A HARMS-based Heterogeneous Human-Robot Team for a Gathering and Collection Function

Abstract—Agriculture production is a critical task in all parts of the world. The process to grow and harvest crops is very human labor intensive in many parts of the world. Much of the difficult labor of crop production can be automated with intelligent and robotic platforms. We propose an intelligent, agent-oriented robotic team which can enable the process of harvesting, gatherin and collecting crops and fruits, of many types, from agricultural fields. This paper describes a novel robotic organization enabling humans, robots and agents to work together for automation of gathering and collection functions. The focus of the research is a model, called HARMS which can enable humans, software agents, robots, machines and sensors to work together indistinguishably. With this model, any capability-based human-like organization can be considered and modeled, such as manufacturing or agriculture. In this work we model, design and implement an application of knowledge-based robot-to-robot and human-to-human collaboration for an agricultural gathering and collection function. The gathering and collection functions were chosen as they are some of the most labor intensive and least automated processes in the process acquisition of agricultural products. The use of robotic organizations to can reduce human labor and increase efficiency allowing people to focus on higher level tasks and minimizing the back breaking tasks of agricultural production, in the future. In this work, the HARMS model was applied to three different robotic instances and an integrated test was completed with satisfactory results that show the basic promise of this research.

I. INTRODUCTION

Basic fruits are an important crop utilized and depended on as a staple food in the diet of many people. This dependence spans many parts of the world and is of particular importance in many parts of Asia. The process to harvest fruits and some vegetables is not currently as automated as are many other dependent staple crops across the world such as wheat, corn or sorghum. These crops are produced in large fields with efficient techniques, large machinery and mass production. Unlike these crops, the production of fruits and other gathered crops remains a very human labor intensive practice. Given the statement of belief by the IEEE Robotics and Automation Society [15], the automation of agriculture can be enhanced with robotic, and therefore agent-oriented systems.

Agriculture is humankinds oldest and still its most important economic activity, providing the food, feed, fiber, and fuel necessary for our survival. With the global population expected to reach 9 billion by 2050, agricultural production must double if it is to meet the increasing demands for food and bio-energy. Given limited land, water and labor resources, it is estimated that the efficiency of agricultural productivity must increase by 25% to meet that goal, while limiting the growing pressure that agriculture puts on the environment. Robotics and automation can play a significant role in society meeting 2050 agricultural production needs. For six decades robots have played a fundamental role in increasing the efficiency and reducing the cost of industrial production and products.

Much of the difficult labor of fruit production can be automated with intelligent and robotic platforms. We propose an intelligent agent architecture which can use sensors to automate the process of gather and collect crops, so that a robot can eliminate labor in the fields. This problem and need for robots is as much of an economic issue as it is a labor or agricultural issue. A short supply of labor can seriously hinder crop production and create massive ripple effects in the food supply.

This agent system will be used to control a platform, such as Bonirob as shown in Fig. 1 the Bonirob [4] is an example of a row crop robotic platform that can be utilized in tillage and general crop production.

As the human drive to speak directly to robots [17] increases, and becomes more common place in some configuration of natural language communication, it is equally important to speak with other classes of cyber-physical systems, as shown in [10]. In the near future, ubiquity in this technological area will provide communication with a range of devices in which humans interact on a daily basis. The foundation is set by Kim et al [12] by defining the need for ubiquity in robotics. A ubiquitous future leans to the notion that we can give commands or request services [14] to any range of cyber-physical systems capable of accomplishing a specific task without regard to the physiological or cognitive definition of
the system. In terms of multiagent, or artificial, organizations, there is typically a goal to accomplish. If an actor in the organization announces a task to all other actors, the first actor only cares that the task is accomplished, but not necessarily concerned with who carries out execution, within reason. The actor who executes the task must be capable of executing the task to accomplish the goal. There may be many actors, in the organization, with this capability and these actors may all be different, in terms of class, embodiment, mobility and physiology. For example, a person can ask for someone to get a morning newspaper from the lawn. A human, dog or robot can accomplish this task, as all have capability. If the requester can give a general command that each can understand then any can execute this task to satisfy the goal of getting the paper. Thus, given that all can understand the command, it is a request ubiquitous to all actors but indistinguishable who must execute the task, given that all are equally or necessarily capable.

This research is an extension of a previous work [11]. This remainder of this paper is organized with section 2 showing the HARMS model to enable work in large cyber-physical organizations. Section 3 shows the realization of a robotic system to implement and allow testing of the model. Section 4 provides some initial results of the system and section 5 and 6, respectively, provide conclusions and planned future extensions to this research work.

II. HARMS LAYERED MODEL

As robots become more pervasive and ubiquitous in agriculture, they become increasingly involved in the lives of humans. Farmers and those involved in production agriculture expect robots will take on tasks to ease their lives, by working with humans just as other humans do, in normal organizations and teams. This labor specialization, by ubiquitous robots, allows humans more comfort, time or focus to concentrate on higher level desires or tasks. To further this unification of relationships, the defined line between humans and other robots must become somewhat indistinguishable. This ever increasing degree of indistinguishability provides that we care less about who or what executes a task or solves a goal, as long as that entity is capable and available. In this section, we propose an on-going developing model and a simple example implementation which minimizes the strict line between humans, software agents, robots, machines and sensors (HARMS) and reduces the distinguishability between these actors, which can be applied to many task domains, specifically gathering and collection.

The development of an organization which supports indistinguishability within it members requires a model definition enabling these actors to connect, communicate and interact. Secondly, the model must support actors of many different cyber-physical definitions. In this research, we have defined a model to connect humans, software agents, robots, machines and sensors HARMS Model where each layer of the model, previously introduced [2] [1] integrates with the layers above and below it, as shown in Fig. 2.

![HARMS Layered Model Concept](image)

Each of the layers in Fig. 2 is connected to the layers immediately below or above it in the model. Layers higher in the model depend on the lower layers function and service. For example, the ability of an agent to communicate depends on its ability to network with another agent. The layers are presented from the lowest layer to the highest highest. The level does not represent the level of abstraction in the model.

A. Network

The Network layer represents the basic communication between the system actors. Each system actor must have basic capabilities to connect to other actors. In this case, via actors will connect via a wired or wireless network to any other system actor. In this model, networking represents the physical connection between actors. The actors have the capability to communicate via sending TCP/UDP messages using unicast, multicast or broadcast, depending on the message type and set of actors the communication is directed towards.

B. Communication

The Communication layer enables the basic common exchange capability between any systems actors. Communication is defined by elements such as meaning, syntax, protocols and semantics [16]. This layer is modeled in a generic sense to allow any model of actor to communicate in many ways, including natural language, gestures, simple text and many other possibilities. This layer enables any n actors to communicate via a standardized interface and is dependent on the ability of actors to network.

C. Interaction

The Interaction layer represents a set of common, well understood algorithms and techniques which provide a layer
for group rational decision making with implied intelligence by a set of actors, such as voting, auctions and also some new models, such as hierarchical decisions. Common economically driven and market-based algorithms provide a basis for this layer typical to the description in Weiss [5]. In this system, there is a collective, cooperative interest in which there exists negotiation and bargaining between agents for decision making. This layers depends on the actors ability to effectively communicate from the layer 2 functionality.

D. Organization

The Organization layer uses multiagent systems organization models such as OMACS[3], for example, enabled by networking, communication and interaction services provided by the HARMS Model’s three lowest levels. The organization layer provides for the needed group rational decision making required to organized based on capability around a set of common organizational goals. Each of the actors possess specific capabilities required to play a role in an organization. Each role can work to solve a set of 100N goals. The overall set of organization goals can be accomplished by the aggregate set of roles, played by the diverse actors, available to be active in the organization.

E. Collective Intelligence

The Collective Intelligence behavior in a collection of agents, robots and humans can lean in a number of different directions. In this case, we focus on collective organizations with emergent and planned behavior. Examples are the societal or organizational norms which exist in a collective [7] [8] or models of social agreement in agent societies [9]. The collective intelligence will not only allow emergent behaviors, but also the connection of multiple organizations into higher-level collectives such as societies or organizations, and potentially a definition of consciousness [13].

F. Indistinguishability

The concept of indistinguishability is not a layer in the HARMS Model, but is a concept the HARMS Model will enable to achieve. Indistinguishability enables a system to choose between n different options of minimally capable actors relative to some task or goal. If there are a number of heterogeneous actors, each with the capability, the selection is not dependent on a specific embodiment, physiology or cognitive design. Capability to solve the goal is the only distinguishing factor between indistinguishable actors, in a system.

III. HARMS COMMUNICATION

The HARMS model [1] assumes communication in a semiformalized approach to language format between all actors. To appear indistinguishable, the actors will communicate in English, or another natural language subset, limited lexically for a restricted domain, but fully conforming to the rules of the specific morphology, syntax and semantics. The actors will exchange messages in terms of questions, directives and information messages. Given the actors can send messages via unicast u, multicast m or broadcast b, they can send from actor_a to any actor_b . . . actor_∞. The set of actors Act is defined by Act = {H, Ag, Ro, M, S} where H is a set of humans, Ag is a set of agents, Ro is a set of robots, M is a set of machines and S is a set of sensors.

There are 3 basic communication functions in the system; questions, messages and directives. An example of a common command is:

drive forward then stop and retrieve apple

Realistically, there are three commands here, so it will be broken down into a set of sequential commands. The next three subsections describe the basic structure of questions, messages and directives.

A. Questions

Questions send a message msg to a group of \{n actors|actors ∈ Act, n ∈ ℝ\} and return a msg back to actor_x, the inquiring agent. Content of the functions is dependent on the domain structure and problem.

\[
msg_x ← \text{question}_{actor_x}(actor_y, msg_y, u) \quad (1)
\]

Each question function has unique parameters, based on its problem and audience. Equation 1 represents actor_x asking a message msg of only actor_y as the third parameter u represents a cast function, which can be unicast, multicast or broadcast.

\[
\{msg_x, y|x, y ∈ Act\} ← \text{question}_{actor_x}(\{actor_y_1 . . . actor_y_n\}, msg_y, m) \quad (2)
\]

Equation 2 represents actor_x asking a question message msg of to all actors in the set \{actor_y_1 . . . actor_y_n\} as the third parameter m represents a multicast function, which targets an inquiry to a select group of actors. The function returns a set of messages from each actor.

\[
\{msg_x, y|x, y ∈ Act\} ← \text{question}_{actor_x}(msg_y, b) \quad (3)
\]

Equation 3 represents actor_x asking a question message msg of to all actors in the organization as the third parameter b represents a broadcast function, which targets a question to a select group of actors. The function returns a message from each actor.

B. Messages

The Message function sends a message to a group of n actors, without any return. Equation 4 represents actor_x sending a message to a single actor_y.

\[
\text{Message}_{actor_x}(actor_y, msg_y, u) \quad (4)
\]

Equation 5 represents actor_x sending a message to all actors in the set \{actor_y_1 . . . actor_y_n\}.

\[
\text{Message}_{actor_x}(\{actor_y_1 . . . actor_y_n\}, msg_y, m) \quad (5)
\]

Equation 6 represents actor_x sending a message to all actors.

\[
\text{Message}_{actor_x}(msg_y, b) \quad (6)
\]
C. Directives

The *Directive* function sends a command to a group of *n actors*, without any return. Equation 7 represents *actor* \(_x\) sending a message to a single *actor* \(_y\). The basic assumption is in a cooperative system, all agents will obey and honor the command.

\[
\text{Directive}_{\text{actor}_x}(\text{actor}_y, \text{msg}_y, u) \quad (7)
\]

Equation 8 represents *actor* \(_x\) sending an directive command to all actors in the set \{*actor* \(_y_1\), \ldots *actor* \(_y_n\)\}.

\[
\text{Directive}_{\text{actor}_x}(\{\text{actor}_y_1, \ldots \text{actor}_y_n\}, \text{msg}_y, m) \quad (8)
\]

Equation 9 represents *actor* \(_x\) sending a directive command to all actors.

\[
\text{Directive}_{\text{actor}_x}(\text{msg}_y, b) \quad (9)
\]

D. Implementing a Directive

The example of a common command directive is *drive forward then stop and retrieve apple*. This directive is compound as there are three commands here to drive, stop and retrieve. Also, two directives have a sub-directive, forward and apple, respectively. For example, we can send this directive to a single actor, a group of actors or all actors, in a system.

For a single directive, the robot in question will be named George from the source actor Bob. The message sequence will be:

\[
\text{Directive}_{\text{actor}_B\text{ob}}(\text{actor}_\text{George}, \text{msg}_\text{drive forward}, u)
\]

then

\[
\text{Directive}_{\text{actor}_B\text{ob}}(\text{actor}_\text{George}, \text{msg}_\text{stop}, u)
\]

and finally

\[
\text{Directive}_{\text{actor}_B\text{ob}}(\text{actor}_\text{George}, \text{msg}_\text{retrieve apple}, u)
\]

IV. REALIZATION

To realize this HARMS oriented project, we used a field with different colored balls to represent fruits on the ground. In this case, the robots gathered ground based fruits. The ultimate and final goal is automatically to pick up all of balls, which simulate agricultural products and move them to the human, which is the collection point. The human is also the initiator in this example. To show this three heterogeneous robots are utilized; Darwin, an humanoid, iRobot Create, and bulldozer robot, which are both UGV’s. For all communication at this project we use Zigbee connection and IEEE 802.11 wifi.

A. Goals

Our first goal is to make Darwin can find the ball and then pick it like Fig. 3. And drop the ball into a basket on top of the iRobot Create, which is the transporter, or the basket which is located at designated end spot.

Our second goal is to control automatically the iRobot create in order to carry balls from Darwin to a human. When Darwin want to use this, it can be used for Darwin or human need this, and then it will help human as a carrier. The final goal is to enable the bulldozer to gather and assemble balls at the designated spot and go back to the original starting location. As the robots go through these steps to goal completion they will network together and communicate using a HARMS model instance in each robot. This will enable the cooperative work. Fig. 4 shows the technical flow chart of activities for the ball picking system as it goes through accomplishment of the system goals.

B. Approach

To start the system, the human gives an order to the robot team to collect all of balls and bring the to the collection point. The bulldozer robot begins to assemble all of balls at the designated spot. When the task is completed, the bulldozer robot sends a message, “I finish the task” to all others. Then the robot moves back to its origin point.

Then, the Darwin initiates motion and vision to to find the a ball as shown as Fig. 5. Once Darwin finds the ball, he sends a message to the iRobot to follow me to let iRobot receive the ball from Darwin. When Darwin retrieves the ball, he then locates the iRobot to drop it into the moving basket(iRobot). If there is no basket in Darwin’s range of view, he will rotate until he finds the iRobot.

The picking-up capability is to take the ball up with Darwins gripper as shown in Fig. 6. For this function, we built and installed two custom boxes on each arm end. This forms a box that allows scooping the ball and ease of picking without damaging the source of the ball (or fruit).

Then, Darwin conveys the ball to a ball basket installed at iRobot. To do this step, the Darwin sends a message to iRobot to call and position next to Darwin. iRobot will receive that message follow me) from Darwin and then he will start to move to find the Darwin and calculate the distance and angle between Darwin and iRobot to go near Darwin as shown in Fig. 7. When the Darwin approaches to place the ball,
the moving basket(iRobot) will wait the ball is fully dropped off into his basket. When the set number of balls have been collected, the iRobot will move to the initial collection point, co-located with the human. By using this approach the balls, randomly distributed on the floor, can be collected quickly and efficiently.

V. EXPERIMENTS AND EVALUATION

Testing was done on the controlled environment, restricting the size of the field to 220cmX200cm. We had 4 experiments in same initial state but only differs the number of robots. Every experiment has same initial state as shown Fig. 8. Each experiment has 3 red balls in a row and put a yellow basket to the top right-hand corner. The goal of the experiments is to put all balls to the yellow basket.

In the test environment, there are a total 4 agents, human, Darwin, bulldozer robot and iRobot. Each robot has its positives and negatives, as shown in Table III, and that characteristic affects to HARMS multi-robot system. It enables the HARMS system efficiency by providing suitable positions for each robot. We attached the yellow basket to iRobot in order to enable moving basket because of consistent movement. In each experiment, Darwin picks a set of balls, the bulldozer robot collects the balls to the end point, human controls

Fig. 4. HARMS Flow chart for the ball picking system

Fig. 5. DARwIn vision

Fig. 6. Ball picking DARwIn
iRobot, so iRobot waits for Darwin until Darwin drop a ball. Wireless network is connected to the Zigbee, sending only simple commands, in broadcast format in TCP. The experiment is started when the Darwin receives the initial Zigbee signal. Each experiment is conducted 5 times.

A. Results

4 different experiments performed (only Darwin, Darwin with Bulldozer robot, Darwin with iRobot and Darwin with Bulldozer and iRobot), while experiment 4 was not successfully completed. So, overall system shows that the closer organization robots, the faster the work can be done, relieving the human of labor.

Experiment 1 took 119.8 seconds in average to carry 3 balls, and it showed that the Darwin itself is hard to achieve great performance because of its weakness, slow walking, inaccurate vision processing, etc. In experiment 2, we added the bulldozer robot to complement of the Darwin. According to table 2, it communicates with the Darwin and it collects three balls to one place, so it reduces the total elapsed time to shorten the Darwins walking distance. Therefore, experiment 2 took 106.2 second in average which has better result than experiment 1. In experiment 3, we used Darwin and iRobot and iRobot runs after Darwin has noticed a ball. It follows the Darwin with ball with the basket, so Darwin does not have to travel to the basket. iRobot following function saves Darwins walking time, so experiment 3 also has better result (104.8 sec) than experiment 1.

In experiment 4, we used all agents to maximize the efficiency of HARMS system. So, the human sends a signal to Darwin, then Darwin starts to move. The bulldozer starts to collect all balls after 2 seconds. When Darwin sees the ball, the iRobot starts to follow the Darwin. In this scenario, the HARMS system minimizes Darwins slow walking time, so it maximizes the efficiency. The result of experiment 4, taken the shortest time, shows the best use and capability of this system. Each robot has somewhat maximized their advantages and minimized their shortcomings, and it finally maximizes the relative efficiency.

VI. CONCLUSIONS

The goal of this system is to create a prototype that uses HARMS to demonstrate a multi-actor system integrated to save labor. Overall, while it is a small implementation and test, it shows the basic capability of the system to enable a team to perform a function. Given the results, it also shows that the time can be decreased as the actors capabilities are more efficiently utilized. Showing a trend to greatly utilize capability of non-human actors in a agricultural collection, shows basic economic promise to employ robots, in this future function.

REFERENCES


TABLE I. POSITIVES AND NEGATIVE ATTRIBUTES OF EACH ACTOR

<table>
<thead>
<tr>
<th>Robot</th>
<th>Positives</th>
<th>Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARwIn</td>
<td>Can pick up the ball</td>
<td>Slow walking</td>
</tr>
<tr>
<td></td>
<td>Find the basket</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Drop the ball into the basket</td>
<td></td>
</tr>
<tr>
<td>Bulldozer</td>
<td>Control multiple balls</td>
<td>Cannot pick up a ball</td>
</tr>
<tr>
<td>iRobot</td>
<td>Create Fast movement</td>
<td>Cannot pick up a ball</td>
</tr>
</tbody>
</table>

TABLE II. TOTAL ELAPSED TIME OF BALL PICKING AS A RESULT OF DIFFERENTIAL ACTOR COMBINATIONS

<table>
<thead>
<tr>
<th>Experiment 1</th>
<th>Experiment 2</th>
<th>Experiment 3</th>
<th>Experiment 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial 1</td>
<td>124 sec</td>
<td>103 sec</td>
<td>114 sec</td>
</tr>
<tr>
<td>Trial 2</td>
<td>124 sec</td>
<td>112 sec</td>
<td>110 sec</td>
</tr>
<tr>
<td>Trial 3</td>
<td>113 sec</td>
<td>110 sec</td>
<td>101 sec</td>
</tr>
<tr>
<td>Trial 4</td>
<td>119 sec</td>
<td>104 sec</td>
<td>101 sec</td>
</tr>
<tr>
<td>Trial 5</td>
<td>119 sec</td>
<td>102 sec</td>
<td>98 sec</td>
</tr>
<tr>
<td>Average</td>
<td>119.8 sec</td>
<td>106.2 sec</td>
<td>104.8 sec</td>
</tr>
</tbody>
</table>

TABLE III. POSITIVES AND NEGATIVE ATTRIBUTES OF EACH ACTOR

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Human, DARwIn DARwIn moves only when human sends a signal to start OOOO</td>
</tr>
<tr>
<td>2</td>
<td>DARwIn, Bulldozer Bulldozer moves 2 secs after DARwIn has started to move XOXO</td>
</tr>
<tr>
<td>3</td>
<td>DARwIn, iRobot iRobot moves only when DARwIn has noticed a ball XXOO</td>
</tr>
<tr>
<td>4</td>
<td>Human, iRobot Human controls iRobot XXOO</td>
</tr>
</tbody>
</table>


