Optimizing hardware

Possibilities and challenges of evolvable hardware

March 2012

Author: *Arne Staphorsius*
Supervisors: *Guszti Eiben and Berend Weel*

Abstract
Evolvable hardware distinguishes itself from conventional hardware by its ability to adapt to its environment, and change its structure to perform better on a given task. The principle of adaptation is based on evolution, and has great potential to be used widely in the future, in many areas. This paper discusses the progression in the field of evolvable hardware. It covers the current state of the field by describing the origin of evolvable hardware and its current uses. Furthermore the paper speculates about the possibilities and promises of future technological advances in the field, and it describes technologies that would enable the field to realize the mentioned possibilities. It also indicates challenges that still lie ahead and that we must meet in order to make the promises of evolvable hardware come true.

Keywords
Evolvable hardware, evolutionary computation, autonomous adaptive systems, self-reconfiguration, modular robotics, 3D printing.

VU UNIVERSITY AMSTERDAM

Preface

This research paper is a compulsory part of the Master program Business Analytics at the Vrije Universiteit Amsterdam. The purpose of the paper is for the student to use the knowledge and techniques that he accumulated over the years to do research into a subject of his choice. The paper should contain the pillars of the Master program: business, computer science and mathematics.

I would like to thank my supervisors, professor Guszti Eiben and Berend Weel, for the insights, the advice, and the support they gave me.

Arne Staphorsius

March 2012, Amsterdam

Index

Introduction

Evolution is a powerful tool. Nature itself has been using it since the beginning of time to solve its most difficult problem: the survival of species. Every specie on Earth is tested by its environment. If it performs well, it will stay alive and reproduce. If it does not perform well, it will die off. The successful individuals of a population will pass on their traits to the next generations, and by this the species as a whole are able to grow stronger over generations.

It is not at all strange that people have been trying to harness the power of this problem solver of nature. The idea dates from the forties, and it has developed ever since. At first, evolution was simulated as software in computers to find solutions to problems. But the last few years, the field of evolutionary computing has been trying to bring evolution to hardware.

This research paper takes a look at this development, and discusses the possibilities and also the promises of this evolvable hardware.

The first chapter of the paper will introduce the idea of evolutionary algorithms. The origin of the field and the characteristics of evolutionary algorithms are discussed, as well as the current uses of evolution in hardware. The second chapter give an overview of the possibilities of evolvable hardware, and gives an idea of how evolvable hardware could be used in the future. The third chapter will give multiple examples of enabling technologies that can be a stepping stone towards fully evolutionary hardware. Finally, the last chapter will outline the problems and challenges in the field of evolvable hardware.

Chapter 1: Current state of evolvable hardware

1.1 What is Evolvable Hardware?

Before we get started about the promises and possibilities of evolvable hardware, it is a wise idea to give a short introduction into the basics of evolutionary computing, and the idea behind evolutionary algorithms.

On the origin of species

The notion of evolutionary algorithms dates from as early as 1948, when Turing suggested a range of ideas for machines that could modify their own program and learn by themselves. [20] With the term 'genetical or evolutionary search' he already hinted, long before the emergence of computers, that evolution could be used to solve problems. [17]

The connection between the evolution theory of Darwin and stochastic search algorithms is the key behind evolutionary algorithms. In his book 'On The Origin Of Species', Darwin introduces the concept of natural selection, which is better known as the phrase 'survival of the fittest'. When a population of individuals resides in an environment that has limited resources, then the individuals will have to compete for the those resources in order to reproduce. The stronger individuals will be able to get to more resources than the weaker individuals in the population. In this setting natural selection is inevitable, since the individuals that are able to get hold of the limited resources in the most effective way, will be the ones that are going to reproduce, and the weaker individuals will effectively die off. This is where the phrase 'survival of the fittest' comes into play. [16]

He also gives evidence of the existence of heredity, and he wanted to show that species did not emerge separately. This was the second force behind evolution that Darwin discovered. [16] Individuals possess traits that are described within their phenotype. These traits determine how they react on their environment, or from a different point of view, how the environment reacts on them. If a phenotypic trait works in favor of the individual, the trait will be carried on to the individual's offspring, while traits that are not beneficial for the survival of the individual will be lost over generations since the individuals that carry it will lose the battle for survival. [17] Darwin saw that small mutations on these traits occurred, and that these random changes are the basis of new traits that are slight variations of former ones. This phenomenon creates a certain diversity in a species, where different individuals can have new combinations of traits, that will be tested by the environment again. Over generations this means that evolution can progress over time, searching for new and better, more successful combinations of traits and getting rid of the bad ones. [17]

Natural selection and heredity are the cornerstones of evolutionary progress as Darwin described it [17], and at the same time they are also the key elements of every evolutionary algorithm.

The evolutionary algorithm

With the insight into the origin of the evolution theory and the link made between the aspects of Darwinian evolution and the elements of an evolutionary algorithm, this section will describe the components and the basic scheme of an evolutionary algorithm.

Where natural, or Darwinian, evolution is about a population of a certain species, and individuals in the population are single organisms, in evolutionary algorithms the population consists of single solutions to the problem that the algorithm is designed for to solve.

Every solution in the population has different traits, and therefore every solution to the problem is different itself. In order to be able to measure the quality of every solution, the algorithm has to appoint a fitness value to every solution. This fitness value gives the algorithm a way to let the weaker solutions of the population die off, and the stronger solutions to reproduce. This works in exactly the same way as Darwin described it in natural evolution. The fittest examples in the population survive. This mechanism of selection is a fundamental part in any evolutionary algorithm, since it will give it a way to keep increasing the overall quality of solutions. [17]

The second important force behind an evolutionary algorithm are what often are called the variation operators. The variation operators, mutation and recombination, provide for the necessary diversity in the population, and thereby facilitate novelty. [17] Because of the variation, the algorithm is able to keep refining the traits of the individuals and improve the quality of the solutions. The variation can also come up with novel solutions, and through those novel solutions the algorithm is able to explore all of the solution space.

Selection and variation are the two components that form the core of evolutionary algorithms. Although there are many different kinds of evolutionary algorithms, these two components are always a part of it, and all of them normally follow a general scheme.

The general scheme of an evolutionary algorithm is shown in Figure 1.

The algorithm is started by creating an initial population of candidate solutions. These randomly generated solutions are then evaluated by the fitness function of the algorithm, in order to determine how 'fit' these starting solutions are. When the initial population is set, the algorithm will repeat the same loop until the termination conditions of the algorithm are met.

This repeating loop contains the variation and selection operators that were discussed before. The recombination operator creates new offspring by combining traits of parents from the existing population. After that the offspring are mutated by the mutation operators, and finally, after evaluating the new solutions, the survivors for the next generation will be selected.

As said, the cycle of parent selection, recombination, mutation and survivor selection will continue until the termination condition of the algorithm is satisfied. Often the algorithm is terminated when a certain number of generations is reached, or the quality of the solution has reached a satisfying value.

```
BEGIN
    INITIALIZE population;
    EVALUATE population;
    WHILE ( termination condition is not satisfied )
        SELECT parents;
        RECOMBINATION;
        MUTATION;
        EVALUATE offspring;
        SELECT survivors for next generation
    END WHILE
END
```

Figure 1: The general scheme of an evolutionary algorithm.

Evolution in hardware

The above sections are completely about the software implementation of the evolutionary algorithms, and since this paper lays its focus on the hardware equivalent, the next section will introduce the use of evolution in hardware.

Even though the idea of using the ways of natural evolution in a stochastic algorithm goes back to Turing in 1948 [t], it was not until around twenty years ago that people took an interest in using it in hardware. [12, 3]

The interest for evolution in hardware started in the area of digital circuit design. The last few decades there have been huge improvements in the field of circuit embodied technologies, and these developing technologies demanded increasingly more complex circuits.[3, 10] Traditionally however the process of designing circuits has been one that is laid out by methodologies that rely on rules, and they require human input. [10]

The need for human input in the design is a problem when the circuits become too complex, and it often leads to the choice between employing more designers or developing less complex circuits. [10] The first option is expensive, and will get even more expensive every time the circuits increase in complexity. The second option is plain and simple a loss of potential.

In order to overcome these consequences of the increasing complexity of circuits, new and more automatic design methods were needed. Methods that were able to deal with complex problems without requiring more designers. Evolvable hardware was one of those methods. [3]

1.2 Applications of evolvable hardware

In this next chapter we will look at some applications of evolvable hardware. First the use of evolution in digital circuit design, which is now a widespread methodology. Afterwards some possible areas of application are discussed.

Digital circuit design

It is a common practice these days to use evolution when designing digital circuits. The hardware design process is transformed to a software design process, and this allows the intelligent evolutionary methods to deal with the problem. [18]

The designer of a circuit still decides on the type of platform he wants the circuit to be put on. Application Specific Integrated Circuits (ASIC) are made to fit a single specific purpose, and they are not reconfigurable. Evolutionary methods are only used to produce the optimal design for such a circuit, before turning it to hardware.

A different approach is the use of reconfigurable hardware as a platform, and Field Programmable Gate Arrays (FPGA) are frequently used for it. In many situations, the FPGA can simply be used as a cheaper version of an ASIC. A simple FPGA only costs a few dollars, but the application-specific design can still be loaded on to it. In general, FPGA's are slower, larger and less energy efficient than ASIC's, but also much cheaper and flexible. [18]

The biggest strength of a FPGA however is the reconfigurability. The ability to reprogram it makes them suitable for rapid prototyping, but much more interesting is when a system is able to change the configuration autonomously. With multiple configurations prepared beforehand, a system can switch between configurations to increase performance by adopting to the task at hand.

This way of looking at circuit design is what is often called reconfigurable computing. [18] It is not only simply a powerful way to put reconfigurable hardware into practice, but it is even seen as a new and better general computing paradigm, and predicted to revolutionize computer science. [18]

While the field of circuit design is not likely to take the use of evolution further than they are doing now with reconfigurable hardware , the impact of this reconfigurable computing shows the power of hardware that can chance or evolve itself autonomously.

Possible applications of evolvable hardware

As was explained in the section above, evolution in the design of circuits is widely used. There are more areas however where evolvable hardware can bring important benefits. This section will go into detail on five important areas where evolvable hardware can be applied. These are: [10]

- Automatic design of low cost hardware
- Coping with poorly specified problems
- Innovation in poorly understood design spaces
- Creation of adaptive systems
- Creation of fault tolerant systems

Automatic design of low cost hardware

The first application is the use of evolution to come with solid but low cost solutions in the design of hardware. Often this works with reconfigurable hardware that has a low cost to make, but can embody designs that are evolved. Using low cost reconfigurable hardware has multiple advantages: the hardware can be reconfigured to reduce replacement costs, the hardware can be tailored to suit individual needs, and it can later be improved either by hand or further evolution.

Coping with poorly specified problems

The variation in evolutionary algorithms is an important feature when it is being used for problems that are poorly specified. It can sometimes be difficult to specify a certain problem, and this can be problematic if you are trying to solve that problem with classic methods. For evolutionary problems however you only need to specify how a solution to that problem looks like, and the algorithm will try and find better solutions no matter what the problem is. Noisy pattern recognition is a common application for evolvable hardware with this purpose. It is impossible to predict the noise, and therefore to specify the exact problem, but an evolutionary system could learn to cope with it.

Innovation in poorly understood design spaces

Another area where evolvable hardware can be of use is the design innovation in poorly understood design spaces. Especially in the field of electronic engineering, new kinds of circuits are generated frequently, and new technologies like those are more often than not poorly understood by people. Because of the creativity of an evolutionary algorithm, optimal designs for these circuits can be created without a thorough understanding of the whole design space at all.

These first three applications are examples of what is called offline evolution [1,2], static evolution [3], or extrinsic evolution [1]. The evolution happens before the actual hardware is manufactured, or in case of reconfigurable hardware the solution is evolved before it gets loaded on to the piece of hardware. The next two applications come closer to the idea of hardware that is able to evolve itself while it is active. This is what is called online [1,2], dynamic [3] or intrinsic [1] evolution.

Creation of adaptive systems

Adaptive systems are systems that can react autonomously to their environment and changes in that environment. These kind of applications are in great interest in areas where human intervention is perhaps impossible, like deep space missions. It can also be very useful in environments where the system encounters changes that cannot be anticipated on forehand. Research and development in adaptive systems is also very large in the field of robotics, where it is often the aim to develop autonomous systems or robots. A deep space example of these systems would be an exploring robot that gets sent to a distant planet, and will be able to adapt to any environment it faces there. On forehand it is impossible to program every possible scenario the robots could face, but an adaptive system could intelligently find a solution itself.

Creation of fault tolerant system

A slightly different approach to adaptive systems are the fault tolerant systems. Fault tolerant systems are, like their adaptive counterparts, able to react to changes. Fault tolerant systems however do not react to their environment, but to faults in its own hardware. This gives them a level of fault-tolerance and can also be used to give robots a self-repair mechanism. A working fault tolerant system has already been created by Higuchi et al. The system learned the behavior of an expert system and could take over its functions as a backup whenever the expert system failed. [10]

1.3 Summary

In this chapter we have looked at the power of evolution and its properties: the natural selection taking place and the heredity from generation to generation.

These features of evolution have been used in algorithms to come to a powerful new way of solving problems. Starting with a set of solutions, an algorithm will keep evolving and creating new and better solutions until a termination condition is met.

It has taken some time after the idea of evolutionary software to make the step to hardware, but the use of evolution in the design process of digital circuits is the first example of evolvable hardware.

Evolution can be used for much more than only the design of digital circuits, and the last section of the chapter has given some possible areas of application. Adaptive and fault-tolerant systems being the ones that stretch imagination the most.

Chapter 2: Future possibilities

It should be clear from the previous chapter that evolution is a very powerful problem solver indeed. The software applications of evolutionary algorithms have unique characteristics compared to other, more classical, problem solving algorithms, and they have been used successfully to solve many problems.

With the use of evolution in hardware, a whole new range of possibilities has become available to us. This chapter will take a look at the major areas where evolvable hardware could bring a lot of benefits, and speculates on how evolvable hardware could look in the future.

2.1 Medical applications

Since the practice of evolutionary computing has been derived from nature itself, it is perhaps logical that evolvable hardware is used in the medical field, which stands close to nature. And there are indeed a lot of possibilities in the medical field to be seen.

Biomedical nano-robots

One possibility to speculate is the ability to build evolutionary robots on such a small scale, that they could be introduced to the human body, for example in the blood circulation. Through evolution, these nano-robots would be able to learn the body, the metabolism of an individual person, and help keep this person healthy.

These "medibots" could be equipped with a way of scanning cells of the body, and serve as a warning system that alerts you when a virus or a disease has been found.

Personalized medicine

Another possibility in medicine, that also follows the global trend of personalization, is medicine that is evolved for a single individual. A development like this could be widely used with implants of prosthetics. Not only could a machine or computer learn about the body of any person and shape implants or prosthetics to the specific needs of that individual, but the implant or prosthetic could possess evolutionary intelligence as well, adjusting itself while in use to suit the user even better. Imagine a prosthetic leg that would learn how its user walks, and adjust itself to suit that individual better.

Evolving body parts

Another dream of many scientists, and something that could become reality with evolvable hardware, is the creation of body parts that could be used as a donor organ in a human body. The organs could be printed by a technique similar to 3D printing, and the created organ would use evolutionary adaptation inside the human body to behave the way that the rest of the body expects it to do, making sure that the artificial organ is accepted in the body and can take the place of a natural organ.

2.2 Advanced technological applications

Besides the medical field, evolvable hardware could have many applications in the field of advanced technology as well. What is meant here with advanced technology are mainly things like space exploration, security and warfare.

Deep space exploration

In the first chapter the adaptive nature of evolvable hardware was discussed. This particular property is of great use in the area of space exploration.

A big problem with space exploration is that it is currently not possible to send real people towards far planets, or even to the moon or to Mars, because the human body does not react very well to a long time without gravity, and of course bringing food and water is a problem too.

Evolvable hardware, in this case mostly robots, could be of great use here. The adaptive nature of evolutionary robots would help them deal with unexpected problems and obstacles, which are bound to occur on a distant and unknown planet. After being dropped on for example Mars, the robots would be able to start exploring and mapping the surface of the planet, while learning to deal with obstacles like craters they encounter. Perhaps the robots would even be able to merge with each other and split up if the circumstances would require it.

Sending intelligent evolvable robots into space would also be the best way to start a space colony, since it would not be possible for astronauts to stay in space for so long. The robots could start building houses and colonies for us, and since they have the ability to adapt, they could build anywhere they would seem suitable.

Search and rescue

Another advanced technological application of evolvable robots would be search and rescue. The idea behind it would be similar to that of deep space exploration. The evolutionary algorithm behind the robots would not only be able to learn the terrain and how to cross it, but also be able to find missing objects or people in the surroundings.

Unmanned aerial passenger flights

Recently featured in the BusinessWeek [24] was the development of a system that can visually detect other airplanes when deployed on a flying drone or airplane. The system is able to detect other flying objects from twice the distance that a human pilot can, and would be a great extra tool for a pilot to have at its disposal.

This system could have another great potential however, if it would be combined with evolutionary intelligence. If the system would be able to adapt well enough to obstacles in front of it, leaving no chance of errors, it could be used in unmanned aerial passenger flights.

Security and warfare

The areas of security and warfare are unavoidably linked to each other, specifically when it comes to evolvable hardware. The possibilities that lay with intelligent and evolvable weaponry or autonomous monitoring systems can be used for both.

The latter, autonomous systems that can monitor an area and learn and adapt to situations that take place in front of them, are an application that could be used to develop smart security systems.

Intelligent and autonomous weaponry could be used in advanced warfare, sending robots into unsafe areas to deal with enemy forces, and able to distinguish civilians and friendly soldiers.

2.3 Commercial applications

Apart from the use in the fields of medicine and advanced technology, evolvable hardware can also be used for commercial applications. Below are a few examples.

Domestic robots

Evolutionary robots could also be used in a domestic setting. The evolutionary intelligence behind it would be able to adapt to the behavior of its "master", and learn to serve its owner as good as possible.

These domestic robots could help people in everyday chores, but also with more difficult tasks, or decision making. Perhaps women would like the robot to learn their preferences for clothes, and help them pick out a dress for a dinner that evening.

Personal creation of objects

Evolvable hardware is perfectly suited for the creation of novel objects or items. In the future this could be implemented by a machine in every household that would be able to use evolution to create objects that you like. The owner of the machine would be able to steer the evolution by telling the machines which objects or which designs he likes, and the machine would use that knowledge to evolve object specifically to your own taste. If enough materials are at hand, the machine could make anything, from tableware to sunglasses or engine parts of your car that broke down.

Personalized manufacturing

The full customization of products is another likely feature that could be established with evolvable hardware. Computers and evolutionary algorithms would learn the exact preferences of a customer, and could build products that are completely customized to that customer. Products could include clothing that fits perfectly, or even larger products like cars could be made different for every individual customer with the technology of evolution and 3D printing for example.

Chapter 3: Enabling Technologies

There is a long way to go before the possibilities of evolvable hardware that have been discussed in the previous chapter can become reality. However, a lot of time and effort is going in to the research and development of technologies that could be a stepping stone towards real autonomously evolving hardware.

In this chapter I would like to give a few examples of technologies that could be that stepping stone. Some of the examples should be seen as small prototypes of future evolutionary hardware, and others are technologies or appliances that could be used with evolution to realize the possibilities of the previous chapter.

3.1 Modular Robotics

The field of modular robotics is very diverse. The research in to the use of modular robots covers areas from deep space exploration to nano-biology. In this section different of these uses of modular robotics will be discussed, as well as the progression in the field, and the way that the promises of last chapter can be realized.

Introduction to modular robots

Modular robots are constructs that consist of multiple identical building blocks, called modules. These building blocks are usually small, have little functionality by themselves, have a uniform way of connecting to other modules and can transfer power, momentum and information to other modules. [6]

In general, modular robots are classified in to different groups based on two characteristics. First, there is the architecture in which the different modules are arranged. This can be done in three different ways: [6]

- **Lattice architecture**: lattice architectures have modules that are arranged in a three-dimensional space, usually a cubic or hexagonal grid. All possible positions of modules is known beforehand, and therefore reconfiguration is relatively simple, and can be scaled more easily to larger and more complex systems.
- **Chain or tree architectures**: these architectures have modules that are connected in a tree or string. The architecture is always serial. They can potentially reach any location in space, and are therefore more versatile, but also computationally more difficult.

- **Mobile architecture**: mobile architectures are the most difficult architecture. Modules can use the environment to connect to each other, and the modules can work in multiple groups and coordinate their movements, being interconnected in a virtual network.

The second way of classifying modular robotic systems is the way modules connect to each other and reconfigure. [6]

- **Deterministic reconfiguration**: modules are directly manipulated to be put into place. The system knows the location of all modules at all times, or has at least a way to discover it, and the system can guarantee a time for reconfiguration.
- **Stochastic reconfiguration:** the location of modules is only known when the module is connected to the system. Modules move around using a stochastic process, usually through energy from the environment. Reconfiguration times can only be statistically guaranteed.

Motivation

To get an idea of the progression in the field of modular robotics, it is a good idea to understand the motivation behind creating robots that can change their shape. There are three key motivations for the creating of modular robots. [6]

- **Versatility**: robotic systems that can change their shape in order to perform better at different tasks are much more versatile than the conventional robot.
- **Robustness**: modular robots consists of the same building blocks, and all parts are interchangeable, making it easy and cheap to replace broken parts, as well as making it possible for the robot to repair itself.
- **Low cost**: because the robotic systems consist of the same modules, mass production and economies of scale become possible. On top of that the reconfigurability makes it possible for a system to perform multiple roles or tasks, thereby saving the cost of having to buy a new system for new tasks.

Achievements made with evolutionary robotics

A lot of different teams of scientists have worked on their own modular robots, and most of the time every team has a different goal with their project. In order to get a general idea of the progression in the field of modular robotics, Yim et al. [6] made a table with different accomplishments made in the field over the years. Figure 2 shows this table:

Accomplishment	Robot	Author	Affiliation	Quantity	Units
Most active modules in connected system	PolyBot	Yim et al.	PARC	56	modules
Smallest actuated module	Miniature	Yoshida et al.	AIST	40 × 40 × 50	mm
Largest actuated module	Helium Catoms	Goldstein et al.	CMU	8	m^3
Strongest actuation	Polybot	Yim et al.	PARC	5	modules cantilever
Fastest modular robot system	CKBot rolling	Sastra et al.	U. Penn	26	module lengths/s
Longest distance running, one charge	SuperBot	Shen et al.	USC/ISI	750	m
Mobile unconnected modules docking	Swarm-bot	Mondada et al.	IRIDIA	16	connecting components
Most robust self-reconfiguration	MTRAN II	Murata et al.	AIST/TiTech	14	nonrepeating attach/detach steps

Figure 2: Hardware achievements in modular robotics.

The table shows that there is still a lot of progress to be made, but it also indicates that a lot of research is being done into the different aspects of modular robotics. The number of modules, the size of modules systems, the robustness and velocity are all being developed and improved.

Three dimensional stochastic reconfiguration

Stochastic modular robots have not only been simulated, but physical implementations have already been developed, built and tested. Paul White et al. [7] created two different modular systems: one where the connection between modules is established by magnets, the other where bonding uses a fluid flow.

Both systems use individual units that are not powered when they are unattached from the main structure. Instead, the modules have to be connected to the structure through random motion in the environment.

In the first implementation, a first module was attached to a magnetic plate, from where it can draw power. The plate and modules are submerged in a tank with vegetable oil which is stirred to create random motion. Another module is floating in the oil and can be attached and detached to the main module, in order to get self-assembly and self-reconfiguration. A 24% success rate was observed in showing this self-assembly and self-reconfiguration, giving a lot of room for improvement in future experiments and research. Photographs of a successful experiment are shown in figure 3.

The second physical implementation of a stochastic modular system used fluid flow to connect modules with each other. The aim for this implementation was to provide a way of bonding modules that can be implemented on a micro scale. The problem with the magnetic bonding on a smaller scale was the bad reliability, but also the difficulty to make electromagnets on such a small scale. The experiment with fluid flow starts with a substrate on the bottom of a tank. The substrate works as a sink, where fluid exits the tank to an external pump. The pump sends the fluid back in to the tank at another location to create motion, and the flow of fluid draws modules to the substrate. The substrate is shaped like one of the sides of a module, so that modules can attach to it. A module that is attached to the substrate can open various valves to direct the fluid flow and draw other modules towards itself, making self-assembly possible. When a valve is closed, a module can be detached again, creating self-reconfiguration. Figure 4 gives an overview of the system.

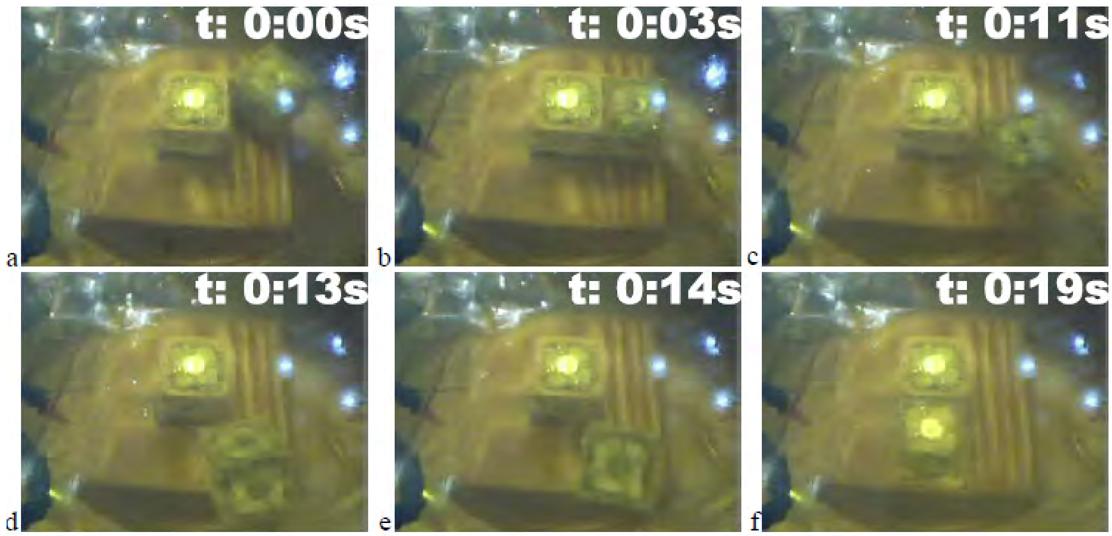

Figure 3: Self-assembly and self-reconfiguration in an experiment with magnetic bonding.

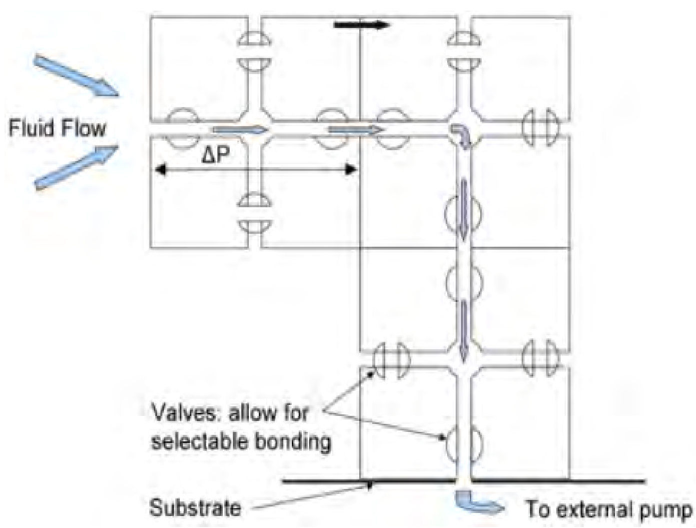

Figure 4: Overview of the fluid flow bonding system.

Distributed algorithms for modular robotics

Progression is also being made with the algorithms that operate modular robotic systems. So far almost all algorithms are centralized, working from the main structure, but it would be far more beneficial if algorithms would be distributed over all modules, meaning that the computational power of all combined modules can be used, and making it possible for the robot to partition itself into multiple functioning robots, which is called self-replication.

Butler, Murata and Rus [9] have made a set of algorithms that allow for simple division and locomotion of modular robots. They have made simulations of the distributed algorithms for both 2D and 3D situations. The robotic systems in the simulations can split up to move in different directions, and recombine if two groups meet up again. The figure below shows the simulation of a modular system that 'replicates' into four different systems. The system determines from the outside to the inside which modules will belong to which part, and then proceeds to move in different directions.

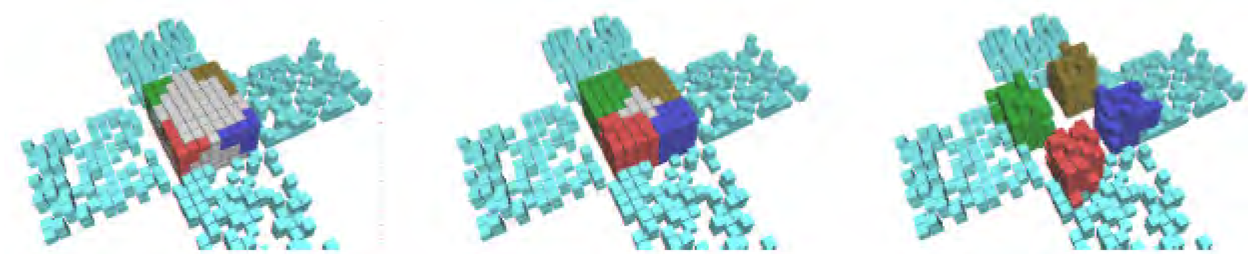

Figure 5: 3D Simulation of self-replication and locomotion of a modular robotic system.

The algorithms provided by Butler, Murata and Rus are only a very basic beginning to algorithms that would have to operate complex autonomous modular systems. But these algorithms can form a basis for more, and the authors themselves are already working on a follow-up study.

3.2 3D printing

A very good example of an upcoming technology that can be combined perfectly with evolutionary algorithms is 3D printing. The idea of 3D printing is very simple: just like a normal inkjet printer, the 3D printer hovers over the place where printing has to take place, and releases its printing material. [15] In case of a 3D printer, this is not ink, but a material that solidifies. Since the printed layer is solid, it becomes possible to print another layer on top of it, and this process continues until a three dimensional object is created.

The technology of 3D printing started as a gimmick embraced by a small group of enthusiasts, but at present day it has become a widely used practical application in many industries, and by many large companies. The areas where 3D printing is used the most can be roughly divided into four categories [15]:

- Manufacturing
- Design
- Architecture
- Medical

In this section the rapid progression of the technology of 3D printing will be shown with examples of applications in the different areas listed above. These examples include small upcoming businesses as well as large companies that see 3D printing as an important new market or use it to replace more classical business processes.

Objet

Objet is a good example of the rapid progression and the many possibilities of 3D printing. The company began in 1998 with the early simplest 3D printers, but today they sell the most advanced 3D printers available to us now.

The most interesting products of Object are their multi-material printers. The use of multiple different materials in a product allows 3D printers to create much more complicated items. The printed items can be assembled goods, or even products with moving parts.

Objet supplies over 60 materials for their 3D printers: 14 materials can be bought in cartridges to use in the printers, and on top of that they have 51 so called digital materials that are a combination of two of their other materials in specific concentrations and structures, to provide different properties, for instance resistance to heat, transparency or bio-compatibility.

In less than 14 years, Objet has developed their products to get from the first simple 3D printers to high tech machines that can print super-thin layers of very diverse materials and create complex assembled products. The products of Objet are mostly used for things like rapid prototyping, rapid tooling, fit testing and concept modeling.

Hewlett Packard

Early 2010, Hewlett Packard decided that the time was ripe for them to get in to the market of 3D printing as well. They made a deal with Stratasys, to begin selling their printers under their own brand. Hewlett Packard has been one of the largest name in the printing industry for a long time, and their initiative to start up their own range of 3D printers shows that 3D printing is becoming a serious business.

So far, Hewlett Packard has brought two printers unto the market. Like their 2D printers, they focus on desktop sized models, mainly targeting designers that want to being their designs to life.

Apart from 3D printers, Hewlett Packard has also developed a 3D scanner, that can make a 3D scan of real objects. A combination of a 3D printer and scanner seems not far away, making it possible for consumers to replicate objects at home. [21, 22]

Google SketchUp and LGM

LGM is a company that provides models of buildings and resorts for architectural firms. They use an engine called CADspan that takes a design made on a computer and transforms it to a file that can be sent to a 3D printer to print out a model of the design. According to the founder of LGM, this technology has enabled them to build architectural models overnight for a price of $2000, instead of a $100000 model built in two months. Because of the speed and the low cost of creating a real model, this enables developers to make lots of changes during the design process, and give a client exactly what he wants.

Figure 6: A model from LGM

The CADspan technology also caught the interest of Google, and they created a plug-in for their design platform SketchUp. Google SketchUp is a tool for designing 3D objects, and the plug-in lets users turn their designs into real models.

The CADspan technology used by LGM and Google shows how 3D printing can take over tasks that previously were expensive on both time and money. It also makes it possible to fully shape a customer's wishes to a unique product, without making it ridiculously expensive. [21]

Boeing and DaimlerChrysler

Large companies that use 3D printing to perfect product design include Boeing and DaimlerChrysler. For both these companies, aerodynamics and fuel consumption are important aspects when designing a part of an airplane or car. The use of 3D printers during design offers considerable benefits.

Before 3D printing, designers would have to make a design, send it to be built elsewhere, and wait for the model to be delivered back to them, before being able to test it. Now, designers can print their own models and test them on the spot. And instead of altering the models by hand to perfect them, they can simply print the altered model straight away.

This technique makes the evolutionary process tangible. Products or designs can be slightly altered, or mutated, during the evolutionary process, but every new solution can be printed and physically tested. It takes a big part of evolution out of the computer and makes it real.

Contour Crafting

Contour Crafting takes 3D printing to a whole new level. Aided by research done on the University of Southern California, they have created a large 3D printer that can build entire houses or even rows of houses.

Figure 7: Impression of the device at work.

The possible applications are emergency housing and low income housing, but also space colonies and commercial applications. Putting a device to print housing on the Moon or on Mars would be a cheap and easy way to set up a colony without humans having to spend extended time in space. Commercial construction of houses could be reduced greatly in cost by 3D printing. There is no waste of materials, there are no labor constraints, and all conduits for electrical, plumbing and air-conditioning can be automatically embedded.

Contour Crafting sees the construction industry even turning in to a consumer market, where people can completely design and build their own house. [21]

Medical applications

3D printing has had a significant influence on the medical field as well. Research in the field aims mostly at surgical aids, drug delivery systems and the creation of bone implants, tissue and organs. [15]

Three dimensional models are used as surgical aids to visualize an operation before starting it. The aids can help with the planning of an operation , the communication with patients, but also the fitting and customization of implants.

Another interesting use are drug delivery systems. Printers for making these are already commercially being used. 3D printing allows pills or capsules to be built from multiple materials and to contain different types of medication, offering a whole new way of designing. This creates a controlled way of accurately releasing medication into the body. The drug release time can be controlled by using alternate layers of powder in the capsule.

The most futuristic use of 3D printing in medicine is the printing of bone implants, tissue or even organs. Implants made from both bone and soft tissue are already been realized. Experiments done [15] show that the use of different kinds of materials in the implant can stimulate cell migration in specific areas, which has major advantages in reconstructive surgery.

The pinnacle of 3D printing in the medical field would be the printing of organs. A printer was already developed that can print single cells and cell aggregates into a 3D gel. The last step of the process of printing on organ, after making a model from scanned date and printing, would be the perfusion of the organ to make it function in a body. So far there has not been any reports of animal or human testing, but the implications are astounding.

Bespoke Innovations

The last example of 3D printing I would like to present is Bespoke Innovations. This company was founded by a designer and a surgeon, that set as their mission to bring personality and humanity to people that suffer from limb loss.


Figure 8: A fairing in use.

Bespoke Innovations offers Fairings, which is their name for the product they sell. Fairings are coverings that surround a prosthetic leg, making it possible to fully personalize and individualize a prosthetic. They use 3D scanning to make a scan of the healthy leg of a client, or use a stand-in if a client is a double amputee. The information from the scan combined with the customization preferences of the client are put in a 3D printer, which then creates the fairing.

Clients of Bespoke will be able to emotionally connect to their prosthetic limbs, and feel like they are part of their own body, of their own personality. [21]

RepRap

The RepRap is a 3D printer. It has a very special feature however that makes it worth naming it. The RepRap is described by its creator as a self-replicating machine. The printer is able to print over half of the components it consist of, making it possible for someone who owns one to build a second RepRap. The other half of the components are available in any small hardware shop, so that practically anyone can build it.

The notion of the customer being able to reproduce a product he has bought, and even the machine that does the reproducing in the first place, has already started raising some concerns and discussions, something that is touched upon in the next chapter. It is also step forward however in the development of self-replicating machines. [21]

3.3 Summary

There is a lot of technological progress to be made to get to the realization of the possibilities of evolvable hardware. Researchers are rapidly making progress however, and so does the commercial industry.

One of the technologies that can be a major part of future developments on evolvable hardware are modular robots. Researchers have been able to build robots that consist of multiple modules, demonstrate self-assembly and self-reconfiguration of these robots, and basic algorithms have been written that make it possible for modular robots to self-replicate, move and reconfigure.

Another promising technology is 3D printing. 3D printing brings an evolutionary approach to a lot of design and manufacturing tasks, and it has been widely used in the medical field as well. The printers today are quickly becoming fully commercialized, and many large companies like Hewlett Packard and Google have entered the market.

Chapter 4: Difficulties of evolvable hardware

As we have seen in the previous chapters, evolution has already been successfully used in computer science, for various applications. Even more thrilling are the promises that evolvable hardware holds, and the possibilities that we have when real autonomous evolutionary hardware arrives.

But despite all these prospects, things like autonomous hardware evolution and self-reconfigurable robots have yet to be realized. [8] This chapter will lay out the difficulties and challenges that are present in the field of evolvable hardware these days.

4.1 Scalability

One of the major problems with evolvable hardware is the scalability. Scalability is a common problem in computer science, and not only research in to evolvable hardware, but also fields like evolutionary computation, artificial neural networks and artificial intelligence in general have that problem. [1] Because of this problem, evolvable hardware has so far only been developed on a smaller scale.

There are two different aspects of evolvable hardware where scalability is an issue. We will go through these two aspects and discuss them further.

Chromosome string length

To be able to evolve a system, the algorithm behind the system must be given something to work with. The system itself, for example the design of a circuit, is captured in a string of bits, called a chromosome after its counterpart in natural evolution. All the information about the system is stored in this chromosome. [17]

When a system grows more complex, the length of the bit string grows, up to the point where it becomes too large to work with. The solution space simply becomes too large, and the evolutionary algorithm cannot work with such large chromosomes efficiently. [1,3] At present, a digital circuit with 100 logical gates, requires a chromosome of about 1000 bits to describe. [1] It is easy to see that this order of complexity, $O(n^2)$, becomes problematic when we want to evolve circuits with much more gates.

Research has been done to reduce the length of chromosomes in more complex systems, both for digital circuits and biological systems. [3, 4] Possible solutions include the use of variable chromosome lengths, computation in parallel and the evolution at functional gate level. These solutions have mainly been applied to circuit design, and have not been able to effectively counter the problem. [3]

Torresen offers an interesting approach that is also applicable to biological systems. Experiments where done with an divide-and-conquer strategy: the evolution was not done on the entire system, but on sub-systems. It was shown that this approach significantly reduces the amount of generations that is needed to come to a solution. [4] The chromosome length does not become smaller however, and the scalability problem has yet to be solved.

Fitness computation

Another difficulty with the scalability of evolvable hardware is the computations of the algorithm itself. The more complex a piece of evolvable hardware becomes, the more computation needs to be done to determine the fitness for all the generated solutions. Evolving a solution for relatively small circuits can already take weeks. [1] With that in mind, it is difficult to imagine much larger circuits to be able to evolved, and it also resembles a great challenge when the goal is to create autonomous systems or robots that can evolve online.

4.2 Relying on hardware speed

It might seem strange that relying too much on hardware is a problem when trying to build evolvable hardware. What is meant here, is that the danger lies in trying to replace the development of smarter and faster algorithms with superior hardware speed. At present there is not much statistical or mathematical theory behind evolutionary algorithms, and research done in this area does not make it likely that there will be any major breakthroughs done there in the near future. [1, 17] Therefore it does seem logical to make improvements to the speed of hardware itself in order to make computations easier and faster, and faster hardware sure offers some relief to the computational cost. But the complexity and computational cost of future evolutionary systems will increase just as much, and the speed of dedicated hardware is not the answer to a time complexity issue. [1]

4.3 Working with a black box

When an evolutionary algorithm has been set to work, it is impossible to predict the outcome. It has been given an input, and when the termination condition is met, there will be an outcome. What you do not know, is how the algorithm will come to the outcome. The process in between is a so called 'black box'. This black box is part of the beauty of evolution: it explores corners and niches where people normally would not have looked for a solution. It can however also be a difficulty, because sometimes you do want to know what happens in between. Below are two examples.

Fitness evaluation and verification

An issue that arises when creating evolvable hardware is the question of how to test the piece of hardware to verify that it found a good solution to its problem. It seems so simple: when the algorithm reaches the maximum fitness, it has found the perfect solution. But this just shifts the problem, because how would you calculate this fitness then? [1]

The behavior of robots is a good example of the difficulty with fitness evaluation. First of all a lot of behavior and movement of robots is inherently noisy, which must be taken into account in fitness evaluation. If a robot moves in a way that seems strange in human eyes, that does not mean it will not perform well on the task it was given. [13]

Caution also has to be exercised not to give too much credit to a controller that seems to be working perfect under certain circumstances. [13] The same controller of the robot may work very badly in another scenario, since all conditions and variables can be adopted by the hardware and used in its evolution. Thompson pointed out that the performance of evolvable hardware may depend on factors such as temperature and power supply. [1]

Maintainability

Related to fitness evaluation is the maintainability of evolvable hardware. As said before, the evolution is a black box, and since the output can be incredibly complex, it is very hard to understand, and thus maintain evolved systems if it would have to be done by human experts.

A different approach to maintainability is the use of self-repair. If evolvable hardware would be able to repair itself, and to do that of course also detect its faults, this would entirely remove the need for humans to understand the systems as a whole. [1]

4.4 Ethical concerns

It is not surprising that there are some ethical concerns with the creation of evolving hardware and autonomous robots. Evolvable hardware has a huge potential, and could bring a revolution in many industries, but there are a few concerns someone could have about these revolutions.

Designing and manufacturing jobs

As we have seen from previous chapters, evolution can be incredibly useful to make designs for things like digital circuits. If research can come up with ways to bring down the high computational costs and the time it takes to evolve a design, it could become a common practice to let evolution make all designs. This would mean that human experts would no longer be needed at all.

The same argument goes for manufacturing jobs. 3D-printing is an upcoming industry, and machines like the RepRap printer have shown that these printers can be very cheap, and are even capable of reproducing themselves. Of course it is too early to put a 3D printer in every household, but there have been concerns already about consumers being able to produce whatever they need at home. This would put a lot of manufacturing jobs at stake.

Grey goo

Other concerns about evolvable hardware include some end-of-the-world scenarios, of which I would like to discuss two. The first one is the grey goo scenario, which was named by Eric Drexler in his book about molecular nanotechnology. The scenario involves the uncontrollable self-replication of robots, eventually leading to the consumption of all matter on Earth. Research into the possibility of this doomsday scenario was done by the Royal Society in 2004. [14] They concluded, based on the opinion of Drexler himself as well, that this scenario is currently very unlikely, both because scientist are not able to build such a self-replicating nano-robot, and because there would be no practical use to build such a machine at all. Debate about this is still open though, and it is something to consider in the future when the creation of such nano-robots could become possible.

Runaway scenario

The runaway scenario is the most popular end-of-the-world scenario concerning robots, both in science and popular science fiction . The scenario does not need much explaining: humans build machines capable of evolving, and eventually these machines become so intelligent that they revolt against humanity and take over control over the planet. In reality, these machines do not necessarily need to become completely self-aware and overly intelligent, and they certainly do not need the voice of Arnold Schwarzenegger. In his manifest from 1995, Kaczynski argues that it would be enough if evolutionary systems would be made to take over the decision-making and problem-solving of such complexity, that humans would no longer be able to retain sufficient control over the systems, because the problem are too difficult to understand, but it would become impossible to shut them down too, because we have become dependent of the solutions. This would result in a situation where humans would have no choice but to accept any decisions the autonomous evolving systems would make. [19]

Misuse of advanced technology

Evolvable hardware has all sorts of fantastic future possibilities, one even better than the other, but a valid concern of many people is the intentions of the people that might be using this technology. In this case, the culprits would most likely be governments who, unlike terrorist organizations, have the resources to be able to develop this advanced technology.

Just like the possibilities of evolvable hardware seem endless for medical purposes or use in industries, just as endless are the possibilities of governments to misuse these technologies. It is scary to think of malicious nano-robots to be used in warfare to kill enemy soldiers, or evolved systems and robots to be used as weapons.

On top of that, these technologies could also be used at home, in the name of security. Governments would use evolved autonomous systems to monitor streets and households, which could lead to heated debates on civil liberties and privacy. [19]

Intellectual property and copyright

Specially the idea of self-replicating or self-reproducing machines has become a concern for people that deal with the laws of intellectual property and copyright. There are plenty of laws that protect copyrights, patents and trademarks alike, and many of these laws already had to be expanded to include intellectual property, but 3D printers will start the discussion all over again.

Consumers will be able to repair machines they bought with spare parts they print themselves, or scan and create a few new dinner plates when they break some while doing the dishes. Maybe they even decide to improve a design from a manufacturer and start selling the results themselves, or they completely make their own designs.

It will be difficult to fit these new possibilities in current laws, and the large incumbent companies will try to counter and restrain these possibilities as much as they can, to avoid lost sales, lower profits and reduced employment. It will come down to policymakers and judges to weight these new possibilities and developments against the inevitable losses in the current industries, and make a good judgment. [23]

4.5 Summary

Evolvable hardware is a promising field of research, and the possibilities seem endless. Before scientists would ever get to the point where evolvable hardware and evolving autonomous systems are build in reality, there are quite a few obstacles to take care of.

First there are the obstacles that relate to the complexity of building evolving systems, and the high cost of the computations in evolutionary algorithms. Smarter and faster algorithms will have be developed to be able to cope with the increasing complexity of these systems, and to avoid the dependency on hardware speed.

Another problem is the black box design of evolutionary systems. Currently it is already impossible for human experts to verify circuits that can be created by evolution, and the complexity will only keep increasing.

Finally there are the ethical concerns with these advanced technologies. There are the issues with evolution taking over manufacturing, which would lead to lots of jobs being lost. Furthermore we have the doomsday scenarios that might seem far away, but should be kept in mind when creating more complex and intelligent evolving systems. And finally there is the potential of misuse of these technologies.

Concluding remarks

Reading this paper should leave no doubt that evolution is a powerful tool that can be used both in software as in hardware to come up with smart problem solvers and intelligent evolving systems. The last few years the development of evolvable hardware has taken a flights, and a scientific breakthrough could be coming soon.

It should be noted however that the future possibilities in the second chapter are, although very plausible and likely to become true in perhaps twenty or thirty years, so far still speculations of the author and so far not yet realized.

In 1877, the Western Union company declared, in a response to an offer for all rights to the telephone that:

"This 'telephone' has too many shortcomings to be seriously considered as a means of communication. The device is inherently of no value to us."

In 1943, the chairman of IBM Thomas J. Watson allegedly said about computers:

"I think there is a world market for maybe five computers."

In my personal opinion, the eventual possibilities of evolvable hardware could have the same impact on our lives as the telephone and computer turned out to have. And with the above quotes in mind, it is going to be very interesting how the world looks like in twenty or thirty years.

Literature

[1] X. Yao and T. Higuchi, "Promises and Challenges of Evolvable Hardware." *IEEE Transactions on Systems, Man, and Cybernetics, C, 29.* February 1999.

[2] T. Higuchi, M. Iwata, D. Keymeulen et al., "Real-World Applications of Analog and Digital Evolvable Hardware." *IEEE Transactions on Evolutionary Computation, 3.* September 1999.

[3] J. Torresen, "An Evolvable Hardware Tutorial." In proc. of *14th International Conference on Field Programmable Logic and Applications.* August 2004.

[4] J. Torresen, "A Divide-and-Conquer Approach to Evolvable Hardware." *Second International Conference on Evolvable Hardware.* September 1998.

[5] O.M. Querin, V. Young, G.P. Steven and Y.M. Xie, "Computational efficiency and validation of bi-directional evolutionary structural optimisation." *Computer Methods in Applied Mechanics and Engineering, 189*, *559-573.* September 2000.

[6] M. Yim, W. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins and G.S. Chirikjian, "Modular Self-Reconfigurable Robot Systems." *IEEE Robotics and Automation Magazine, 14, 1, 43-52.* March 2007.

[7] White, P.J., V. Zykov, J. Bongard and H. Lipson, "Three dimensional stochastic reconfiguration of modular robots." In *Proc. of Robotics Science and Systems.* 2005.

[8] K. Stoy, H. Kurokawa, "Current Topics in Classic Self-reconfigurable Robot Research." *Proceedings of the IROS Workshop on Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems*. 2011.

[9] Z. Butler, S. Murata, and D. Rus, "Distributed replication algorithms for self-reconfiguring modular robots." *Distributed Autonomous Robotic Systems*, *5,* 37-48, 2002.

[10] T.G.W. Gordon and P.J. Bentley, "On Evolvable Hardware." In *Soft Computing in Industrial Electronics*, *279-323*. 2002.

[11] F. de Toro Negro, J. Ortega, E. Ros, S. Mota, B. Paechter and J.M. Martin. "PSFGA: Parallel processing and evolutionary computation for multiobjective optimisation." *Parallel Computing, 30, 5-6, 721-739.* 2004.

[12] C. Lambert, T. Kalganova and E. Stomeo. "FPGA-based Systems for Evolvable Hardware." *Proceedings of World Academy of Science, Engineering and Technology, 12.* March 2006.

[13] J.-A. Meyer, P. Husbands and I. Harvey. "Evolutionary Robotics: a Survey of Applications and Problems", *Proceedings of The First European Workshop on Evolutionary Robotics - EvoRobot98.* 1998, reviewed March 28, 2008.

[14] The Royal Society & The Royal Academy of Engineering (RS&RAE). "Nanoscience and nanotechnologies: opportunities and uncertainties", *London: The Royal Society & The Royal Academy of Engineering*. 2004.

[15] D. Dimitrov, K. Schreve and N. De Beer. "Advances in three dimensional printing- state of the art and future perspectives." *Rapid Prototyping Journal, 12, 3, 136- 47*. 2006.

[16] C. Darwin. "On the Origins of Species." *John Murray.* November 1859.

[17] A.E. Eiben and J.E. Smith. "Introduction to Evolutionary Computing." *Springer. Natural Computing Series.* 2003.

[18] L. Sekanina. "Evolvable Components: From Theory to Hardware Implementations." *Springer. Natural Computing Series.* 2004.

[19] A. Krishnan. "Killer robots: legality and ethicality of autonomous weapons." *Ashgate Publishing Limited.* 2009.

[20] A. Hodges. "Alan Turing." *The Stanford Encyclopedia of Philosophy.* June 2011.

[21] A. Vance. "3-D Printing Spurs a Manufacturing Revolution." *New York Times*, September 13, 2010. Online. http://www.nytimes.com/2010/09/14/technology/14print.html?_r=1 , March 19, 2012.

[22] P.-E. Gobry. "The Next Trillion Dollar Industry: 3D Printing." *Business Insider,* February 25, 2011. Online. http://www.businessinsider.com/3d-printing-2011-2?op=1 , March 19, 2012.

[23] M. Weinberg. "It will be awesome, if they don't screw it up." November 2010.

[24] A. Batey. "2Excel Aviation Gives Drones Virtual Eyes." *Business Week,* March 22, 2012. Online. http://www.businessweek.com/articles/2012-03-22/2excel-aviation-gives-drones-virtual-eyes , March 23, 2012.