Master Thesis
Computer Science
Thesis no: MCS-2006-17
Jan 2007

Exploring the Implementation of Complex Appearances on Small Robots

Jeffrey Knurek

Department of Intelligent Software Systems
School of Engineering
Blekinge Institute of Technology
Box 520
SE 372 25 Ronneby
Sweden

This thesis is submitted to the Department of Intelligent Software Systems, School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author:
Jeffrey Knurek
E-mail: jknurek@yahoo.com

External advisor(s):
Mattias Jacobsson
Viktoria Institute
Address: Hrselgngen 4, 2nd floor, Gteborg, 417 56, Sweden
Fax: +46 31 772 89 63

University advisor(s):
Stefan Johansson
Department of Intelligent Software Systems

Department of Intelligent Software Systems
Blekinge Institute of Technology
Box 520
SE 372 25 Ronneby
Sweden

Internet: www.bth.se/tek
Phone: +46 457 38 50 00
Fax: + 46 457 102 45

ABSTRACT

Abstract

The purpose of this project is to explore how autonomous robots could develop a language to communicate visual patterns. The idea is that each robot should be able to change their visual appearance depending on what its neighbors are trying to communicate. Thus, the robots should talk about their patterns, trying to influence each other. For this project we used the e-puck robot, a small mobile robot developed by EPFL (Ecole Polytechnique Fédérale de Lausanne) in conjunction with the see-puck display. The display consists of a matrix of 148 LEDs in a circular shape. We looked into several methods of achieving communication though the sensors and actuators of the e-puck robot. An additional area which was explored is the process user interaction with the robots.

Keywords: Mobile Robots, Complex Patterns, Agent Communication.

Acknowledgments

Though its not in the scope of this work I find it fascinating that we as humans are simply a highly evolved embodied and communicating agent. I can only wonder if the e-pucks that were turned on and off during the course of this project could have reached the same feelings of “self-worth”.

Thanks. Thanks to the friends and family that helped me get to where I am today.

Contents

1 Introduction

1.1 Preface

For decades there have been plenty of broad expectations for the functionality of robots. In recent years, there has been work on determining how robot applications can affect peoples everyday lives and habits. One example of such work involved creating personas to illustrate possible relations between a person and his or her personal robot [1]. An example of one scenario developed during this work was based on a type of agent that communicates through colors. These agents would communicate to each other and then change their color patterns. Each agent would begin with a specific individual pattern, and when put closely to another agent, they would each start to change their individual patterns. Surrounding light, sounds, and movement would also affect an agents pattern.

The work presented in this paper will be an extension of the scenario described. The remaining parts of the Introduction will cover the research area and how we approached it, along with the goal of our project. We also provide a review of similar work that has been performed. In the later sections of this paper, we will cover the scope of the project and theories proposed. We will then step through the methods used and their results as we discuss each of the techniques we explored. We finish the paper with a conclusion of our work.

1.2 The Research Purpose and Questions

The purpose of this study is to explore how autonomous agents could develop a language to communicate visual patterns. The idea is that each robot should be able to change their visual appearance depending on what the neighbors are trying to communicate. Thus the robots should talk about their patterns, trying to influence each other.
We feel that exploring this issue has importance within three different areas. From a scientific point of view, we hope to provide more insight into the manner in which communication has emerged in natural species. From a technological point of view, there is the possibility of defining new methods or even improving existing methods for handling communication between mobile robots. And lastly, we foresee that the pattern creation and sharing of our robots might have an affect on the artistic community.
With our method being an exploration of the area described, we have formulated questions that this research may be able to answer. They are as follows:

- What method of agent to agent communication is most effective? (both for communicating information about the external environment, and for sharing information on pattern creation)
- How can robot driven movement be used to assist with this communication?
- What kind of patterns produce interesting results to a user?
- How can the user inform the robot when an interesting pattern is created?
- How does the robot evolve as it receives new information?

1.3 Goals

The aim of the project is the development of agents that are able to interact directly with the physical world and to communicate between themselves and with other agents. The final outcome from this research should be a programed robot that it is capable of developing communication with other robots in its environment. The user should be able to acknowledge this communication through the patterns drawn upon the robots' display. Our objective is to explore the capabilities of the platform used and how it might be implemented to achieve our goal. As is typical when working with complex systems, we began with an open-ending concept which we wished to achieve and explored the different ways in getting there.

1.4 Review of similar work

From the inception of mobile robots, programming their controls followed a paradigm of sense-plan-act (SPA) [2]. This approach was computational intensive in its process of collecting everything from its sensors; then calculating a plan based on the world view created; and finally performing the action of the plan. One of the greatest drawbacks to this approach is that when it comes time for the robot to act, the environment may have already changed.
A new architecture for controlling mobile robots came in 1986. The concept would become broad and take on many different names, including behavioral-based, but the architecture is based on Brooks' subsumption design [3]. The concept suggests that there are different layers of control that influence an action by the robot. Lower level layers will continue to run while higher level layers have the option to override and take control of the action being performed. An example of this would be that a low level layer of a simple mobile robot would consist of the action of moving forward. A higher layer might involve sensors that detect obstacles in front of the robot and turn it in another direction. So by following the behavioral-based approach the robot would continue moving forward until the sensor layer detected an obstacle, at which time it would override the forward movement and turn it in the proper direction.
Programmers are now extending this behavioral based approach with the use of AI techniques. It involves implementing robots that can choose by themselves what layers have hierarchical control. They can achieve this determination on their own through evolution. In the work of Marocco et al, they implement an example of this with simulated agents which are left free to determine for example whether they will use their signaling capability or not, or the behavior they display when their sensors are not affected by other agents, or even the circumstances in which signals will be produced [4].

This brings us to the concept of embodiment. Although the term can simply refer to the existence of a physical body, embodiment is often thought of as the awareness of ones own body and its surrounding environment. Duffy and Joue go further into defining what the term means, even stating that "in order to achieve cognitive capabilities or a degree of intelligence in an agent, a notion of embodiment is required where there is cohesive interaction between the environment and the body [5]."
MacDorman et al point out that an agents ability to perceive certain things depends on having a particular body. The perceptions change as that body or its environment changes [6]. They also point out that as humans we are able to disembody ourselves, shift perspectives, and ponder the future. Their claim is that these abilities are capable based on the power of words and symbols to stand in for something which is absent.

In recent years there have been many different methods developed for evolving communication. We will provide a few examples here. This is not meant to be an exhaustive review, but rather to present a few of the ways in which communication can exist. For example, Quinn has evolved multiple robots to work together in teamwork [7]. Through the use of neural network controllers, the robots evolve to solve a coordinated movement task. The behavior that two agents follow is that first both agents rotate counter-clockwise till it is facing the other. The agent that first faces the other agent with its front (agent B) sends a signal. If an agent perceives the signal while rotating it becomes the leader. Agents A and B thus assume the roles of leader and follower respectively. Agent B moves toward and remains close to agent A while A moves backward. Di Paolo developed an approach similar to Quinn's work in which the agents stay as close to each other for as long as possible [8]. Iizuka and Ikegami evolved simulated agents that exchange their roles (chaser/evader) so as to produce a form of turn-tacking behavior [9]. Nolfi created robots which, as a group, are asked to find and remain on two feeding areas by equally subdividing themselves between the two areas. These examples are simple implementations of getting mobile robots to evolve a communication. Nolfi looks at these pieces of work and finds different ways in which to model how a population of agents might develop complex forms of communication and a shared language [10]. But the difference in our work is that we don't have any method to supply reinforcement (to our agents. Our objective is to utilize the techniques available for simple robot communication in an attempt to evolve a more appropriate 'language' for our agents.

One of the major aspects to consider with the 'language' of our agents consists of the patterns which the agents make. There are many methods for the creation of patterns. One such method is based on cellular automata. Cellular automata started with the work of Alan Turing in the late 1930s, but it reached its popularity through the Game of Life, developed by John Conway in 1970. The rules of the game are what control the complexity of the patterns created. These are the rules as presented on wikipedia:

> The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, live or dead. Every cell interacts with its eight neighbours, which are the cells that are directly horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:
>
> - Any live cell with fewer than two live neighbours dies, as if by loneliness.
> - Any live cell with more than three live neighbours dies, as if by overcrowding.
> - Any live cell with two or three live neighbours lives, unchanged, to the next generation.
> - Any dead cell with exactly three live neighbours comes to life.
>
> The initial pattern constitutes the first generation of the system. The second generation is created by applying the above rules simultaneously to every cell in the first generation – births and deaths happen simultaneously, and the discrete moment at which this happens is sometimes called a tick. (In other words, each generation is based entirely on the one before.) The rules continue to be applied repeatedly to create further generations [11].

Stephen Wolframs' New Science goes even further into the complexities that can be evolved when implementing simple rules over an extended period of time [12]. The complexity produced with these rules however expand beyond our 14 by 14 display. We would need more area in order for the complexity and interest to emerge for the user.

2 Scope

The overall concept of this project stemmed from the work being done with ECAgents: Embodied and Communicating Agents (`http://ecagents.istc.cnr.it`). Their aim is to identify new methods and algorithms that allow systems to self-organize and to display properties emerging from the interactions between themselves and with the external environment. The project idea specifically came from the Future Applications Lab (FAL) at the Viktoria Institute (`http://www.viktoria.se/fal`). Their aim in the thesis project is to develop a prototype that can illustrate how patterns can be presented and communicated, based on the agents perception of their environment and interaction with each other. It was these objectives that helped focus this research.

One essential aspect of this project is the use physical robots. While its conceivable that our goal could have been reached through simulation within a shorter development period, there are several draw backs to simulation. The most convincing aspect toward this choice is the complexity involved with using physical agents. The noise that could be implemented within simulation would be predetermined. Whereas within a real environment, noise is inherently unique and provides users with more potential for unexpected results.

Throughout this paper the terms robot, agent, platform, and e-puck will be used often. While each of these words has a unique meaning, the concept they represent will be used interchangeably. It is key to note that our work focuses on the idea that agents interacting together exhibit a collective performance that no single agent would be able to generate acting alone.
Throughout the process of our work we would analyze our situation to generate ideas for continued work. Within this paper the term brainstorm is often used to describe this process. There was a particular brainstorm session which was held and the details of this activity will be discussed later in this work.
The hardware used for this project consists of the e-puck robot and an additional component called the see-puck.

(a) The e-puck and an e-puck with the see-puck display attached standing next to an ordinary coffee mug.

Figure 1: *E-Puck and See-Puck*

E-Puck The e-puck is a small mobile robot developed by EPFL (Ecole Polytechnique Fédérale de Lausanne) intended for education prototyping. The e-puck is built with a transparent plastic body supporting the motors, the battery and the electronics. The body measures in at a diameter of 70 mm and height of approximately 50 mm. There are two wheels on the robot allowing it to run on flat surfaces, such as a table top. The processor onboard is designed to support C programming. The e-puck is equipped with 8 infrared (IR) proximity sensors with a detection distance of 3-4 centimeters. There is an accelerometer which measures along the x, y, and z axes. The other sensors, which will be discussed more later, are 3 microphones and a color camera with a resolution of 640x480 pixels. As output, in addition to the wheels, the e-puck is equipped with a speaker, 8 red LEDs around the body, 1 red LED in the front of the body, and a green LED located inside the transparent body. The communication links supported by e-puck are a standard RS232, an infrared remote control and Bluetooth [13].

See-Puck The optical display consists of two circuit boards. The top card, the see-matrix card, is a matrix of 148 LED's in a circular shape. The second card, the see-controller card, is made up of a micro controller, a few amplifiers, and an RS232 interface. The see-puck is connected to the e-puck using an RS232 connector and mounted on top of the extension board, making the total height of e-puck and see-puck approx. 80 mm. The display has commands to draw primitives such as lines and circles, along with more low-level commands to manipulate single pixels or scrolling. The display also has commands for running the game of life. There is one front buffer and one back buffer for double buffered graphics providing smooth animations [14].

To program the robot in C, we used the integrated development environment (IDE) of the microcontroller of e-puck. We used the MPLAB IDE for editing, the C compiler MPLAB C30, and the TinyBootloader as our downloader. With the Bluetooth on the e-puck, there is a serial line emulation which supported by any PC, making the communication and the development of PC software simple. The e-puck came with library files and example programs that helped in the understanding of its capabilities.

Test Environment When testing our applications on the e-pucks, we used a 70x100 cm flat area with 3 cm borders to prevent the e-pucks from leaving the area. We had 20 e-pucks available, with 10 working see-puck displays. All tests were run under supervision to monitor for dead batteries and faulty displays.

3 Theory

The ultimate goal of our work is to establish a communication system for the agents which is not predesigned and not fixed but rather emerges spontaneously from the agents interacting among themselves and the external environment. In this section we will detail some of the limitations we imposed upon ourselves earlier in the project, and also highlight some of the theories we envisioned to be part of our work. This section should help define those things that both helped and hindered us from reaching our goal.

3.1 Limitations

Sound
The speaker of the e-puck is capable of playing any kind of WAV or tone sounds. However, the microphone and speaker combination of communication was ruled out immediately. This was decided due to the likelihood that the final application would need to be presented in an environment of uncertain noise levels. For this reason, we felt that to develop something that would ultimately be unpresentable at a later date was not desired, and thereby eliminated the use of speaker to microphone communication.

Camera
The camera provides a much larger amount of data than the processor can store in its memory. The camera can therefore only be used if small portions of the image or lower quality images are acquired. While there were several ideas reached through brainstorming solutions that incorporated video, none of them provided enough value to implement into a functional application. One such idea was to present an image to the e-puck video input and have the agent process that image into a simplified version that would then be displayed upon the see-puck. Besides the resource consumption of this task, the ability to inform the e-puck when to take a snap shot without a combination of sound and movement inputs would be an extreme task.

Interrupts
To run different process simultaneously, the e-puck uses interrupts. With five available, and two being dedicated to motors, tasks within our more robust applications had to be combined through some use of behavioral design.

Store Data
Data can easily be retrieved and stored through communication from one e-puck to a master computer. Unfortunately this is the only method available to persistently store data from the e-puck. Each time the e-puck is turned on or reset, all previous information is lost. By our understanding, once the e-puck is restarted, it looses connection to the master computer and connection can only be reestablished by the computer.

Display Feedback
The see-puck display doesnt provide any feedback method to obtain pixel's color information. Thereby, there is no way to determine whether a light is on or off. This limits the use of the Game of Life sequence, because mixing in different inputs creates results which cant be calculated before hand. This severely limits the agents ability to use the Game of Life in conjunction with any learning and/or evolving patterns.

3.2 Theoretical

Most projects that involve simple robots are built around the idea of cooperation to achieve a common task (see 'Review of similar work'). In our project however, agents weren't given any particular task or problem to solve. There was no light to chase or food to grab. We wanted to demonstrate that amongst a group, an individual agent could acknowledge other agents amidst its environment and display its recognition through a visual means to the user.

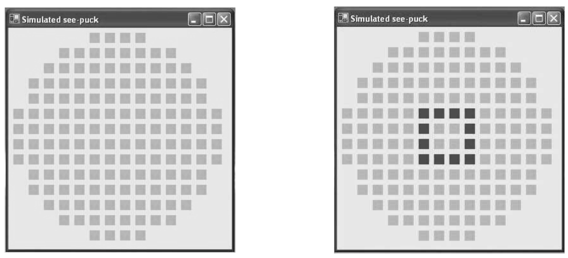
One thing we did was to look into how other communication systems, such as human, animal and insect, generate their messages. Due to our limitation of no sound, this allowed us to narrow down some of these possibilities. Since most of the remaining communication channels that don't use sound rely on scent, something our e-pucks don't have a sensor for, we then narrowed our options further. One of the possibilities that evolved during brainstorming was that of male peacocks and their use of plumage to attract females. Another was of lightning bugs, or fireflies, and their use of blinking lights to attract mates.
The research of Richards and Hanson brought a tested theory to our process. They present that "each firefly lengthens or shortens his next interflash period according to whether he had flashed earlier or later than the average during the previous concerted emission. It was thought, in other words, that each firefly must be capable of distinguishing the flash sequence and of controlling the period of his endogenous timer [15]." What they found was that the individual firefly resets its flash period to that of the group. The resetting occurs cycle by cycle, with the firefly reverting immediately to his free-run period when there is no influencing input. In a single period, a firefly can synchronize his timing with a group he has just joined, and reset the timer each period if his clock tends to drift with respect to the group [16].

Developing embodied agents meant that movement of the robot would allow us to take advantage of different results. There exist several methods of controlling movement of mobile robots including swarmed, following, obstacle avoidance, and random/chaotic. It was here agian that we found our work to differ from that of the majority of mobile robot applications. Without a specific task to solve the use of following or swarmed movement was unnecessary and in some ways inappropriate. It was because this that we chose to use a random/chaotic approach. Of course we also needed to keep in mind the aesthetics of the movement to interest the user. "Behaviors of elegance and sophistication, as seen in the mind's eye, often degenerate into mysterious bumps and gyrations when executed by the robot [17]."
This logic is what defined our development process. Our strategy was not to set all requirements for the desired application and then design and develop. Instead we developed one piece of our design and upon learning of its effectiveness, we would then alter our design accordingly. We considered this to be a bottom up approach that had top down tendencies.

4 Exploration

This section describes the methods for exploring the capabilities of the platform used from the start of this project to the point where it is to date.
"Robots have a knack for exposing the ignorance of their creators [17]." This eloquent phrase from Jones describes the need to iteratively explore the design our project. Each idea created needs to be tested to see how it performs before moving on to the next stage of development. By following this iterative approach, we would explore the possibilities with the implementation of a particular method and analyze the results. Based on these results, a new method would be created extending the successful techniques. Jones states in his book that when trying to program something new on a robot, we as programmers have to learn something new. This meant that for "each time we seek to make a robot do something new, we as robot builders are obliged to learn something new [17]."
For each method explored, we will detail the results obtained and discuss how they effected our project. In some instances, the methods decided upon where in direct correlation to the theories previously discussed.

Develop Simulated See-Puck
The see-puck would be getting its first complete implementation since being designed and built at FAL the previous year. In fact, at the start of this project the only one in existence was the prototype. At the beginning of the project, while waiting for the shipment of see-puck displays to arrive and the prototype temporarily unavailable, we created a simple GUI tool to practice with the light patterns. The tool was based on the physical design of the see-puck with 14 by 14 grid and the corners rounded off. Given the documentation that came with the see-puck, we also wanted to implement the commands that could create simple patterns, for example lines, squares and circles.
The GUI application consisting of 148 square images fitting the layout of the see-puck was created. Light



(a) Simulated See-Puck (b) Simulated See-Puck with square drawn

Figure 2: *Images of the Simulated See-Puck.*

red and dark red were chosen as the colors to represent the light being off or on respectively. A two dimensional array was created with 14 by 14 capacity, and each position in the area was given a number to represent the corresponding light, or zero if the position didn't have a light to correspond to. Another array was created, single dimension with capacity of 148, that stored instances of the image of each light. Two test runs were developed, one to scroll through each light and turn it on then off in a consecutive pattern. The other two expand and collapse a box pattern. The patterns created were hard coded at this point, but the goal was to provide ideas on how to continue working with it once switched to the see-puck.
When creating this simulation, we had to question how the actual see-puck was structured. For example, were there considered to be 196 entities, that is was [0,0] a non-light parameter that was physically located in the top corner, or lower right corner even, and so on. Or were there only 148 entities, and if so where would [0,7] be located. With this simulation, we actually implemented both techniques. Although we were able to implement both in our simulation this didn't answer the question of how the see-puck would function. Another question was that if it functioned on the two dimension coordinate system, what would happen if trying to light pixel [0,0], or [123,56]. Did the display have error handling built in for faulty inputs?

Test Libraries (LED, motor, sound)
The first step with working on the e-puck was to learn how the sensors and outputs functioned. The e-puck was a new interface to work with. We considered the outputs first since they would tell us what the e-puck could 'do'. And then during this process we would also be able to learn what input the e-puck could receive. The e-puck had several libraries and example programs created for it so our goal was to implement each of these and learn how the components of the e-puck functioned on an individual basis. Even though we had originally decided (see **Limitations** in the previous section) that using sound would not be effective for our final application, it was still something that needed to be tested.
As work began there was no trouble getting the e-pucks to implement the simple libraries, including lighting the LEDs and driving with the motor back and forth. We found that speaker pointed upward was not an efficient means for robot to robot communication. Though later when adding the see-puck display cards, the sound reverberated enough to make it slightly more effective.
While the lights and movement were no trouble, integrating the IR signaling and how to read in sensor information from the e-puck proved more difficult. Below is a sampling of the code used to monitor the proximity sensors:

```
static int ir_phase=0; // phase can be 0 (ambient) or 1 (reflected)
static int ir_number=0; // number goes from 0 to 3 (4 couples of sensors)

switch (ir_number)
{
  case 0: // ir sensors 0 and 4
  {
    if (ir_phase == 0)
    {
      PR1 = (350.0*MICROSEC)/8.0; // next interrupt in 350 us
      ambient_ir[0] = ReadAd(IR0);
      ambient_ir[4] = ReadAd(IR4);
      pulse_IR0 = 1; // led on for next measurement
      ir_phase = 1; // next phase
    }
    else
    {
      PR1 = (2100.0*MICROSEC)/8.0; // next interrupt in 3 ms
      ambient_and_reflected_ir[0] = ReadAd(IR0);
      ambient_and_reflected_ir[4] = ReadAd(IR4);
      reflected_ir[0] = ambient_ir[0] - ambient_and_reflected_ir[0];
      reflected_ir[4] = ambient_ir[4] - ambient_and_reflected_ir[4];
      pulse_IR0 = 0; // led off
      ir_phase = 0; // reset phase
      ir_number = 1; // next two sensors
    }
    break;
  }
  case 1: // ir sensors 1 and 5
  ...
```

See-Pucks Arrive
When the see-puck displays did arrive, we learned that a portion of them were defective. A simple display test program was created to test the functionality of each display. Building this test program also helped with answering the questions which arose when developing the simulated see-puck. By running the test, we found that all of the red displays were dysfunctional, along with a few of the yellow and green ones. Those that did work, implemented the test patterns without issue. Throughout the life of the project, we would need to adjust the application to provide a display that continually made changes so that defective e-pucks can be singled out from the group. This also included using the body light to indicate that the battery level was low.
With the ability to now work directly with the see-puck, we were able to answer the questions posed during **Develop Simulated See-Puck**. There turned out to be 196 entities, with areas such as [1,1] existing in a non-light sector. There weren't any issues when trying to light any of the coordinates within

the [14, 14]. However, if the pattern went out of these bounds it would then extend back through [0,0], which would cause odd results. After we filtered out the dysfunctional displays, we still found irregularities. As we continued working and performed tests on our development, strange anomalies would occur with the displays. As this first began, it couldn't be determined whether the issues were caused by the applications functioning improperly, the applications being designed improperly, or the hardware functioning improperly. As this issue continued we were able to eventually narrow it down to a portion of the see-puck controller boards being faulty.

Actuators and Sensors
There are several methods of achieving communication through the sensors and actuators of the e-puck robot. Our first step was to explore what sensors were available on our robot platform and the type of input which could be received from the environment and other robots. The results from **Test Libraries**

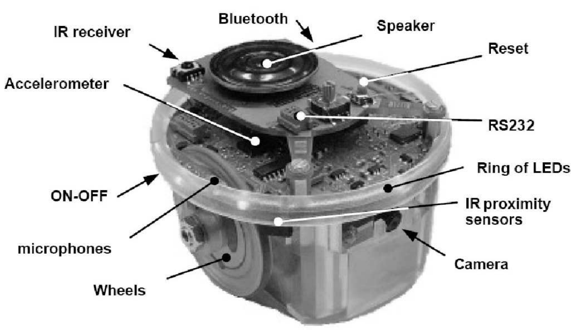


Figure 3: *E-puck with actuators and sensors labeled.*

helped in evaluating the actuators and sensors. We also took into consideration the limitations we imposed on ourselves.
The possible combinations that we looked into were: LED to camera; LED to IR; proximity sensors; microphone to speaker; motor to accelerometer; Bluetooth; and IR from remote control.

Input	Output	Human Comparison
LED	Camera	Eyes
LED	IR Proximity	Eyes
IR Proximity	Motor	Touch
Microphone	Speaker	Voice
Accelerometer	Motor	Equilibrium
Bluetooth	Bluetooth	Telepathy
IR Receiver	Remote Control	Sharp stick in the back

Table 1: *Possible communication combinations*

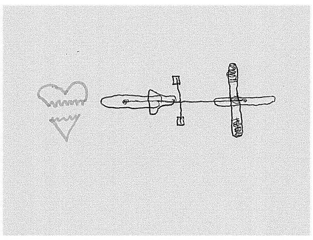
As discussed earlier, the limitations we imposed on ourselves ruled out some of these communication methods, which limited us to the use of LED to IR, proximity sensors, and motor to accelerometer. LED to IR proved to be a very effective tool for establishing communication between robots. As described later, the robots were able to synchronize their flashing as a group.
The accelerometer didn't work. Its settings couldn't be tweaked enough so that it didn't constantly think that little shakes and bumps of driving weren't coming from being picked up and shaken. So due in large part to hardware limitations this method was not effective for communication for embodiment or user to

robot communication.

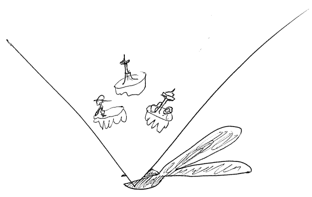
Brainstorm
We went through the process of a brainstorm session to look into unexplored techniques for communication. The session was performed with four people and consisted of 3 stages. The first stage was word association, in which a word was given and each person would write as many words they could associate with the given word within a 2 minute time period. There was a middle step between the first stage and the second. The process of this step involved combining three words at random that were created during stage 1. For example, if the words [shoe, banana, car, round, butter] were created during stage 1, then a possible combination could be car-butter-shoe. The second stage involved taking these combinations and applying a rule to each of them to create images. There were four rules:

- If A feels B then C happens
- A communicates with B using C
- Many of something A becomes B by eating C
- Some A and some B becomes C

To continue our example to stage 2, if we use the first rule on our combination, then we would have: If [car] feels [butter] the [shoe] happens. Each person would be given a word combination and a rule to apply to it, and would then draw pictures representing what that sentence meant to them. The drawing of our example might have an image of a car driving over a piece of butter which causes the car to crash and thereby the driver must now use shoes to travel. Obviously there can be many interpretations of this unique phrase. The third, and final stage, was performed in groups of two. Each group would look at two separate drawings created in stage 2, and trying to create possible agents out of the content of each drawing. The group would then describe how the agents from two different drawings might interact with each other.
The brainstorming session produced seven unique results and an example of one of these results will be detailed here. The first items which should be noted are Figure 4(a) and Figure 4(b). These images show different results of the 2nd stage of the brainstorming session. Figure 4(a) represents the first rule being applied to the words: heart, bike and thorny; If HEART feels BIKE then THORNY happens. The image depicts that if the heart was run over by the bike, it would break and become thorny. And Figure 4(b) also shows the first rule being applied, this time to the words: corner, clipper, and place; If CORNER feels CLIPPER then PLACE happens. This image depicts the idea that places, such as Paris, Liseberg, and Kungsplatsen, are all contained together in a plane of existence, and that once the plane is opened at the corner with the clippers, then the places flow out into the world.
During the third stage, each image was given to a different group. The groups came up with possible agents that were represented by the image. For Figure 4(a), a robot with paint was discussed that performs its 'communication' in a tactile means with the paint. Another agent described was a bike bot that communicates broken hearts. For Figure 4(b), there were an architect robot which has the power to create places, a guiding robot that can take one from place to place, and Edward Scissorhands. When the groups finished coming up with these agents, the two were combined to see how they might communicate between each other. Figure 5 depicts some of the possible methods through drawings. What you may be



(a) If HEART feels BIKE then THORNY happens



(b) If CORNER feels CLIPPER then PLACE happens

Figure 4: *Images from stage two of the Brainstorming session.*

able to decipher from this image is the architect robot which creates roads that the bike needs to follow. Another idea that was not drawn was that Edward creates hearts but cuttings them into stamps, which the bike then leaves a track of while it rides along, with Edward sitting on the back making the stamps. What also can be seen in the drawing is the guiding robot providing instructions through tactile means, forward and circle patterns, which the mobile bot then performs on its own. While it was clear that the

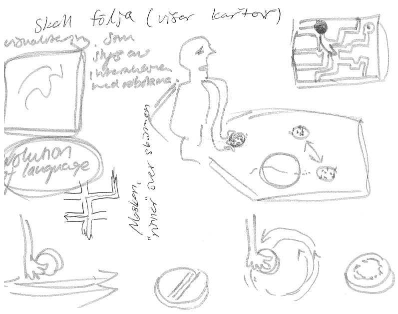


Figure 5: *Example result from the Brainstorming session.*

brainstorm session provided interesting results, we found that they didn't help much in the techniques of our work. The idea was that we would find a unique way to approach the communication of the e-puck. One of the reasons why this session seemed to be ineffective was that in our previous step of determining the communication tools of the e-puck, we eliminated the possibility of such creativity of communication.

Implement UART

When it came to debugging the programs that were written for the e-puck, the process was to use the LEDs to signify when certain parts of the code had been reached. For some applications this was a very effective tool, however if the program was to use the LEDs for other means then this wouldn't work. It was clear that another method was needed one that included variable information. It was this need that led to the implementation of the uart, Universal Asynchronous Receiver/Transmitter (UART).
The success of implementing the UART connection was an important step in development. Besides having another technique available for debugging, we were now able to read sensor data from the e-puck in real time. The print function on the e-puck was now functioning, so that a robot which is still in a Bluetooth connection with the main PC could print to the terminal. This would be one of the most useful features we implemented.

Attempt IR Communication

Based on the conclusions reached with **Actuators and Sensors**, we began looking into getting the e-pucks to recognize each other through the IR proximity sensors. In this process we tried to reach the point where one e-puck can recognize if another e-puck is directly infront or behind. Then the e-puck would be able to send light signals indicating its intentions.
The success of this technique was demonstrated by the e-pucks' embodiment, by sensing the environment around them and acting upon that sensation. It was determined however that the limitation in perception of the IR sensors meant that the e-pucks would often be next to another e-puck but unable to sense it before turning away.

Sync

By using the proximity sensors, if the e-puck acknowledged a flash, it would begin to sync its own LED flashing to its neighbor. This was to be performed in a similar way to that of the firefly, so that it wasn't flashing its light when it saw another flash, but rather it would adjust its interflash period accordingly.
The implementation of the syncing flashes of the LEDs worked very well. There was clearly a connection between to result and the concept intended. It was effective in that not just single robot to robot, but entire groups of robots could all coordinate their flashes. What also proved interesting is that the flash time that was established would be kept and then have an influence on the next 'moment' of communication.

Game of Life Gliders

We divided the e-puck into four quadrants, four corresponding directions (i.e. front 45°, right 135°, left 225°, and rear 315°). During the **Sync** process the e-puck would check to see if it recognized anything within its proximity in each of the quadrants. If there was a detection within a quadrant, the display would create a glider (see Figure 6) from the Game of Life in that corner. Gliders would continue to be created during the detection interval while there continued to be something within that quadrant.

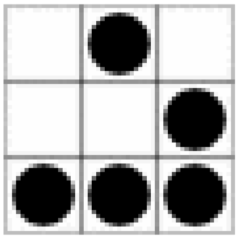


Figure 6: *Glider in its initial phase*

Because having a single glider across the screen doesn't represent the fact that the e-puck could be surrounded by other agents, a glider would be created in each quadrant where there was a detection. Since detection was often made at the same time, gliders would be created almost certainly to collide. When they collided they spread of their pattern and provided the complexity that cellular automata is based upon.
After attempts at creating static patterns for the e-pucks consisting of squares and lines, we determined that the most interesting patterns to the viewer were achieved through the use of the Game of Life. This wasn't because the static patterns weren't interesting themselves, but because the Game of Life provides the complexity in appearance that is difficult to achieve with static patterns. We did approach the static pattern again when using the **Bluetooth Communication**.

Movement
The movement method used to achieve something that to the user appeared random while keeping a form of elegance was achieved with this algorithm:

```
tmp_l = (float)i / 6.1;
speed_l = (int)(1000.0 * sin(tmp_l));
tmp_r = (float)i / 4.9;
speed_r = (int)(1000.0 * sin(tmp_r));
```

Using the sine function provides a smooth elegance that appeared random. And since the wheels would occasionally be effected by slip, this method actually turned out to be entirely random.

Bluetooth Communication
Our next step was to explore the Bluetooth capabilities deeper so that the e-pucks could achieve robot to robot communication. Our logic was that if there was Bluetooth capabilities to connect one e-puck to the computer and share information, then there should be a method to connect one e-puck to another.
Our intention was that one robot will send another robot information about the pattern it is currently making, and the receiving robot will take this information in, along with other environment data, to change its own pattern accordingly. The idea is that each agent would begin with an initial pattern unique to itself. Over time, each agent in the environment would be capable of sharing its pattern with another. Once a pattern is received it would then make a decision on how to integrate the received pattern with its own current pattern. This could involve three options; keeping its own pattern as is, replacing its own pattern entirely, or producing a blend of the current and received pattern. Another influence upon the pattern would be what is around them, considering that they should be embodied robots. They should be able to detect if anything lies within their proximity when choosing how to incorporate a new pattern information. A possible idea is that they can detect if a neighbor of theirs is part of three groups (green, red or yellow displays).
We located the Bluetooth library after much searching, and were able to implement sending simple commands from one 'speaker' e-puck to another 'listener' e-puck. These consisted of sending the 'listener'

commands that made it turn specific LEDs on and off. Unfortunately we were unable to disconnect the connection of one robot to another. For example, if robot A sent robot B a command to turn LED2 on, then robot A would be unable to disconnect its Bluetooth connection to then tell robot C to turn LED3 on.
Since Bluetooth presents sort of an anonymous connection, as far as knowledge of the environment is concerned, the robots would need to use their other sensors to identify information about themselves.
This approach limited the possibilities for testing effectiveness of communication, but went along the idea of testing the possibility of emergence in communication. It is reasonable to assume that progresses might be only achieved by predefining, in the starting conditions, crucial elements of the communication protocol that although in theory could spontaneously emerge in the course of the process, but in practice, would very unlikely do so. It becomes possible that these minimal set of prerequisites might trigger the emergence of complex forms of patterns and communications. For the reason that the communication between e-pucks couldn't be implemented, we were unable to test to see how effective the use of a neural structure was in managing pattern data.

Random Patterns

The process of using the **Bluetooth Communication** started with random patterns generated from the API of the see-puck display. A random sequence of eight different options was generated. The options used were: line, pixel, circle, filled circle, box, filled box, horizontal scroll, and vertical scroll.
While testing the outcome of these, it appeared that a single pixel wasn't very effective. Nor were the filled circles and boxes. An individual pixel didn't add any significant interest into the pattern, and the filled shapes often took up most of the screen.
In the last phase of working with random patterns, the options were limited to five. In these options a random selection of 4 to 100 pixels would be lit, or random amount of lines between 2 to 10, or random amount of boxes or circles from 1 to 5, or any combination of the previous for patterns. This technique improved the results of patterns created.

5 Conclusion

5.1 Research questions answered

What method of agent to agent communication is most effective? (both for communicating information about the external environment, and for sharing information on pattern creation) This has been a difficult question to answer. Entering the project we had a bias that Bluetooth, with its data transferring capabilities, would be the best option. However, as the project progressed and we looked at embodiment of autonomous agents, there developed flaws in the Bluetooth technology. Particularly with the issue that there is no way to determine the physical presence of a communicator. Now in answering this question, we must also take into consideration that not all possible methods of e-puck communication were analyzed, such as sound and video. So in the end, it must be stated that the IR proximity sensors were most effective in communicating the external environment. And though these sensors didn't directly share information related to patterns on other agents, they did influence the information used to create patterns. At this point there is still a bias toward the possibilities available with Bluetooth, but it is now clear that this method is uncapable of being truly efficient on its own.

How can robot driven movement be used to assist with this communication? As discussed earlier in this paper, there exist several methods of controlling movement of mobile robots. In our work we eliminated a few of these methods. We did find that movement was essential to our application, for the e-pucks couldn't have found each other with moving about within their environment. In connection to our previous question, if the communication was performed solely through Bluetooth, the use of movement would become unnecessary. With a range of 10 meters, the amount of movement needed to take the e-puck in and out of range would be epic in comparison to its size.

What kind of patterns produce interesting results to a user? Our work didn't take into account all the parameters needed to answer this question. Our approach toward this issue did not use any scientific methods. For example, our sample population of users never exceeded a group of ten, and the feedback generated was kept informal. Another issue preventing a quality answer to this question is that throughout the development phase, the types of patterns used were in constant flux. One tendency that was noted was that there was seldom a pattern that a user didn't find interesting. Even when patterns and shapes weren't clearly defined, they still generated interest from the users. In those instances, it was often compared to cloud gazing, the act staring up at the sky, looking at cloud formations and visualizing objects.

How can the user inform the robot when an interesting pattern is created? In the application that we created, the answer to this is simple, they can't. The method we found to be most appropriate was for the user to pick the e-puck up and shake it. However, based on the settings we used, the e-pucks natural shaking while moving set this feature off, causing the need for it to be removed from our application. Another possible method for user interaction is the use of the proximity sensors. In our application these sensors were essential for other tasks, however the user can interact with the e-puck by placing a hand over specified sensors. Sound could also be used, such as a clap from the user to indicate like or dislike of a pattern. As you can see there are different ways in which user interaction can be received, only in our application we were unable to successfully incorporate any such method.

How does the robot evolve as it receives new information? Although this was a key concept in which we hoped to have an answer for when we started this project, it remains unanswered. One reason for this could be that its evolution should not be predetermined, but rather something that it accomplishes on its own. But the most likely reason for this continuing to be such an unknown is that without the use of persistent data, the life cycle available to produce interesting results of evolution is severely limited and has to be approached on a application by application basis.

5.2 Future work

The work we performed is considered to be a starting point for development of the see-puck. We explored the areas that we were able to within the time period of this project, but there is room for more contributions to be made by continuing this work. Throughout this paper ideas have been discussed which would be inconceivable to have been accomplished within the time frame for this work. Also under consideration should be the fact that as time goes by, technology advances exponentially. It is possible

that some of the limitations we faced could be solved by the time this is read and would make our work ready for improvement. That being said, there are a few ideas that should be discussed in regard to continued work on this project.
One possibly technique that could be implemented is to use a simulation to evolve a neural structure for handling the patterns communicated through Bluetooth. This could be performed by having the e-pucks individually connect to a master computer and provide pattern data. The master computer could use information gathered in various methods to then return a pattern structure to the e-puck. This solution could resolve the issue of persistant data along with the limited processing power of the robot. This might even be possible through the use of tools such as Evorobot, a software for running evolutionary robotics experiments [18].
Another area that should be looked into more is the different possible ways of creating patterns. - and how one pattern could be shared/integrated with another pattern

The use of IR proximity sensors could be explored more. One technique that was thought of is that each agent, or possibly small groups of agents, could have their own distinguishable signal, not unlike Morse code. So that when one agent approaches another, it would be able to determine if that agent was part of its group or a possible new agent.

To close this paper we shall leave you with a thought from Jones in which he describes the gap between expectations and experiences in robotics:

> The crux of the problem is that humans are just very good. We take many things for granted in our biological selves: the acuity of our eyesight, the fine dexterity of our fingertips, the amazing power-to-weight ration of our muscles, and the efficiency of our energy conversion system, to name a few. Instilling human-level equivalence in a robot is quite a challenge!
> In fact, the disparity between expectations and experiences grows wider if we think about the tiniest insects. Even their perceptual-motor skills are amazing. Common houseflies can land upside down on ceilings, spiders can assemble the most intricate homes, and ants can carry loads many times their weight. Robots have a lot of catching up to do [17].

References

[1] S. Ljungblad, K. Walter, M. Jacobsson and L.E. Holmquist. "Designing Personal Embodied Agents with Personas." Manuscript received March 15, 2006.

[2] T. Bräunl, *Embedded Robotics*. Berlin, Heidelberg: Springer-Verlag, 2003.

[3] R.A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 1423, March 1986; also MIT AI Memo 864, September 1985.

[4] D. Marocco, and S. Nolfi, "Emergence of Communication in Embodied Agents: Co-Adapting Communicative and Non-Communicative Behaviours." In A. Cangelosi et al., editor, *Modelling Language, Cognition and Action: Proceedings of the 9th Neural Computation and Psychology Workshop.* Singapore: World Scientific, 2005.

[5] B.R. Duffy and G. Joue, "Intelligent Robots: The Question of Embodiment," *Brain-Machine*, Ankara, Turkey, December 2000.

[6] K. F. MacDorman, K. Tatani, Y. Miyazaki, M. Koeda and Y. Nakamura. "Protosymbol emergence based on embodiment: Robot experiments," In Proceedings of the *IEEE Int. Conf. on Robotics and Automation*, pp. 1968.1974, 2001.

[7] M. Quinn, "Evolving communication without dedicated communication channels." In Kelemen, J. and Sosik, P. (editors) *Advances in Artificial Life: Sixth European Conference on Artificial Life (ECAL 2001).* Springer Verlag, 2001.

[8] E.A. Di Paolo, "Behavioral coordination structural congruence and entrainment in a simulation of acoustically coupled agents," *Adaptive Behavior*, vol.8 no.1, p.27-48, Jan. 2000.

[9] H. Iizuka and T.Ikegami, "Simulating turn-taking behaviours with coupled dynamical recognizers," Proceedings of *The eighth international conference on Artificial life*, p.319-328, December 2002.

[10] S. Nolfi, "Emergence of communication in embodied agents: co-adapting communicative and non-communicative behaviours." *Connection Science*, 17(3-4):231–248, 2005.

[11] Conway's Game of Life. (2007, January 4). In *Wikipedia, The Free Encyclopedia.* Retrieved January 9, 2007, from `http://en.wikipedia.org/w/index.php?title=Conway%27s_Game_of_Life&oldid=98324392`.

[12] S. Wolfram, *A New Kind of Science.* Champaign IL: Wolfram Media, Inc., 2002.

[13] e-puck Robot pages on EPFL's Web site, `www.e-puck.org`.

[14] see-puck - LED Matrix display for the e-puck robot (and other RS232 compatible devices) Complete reference manual, 2006.

[15] F.E. Hanson, J.F. Case, E. Buck and J. Buck, Synchrony and flash entrainment in a New Guinea firefly. *Science*, vol. 174, pp. 161-164, 1971.

[16] D. Richards, "Biological strategies for communication," *Communications Magazine*, IEEE vol.23, no.6, pp. 10-18, Jun 1985.

[17] J.L. Jones, A.M. Flynn and B.A. Seiger, *Mobile Robots*, Natick, MA: A K Peters, 1999

[18] S. Nolfi, *Evorobot* (Version 1.1) [Computer software], 2000.