even though it has not been programmed for that particular event) by requesting for aid from other robots. If all solutions prove futile it tries a

random move or assumes it is stuck and contacts its human master.

Case 2 Faith Query

The building of faith amongst the entities transforms the network into a social one. Whenever a robot obtains a correct solution from either of its siblings (LQ) or from its cousin entities (GQ) it builds a faith on the entity that provided the solution. This faith is maintained in the form of counters for each robot in their respective hosts. These counters are increased or decreased based on whether the solution yielded a good or a bad result. The following equations were used to test the generation of faith among the entities in the network. The value of faith is governed by Equation 2 [8].

$$F_{ijNew} = F_{ijOld} + (R_i^j - P_i^j) / \sum_{k=1}^N I_{ik} \quad (2)$$

where F_{ij} is the faith built so far of robot i on robot j . R_i^j and P_i^j indicate rewards and penalties obtained by robot i after an interaction with robot j . I_{ik} indicates the total number of interactions that robot i had with robot k . A robot i is said to have robot j in its virtual faith network if and only if

$$F_{ij} > \mu \quad (3)$$

where μ is the upper threshold of faith. The faith value is reduced after a timeout period proportional to a Time-To-Live (TTL) governed by the equation 4.

$$TTL_{New} = TTL_{old} + c_1 R_i^j - c_2 P_i^j \quad (4)$$

where c_1 and c_2 are constants. The generation of faith prevents the robot from making a global query. Instead it directly contacts the robot on whom it has built the faith thereby (hopefully) gaining a solution in lesser time. Such queries have been termed as Faith Queries (FQ). These queries naturally consume far less time than their global counterparts.

3.2. Human to Robot

A human user may also participate in being an entity by the interface provided by the web server

hosted at the IAP. He/she can send commands either to control a robot or get information on its status. A set of commands constituting a small program may also be sent.

3.2. Robot to Human

When a robot wishes to interact with a human being or its deployer to get some assistance, it can request the IAP to send a message to the specified person. The IAP forwards this message to the deployer's *inbox*, who in turn can check it as and when he is online. Such a scenario can occur when a robot does not get a feasible solution from any robotic entity on the network.

4. APPLICATION TO CONSTRUCTION SCENARIOS

Large construction sites require constant interactions between the people involved as also the monitoring of the machinery used. Robots forming the network described herein could be appropriately programmed to monitor precarious construction scenarios. Material handling as well as requirements at various sections of the site could be monitored and managed using such networked robots. The need for a particular material (for instance bricks, tiles, cement, etc.) could be identified by the robot on the job and relayed to other entities (robots and/or humans) in the network. Scenarios where a demolition is to take place could require robots to operate in co-operation to find the strength of a particular region which could otherwise be dangerous for human beings. Sensory probes mounted on robots could find the relevant parameters and relay them through other robots in the ad hoc network to reach the human user. Since the networked robots described herein can be programmed to co-operate and learn, it is possible for a human user to tailor them to suit the needs of the construction scenario. A large task force of robots comprising the nodes of the network can prove to be a vital aid to humans working in a construction scenario. Since the humans also comprise of nodes in the network, the degree of interaction with the robots within the person's vicinity increases, effectively enhancing throughput.

4. CONCLUSIONS

In this paper we have introduced a framework that can facilitate an easy communication between networked robots and human beings. Such systems can sustain only if proper communication channels are thrown open between every entity that constitutes the network. It was found that the IAP acts as a central server only initially. As inter-entity faiths are strengthened with the passage of time, the IAP seemed to play a lesser role in communication. With a decrease in the server load new entities can get their requests processed at a faster rate. As part of the future work, we are intend to incorporate more features to facilitate efficient remote programming methodologies as also adding devices as entities of the network.

5. REFERENCES

- [1] Goldberg, K. & Mascha M (1995) Desktop teleoperation via the World Wide Web, Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 21-27 May, Vol. 1, pp- 654-659.
- [2] Michel, O.; Saucy, P. & Mondada, F. (1997) Khep-on the Web : An Experimental Demonstrator in Telerobotics and Virtual Reality, Proceedings of the 1997 International Conference on Virtual Systems and MultiMedia, IEEE Computer Society, Switzerland, 10-12 September, pp. 90.
- [3] Goldberg, Ken., Chen, B., Solomon, R., Bui, S., Farzin, B., Heitler, J., Poon, D. & Smith, G. (2000) Collaborative Teleoperation via the Internet, Proceedings of the IEEE International Conference on Robotics & Automation, San Francisco, USA, 24-28 April, pp. 2019-2024.
- [4] Schulz, D., Burgardz, W. & Cremers, A.B. (1999) Robust Visualization of Navigation Experiments with Mobile Robots over the Internet, Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems, Kyongju, Korea, 17-21 October, pp. 942-947.

- [5] Badapanda, R.P., Nair, S.B. & Kim, D.H. (2004) A Framework for Rapid Deployment of Devices and Robots on a Network, Proceedings of the 1st International Computer Engineering Conference, Cairo, Egypt, 27-31 December, pp. 625-629.
- [6] Toppo, N.K. & Nair S.B. (2005) A Framework for Sharing Intelligence Among Mobile Robots on a Network, Proceedings of the 2nd IASTED International Multi-conference On Automation, Control, And Information Technology on Network Based Intelligence, Novosibirsk, Russia, 20-24 June, pp. 93-98.
- [7] Nair S. B., Pradeep Kumar K. V. D. & Saravanan M. (2006) A Communication Protocol for a Mobile Adhoc Network of Robots, Proceedings of the International Conference on Emerging Applications of IT (EAIT 2006), Published by Elsevier, Science City, Kolkata, India, 10-11 February, pp. 223-226.
- [8] Dutta Baruah R., Rathore R.K., Soni. P, Nair S.B. (2007), RoboSapienNet - Towards Building A Social Network of Human Beings and Robots”, The Fourth IEEE and IFIP International Conference on wireless and Optical Communications Networks, WOCN 2007, Singapore, 2-4 July, 2007, (Accepted).