TECHNISCHE UNIVERSITÄT DRESDEN
Fakultät Informatik

TECHNISCHE BERICHTE
TECHNICAL REPORTS

ISSN 1430-211X

TUD-FI15-05-Oktober 2015

Dr. Frank J. Furrer, Georg Püschel (Eds.)
Institut für Software- und Multimediatechnik

Cognitive Computing (Collected Papers)

Technische Universität Dresden
Fakultät Informatik
D–01062 Dresden
Germany
URL: http://www.inf.tu–dresden.de/

Hauptseminar im Sommersemester 2015

Dr. Frank J. Furrer

Cognitive Computing

Collected Papers

Editors: Dr. Frank J. Furrer, Georg Püschel

Technische Universität Dresden
Technical Report TUD-FI15-05-Oktober 2015

"Cognitive technologies are products of the field of artificial intelligence.They are able to perform tasks that only humans used to be able to. [1]"

Table of Contents

Introduction

Dr. Frank J. Furrer

Context

"Cognitive Computing" has initiated a new era in computer science. Cognitive computers are not rigidly programmed computers anymore, but they learn from their interactions with humans, from the environment and from information. They are thus able to perform amazing tasks on their own, such as driving a car in dense traffic, piloting an aircraft in difficult conditions, taking complex financial investment decisions, analysing medical-imaging data, and assist medical doctors in diagnosis and therapy [2][3][4]. Cognitive computing is based on artificial intelligence, image processing, pattern recognition, robotics, adaptive software, networks and other modern computer science areas, but also includes sensors and actuators to interact with the physical world [1].

Cognitive computers – also called "intelligent machines" – are emulating the human cognitive, mental and intellectual capabilities. They aim to do for human mental power (the ability to use our brain in understanding and influencing our physical and information environment) what the steam engine and combustion motor did for muscle power. We can expect a massive impact

of cognitive computing on life and work. Many modern complex infrastructures, such as the electricity distribution grid, railway networks, the road traffic structure, information analysis (big data), the health care system, and many more will rely on intelligent decisions taken by cognitive computers.

A drawback of cognitive computers will be a shift in employment opportunities [5]: A raising number of tasks will be taken over by intelligent machines, thus erasing entire job categories (such as cashiers, mail clerks, call and customer assistance centres, taxi and bus drivers, pilots, grid operators, air traffic controllers, ...). A possibly dangerous risk of cognitive computing is the threat by "super intelligent machines" to mankind [6]. As soon as they are sufficiently intelligent, deeply networked and have access to the physical world they may endanger many areas of human supremacy, even possibly eliminate humans. Cognitive computing technology is based on new software architectures – the "cognitive computing architectures" [7][8]. Cognitive architectures enable the development of systems that exhibit intelligent behaviour.

Seminar Work

This seminar worked on answers to the central question:

> *Which are the situation, the challenges, and the impact of cognitive computing in the year 2025?*

The focus lies on 3 relevant areas:

Q1 Which are the promising software architectures for cognitive computing?

Q2 How does cognitive computing enable future applications?

Q3 What is the impact of cognitive computing on people, work and society?

The Hauptseminar work was structured as follows:

- An introduction day: Cognitive Computing was introduced in an initial lecture by Prof. Dr. Frank J. Furrer (seminar kick-off day),
- Individual, guided research in the selected area and authoring of a scientific paper. Feedback from peer reviewers,
- A first seminar day: The participants presented their results and received feedback from the audience,
- Improvement of the paper and the presentation, based on the peer review feedback,
- A second seminar day: The participants presented their improved results and received feedback from the audience,
- Delivery of the final paper.

The participants learned: (a) to do focused research in a specific area ("Cognitive Computing"), (b) to author a scientific paper, (c) to improve their LaTex expertise, (d) to experience the peer-review process, both in giving and receiving peer review advice and (e) to hold convincing presentations, and (f) to benefit from a considerable broading of their perspective in the field of technology, software, applications, and impact.

As a final outcome of the seminar, a proceedings volume including all the papers produced by the participants was assembled (= this report) and is made available in electronic form to anybody interested. Seminar language was English. Three seminar days were held and 3 ECTS credits were awarded for the successful participation. Audience was limited to 8 participants.

My sincere thanks go to **Georg Püschel** for his highly valuable assistance during the seminar and for assembling this Technical Report. Thanks also to the 7 participants, whose active collaboration and innovative work is greatly appreciated.

References

[1] David Schatsky, Craig Muraskin, and Ragu Gurumurthy. Demystifying artificial intelligence – what business leaders need to know about cognitive technologies. Downloadable from: http://dupress.com/articles/what-is-cognitive-technology/ [last accessed: 31.12.2014], 2014.

[2] John E. Kelly III and Steve Hamm. *Smart Machines: IBM's Watson and the Era of Cognitive Computing*. Columbia University Press, N.Y., USA, 2013.

[3] Erik Brynjolfsson and Andrew McAfee. *The Second Machine Age – Work, Progress, and Prosperity in a Time of Brilliant Technologies*. WW Norton & Company Inc., N.Y., USA, 2014.

[4] Eric W. Brown. Cognitive computing ushers in new era of it. Downloadable from: http://www.forbes.com/sites/ibm/2014/02/03/cognitive-computing-ushers-in-new-era-of-it/ [last accessed: 8.9.2014], 2014.

[5] Harry Rudin. Will the it revolution cost our children their jobs? Downloadable from: http://ercim-news.ercim.eu/en99/challenges-for-icst/will-the-it-revolution-cost-our-children-their-jobs [last accessed: 7.10.2014], 2014.

[6] James Barrat. *Our Final Invention – Artificial Intelligence and the End of the Human Area*. Thomas Dunne Books (St. Martin's Press), New York, N.Y., USA, 2013.

[7] Jerzy W. Rozenblit. Cognitive computing: Principles, architectures, and applications. In *Proceedings of the 19th European Conference on Modelling and Simulation (ECMS)*, 2005.

[8] John E. Laird. *The SOAR (State, Operator And Result) Cognitive Architecture*. The MIT Press, Cambridge MA, USA, 2012.

Participants

Each participant had to choose one focus and elaborate on one specific theme related to the focus question. The following choices were made by the participants:

Name	Q1	Q2	Q3
Mai, Felix	X		
Schröppel, Philipp	X		
Nett, Tobias	X		
Butler, Tyler	X		
Blau, Ben		X	
Richter, Robert			X

Hauptseminar Papers

The following papers were authored, peer-reviewed and presented during the Hauptseminar. All papers are reproduced in full on the following pages.

Applying the Subsumption Architecture to the Genesis Story Understanding System

A Notion and Nexus of Cognition Hypotheses

Felix Mai

TU Dresden
felix.mai@tu-dresden.de

Abstract. Language and particularly story understanding is a fundamental component of human-level intelligence. It enables us to predict consequences of actions without examining these actions before, making investigations according to story understanding a necessary path on our way to artifical intelligence. This paper aims to improve the story understanding system *Genesis* by applying the layer concept of the *subsumption architecture* to the system's variable interpretations capability. Therefore the underlying *cognition hypotheses* as well as their implementations are described. Additionally, a notion of cognition hypotheses is given. The paper finally concludes that the layer concept improves the system's extensibility and paves the way for real-time behavior. Also, further investigations of combinations of cognition hypotheses are motivated.

Keywords: Cognitive computing, artificial intelligence, cognitive science, hypothesis, cognition hypothesis, story understanding, subsumption architecture, nexus

1 Introduction

Cognitive science can be seen as the intersection of cognitive neuroscience, cognitive psychology and computer science, namely artificial intelligence (AI). While neuroscience is focussing rather on the brain's hardware and psychology explores the brain's software, AI combines their results in form of cognitive architectures—or systems—to investigate the implications of their outcomes. In fact, emerged theories are (partially) implemented in such architectures in form of *cognition hypotheses*.

1.1 A Powerful Feedback Loop

A cognition hypothesis can be seen as an interface between neuroscience or psychology, respectively, and AI in a powerful feedback loop: Once a cognition hypothesis is proposed it can be implemented as a software artifact within a cognitive architecture. That artifact can then be evaluated which leads eventually

to the adaption of the proposal by incorporating the observed results. This in turn starts the loop again which allows to refine the hypothesis iteratively. As can be seen in figure 1 the feedback loop constitutes a duality of proposing and adapting as well as implementing and evaluating a cognition hypothesis.

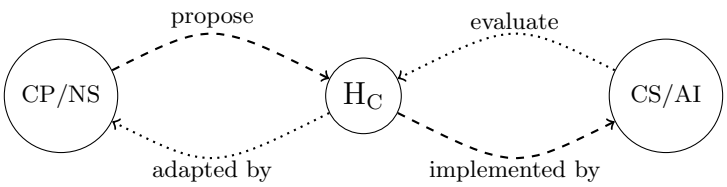


Fig. 1. The feedback loop between cognitive psychology (CP) and neuroscience (NS) on the left side and computer science (CS), i. e. AI, on the right side with the cognition hypothesis (H_C) as mediating interface in the middle.

Furthermore, the observed results allow conclusions regarding the reflected human's counterpart. Another reason why implementing cognitive architectures according to cognition hypotheses is given by the computational theory of mind—also referred to as computationalism—which states that a computational model that embodies a certain theory eases the determination of possible implications [5].

Since cognitive science tries to reproduce the large range of the human's high-level cognitive abilities a particular cognition hypothesis can be interpreted as a (partial) representation of a cognitive capability. Examples for cognitive capabilities might be reasoning, problem solving and understanding of natural language [4,5,6].

1.2 Story Understanding

As stated above, language seems to be a fundamental ingredient when attempting to reproduce human-level intelligence [5,4]. Furthermore, Patrick H. Winston argues in his well-elaborated paper [8] that language empowers humans to tell stories—ranging from fairy tales over teaching lessons to recipes—and that this ability enables us to predict consequences of changes in our real world. For example, one can predict what would happen if a golf ball is dropped into a cup of water without doing an experiment before [5]. He further hypothesizes that *story understanding* plays a key role in the process of human thinking. This paper also focusses on the process of understanding stories and tries to contribute to that topic by suggesting an alteration of its exploratory implementation called *Genesis*.

1.3 Structure of the Paper

The remainder of this paper is organized as follows: Since cognition hypotheses seem to be a reasonable basis for investigations concerned with human cognition section 2 tries to give a notion of *cognition hypothesis*. Then in section 3 an attempt to improve the process of story understanding is described. Therefore, first the underlying cognition hypotheses are presented and their implementations described. After that, the improvement is proposed and its consequences discussed. The paper is finally completed by concluding the contributions of this work and suggesting further work in that direction.

2 Cognition Hypotheses—A Notion

A clear definition of terms is important. Moreover, two other reasons encourage to define what a *cognition hypothesis* denotes. First, a definition in the context of cognitive architectures disambiguates the term "hypothesis" in the field of AI[1]. Second, a definition provides a uniform name for referring consistently to explanations concerning cognitive capabilities in the future.

The term "cognition hypothesis" apparently consists of two parts: The "cognition" part of the term indicates that the hypothesis belongs to the field of cognitive sciences and provides a delimitation to other meanings in the field of AI, e. g. abductive reasoning. On the other hand, the "hypothesis" part denotes a tentative explanation which can be validated by further *investigations*. The term "investigation" in the context of computer science can be interpreted as an experimental evaluation which exhibit information about the correctness of a certain subject—in this case that of a cognition hypothesis.

This clearly implies that cognition hypotheses should be evaluated by means of a cognitive architecture as stated earlier in section 1. Thus, a cognition hypothesis embodies a) the (partial) reflection of a cognitive capability in the context of cognitive psychology and neuroscience, and b) a software artifact implementing that cognitive capability to a certain extent in the context of a cognitive architecture. This again constitutes the duality between cognitive psychology and neuroscience as well as artificial intelligence.

3 Improving Story Understanding

The ability to understand natural language is considered as a fundamental cognitive capability. Furthermore, language seems to have direct impact on human thinking [4]. Winston [8] argues that especially the ability to understand stories contributes crucially to the process of human thinking. An example he gives involves the apparent conflict between Estonia and Russia in 2007: *After the Estonians relocated an old Soviet war memorial the country's computer network*

[1] Abductive reasoning—a form of logical inference—uses that term, too, to describe an intermediate step.

was extensively attacked. When we consider only the last sentence we are able to interpret that this was maybe an act of revenge by Russia in reference to the controversal relocation of the memorial. The result of this interpretation can be extended by judging the attack regarding one's justification. This judgement would clearly depend on the cultural background as discussed in section 3.3.

This section is dedicated to suggest an improvement of the process of story understanding. Therefore the underlying hypotheses, namely the *Strong Story Hypothesis* and the *Physical Symbol System Hypothesis*, are described as well as their associated implementations explained. The last part of this section presents a possible improvement of the story understanding process and discusses probable consequences.

3.1 The Strong Story Hypothesis

As mentioned above Winston hypothesizes that story understanding is a crucial concept for human thinking which eventually resulted in his *Strong Story Hypothesis* [8]:

> *The mechanisms that enable humans to tell, understand, and recombine stories separate human intelligence from that of other primates.*
>
> — Winston, 2011

In this context a story can be understood as a generic concept which encodes information and their relations by means of continuous text. Further, a story provides—beside its explicitly stated facts—implicit information as the example above has shown.

To investigate that cognition hypothesis an exploratory implementation was developed—called *Genesis*—which is actively developed at the MIT under the direction of Patrick H. Winston[2].

The system applies three subsequent steps to extract an interpretation from a simplified story[3]. First, the text is parsed and the given plot translated into a semantic net. Then, *commonsense rules* are applied to enrich the semantic net which produces a so called *elaboration graph*. Finally, a particular set of *reflection patterns* derive an *interpretation* of the given story. Figure 2 depicts that workflow schematically.

The commonsense rules as well as the reflection patterns can be seen as inference rules. While commonsense rules are represented as simple if-then statements reflection patterns instantiate much more complex reasoning rules. The elaboration graph—as an intermediate result—arises from the initial semantic net and the additional inferred knowledge from the commonsense rules.

The usage of a semantic net, i. e. a set of entities related to each other, and inference rules, which are used to produce additional entities and relations, remind of Allen Newell and Herbert A. Simon's *Physical Symbol System Hypothesis* which is explained in the following.

[2] `https://groups.csail.mit.edu/genesis`

[3] Genesis can only process simplified English, although there are efforts to overcome this limitation [8]

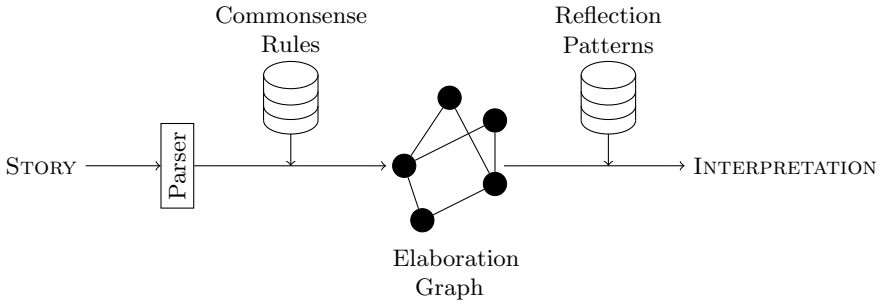


Fig. 2. The workflow of story understanding in Genesis (schematic)

3.2 The Physical Symbol System Hypothesis

In 1976 Newell & Simon posited the *Physical Symbol System Hypothesis* as follows [7]:

> *A physical symbol system has the necessary and sufficient means for general intelligent action.* — Newell & Simon, 1976

As the cognition hypothesis states, physical symbol systems represent another fundamental pillar of human-level intelligence though it is not sufficient enough to provide high-level cognitive capabilities by its own—much like the Genesis system which cannot succeed without a physical symbol system. The elaboration graph as well as the commonsense rules and reflection patterns can be considered as an instance of a physical symbol system. Without these representations understanding and interpreting stories would obviously not be possible, thus couples the Strong Story Hypothesis tightly to the Physical Symbol System Hypothesis.

Newell & Simon define a symbol system to be consisting of symbols which are related to each other and thus forming symbol structures. The system is complemented by a collection of processes which are able to manipulate these symbol structures in terms of creation, modification, reproduction and destruction. The term "physical" indicates that such a system can be physically constructed by engineering [7].

The purpose of such a system is to provide knowledge in a machine readable way. Popular and widely used implementations of such a system might be RDF[4] and OWL[5] ontologies, where symbols correspond to URIs and literals, symbol structures to triples, and where processes are represented by axioms and rules.

3.3 Variable Story Interpretations

As mentioned above, in the context of the 2007's conflict between Estonia and Russia interpretations are possible which depend on the cultural background. For

[4] `http://www.w3.org/RDF`
[5] `http://www.w3.org/2001/sw/wiki/OWL`

example, a proponent of Russia might see the attack as "teaching a lesson" while a proponent of Estonia probably argues towards revenge [8]. Winston suggests that by alterating the reflection patterns to include cultural background the above presented variation of an interpretation can be achieved. Other examples of culturally biased story understanding is given in [1] and [9].

Similar to the cultural background, the social and situational context can bias an interpratation of a story as well. For example, telling that a taxi turns around the corner maybe entails that a Briton expects a black car while a German expects a pale yellow car. These expectations clearly depend on the cultural background. But if a taxi—no matter of which country—is nearby, both might agree on the same car color which then is most probably an interpretation based on the situational context. By contrast, a discussion in a pro-Estonian or pro-Russion group of people might influence the opinion as well which then would be a socially effected interpretation of the conflict story.

The process of deriving such interpretations seems to be done in a layered way as the examples above suggest. Thereby the cultural context might be subsumed and extended by the social context while the social context might be subsumed by the situational context. This is reminiscent of Rodney A. Brooks' *subsumption architecture* which is described in the following.

3.4 The Subsumption Architecture

As Brooks argues in his paper [2] a robust robotic system should be constructed by layers each of which extend the competence of that system. By competences Brooks denotes the system's abilities to manage a certain task ranging from low-level competences, like object avoiding and wandering, to more sophisticated abilities, like exploring and seeking. Thereby each level subsumes the capability of the preceeding one which gives that architecture its name. The computation of any action is computed for each layer independently and in parallel. Furthermore, any level can interfere with its preceeding competence level by suppressing its input signal or inhibit its output signal. As a consequence a higher level capability, such as route planning, can suppress a low-level behavior, such as avoiding objects. For example, stopping in front of an object can be suppressed as the robot might have found an alternative route when the object was encountered. In turn, if a high-level capability does not suppress a low-level capability, e. g. the hardware is not sufficiently fast enough to compute the alternative route, the low-level capability will proceed, thus providing implicit robustness.

Although Brooks describes that architectural type for the purpose of building robots, the concept of subsuming layers can be assigned to cognitive architectures in general. Considering figure 3 which shows the layered structure of the subsumption architecture, the input and output could be generalized to abstract signals. In terms of story understanding these signals would be a story as input and an interpretation as output. Each competence level then can be seen as a higher level of abstraction which implements a more sophisticated cognitive capability. For example, interpreting a story in the situational context, i. e. taking additional cultural, social and environmental knowledge into account instead of

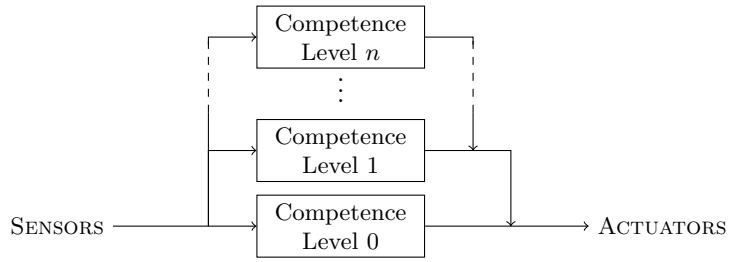


Fig. 3. The general structure of the subsumption architecture

only using cultural background information, seems to be a more evolved cognitive capability.

3.5 The Nexus—A Tought Experiment

After all necessary components were described I propose a nexus of Winston's Strong Story Hypothesis, i. e. its implementation in Genesis which also involves Newell & Simon's Physical Symbol System Hypothesis implicitly, and Brooks' subsumption architecture. As stated above different context levels, e. g. situational, social and cultural, can be implemented by means of layers.

In [3] Brooks argues that intelligent systems should be constructed incrementally. He also claims that at each stage the system's capabilities should be built step by step while remaining self contained to ensure their validity without the need to pay attention to possible sideeffects. Extending these concepts from robots to cognitive architectures in general and story understanding in particular, I believe that the workflow of story understanding (cf. figure 2) can benefit from a construction in a similar way. In fact, the application of the reflection patterns which provide the actual interpretation based on the intermediately produced elaboration graph should be restructured to fit the subsumption architectural style of layered competences which could be seen also as level of cognitive competence or interpretation competences. Instead of applying an exchangeable set of reflection patterns depending on the contextual background each layer can derive its interpretation independently. Since the layers are of subsuming nature the most sophisticated competence which can be applied in a certain context produces the final result probably containing an extended version of the interpretation of the layers below. Figure 4 depicts the small structural change according to the subsumption principle. Thereby the intermediate elaboration graph is passed to each layer which then derives an interpretation. Finally, the competence with the highest level can suppress the output of all lower competences resulting in the most sophisticated interpretation which is possible.

Possible implications of that adjusted workflow are discussed in the following. First, through the layered structure the architecture of the Genesis system might get more extendable since the the subsumption architecture is very extensible by default [2]. This also could ease maintainability since the layers are indepen-

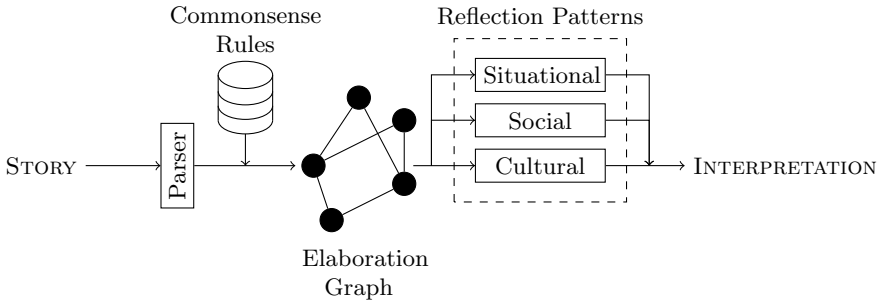


Fig. 4. The nexus of the story understanding workflow and the subsumption architecture

dent of each other. For example, if a new interpretation competence should be introduced a new layer is inserted without the need to touch any other layer. The only thing to do would be to determine what levels are subsumed by the new layer and which level subsumes the new layer. Second, due to the fact that the computation of a particular interpretation can be done in parallel the Genesis system might get a little closer to real-time performance without paying attention to concurrency between the interpretation contexts.

Although the suggested improvement is only theoretical, the idea of applying the subsumption architecture to cognitive architectures in general instead of only robotic architectures convinces through its robustness and extensibility properties which is a strong requirement for complex systems in general.

4 Conclusion

After motivating language as one of the fundamental pillars of human-level intelligence a notion of *cognition hypotheses* was given. The notion is meant to provide a consistent reference for further work while also motivating experimental investigations in the context of cognitive architectures. Therefore a cognition hypothesis was introduced as an interface between cognitive psychology and neuroscience on the one side and artificial intelligence on the other side—constituting a powerful feedback loop.

Diving deeper into the cognitive capacity of processing natural language revealed that story understanding as a subdomain seems to be a crucial concept for the process of human thinking. As a contribution to that field a possible alteration of the Genesis story understanding system was presented. In fact, the concept of layered competences from the subsumption architecture was adapted to the interpreting part of the Genesis workflow. The theoretical consequences discussed include the adoption of the subsumption architecture's robustness and extensibility properties as well as the ability to derive possible interpretations in parallel which paves the way for real-time behavior.

The new knowledge gained in this paper includes that combinding different hypotheses is a) sometimes necessary and b) does almost always emerge to an improvement of the cognitive architecture. The Genesis system is a good example for both cases. On the one hand, the system uses the *Physical Symbol System Hypothesis* as basis for the *Strong Story Hypothesis* which in fact incorporates a physical symbol system in order to interpret stories. On the other hand, incorporating only the *Strong Story Hypothesis* does not necessarily lead to an intelligent system which is able to interpret stories.

4.1 Further Work

As next steps, following tasks may be addressed: First, the theoretical proposal of the workflow alteration needs to be evaluated in practice. This might also reveal whether the subsumption architecture can be easily adapted to other more general cognitive architectures or not.

Second, I suggest to create a catalogue of existing cognition hypotheses. That catalogue then can be used to a) provide a comprehensive overview about the field of cognitive science and b) indicate compatibilities and further combinations of cognition hypotheses. Beside that, a comprehensive list of emerged cognition hypotheses would be useful to guide research in that field. And how knows? Maybe we are facing a new development paradigm in the context of cognitive architectures, called *hypothesis-driven* or *-centered development*.

Acknowledgement. This paper was elaborated during the research seminar "Cognitive Computing" which was directed and supervised by Dr. Frank Furrer. Additional remarks of other reviewers improved the paper as well.

References

1. H. Awad. *Culturally Based Story Understanding*. Master thesis, Massachusetts Institute of Technology, June 2013.
2. R. A. Brooks. A robust layered control system for a mobile robot. Mar. 1986.
3. R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1–3):139–159, 1991.
4. J. D. Friedenberg and G. Silverman. *Cognitive Science: An Introduction to the Study of Mind*. Sage Publications, Inc., 2006.
5. D. Kolak, W. Hirstein, P. Mandik, and J. Waskan. *Cognitive Science: An Introduction to Mind and Brain*. Routledge, 2006.
6. P. Langley. The cognitive systems paradigm. *Advances in Cognitive Systems*, 1:3–13, July 2012.
7. A. Newell and H. A. Simon. Computer science as empirical inquiry: Symbols and search. *Commun. ACM*, 19(3):113–126, Mar. 1976.
8. P. H. Winston. The strong story hypothesis and the directed perception hypothesis. *Advances in Cognitive Systems: Papers from the 2011 AAAI Fall Symposium (FS-11-01)*, 2011.
9. W. V. H. Yarlott. *Old Man Coyote Stories*. Master thesis, Massachusetts Institute of Technology, June 2014.

Benefits and Drawbacks of Hardware Architectures Developed Specifically for Cognitive Computing

Philipp Schroppel
philipp.schroeppel@mailbox.tu-dresden.de

TU Dresden

Abstract. The Von Neumann computer architecture has been the dominant architecture for all kinds of computer systems over the last 70 years. The upcoming idea of cognitive computing basically meaning that computers act with their environment in new, more natural ways raises the question if new hardware architectures, inspired by the human brain, could be more efficient for its implementation.
By describing and comparing two existing cognitive systems, one based on the Von Neumann architecture and the other one on a brain-inspired architecture, I want to give a prospect on which approach will be more promising for the future.
This paper will provide a summary of the benefits and drawbacks of both architectures and analyze how they are suited for dealing with the key challenges for computer systems in the future.

1 Introduction

Most of the currently used computers are based on the Von Neumann architecture: There is a memory where data is stored and a central processing unit which can perform calculations based on the data. The list of instructions that tell the computer how to process the data has to be provided by the programmer.

The wish for making computers act more natural, simplifying interaction between humans and computers, is not new, but apparently technology is still struggling with achieving this goal.

One promising approach is called cognitive computing, which diverges from the idea of computers that strictly follow predefined instructions given by a programmer. Basically cognitive computing means that computers interact with the environment in new, more natural ways and draw their own conclusions. They learn by themselves and develop new strategies to deal with problems. They can work on data, even if the provided information is not consistent or even contradictory [1].

Today there are several challenges where cognitive computers might be helpful. One example is handling big amounts of data: As more and more data is available today, it is difficult for the programmer to write programs which solve problems while considering all the given information. With cognitive computing

the programmer doesn t have to specify how to process every single piece of information, but the computer itself can decide what is important and find a solution.

2 State of the Art

Today there are cognitive computing systems for Von Neumann machines, but other hardware architectures with completely new software ecosystems have been developed for cognitive computing as well.

Some of the more successful and influential cognitive architectures for Von Neumann machines are SOAR, ACT-R, CLARION and EPIC [2], which have been started to be developed decades ago. Although newer systems, like Google DeepMind, are recently producing impressive results, I want to give an introduction to SOAR as an example for the classical approaches.

On the other hand, there are not too many systems with hardware architectures developed specifically for cognitive computing. One such system is TrueNorth, a brain-inspired architecture developed by IBM. I will give an overview to TrueNorth, because its development has already reached a state where results can be seen and first applications are implemented.

Up to now, it is not clear whether new hardware architectures will become established, or if the Von Neumann will continue to be the leading architecture. Therefore I want to present my opinion on which architecture will be better suited for cognitive computing in the future.

3 SOAR: A Cognitive Architecture for Von Neumann Machines

A cognitive architecture tries to put results of cognitive psychology into a computer program. One of the most successful cognitive architectures is SOAR, which was started to be developed in the 1980 s, when Allan Newell and his students tried to build and implement a Unified Theory of Cognition: A theory that brings together all the research results and theories on cognitive science by different psychological disciplines [3].

3.1 How SOAR Works

Simply put, every behavior shown by SOAR is the result of trying to get from a current state closer to a defined goal state, following the principle of rationality: if an agent has knowledge that an operator application will lead to one of its goals then the agent will select that operator [4].

A state describes a situation the system could be in and consists of all features and values that are important to model the situation.

As the principle of rationality suggests, SOAR will – like a human being mostly does – perform actions in order to achieve a specific desirable situation,

here described by, one or more, goal states. Doing so, the system moves from state to state, trying to get closer to the goal.

Internally, operators are used for these movements between states. An operator can be applied when predefined conditions – expressed by specific features and values – are fulfilled and will cause a state change either by just internally changing some features and values or by making the system perform an action in the real world, which in turn leads to a change of perception that eventually changes features and values as well.

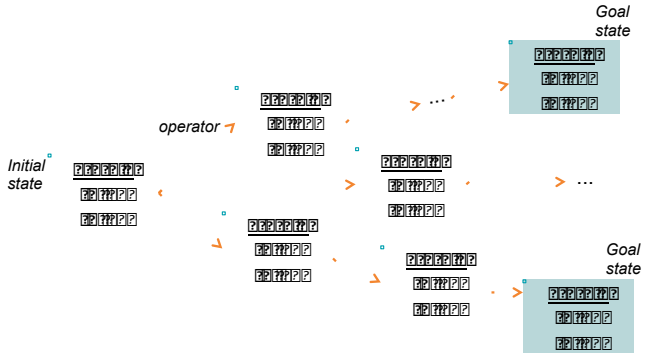


Fig. 1: Working principle of SOAR: The system tries to reach a goal state by moving from state to state using operators. States have features (*f1*, *f2*) with specific values (*v1*, *v2*). Adapted from Fig. 4 in [3].

3.2 How SOAR Stores Knowledge

Although this describes the working principle of SOAR, one more thing is missing for determining the behavior of the system: To decide if an action could be appropriate for the current situation and the current goal, knowledge about the real world has to be included in the system.

In SOAR this knowledge is expressed in terms of states, operators, goals and problem spaces and is – inspired by the human brain – stored in two memory structures: The long term memory (LTM) stores general knowledge and the working memory (WM) has knowledge about the current situation [3].

4 TrueNorth: IBM's Brain-inspired Cognitive Computing System

Even though the Von Neumann architecture is used in almost every modern computer, the human brain has some qualities which make it worth to research in building computers that mimic its structure and working principle.

While state of the art computer chips have the problem of consuming more and more energy and therefore producing too much heat, the brain works with approximately 20 Watt [5]. One single damaged transistor in a chip can mean the end of the whole system, but the human brain, providing a way higher fault tolerance, can still work after being damaged. On top of that, the brain is able to organize itself and find new ways to solve problems instead of just executing predefined program code like computers [6].

A promising attempt to build a brain-inspired computer is a system called TrueNorth which is developed by IBM in the SyNAPSE program and includes a hardware chip and a software ecosystem.

TrueNorth consists of 4096 newly developed cores, which are called neurosynaptic cores by IBM. Each core simulates 256 neurons and 256 synapses per neuron and handles processing power, memory and communication by itself, so there is no need for memory shared by all chips, like in Von Neumann architectures [7]. Communication between the cores happens through spike-events.

While designing TrueNorth, the scientists at IBM have focused on low power, rather than compact size. Because the system uses no clock, but performs computation based on events, no energy is wasted while cores are idle [7].

The following chapter will explain the basic hard- and software architecture of this system.

4.1 The Human Brain

While computer chips have specialized units for different tasks, the computation in the brain is executed by billions of more or less equal neurons, which are strongly connected, forming so-called neuronal networks, and communicate through electrical and chemical signals [8].

Neurons are attached to axons for sending and dendrites for receiving signals [9]. In order to transmit information, the neuron sends electrical signals over the axon. The axon is connected to synapses where the incoming electrical signal results in the release of neurotransmitters. Those neurotransmitters are chemicals that in turn cause electrical activity in the dendrites of other cells, making those cells receive the sent information.

4.2 Hardware Architecture of TrueNorth

Given this brief introduction to the main components of the human brain, we can start regarding how they are implemented in TrueNorth s hardware architecture.

This will be done with a bottom-up approach: First we have a look at the implementation of a single neuron, then, going one step further, how the single neurons are linked together to form big networks.

The Neuron Model. Each of TrueNorth s 4096 cores simulates 256 neurons. This simulation has to be based on a neuron model: A model which mathematically describes behavior of a neuron in a more or less accurate way.

Over the years lots of neuron models have been developed. For the implementation in a computer system it is crucial to find a model which is detailed enough for sufficiently imitating the behavior of a neuron, while being not that detailed that it needs too much computational power [10].

One such model is the leaky integrate-and-fire model, which was developed in 1965 and is used in the TrueNorth system today.

As the name indicates, this model consists of three parts: Integration, fire and leak. The membrane potential of a neuron increases over time by summing up the synaptic input (*synaptic integration*) until it reaches a threshold. When this happens, the neuron fires, which means that it sends out an action potential and resets the membrane potential. For taking leak currents into account, every time step the membrane potential diminishes by a fixed value (*leak integration*) [10].

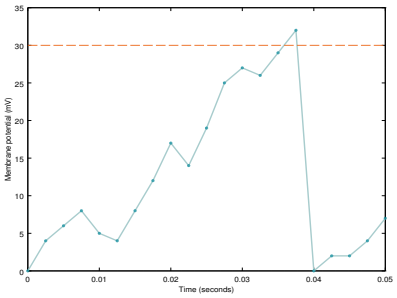


Fig. 2: Membrane potential changes over time by synaptic integration and leak integration. Adapted from Fig. 3 in [10].

Starting out from this model, the scientists at IBM have added several extensions to make their simulated neurons more versatile [10].

The first extension was adding stochasticity. It is possible to switch neurons between a deterministic and a stochastic mode, which affects synaptic and leak integration: While in deterministic mode, synaptic and leak integration happens normally every time step, in stochastic mode it is randomly determined if synaptic and leak integration will happen at all, or if they will be skipped for the current time step. Furthermore stochasticity is also introduced for the threshold, making the threshold value vary by a random number.

Secondly, four different leak modes were supported. The leak can be positive or negative, thus increasing, or decreasing the membrane potential, or it can be divergent or convergent, meaning that it causes the membrane potential to converge towards 0 V or diverge away from 0 V.

The last extension affects the resetting of the membrane potential by providing three different reset modes. The first mode, the normal one, resets the

membrane potential to a fixed value after the threshold is crossed. The second mode resets the membrane potential to the value that the potential was exceeding the threshold. The third mode does not reset the potential at all, relying on synaptic integration and leak integration to diminish the membrane potential.

Building a Network of Neurons. Computation in TrueNorth happens through networks of neurons connected by axons and synapses [5]. Axons transmit output signals from neurons to synapses, which are the connection to other neurons, as it is shown in Fig. 3, where K axons are connected to N neurons by $K \times N$ synapses. All those connections are implemented by separated circuits, allowing parallel communication [5]. The neuron that belongs to an axon can be on the same chip, or on another one in the system.

It is possible to assign different weights for a neuron to synaptic inputs from specific axons. Those weights can be negative as well, making it possible that synaptic inputs decrease the membrane potential.

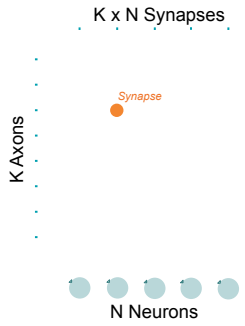


Fig. 3: Synapses as connections between neurons and axons. Adapted from Fig. 1 in [5].

4.3 The Corelet Software Ecosystem for TrueNorth

Today we have several well-established programming paradigms, describing different approaches on how to write programs.

But actually, although they represent fundamentally different styles of computer programming, all of them have in common to express programs through basic blocks consisting of sequential lists of instructions.

In contrast to this – given the hardware architecture of TrueNorth – it should be clear that programs for the new system have to be specified in completely different ways and therefore new programming paradigms are necessary [11].

As every computation in the TrueNorth system is performed by a network of neurons, a program has to be a complete specification of this network, exposing

only external inputs and outputs [11]. On the one hand, the specification of the network includes parameters for the neuron model, as described in 4.2, and on the other hand it provides the anatomy of the network, meaning the connections between neurons within one core and the connections between different neurosynaptic cores.

Therefore scientists at IBM have developed a programming paradigm for TrueNorth which simplifies writing programs, that can be efficiently executed by the hardware [11]. This programming paradigm introduces Corelets, which are abstractions of a TrueNorth program, and hence encapsulate the specification of a network of neurosynaptic cores, exposing only external inputs and outputs. Furthermore the paradigm consists of the Corelet Language for creating Corelets, the Corelet Library, which is a repository of reusable Corelets, and the Corelet Laboratory, a programming environment for the whole programming cycle.

Obviously, as soon as the neuronal networks used for a program get bigger, it is difficult for the programmer to specify the whole network. For this reason, it is possible to compose simple Corelets in order to build more complex ones.

Axons on a Corelet that receive their input from outside of the Corelet are called input connectors and neurons that send their output to destinations outside of their Corelet are called output connectors.

For composing Corelets, the programmer can connect output connectors from one Corelet to input connectors from another one.

Definition and composition of Corelets is done with the Corelet Language. This is an object-oriented language implemented in Matlab OOP that provides classes for neurons, neurosynaptic cores, connectors and corelets.

5 Comparing the Two Architectures

5.1 Key Challenges for Future Systems

After this introduction to two systems which are based on completely different hardware architectures, the question arises if one of them will be superior when it comes to building cognitive systems and computers in general in the future. The Von Neumann architecture has been more or less the leading architecture in the last 70 years, so why should we suddenly need a change? Does the future hold challenges that Von Neumann machines can t cope with?

To answer those questions and eventually compare both systems, it is necessary to identify challenges and key requirements for computing in the future, particularly with regard to cognitive computing.

Although there are multiple upcoming challenges, dealing with a growing amount of available data, the steadily increasing power consumption of computers and the need for new ways of interacting with computers, are probably the most urgent ones for cognitive computing.

5.2 Benefits and Drawbacks of the Two Architectures

Of course the Von Neumann architecture has not been out there for such a long time without a reason. When it comes to performing calculations and applying logic, neither humans nor cognitive systems can keep up with Von Neumann machines today. One also should not forget all the concepts, systems and technologies developed for this architecture over the last decades, which can not be just ported to other systems.

Nevertheless the Von Neumann architecture has some drawbacks, which are long known, but get more and more relevant now and in the near future.

For taking big amounts of data into account for computations, enormous computing power is needed. Until today the computing power has steadily increased, but this process is slowing down. Companies had to stop putting up the clock rates of processors already years ago and for the first time since the seventies, they are struggling to keep up with Moore s Law. The reasons for both of this basically is, that processors consume more and more power and therefore produce more heat. And this heat is starting to get a problem for making computers even faster [12].

In contrast to this, brain-inspired systems have a remarkably small power consumption: While CPUs have a power density of around $50 - 100\,\frac{\mathrm{W}}{\mathrm{cm}^2}$, TrueNorth only needs $20\,\frac{\mathrm{mW}}{\mathrm{cm}^2}$. Even more: TrueNorth can run a typical recurrent network consuming only 70 mW. This is about four orders of magnitude lower than a simulation on a Von Neumann computer would need [7].

Probably Von Neumann machines will always be better than neuronal systems at plain calculations, but for learning new behavior and interacting with their environment they might be left behind pretty soon. Already now, applications like speaker recognition or digit recognition have been developed and work well for the still young TrueNorth system [13]. This gives confidence that brain-inspired architectures can and will deliver fundamentally new ways of interaction between humans and computers.

A major problem for neuronal systems is the development of software. As described in 4.3, the programmer has to specify the anatomy of the neuronal network, meaning he has to define all the connections between the neurons. Although the behavior of single neurons is quite well understood today, it is still not clear how neuronal networks are formed and how the neurons work together [14]. Naturally, this makes it difficult for software developers to set up networks that solve the problem he is dealing with.

6 Conclusion

As Von Neumann computers are used in all areas of technology all over the world, they will definitely be present for the next decades and probably won t disappear at all. Completely discarding the Von Neumann architecture would mean leaving years and years of research behind and it is likely that it will take very long to develop equivalent technologies for new hardware architectures.

But one should also be aware that Von Neumann computers have several advantages, as described in the previous chapter, that justify to continue using them, not just for the reasons of compatibility with old technologies.

However, regarding cognitive computing, apparently brain-inspired systems are better suited. IBM describes them as slow, sensory, pattern recognizing machines , which makes them good at working in real time with all kinds of – probably noisy – data from sensors, and therefore well-suited for interacting with their environment and humans.

In the future, there won t be just one dominant system, but it will rather be important for researchers and engineers to know benefits of both systems and decide depending on the problem which system is more efficient.

One can also imagine combining both architectures, to get the best out of both of them to build powerful computers that can easily learn new behavior, adapt to their environment and interact naturally with humans.

Brain inspired computers are still a very new technology, but one can be sure that after 70 years of having the von Neumann architecture as the dominant architecture for all kinds of computer systems, brain-inspired architectures will provide a promising alternative in the future, at least for specific areas like cognitive computing.

References

[1] IBM Research. Cognitive computing. [Online] Available from: http://www.research.ibm.com/cognitive-computing/#fbid=hD9eIGrGTwH. [Accessed 20-May-2015].

[2] Wikipedia. ACT-R. [Online] Available from: https://en.wikipedia.org/wiki/ACT-R#What_ACT-R_looks_like. [Accessed 01-August-2015].

[3] Jill Fain Lehman, John Laird, and Paul Rosenbloom. Gentle introduction to SOAR, an architecture for human cognition: 2006 update. [Online] Available from: http://ai.eecs.umich.edu/soar/sitemaker/docs/misc/GentleIntroduction-2006.pdf. [Accessed 01-August-2015].

[4] Allen Newell. The knowledge level. Technical report, Carnegie Mellon University, 1982. Available from: http://repository.cmu.edu/cgi/viewcontent.cgi?article=2616&context=compsci.

[5] Paul Merolla et al. A digital neurosynaptic core using embedded crossbar memory with 45pj per spike in 45nm. *IEEE Custom Integrated Circuits Conference (CICC)*, Sept. 2011. Available from: http://www.modha.org/papers/012.CICC1.pdf.

[6] Karlheinz Meier. Computer nach dem vorbild des gehirns? *Ruperto Carola*, 1, 2007. Available from: http://www.uni-heidelberg.de/presse/ruca/ruca07-1/vorbild.html, [Accessed 01-August-2015].

[7] Dharmendra S. Modha. Introducing a brain-inspired computer. [Online] Available from: http://www.research.ibm.com/articles/brain-chip.shtml, 2015. [Accessed 07-August-2015].

[8] Eric H. Chudler. Millions and billions of cells: Types of neurons. [Online] Available from: https://faculty.washington.edu/chudler/cells.html. [Accessed 07-August-2015].

[9] Eric H. Chudler. Making connections: The synapse. [Online] Available from: https://faculty.washington.edu/chudler/synapse.html. [Accessed 08-August-2015].

[10] A. S. Cassidy, P. Merolla, J. V. Arthur, S. Esser, B. Jackson, R. Alvarez-Icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman, A. Amir, D. Rubin, F. Akopyan, E. McQuinn, W. Risk, and D. S. Modha. Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores. *International Joint Conference on Neural Networks (IJCNN)*, 2013. Available from: http://www.research.ibm.com/software/IBMResearch/multimedia/IJCNN2013.neuron-model.pdf.

[11] A. Amir, P. Datta, A. S. Cassidy, J. A. Kusnitz, S. K. Esser, A. Andreopoulos, T. M. Wong, W. Risk, M. Flicknerand R. Alvarez-Icaza, E. McQuinn, B. Shaw, N. Pass, and D. S. Modha. A corelet language for composing networks of neurosynaptic cores. *International Joint Conference on Neural Networks (IJCNN)*, 2013. Available from: http://www.research.ibm.com/software/IBMResearch/multimedia/IJCNN2013.corelet-language.pdf.

[12] John E. Kelly and Steve Hamm. *Smart Machines: IBM's Watson and the Era of Cognitive Computing*. Columbia University Press, New York, NY, USA, 2013.

[13] S. K. Esser, A. Andreopoulos, R. Appuswamy, P. Datta, D. Barch, A. Amir, J. Arthur, A. S. Cassidy, P. Merolla, S. Chandra, N. Basilico, S. Carpin, T. Zimmerman, F. Zee, M. Flickner, R. Alvarez-Icaza, J. A. Kusnitz, T. M. Wong, W. P. Risk, E. McQuinn, and D. S. Modha. Cognitive computing systems: Algorithms and applications for networks of neurosynaptic cores. *International Joint Conference on Neural Networks (IJCNN)*, 2013. Available from: http://www.research.ibm.com/software/IBMResearch/multimedia/IJCNN2013.algorithms-applications.pdf.

[14] Wikipedia. Brain. [Online] Available from: https://en.wikipedia.org/wiki/Brain. [Accessed 07-August-2015].

Language Workbench Technology For Cognitive Systems

Tobias Nett

Technische Universität Dresden
Institut für Software- und Multimediatechnik
Lehrstuhl für Softwaretechnologie
tobias.nett@tu-dresden.de

Abstract The combination of cognitive technologies such as artificial intelligence and robotics and the advent of cognitive computing usher a new generation of complex software systems. In the last years, various approaches on cognitive architectures have been taken and they established themselves for solving domain-specific problems. As a consequence, the design and implementation of cognitive systems gains importance and new tools are required to ensure quality software.
In this paper, we present language workbench technology as instrument for improving the software development process for cognitive systems for both system developers and domain experts. Language workbenches create an enabling environment for domain-specific languages and supporting tools, e. g., through integrated development environments and concise toolchains. We propose to apply language workbench technology to create a language ecosystem for cognitive technologies which can then take advantage of diverse notations and specialized tools.

Keywords: Cognitive Computing, Domain Software Engineering, Language Workbench Technology, View-based Programming, Domain-specific Language

1 Introduction

Within the last years, computers became capable of doing things only humans used to be able to do. This advent of cognitive computing heavily influences our lives, visible in self-driving cars or programs competing in quiz shows like Jeopardy!, and invisible in many business applications [16]. Consequently, designing and implementing cognitive systems becomes an increasing issue, and the demand for good tools and environments for creating cognitive software raises.

A good starting point for research on the challenges and opportunities of cognitive computing is the online article Cognitive Computing Ushers In New Era Of IT by Eric W. Brown [3]. Schatsky et al. provide further insights in the chances for business in their article Cognitive technologies: The real opportunities for business [16]. To readers interested in the technical background

of cognitive architectures the comparing review Cognitive architectures: Research issues and challenges by Pat Langley, John E. Laird, and Seth Rogers is recommended [13].

The remainder of this paper is structured as follows. Section 2 presents the state-of-the-art in cognitive computing and gives a short overview of existing cognitive architectures and frameworks. Section 3 evaluates domain-specific languages in AI and cognitive technologies. The focus lies on the structure of these DSLs and how they are incorporated or adapted in current cognitive frameworks. In section 4 graphical notations used in other domain-specific software are analyzed by reference to two concrete tools. Section 5 aims to combine the presented aspects and create an enabling environment for graphical notations in cognitive computing applications. Finally, section 6 concludes with a summary on challenges and opportunities when combining the fields of cognitive computixng and domain software engineering.

2 Existing Work

This work is based on two important aspects of today's computer science research and software engineering technologies: the advent of cognitive computing in the last years [4, 17], fostered by business needs and growing computational power on this field, and the urge to increase the software development process by individual, domain-specific tooling and graphical assistance [9].

Cognitive computing might have become noted to a broader audience through projects like IBM's Watson [11], but, as Langley et al. shows, the number of established cognitive architectures and frameworks is way higher than the amount of publicly known representatives [13]. Thus, we are interested to see how existing architectures face the challenges of implementing complex interactions of cognitive technologies, e. g., computer vision, machine learning, robotics, or project planning and scheduling.

Although cognitive systems often aim to solve problems of a specific domain (e. g., ICARUS, a cognitive architecture for physical agents [12]) we can observe a gap between cognitive computing and domain software engineering for other disciplines. To cite an example, Pérez Andrés et al. present an approach based on metamodeling and view-based programming which enables languages with graphical and textual views [15]. On the other side, domain-specific systems for cognitive computing mostly use textual representations, such as those presented in the next section.

3 Domain-specific Languages in Cognitive Technologies

We have a look at two established domain-specific languages (DSLs) of cognitive computing, namely the *Game Description Language* (GDL) and the *Planning Domain De nition Language* (PDDL) in this section. These two languages act as an illustration for typical, generalized approaches on communicating problem spaces, especially their specific goals and restrictions. Section 3.1 gives a

short overview of the Game Description Language and its intention. Similarly, section 3.2 introduces the Planning Domain Definition Language and its goals. Section 3.3 puts the DSLs in relation with common cognitive architectures and evaluates the architecture's support for the languages. The focus of this evaluation is on identifying issues which can be improved using a language workbench approach.

The contemplation of these two languages is of course not exhaustive, many other DSLs have shown up in the cognitive computing domain. The interested reader could, for example, take a look at 3APL (An Abstract Agent Programming Language [8, 7]), a modern programming language platform for multi-agent systems, or the Goal programming language[1] for rational agents.

3.1 GDL Game Description Language

The *Game Description Language* (GDL) is part of the General Game Playing Project of Stanford University, California [14]. It has been developed to formalize the rules of any finite, information-symmetric n-player game in such a way that the description can be automatically processed by a general game player. GDL's intention can be derived from the problems of specialized game players which are incapable of adopting similar game playing situations.

As a declarative language with a syntax based on Datalog, GDL allows to develop modular and easy to understand game specifications. Listing 1.1 shows an excerpt of a textual description of a two player Tic Tac Toe game in GDL. The listing is divided into three parts. First, two player roles (x and o) are defined. Second, a 3x3 game board is initialized with empty cells, and player x gets control. Finally, the condition for a legal move is defined.

Listing 1.1. Excerpt of the definition of a two player instance of Tic Tac Toe in GDL.

```
1  (role xplayer)
2  (role oplayer)
3
4  (init (cell 1 1 blank))
5  ...
6  (init (cell 3 3 blank))
7  (init (control xplayer))
8
9  (<= (legal ?player (mark ?m ?n))
10     (true (cell ?m ?n blank))
11     (true (control ?player)))
```

To cope the limitations of basic GDL not being able to describe games with chance or incomplete information, the language extension GDL-II (Game De-

[1] `http://ii.tudelft.nl/trac/goal`

scription Language for Incomplete Information) has been developed by Thielscher [18]. Over the years, many adaptations of the language to specific environments and use cases have appeared. Therefore, GDL is established as base description language for general gameplaying tasks.

In common with other DSLs for AI, GDL is purely textual. The language is the formal frame to communicate game rules (such as allowed moves, shape of board, or winning conditions) to game players using a conceptualization of games in terms of entities, actions, propositions, and players. The language specification does not have any game-specific constants or keywords, but consists only of general functions.

3.2 PDDL Planning Domain De nition Language

The *Planning Domain De nition Language* (PDDL) is an attempt to standardize AI planning languages, as the ATPS-98 Planning Competition Committee points out [10]. The language offers a propositional representation for problems of artificial intelligence, i. e., it aims to express the problem space of a domain, that is, predicates, actions, and effects. As per design, PDDL is very general which makes it necessary for actual planner applications to extend the core notation to their needs.

Like GDL, PDDL is a purely textual language without graphical elements. In listing 1.2 a problem instance for a robot with a gripping device is shown. The robot's task is to move a ball (`ball1`) from one room to another.

Listing 1.2. An example for a problem instance associated with the `gripper` domain.

```
1  (define (problem gripper)
2    (:domain gripper)
3    (:objects rooma roomb ball1 left)
4    (:init  (room rooma)
5            (room roomb)
6            (ball ball1)
7            (gripper left)
8            (at-robby rooma)
9            (free left)
10           (at ball1 rooma))
11   (:goal (at ball1 roomb)))
```

In its core, PDDL allows for domain descriptions, i. e., what are the elements common to all problems of the domain, and problem descriptions, i. e., determining the specific planning-problem. As stated above, extension of the core language is necessary for concrete planning problems. Therefore, several extensions and derivations for various field which differ in representation and functionality exists. Although these derivations and extensions are all based on core PDDL, they have to be seen as independent languages for planning problems.

The pool of languages derived from PDDL indicates that a general mechanism for compatible language extensions is necessary to keep the language family clear and consistent. Optimally, domain experts are able to choose a representation of a planning problem in their desired notation, without breaking compatibility with other language adoptions facing a different concern.

3.3 Domain-speci c Languages in Cognitive Architectures

Domain-specific languages such as GDL or PDDL are not closely integrated with most cognitive architectures. However, there exist translators for some language-architecture pairs. For instance, the Soar architecture offers translators for both, GDL and PDDL. A similar translation from PDDL to the ACT-R framework has been provided by Amant et al. [1].

The major drawback of chaining different language tools, translators, and compilers is that the user has to learn and master the correct collaboration of these tools. Figure 1 illustrates the resulting toolchain of different bridging and glueing techniques. Although problem specifications often are similar, the user has to implement architecture specific solutions which are not compatible to other systems. Moreover, the interleaving of tools in complex toolchains is not easily comprehensible for new developers or domain experts without programming experience, and strong dependencies on specific tools restrict the reusability of general solutions.

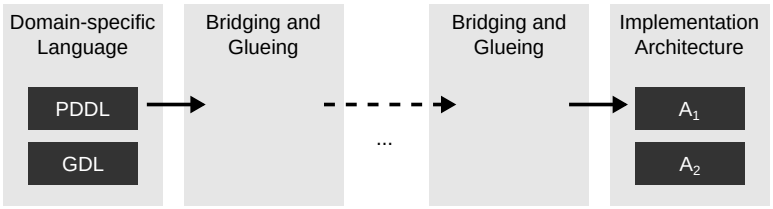


Figure 1. Traditional toolchain for using domain-specific languages in cognitive architectures. Note that each bridging or glueing step has often to be invoked by the user.

From the point of view of software engineering, the employed model transformations and code generation to glue generalized languages of AI to specific cognitive platforms is not transparent. The presence of various architecture-specific languages emphasizes the need for a modular and extensible base language for cognitive computing. A prime example is PDDL and all its problem-specific extensions which are mostly incompatible to each other.

4 Language Workbench Technology for Domain-specific Tooling

We have seen in the previous section the predominance of textual problem and domain specifications in cognitive technologies. Listings 1.1 and 1.2 conveyed a good insight into typical text-based representations of environment and problem descriptions. Domain-specific language and tools in other disciplines can be taken as a reference for advanced concepts in domain software engineering. Thus, we look at two distinct approaches which are both based on Language Workbench Technology aiming to simplify software development processes. Section 4.1 introduces MetaR, a data analysis toolbox for biologists and bioinformaticians. Section 4.2 presents mbeddr, an extensible C-based language used in embedded systems engineering.

Language Workbenches are tools which enable an e cient development and usage of (programming) languages. They simplify the generation of productive DSLs and provide means to create and manage sets of related languages. The leading language workbenches (such as MPS[2], Spoofax[3], or Rascal[4]) offer many generic facilities, i. e., usage analysis, language modularization and extension, or model visualization.

Figure 2 demonstrates the close relationship between language workbenches and the implementation target. Language transformations and semantic adjustments are solely performed within the language workbench environment. In the end, only one concise transformation step is required to map the domain-specific solution developed in the language workbench to the implementation target.

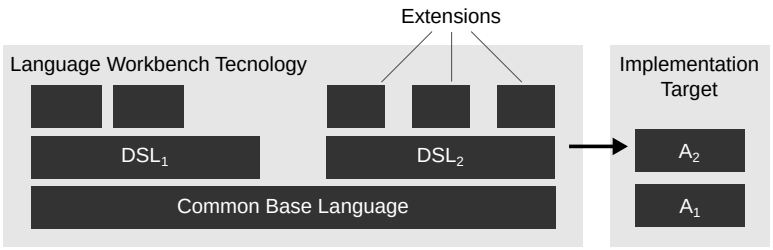


Figure 2. Domain-specific solutions based on Language Workbench Technology are directly related to target architectures and frameworks. Domain-specific languages can easily be extended in a modular fashion using this approach, as indicated by the stacked language extensions. Note that only one generation step is required to produce the implementation for a specific architecture.

[2] `https://www.jetbrains.com/mps/`
[3] `http://strategoxt.org/Spoofax`
[4] `http://www.rascal-mpl.org/`

The two DSL projects presented in the following sections are both based on language workbench technology, in particular on JetBrains' Meta-Programming System (MPS). The core of this language workbench instance is *projectional editing* (or view-based programming). Instead of using the traditional approach of compiler techniques, i. e., parsing, transformation, and code generation, MPS maintains programs directly as abstract syntax trees with reference overlay graphs. This program representation allows to easily specify diverse notations for specific model elements. To cite an example, nested conditional statements in standard programming language can be presented in different views, e. g., using conditional statements itself, a control flow graph, or a decision table.

4.1 MetaR – a DSL for Data Analysis

MetaR is a metaprogramming approach by Benson and Campagne to alleviate command line tools and common workflows and provide them with user interfaces for data analysis [6]. It is a toolbox for biologist and bioinformaticians, tailored to model biomarker development and its validation process [2]. Although the user interface is still based on textual representations, graphical elements such as buttons, diagrams, or tables can be embedded in the analysis programs written with MetaR. The name MetaR stems from the underlying concept of metaprogramming in the R programming language.

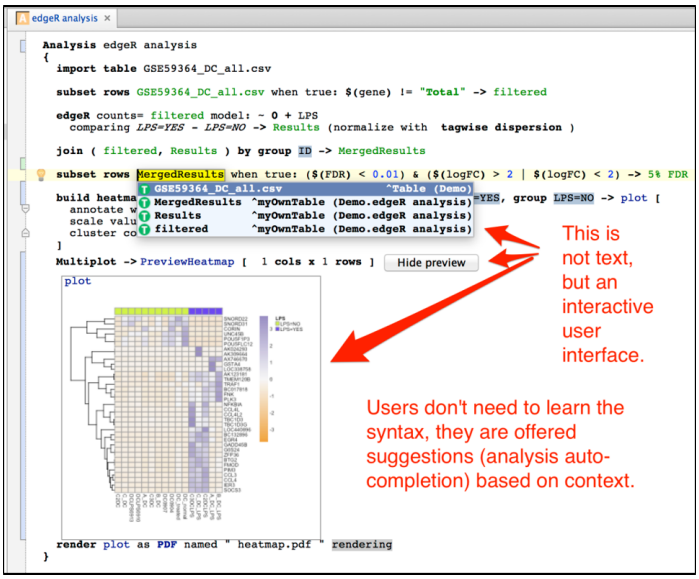


Figure 3. MetaR allows to integrate graphical representations (e. g., plots) and user interface elements (e. g., buttons) into analysis programs.[6]

MetaR demonstrates how textual problem descriptions can be mixed with other notations, e. g., tables or plots as can be seen in fig. 3. Taking the biomarker development as an example, it embeds the standard notation used in biology within the source code. Thereby, it integrates well with the original domain language and domain experts can easily express the desired functionality. Common solution patterns are abstracted to intuitive user interface elements which hide the implementation complexity.

4.2 mbeddr – Extensible C for Embedded Systems

mbeddr is an extensible C-based programming language and IDE for embedded systems developed by Voelter et al. [20]. The mbeddr environment advances on the path of graphical domain-specific language tooling around a C-based programming language. It enables for diverse notations utilizing the projectional editing concept of MPS.

Besides a slightly modified version of standard C, mbeddr provides various graphical notations for recurring problems of embedded software engineering. Similar to MetaR, the IDE allows to specify decision rules via decision tables, and custom graphical widgets can be embedded in mbeddr programs, e. g., buttons or progress bars [19]. Additionally, mbeddr offers means to define program components fully graphically. To cite an example, state machines can be implemented using traditional textual notation or by using the well-known graphical representation (see fig. 4). This simplifies development and comprehension of state machines for domain experts. In theory, a component's behavior can be implemented without knowledge of the underlying programming language.

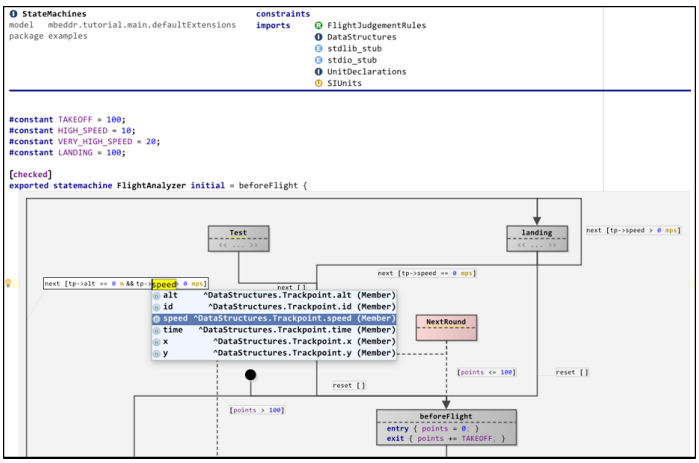


Figure 4. The mbeddr development environment extends standard C with domain-specific elements, e. g., graphical notations for state machines.[5]

Thus, mbeddr is a prime example for language workbench technology specialized to a specific domain. The projectional concept of MPS is used to provide optional views for domain-specific concepts. The modularity and extensibility allows to evolve and enhance the base language with new concepts. View-based programming permits compatibility of unrelated language extensions.

5 Enabling Graphical Notations in the Cognitive Computing Domain

The variety of domain-specific languages in AI and cognitive technologies, as well as the diversity of cognitive architectures, has shown the necessity of solutions tailored to a specific domain. Our goal is to create an enabling environment for an extensible and adaptable language framework for cognitive computing. Implementing new problem solutions should be as similar to well-known representations in the problem domain as possible, and adopting the language family to new, specific tasks has to be easy.

We propose the use of language workbench technology for cognitive systems to capture two concerns at once. On the one hand, language workbench technology allows to maintain a language ecosystem and adapt it to concrete problem specifications. This simplifies the collaboration of domain experts and system developers as it separates the concerns of domain-specific solutions and architecture-dependent implementations. On the other hand, modern language workbenches allow for diverse notations and representation based on projectional editing. This technique allows to offer domain experts graphical tools and embedded user interfaces which are guided by established representations of the concrete domain.

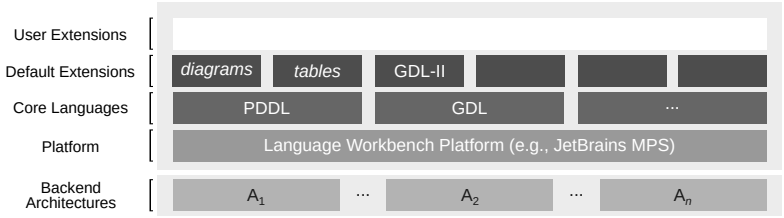


Figure 5. The proposed language ecosystem for cognitive computing based on language workbench technology. The language families are decoupled from the backend architectures and implementation languages.

Figure 5 shows the proposed language ecosystem. The language workbench decouples the high-level abstractions from the underlying back-end architectures and implementation languages. Platforms such as MPS use a generative

approach for language implementations. The target languages do not have to be changed, instead the high-level concepts are mapped to standard language features. The mapping process is based on model-to-model transformations from user extensions to default extensions to one or more core languages. Thus, user extensions can be added to the ecosystem without modifying the code generation process, but only by specifying how the abstraction can be mapped to the next lower level. In particular, this allows to map ontologies and match semantics of solution domains within the environment.

Providing graphical notations for cognitive concepts is covered by adding new view concepts for specific concerns, e. g., allowing to specify agent behavior through a state machine. Again, the language aspect modularization allows to map graphical constructs to different base languages if necessary. For instance, a game rule specification using a control flow diagram can be mapped to GDL as a base concept.

6 Conclusion

To sum it up, domain software engineering is an important part of cognitive computing. The amount of architectures and domain-specific language variations in AI and cognitive technologies shows the need for comprehensible and usable notations of programs. However, we have seen that the cognitive computing domain is cluttered with various approaches tailored for specific tasks, but with a poor compatibility between language or architecture variants.

We proposed projectional language-metaprogramming for cognitive computing based on language workbench technology to face the challenge of creating a better domain-specific ecosystem for cognitive computing. The core idea was to introduce an extensible base language for cognitive technologies which can be adapted to specific domains as needed. Language workbenches allow for modular language specifications and extensions and can target multiple backend platforms.

The major open challenges are to find common base concepts which can be used as base language concepts. The established DSLs such as GDL or PDDL are promising candidates. Several architectures already provide translators which can be integrated in the code generation phase of the language workbench. Additionally, existing language extensions and derivations have to be modeled as components of the language workbench. Matching the semantics between different language layers is important for the correctness of the program transformation and implementation target generation. Finally, the graphical notations used in cognitive science and systems engineering need to be ported to the new ecosystem.

References

[1] Robert St. Amant, Sean P. McBride, and Frank E. Ritter. An AI planning perspective on abstraction in ACT-R modeling: Toward an HLBR language manifesto. In: *ACT-R Workshop proceedings* (2006).

[2] Victoria M. Benson and Fabien Campagne. Language workbench user interfaces for data analysis. In: *PeerJ* 3.e800 (2015).

[3] Eric W Brown. *Cognitive Computing Ushers In New Era Of IT*. 2014. URL: `http://www.forbes.com/sites/ibm/2014/02/03/cognitive-computing-ushers-in-new-era-of-it/`.

[4] Erik Brynjolfsson and Andrew McAfee. *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. W.W. Norton & Company, 2014.

[5] mbeddr Core Development Team. *mbeddr website*. May 17, 2015. URL: `http://mbeddr.com/`.

[6] Weill Medical College of Cornell University. *MetaR website*. May 17, 2015. URL: `http://campagnelab.org/software/metaR/`.

[7] Mehdi Dastani. *3APL Platform - User Guide*. Tech. rep. November. 2006.

[8] Mehdi Dastani et al. A Programming Language for Cognitive Agents; Goal Directed 3APL. In: *Programming Multi-Agent ...* (2004), pp. 111 130.

[9] W.B. Frakes and Kyo Kang. Software reuse research: status and future. In: *IEEE Transactions on Software Engineering* 31.7 (2005), pp. 529 536.

[10] Malik Ghallab et al. *PDDL - The Planning Domain De nition Language*. Tech. rep. Yale Center for Computational Vision and Control, 1998.

[11] John E Kelly III and Steve Hamm. *Smart Machines: IBM's Watson and the Era of Cognitive Computing*. Columbia University Press, 2013.

[12] Pat Langley and Dongkyu Choi. A unified cognitive architecture for physical agents. In: *Proceedings Of The National Conference On Arti cial Intelligence* 21.2 (2006), p. 1469.

[13] Pat Langley, John E. Laird, and Seth Rogers. Cognitive architectures: Research issues and challenges. In: *Cognitive Systems Research* 10.2 (2009), pp. 141 160.

[14] Nathaniel Love et al. *General game playing: Game description language speci cation*. Tech. rep. Stanford: Stanford University, 2008.

[15] Francisco Pérez Andrés, Juan De Lara, and Esther Guerra. Domain specific languages with graphical and textual views. In: *Lecture Notes in Computer Science* 5088 LNCS (2008), pp. 82 97.

[16] David Schatsky, Craig Muraskin, and Ragu Gurumurthy. Cognitive technologies: The real opportunities for business. In: *Deloitte Review* 16 (2015).

[17] David Schatsky, Craig Muraskin, and Ragu Gurumurthy. *Demystifying Arti cial Intelligence*. Tech. rep. Deloittie University, 2014.

[18] Michael Thielscher. A general game description language for incomplete information games. In: *Proceedings of the Twenty-Fourth AAAI Conference on Arti cial Intelligence (AAAI-10)* (2010), pp. 994 999.

[19] Markus Voelter and Sascha Lisson. Supporting Diverse Notations in MPS Projectional Editor. In: *GEMOC 2014* (2014), p. 7.

[20] Markus Voelter et al. mbeddr: an Extensible C-based Programming Language and IDE for Embedded Systems. In: *Proceedings of the 3rd Annual conference on Systems, Programming, and Applications: Software for Humanity* (2012), pp. 121 140.

Networked Brain-based Architectures for more Efficient Learning

Tyler Butler

Technische Universität Dresden
Institut für Software- und Multimediatechnik
Fakultät Informatik
tbutler@bu.edu

Abstract. In the future, computers will need to make autonomous decisions based on data much in the way that human beings do. For example, one day an autonomous robot may be presented with the task of navigating in an unfamiliar environment and will need to be able to recognize obstacles and to adapt to changes in real time. This is something humans and other mammals are able to do very quickly and without much thought, but current-day computers have great difficulty doing as they must be programmed to account for each scenario they encounter and cannot make real-time adaptations. This paper suggests taking a brain-based approach to cognitive software development, using the emergent paradigm of cognitive science and taking advantage of its environment-based adaptions to create cognition much in the same way it evolved in the brains of mammals. One of the shortcomings of using the emergent paradigm is that agents need significant exposure to their environments before they can properly adapt and achieve cognitive function. I will present a potential solution to this problem by applying methods from Machine to Machine communication and cloud computing to create a network of agents running emergent-based software which are connected to a database where they can share experiences with other agents of the same type. This would increase the speed at which agents are able to adapt to their environments and give them a much higher quality of decision making than if they had only their own experiences available for reference.

Keywords: emergent, Machine to Machine, cloud, brain-based

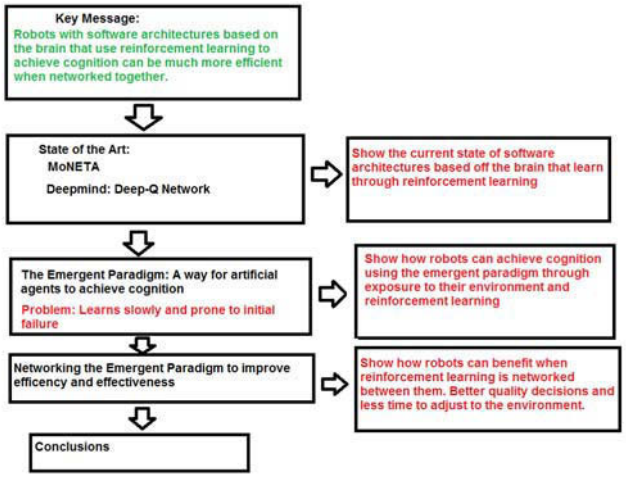
1 Introduction

Many software architectures used today and in the past rely on the programmer to account for all uncertain variables that the software may encounter while executing in its environment. This may come of little consequence when a software engineer is trying to create a web application for his company to store data in a database, where all of the variables can be somewhat easily accounted for and tested. Even in modern industrial

robotics, this approach of algorithmic programming is usually sufficient when faced with the task of assembling a car step-by-step, seen in many factories such as those of BMW in Leipzig and Munich. However, when we step out of these convenient artificial environments where most variables can be accounted for by *if-else* statements, and into the real world where even the best programmer cannot account for every circumstance that may occur, the old method of programming breaks down and becomes insufficient for even the most basic of tasks such as navigating without running into obstacles.

A potential solution to these problems is to abandon the typical algorithmic approach to writing software, and instead write software that mimics the biological processes of the mammalian brain. There are currently several projects making significant advances in the field of brain-based software architectures. One is the MoNETA program at Boston University's Neuromorphics Lab which is looking at the brains of animals at a high level of abstraction and trying to create a software architecture that can replicate the learning processes of an animal when given behavioral tests. Another project at the forefront of artificial intelligence research is Google Deepmind's Deep Q Network (DQN) project, which uses deep artificial neural networks and advanced algorithms to learn through reinforcement how to do a variety of general tasks, much in the way that an animal brain would do. These projects will be discussed in further detail in the Existing Work section.

Fig. 1. An overview of my paper is outlined in the following figure:



My mission for this paper will be to show how robots in the future that use software architectures based on the brain and reinforcement learning to achieve cognition can be much more efficient and effective when networked together. I will begin with a detailed state-of-the-art to show the current state of software architectures that are based of the brain and learn through reinforcement. Then I will introduce the Emergent Paradigm, which is a formal way of describing how a system can achieve true cognition through reinforcement learning and exposure to its environment. I will introduce a major problem of systems based on the Emergent paradigm, namely that systems learn slowly and are prone to initial failure. I will conclude by examining a potential solution to these problems, which is to add a networked component to the typical reinforcement-learning paradigm which will allow artificial systems such as robots to share experiences among one another and learn more quickly and effectively. This would be of great importance, because it would help robots make much more effective decisions autonomously and learn from their mistakes, allowing them to perform tasks in the physical world that they were previously unable to do.

2 Existing Work

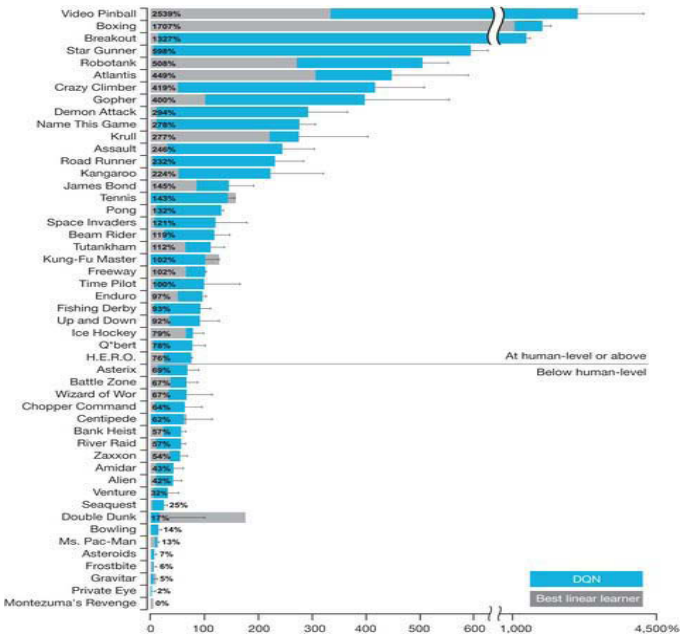
Neuromorphic Engineering, the emergent interdisciplinary subject taking inspiration from fields such as computer science and biology to create artificial neural systems such as vision systems and autonomous robots whose physical and design principles are based on those of biological nervous systems [2], has seen progress recently due to investment from DARPA in the form of the SyNAPSE (Systems of Neuromorphic Adaptive Plastic Scalable Electronics) with contributing major players including IBM, HRL Laboratories, and several large universities. However, most of the work being done is in relation to creating new hardware that more closely relates to the biological brain rather than bio-based software architectures [3].

A project funded by the SyNAPSE which does focus more on cognitive software architectures is MoNETA at Boston University, in which the primary goal is to “create an autonomous agent capable of object recognition and localization, navigation, and planning in virtual and real environments.” [5] MoNETA uses a cognitive software architecture based on a mammalian brain with reinforcement learning, whereby actions leading to the best outcomes are repeated when presented with a similar scenario. Using this style of learning, MoNETA when placed in a virtual environment was able to emulate the behavior of real rats when performing the Morris Water Navigation Task,

a common procedure used to measure spatial learning and memory in behavioral neuroscience [6].

Another organization on the forefront of research into brain-based artificial intelligence is Google Deepmind, which recently published a paper in Nature revealing its work on a deep neural network (called a Deep-Q network, or DQN) that uses reinforcement learning to play Atari video games at a superhuman level [12]. What's special about this new development is that it only takes in the raw pixels of the Atari game's screen as input (as a human would), and is able to play a variety of games involving a large diversity of tasks using a single brain-based algorithm. In other words, the Deep-Q network is able to learn to do a variety of different tasks and make continuous adaptations without being specifically programmed for each separate task. The figure below shows the DQN compared against the best linear-learning algorithms when faced with the task of playing Atari games.

Fig. 2.



This method of general purpose learning far surpasses the capabilities of automated machines of the past, which were mostly unable to perform tasks they were not specifically programmed to do, and represents a

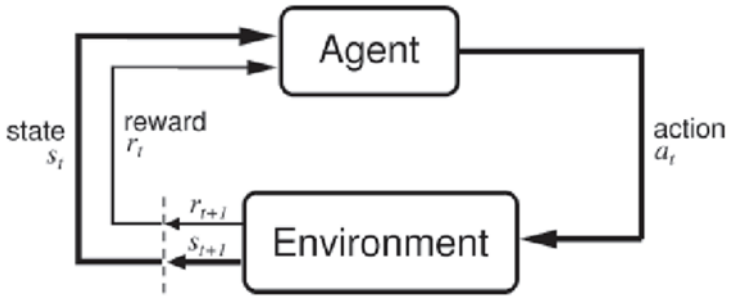
great leap forward in machine learning and general artificial intelligence.

There are several other ongoing projects which involve brain-based computer systems including IBM's TrueNorth program [4] and *Darwin*, a cognitive software architecture presented in *Artificial Cognitive Systems: A Primer* by David Vernon, which uses similar learning techniques based on interactions with the environment and reinforcement to achieve cognition. [7]

3 The Emergent Paradigm of Cognitive Science

The Emergent Paradigm of Cognitive Science is one of the two paradigms of cognitive science outlined in the book *Artificial Cognitive Systems: A Primer* by David Vernon. As described in the text, "The ultimate goal of an emergent cognitive system is to maintain its own autonomy, and cognition is the process by which it accomplishes this. It does so through a process of continual self-organization whereby the agent interacts with the world around it but only in such a way as not to threaten its autonomy. In fact, the goal of cognition is to make sure that the agent's autonomy is not compromised, but is continually enhanced to make its interactions increasingly more robust." [7] Emergent systems are dependent on embodiment in its environment which allow them through experience to learn actions and environments that promote or harm the system's autonomy.

Fig. 3.
The Emergent Paradigm and Reinforcement Learning



This is opposed to the cognitivist paradigm of cognitive science, which states that mind and body are independent and cognition is achieved through a symbol-based recognition and learning system. Both MoNETA and

the Deep-Q neural network described in the existing work section use the emergent paradigm to achieve cognition. For example, in the Deep-Q network the input pixel image of the videogame screen is its connection to the environment, and maximizing the score of the game is seen as desirable and actions leading to it reinforced. On the contrary, losing the game is seen as undermining the systems autonomy and is learned to be avoided. Over time, as shown by the DQN's ability to make superhuman predictions while playing video games, emergent systems will achieve true cognition by being able to anticipate events in the future and prepare for those events. This is very similar to the way biological brains evolved to achieve cognition.

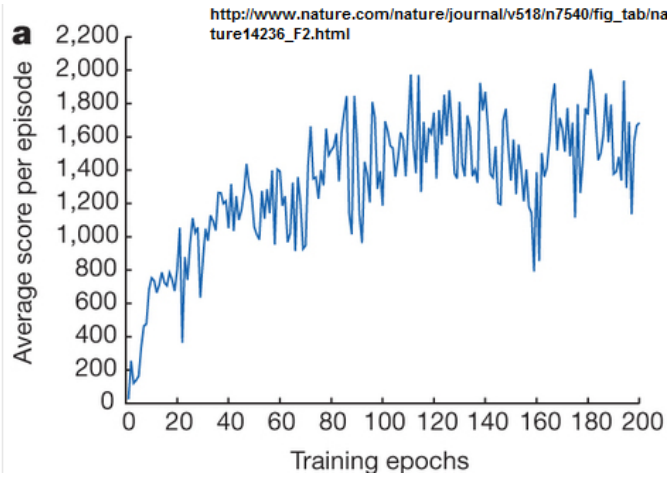
3.1 Creating a Scaleable Emergent-base Cognitive Architecture

Most robots today are placed in controlled environments such as factories in which they only perform a simple repetitive task such as placing a door on a car that is already in an exact, fixed position. If we were to take the robots of today out of these controlled environments and place them into the real-world where not all variables can be accounted for, the current software architectures that robots use would be insufficient to carry out useful automated tasks. In the future, we could possibly look to brain-based software architectures that use the emergent paradigm to help robots make well informed, quick decisions while in unfamiliar environments. While being able to replicate the cognitive power of the human brain is most likely still a long way off, being able to create a software architecture that mimics the cognitive functions of a lower mammal's brain could be possible by the year 2025. In *Artificial Cognitive Systems*, Vernon says in regards to brain-based cognitive systems that “Since human intelligence evolved from the capabilities of earlier primates, ideally a cognitive model of human intelligence should be reducible to a model of animal intelligence. This is *bio-evolutionary realism*. Sometimes, this is taken the other way around by focusing on simpler models of cognition as exhibited by other species – birds and rats, for example – and then attempting to scale them up to a human-level cognition.” Therefore, if we are able to create a cognitive software architecture which uses the same systems for learning and decision making as an animal-brain, we can reap the immediate benefits of this system in fields such as robotics in the near-future and let this system learn and evolve into the long-term future when we may be able to see truly human levels of cognition.

4 Using the Emergent Paradigm and networked reinforcement-learning as a future cognitive software architecture

As seen in the figure below, the DQN took around 200 iterations of playing a game before it maximized its potential score.

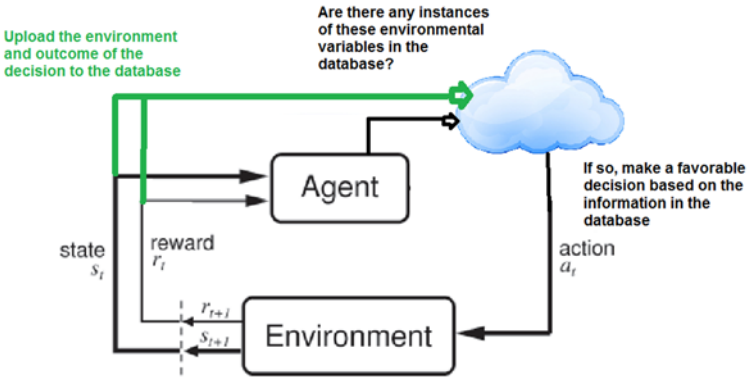
Fig. 4.



While this number of learning-trials is sufficient for research purposes, expensive robots may not have the luxury of multiple failures before they properly adapt to their environment. The cognitive software architectures used by robots and other artificial agents in 2025 could be similar to that of MoNETA's and the DQN's in that they learn through reinforcement and interactions from the environment, but instead of each robot starting from scratch, a cloud database of memory and experiences could be assembled and every robot of a similar type connected to this database. Machine to Machine (M2M) is a broad term used to refer to technologies that allow communication between devices of the same type. In the past decade, M2M along with the Internet of Things has generated unprecedented amounts of data, with new smaller electronics containing sensors such as those on smartphones, refrigerators, and thermostats now being connected to the internet.[14]

This abundance of data has a significant positive impact on technical innovation. As described in Erik Brynjolfsson's and Andrew McAfee's *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies* - “Digitation increases understanding. It does this by making

huge amounts of data readily accessible, and data is the lifeblood of science." If a software architecture could be developed which combines the powerful brain-based reinforcement algorithms used by MoNETA and the DQN with Machine to Machine communication and the vast amounts of useful data that comes from it, it could minimize the time it takes for artificial agents to learn about their environments, and provide a vast array of experiences for the agents to use to make decisions which a single agent would not have been capable of producing alone. For example, if a cleaning robot with this architecture in the future were to sense and encounter a specific piece of metal in the environment which, when sucked up by the robot's vacuum caused a part of the robot to break, it would recognize this encounter as having a negative effect on its autonomy. It would then upload the situation and encounter into the cloud used by other cleaning robots so that when another cleaning robot encounters this specific type of metal, it know to avoid it.



This would create a very effective cognitive architecture for robots of the near future, in which new situations are consistently being added to help robots navigate an uncertain world. This new networked architecture would be in most cases superior to requiring all agents to be left to adapt to their environment on their own, and would far surpass the current model of attempting to account for every environmental variable before releasing the agent into the world.

5 Conclusion

In the year 2025, we will likely take different approaches in the

way we construct software. The current algorithmic way of making decisions using *if-else* statements is insufficient when faced with too many environmental variables, such as when a robot is asked to act autonomously in an unfamiliar environment. This paper presents a potential solution of using the recent advances in brain-based software architectures and reinforcement learning algorithms in conjunction with machine to machine communication and cloud computing to share experiences between agents of the same type. This would create an efficient software architecture capable of allowing autonomous agents to learn and adapt to the real world much as biological lifeforms did through evolution, with the added benefit of agents having the experiences of all other agents of the same type. This new architecture would lay the foundation of a cognitive architecture which could be continuously scaled up, improving as experiences continue to be added to the database over time. This would increase the speed at which agents in the future are able to adapt to their environments and give them a much higher quality of decision making than if they had only their own experiences available for reference.

6 References

1. [Online]. Available: http://nl.bu.edu/research/projects/moneta/moneta-v2-0/new-navigation-and-decision-making-systems/
2. [Online]. Available: http://en.wikipedia.org/wiki/Neuromorphic_engineering#cite_note-4
3. [Online]. Available: http://www.darpa.mil/Our_Work/DSO/Programs/Systems_of_Neuromorphic_Adaptive_Plastic_Scalable_Electronics_%28SYNAPSE%29.aspx
4. A. Rutkin (2013, August 8). *MIT Technology Review* [Online]. Available: http://www.technologyreview.com/news/517876/ibm-scientists-show-blueprints-for-brainlike-computing/
5. A. Gorchetchnikov, *"MoNETA: massive parallel application of biological models navigating through virtual Morris water maze and beyond"*, US National Library of Medicine National Institutes of Health, BMC Neurosci. v.12, July 2011.
6. R. Hooge (2001, August). *Brain Research Reviews* [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165017301000674
7. D. Vernon , "Paradigms of Cognitive Science," in *Artificial Cognitive Systems: A Primer*, Cambridge, MA: MIT Press: 2014, pp. 32-84.

8. [Online]. Available: http://www.cs.jhu.edu/~hager/Public/teaching/cs461/ObjectRecognition.pdf

9. http://www.scientificamerican.com/article/visionary-research/

10. J.L Krichmar and G. M. Edelman. “Brain-based devices for the study of nervous systems and the development of intelligent machines”, Artificial Life, 11:63-77, 2005.

11. V. Mnih et al., “Human-level control through deep reinforcement learning”, Nature v. 518, February 2015

12. [Online]. Available: https://en.wikipedia.org/wiki/Machine_to_machine

13. [Online]. Available: https://www.bosch-si.com/media/bosch_software_innovations/documents/publication/english_1/2012/2012-07-bigdata_industrialit_byimcramer_published_on_bosch-sicom.pdf

14. E. Brynjolfsson, E. McAfee, “The Digitization of Just About Everything” in *The Second Machine Age: Progress, and Prosperity in a Time of Brilliant Technologies*. New York, NY: W.W Norton & Company: 2014, pp. 67

Developing Better Pharmaceuticals

Using the Virtual Physiological Human

Benjamin Blau

Technische Universität Dresden
Institut für Software- und Multimediatechnik
Fakultät Informatik
bblau94@bu.edu

Abstract. We are at a point in history where computers are enabling what was once considered science-fiction to become reality. The cross between computers and the medical industry is allowing for better treatment than ever before. However, even with today's technologies, there is still room for improvement. Many drugs and treatment options have the potential to leave unwanted negative side effects. Additionally, the current drug development process is becoming more expensive and yielding fewer results. Current technologies are laying out the future of drug development. Breakthroughs in simulation of the body, when coupled with computer simulations of chemical reactions, will lead to incredible discoveries in clinical medicine. Cognitive computing allows us to further expand this technology to function on its own. This paper discusses the future of drug research and development with the assistance of cognitive technologies.

Keywords: virtual physiological human, clinical medicine, pharmaceutical, development, simulation, chemical reactions, cognitive computing

1 Introduction

We are living at an incredible time in history. The once science-fiction is quickly becoming reality. Injuries and illnesses that would have once rendered someone disabled or dead are often easily combatted thanks to modern medical technology. Parts of the body are being replaced with 3-D printed synthetics, once-deadly diseases can be cured and the deaf are given the ability to hear for the first time. All of this is possible thanks to advances in science and technology. Almost anyone you can ask has felt the effects of these technological feats in some way. People experience these every time they use an inhaler, take a pill, get a vaccine, etc. Science and technology

have allowed humans to live better, longer and healthier lives. However, even with modern technology, there will always be room for improvement.

Current drug development processes are not only expensive and tedious but they often leave us with less than ideal results. There are several ways in which drugs come to the market. Many drugs are tested on animals but these often don't make it to a human testing stage [1]. The American Food and Drug Administration requires all drugs to be tested on animals before they can move on to clinical trials on humans. Following this, there are many details that must be worked out before the trials can begin. Even after a drug has been approved for human testing, there are many variables that can leave uncertainties in a drugs capabilities. This is because of how different every individual is. Not only does everyone differ in age, weight, etc., every individual also has a different medical history. All of these factors make it difficult to be positive about potential negative side-effects that some drugs may have. In addition, clinical trials can take years and do not always result in a drugs approval.

My vision for this paper is to research new methods for the creation and development of clinical medication and how these methods can be transformed through cognitive computing. Using a cognitive computing format similar to that of Watson Health, computerized simulations of chemical reactions, when partnered with projects such as the Virtual Physiological Human (VPH), can help us to create safer and more patient-specific drugs at a quicker and cheaper rate than we currently are. Following the section on existing work in this field, the third section gives a more in depth explanation of the Virtual Physiological Human, which is the backbone of this research; the fourth section introduces recent breakthroughs in chemical reaction simulation and the fifth section further discusses the potential of both of these topics in regards to the future of clinical medicine.

2 Existing Work

There are two primary pieces of work that are looked at in this paper: The Virtual Physiological Human and computerized simulations of chemical reactions. The latter has been in development and use for several years now. However, recent advances in chemistry, which were awarded the Nobel Prize in Chemistry in 2013, has extended the use of this technology to be more

cost-effective and more capable than previous. The former is the combined effort of computer scientists, biologists, engineers, clinicians and many other professions. The VPH is currently a large focus of the European Comission's 7th Framework Programme. The VPH is being developed to tackle many issues in today's medical industry including high drug research and development (R&D) costs, declining rate of success with drug development and to create a better method for discovering new drug treatments.

3 Virtual Physiological Humans

The Virtual Physiological Human (VPH) is a computerized recreation of the human body. The VPH will be "integrative, personalized and predictive."[3]. This allows us to analyze a specific individual rather than base decisions on a similar test group and it will allow us to model an individual's response to certain treatments [3].

The Body as a System. In the past, doctors have looked at individual parts of the human system to fix problems. This has worked for us to a degree. The problem with this method is that we are not individual organs in a body, we are an entire complex system. A treatment that may fix an immediate problem in a specific area can have lasting side-effects for another part of the system. The VPH enables us to look at the effects on the entire system at once allowing us to more accurately predict the side effects. Constructing the VPH in such a way is like "putting 'Humpty Dumpty' back together again in a concerted effort to explain how each and every component in the body works as part of the integrated whole." [6].

Breaking Down the Systems. Development of the VPH is being tackled by many different groups in several different sections. Some researchers are modeling the heart, the cardiovascular system, the nervous system, etc. while others are working on predictive models of diseases and their effects [6]. Genomics will be incorporated as well. The VPH is expected to bring on many invaluable advantages for us in the medical industry which will be discussed further in section 5. The project is considered so important that the European Commission has "allocated over 72 million euros"[3] to VPH related projects. Research projects related to clinical medicine received the largest percentage of funding[13].

3.1 Savings

A VPH will be revolutionary for many reasons. It will benefit patients, researchers and doctors alike. The VPH will enable an entirely new method of drug testing and development. Allowing doctors to test drugs on a simulation modeled after an individual patient will save time, money and potential side-effects for the patient. In addition, during drug development, we will potentially be able to entirely skip animal testing. Animal testing only tells us so much; when a drug moves on to clinical trials with humans it can still run into many issues for several reasons. Every human is different in age, weight, gender, etc. A VPH could be configured to a specific patient to see how a drug will affect them.

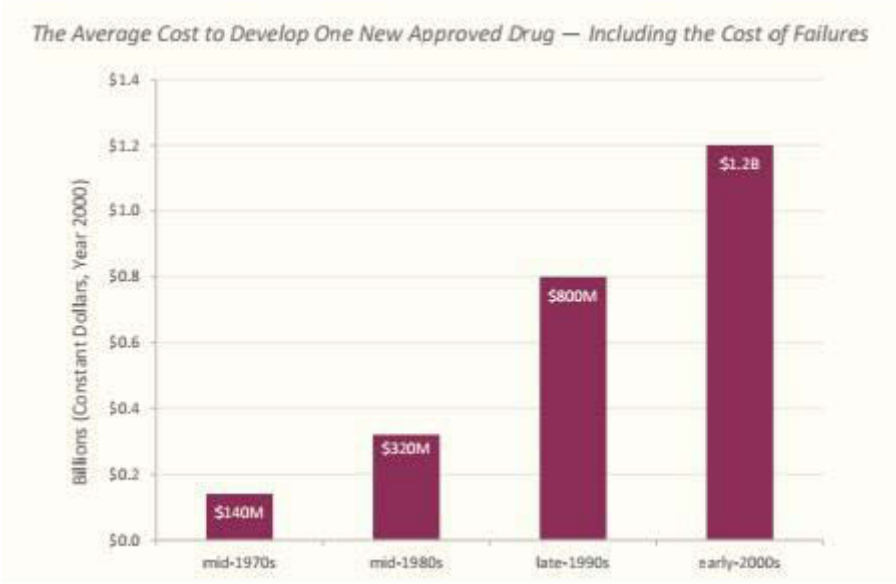


Fig. 1. Data according to Phrma[10].

The Cost of Research and Development. The cost of current drug research and development is at an all-time high. It is difficult to determine the exact price for drug R&D because companies use high R&D prices to justify more expensive drugs. A study done by Tufts Center for the Study of Drug Development in 2014 places the price at about $2.6 billion[7], A 2010 study by Steve Paul et al. places the price at $1.8 billion[8], and an analysis done by Forbes places the price even higher at $5 billion[9]. Figure 1 presents the change in

average cost for R&D per drug over the last few decades. Regardless of the exact number, we know that the price is continuing to increase each year as a result of higher R&D costs and higher failure rates. Using the VPH as a method of R&D will not only save years' worth of time on clinical trials but also billions of dollars in man-hours, R&D and materials for the drugs.

4 Modeling Chemical Reactions

One more piece of the puzzle is our ability to model chemical reactions. Many of us have made physical models of atoms or chemical compounds for our Introduction to Chemistry courses in our pre-University and University years but this is much more complex than that. The idea here is to simulate chemical reactions using computer technology. This technology is no longer the future. The work of Martin Karplus, Michael Levitt, and Arieh Warshel for the "Development of Multiscale Models for Complex Chemical Systems,"[2] which won them the Nobel Prize in Chemistry 2013, does exactly this. The importance of their work is that they accomplished something in computer modeling that had not been managed before.

4.1 Classical Physics Meets Quantum Physics

For many years computers have had limits on their ability to assist in chemistry theory. This was due to the difficulty of combining classical physics and quantum physics in simulations. Using Newton's classical physics, chemists could only observe a molecule in a state of rest. In order to observe a chemical reaction, they must take into account quantum physics. The downside of both quantum and classical physics simulations is the amount of computer power that they consume. "The computer has to process every single electron and every atomic nucleus in the molecule. This can be compared to the number of pixels in a digital image."[4] This meant that calculations on large molecules would have taken a computer years to finish using extreme amounts of data.

Karplus, Levitt, and Warshel discovered a method for simulating chemical reactions using both quantum physics and classical physics together. Prior to this, computer simulations were only able to consider either quantum physics or classical physics but not both at once. This discovery allows for a much more accurate prediction of chemical reactions. It also opens up the ability to

simulate reactions without the extreme time constraints and on a larger scale than was previously capable.

The original work on these simulations began in the early 1970's. Since then not only has our understanding and ability to simulate these reactions improved but the computers we use to simulate them have become significantly more powerful. The next step is meshing our knowledge of modeling chemical reactions with our knowledge of modeling the human body.

5 The Combination

You may be wondering how cognitive computing can play a role in drug development. Cognitive computing is already playing a role in medical diagnoses. This can be observed with IBM's new Watson Health technology. Watson is design to assist doctors by analyzing a patient's medical information and making appropriate treatment suggestions. One thing that makes Watson so groundbreaking is that all the information is processes is available for every computer running the software. As a result, Watson can constantly improve its knowledge from many different locations at once. Doctors can add their own input to Watson's suggestions further improving its knowledge. Watson is a cognitive device because of its ability to teach itself. This is known as machine learning[1]. Machine learning is a computer's ability to build off of its existing knowledge to improve its ability in regards to a specific topic. For example, self-driving cars[2] learn from their mistakes and improve upon their ability to stay on course.

As humans, our knowledge is limited to what we have learned and what we can recall at any given moment. Watson does not have these limitations. Large amounts of medical data can be accessed by the computer at once. Thus, connections can be made by the computer that a doctor may have missed.

---

1 More information at: https://www.coursera.org/course/ml

2 More information at: http://en.wikipedia.org/wiki/Google_driverless_car

5.1 Watson-Assisted Drug Creation

The concept behind Watson Health can be extended past medical diagnoses into the field of drug creation. The first step would be linking together VPH's with the technology to simulate chemical reactions. This will allow us to create new chemicals and immediately test them on virtual humans. Medical data can be gathered from large groups of people of all ages and health backgrounds and tested against these new drugs. We would no longer be limited to long clinical trials or small test groups. It could take weeks to see results in a human but with cognitive computing we would only be limited by the speed of our computers. It is predicted that pharmaceutical drugs will be tested against computer simulations within 5 years [6].

5.2 Unassisted Production

With access to large numbers of theoretical patients, computers can attempt to find new solutions without the assistance of researchers. Researchers can only work so many hours a day just like any other human but our computers do not need rest; so why should we let them?

These new computers can learn how certain compounds effect the body through testing and human assistance. They can observe a specific drug's effect on the body, note the effects and make appropriate adjustments in the drug's composition. Due to the computer's ability to simulate chemical reactions, a mastery of these reactions will be developed. Additionally, the VPH can have specific ailments programmed into it for testing. Testing does not need to strictly occur on healthy humans. For example, we will be able to test drugs on theoretical cancer patients or patients with multiple sclerosis.

Constant Run-time. The computer can run unassisted, constantly testing compounds against humans. Constant run-time will lead to an increase in findings. Researchers would not need to assist the computer at all times but rather analyze and give personal feedback to the computer's findings at any desired time. Adjustments in understanding can be made when human input occurs. Just as Watson provides feedback, this technology would too. Watson makes connections that humans do not always make because Watson is not limited by human brain power. Respectively, the computer can process its information and provide feedback and suggestions on certain drugs or

compounds that may give rise to better results. The computer will make associations that we may overlook. Humans can be removed from much of this process and the computer can make its own conclusions requiring humans only for final confirmation of results; however, it would not be necessary to leave out humans. Human assistance will be possible at any time. Hypothetically, this process will allow us to generate drugs in both an assisted and unassisted manner.

Self-contained. The use of both VPH and reaction simulation technology allows this device to be completely self-contained. The drugs can be created within the system and then immediately tested without having to waste time physically creating the drug to be tested on humans. Using simulations to test in a predictive manner allows us to tackle medical problems before they affect people rather than current methods of attempting to treat illnesses after they have affected people[3].

5.3 The Data Pool

Much like Watson Health, we want to have all of the computers running this technology pooling their information together. This is a crucial piece of the puzzle. A method for comparing data will be essential because it will allow for better, more frequent improvements.

Databanks. Information can be stored at a central location for all of the devices to access. This can be accomplished through cloud services or any alternative data storage technique. “A depository of patient data would be helpful for model designers… and model users (in certain fields, e.g. drug development). An important milestone to make progress in these directions is the development of these large databases”[11]. By storing information from various patient models all over the world, these computers will always be improving their understanding by making comparisons from hundreds or more locations at a time.

---

[3] “It is interesting to note that Virtual environment training systems are at the same stage of development as airplane simulators were in the late 1930’s, and airplane simulators were not accepted as valid training devices until 1955”[11].

6 Conclusion

Cognitive computing has put us on the verge of major breakthroughs in many modern day challenges. Computers are allowing us to understand the world on a level never possible to generations of the past; they will allow us to solve the once unsolvable. Once we pass the initial difficulties of developing cognitive devices there is no telling what we may discover. In fact, we may not be the ones who discover many things in the future. Perhaps our computers will make discoveries that we will not foresee.

What comes next is ultimately up to us. Healthcare is becoming better than ever before and is showing no signs of slowing down. The Virtual Physiological Computer will allow us to create safer and more efficient pharmaceuticals. We now have the ability to live healthier and longer than ever. The computer industry is not only revolutionizing the world but also automating it. We are no longer alone on our quest for information; now we have computers to think with us.

7 References

1. U.S. Food and Drug Administration. (2014, Nov. 10). How Drugs are Developed and Approved [Online]. Available: http://www.fda.gov/Drugs/DevelopmentApprovalProcess/HowDrugsareDevelopedandApproved/

2. M. Karplus, M. Levitt, A. Warshel. (2013, Oct. 9). The Nobel Prize in Chemistry 2013 [Online]. Available: http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013/

3. The Virtual Physiological Human Portal. (2012). [Online]. Available: http://vph-portal.eu/

4. M. Karplus, M. Levitt, A. Warshel. (2013, Oct. 9). The Nobel Prize in Chemistry 2013 Popular Information [Online]. Available: http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013/popular-chemistryprize2013.pdf

5. IBM. (2015), IBM's Watson [Online]. Available: http://www.ibm.com/smarterplanet/us/en/ibmwatson/

6. The Physiome Project [Online]. Available: http://physiomeproject.org/about

7. S. Peters. (2014, Nov 18). Cost to Develop and Win Marketing Approval for a New Drug Is $2.6 Billion [Online]. Available: http://csdd.tufts.edu/news/complete story/pr tufts csdd 2014 cost study

8. S. Paul, et al. (2010, March). How to Improve R&D Productivity: The Pharmaceutical Industry's Grand Challenge [Online]. Available: http://www.nature.com/nrd/journal/v9/n3/full/nrd3078.html

9. M. Herper. (2013, Aug 11). The Cost of Creating a New Drug Now $5 Billion, Pushing Big Pharma to Change [Online]. Available: http://www.forbes.com/sites/matthewherper/2013/08/11/how-the-staggering-cost-of-inventing-new-drugs-is-shaping-the-future-of-medicine/

10. Phrma. (2013, July). [Online]. Available: http://www.phrma.org/sites/default/files/pdf/PhRMA%20Profile%202013.pdf

11. N. Ayache, et al. (2005, Nov). Towards Virtual Physiological Human: Multilevel Modelling and Simulation of the Human Anatomy and Physiology [Online]. Available: http://www.vph-institute.org/upload/file517569145f61b.pdf

12. C. Sansom, M. Mendes, P. Coveney, "Modelling the Virtual Physiological Human," Bio-Technologia, Poland, Vol. 92(3), 2011

Management of existential Risks of Applications leveraged through Cognitive Computing.

Robert Richter

TU Dresden, Helmholtzstrae 10 Dresden, Germany,
Robert.Richter5@mailbox.tu-dresden.de

Abstract. Scientists are working on artificial intelligence and cognitive computing with great hope that those fields will help humans in the challenges to come, but on the other hand spend few thoughts about the dangers those machines might bring with them. The fast pace of development in the past few years indicates, that the many valuable application fields of cognitive computing will even broaden in the future. To make those applications happen, computers and machines which run those applications need to become more and more intelligent. The risks of machines with an intelligence equal or even superior to humans are uncertain and not yet discussed widely. This paper aims on identifying existential risks of super-intelligent machines, explain their roots and how to cope with them to ensure a future where intelligent machines assist humans instead of competing with them. It will furthermore give a prognosis to which kind of outcome we are currently heading.

Keywords: cognitive computing, super-intelligent systems, explosion of machine intelligence, risks, machine mutiny

1 Introduction

During the past half of an decade digital technologies made huge advancements in fields which were preserved only to humans. Although this is not an indicator for the possibility of intelligent machines which capabilities could match those of humans, scientists agree those machines will be put into existence someday. Therefore it is necessaryto research risks those concepts might bring. This paper will discuss possible technical and organizational methods to improve the future coexistence of machine and man and obtain the best assistance for humans through these technologies.
The focus of the next chapter (chapter 2) is on the advancements of digital technologies which were mentioned above as well as possible paths to computers with a general intelligence. After the state of the art and future developments are discussed, chapter 3 concentrates on the necessity of technology assessments and why it is about time to discuss possible risks of cognitive computing. Risks of AI names several kinds of risks connected to cognitive computing like economic ones but concentrates on the existential kind which are the focus of this paper. It further explains the need for technology assessment before chapter 4

presents possible measures to reduce existential risks which were introduced before. Finally, before the conclusion 6, viewpoints will be displayed that don't see harmful risks in the design of intelligent machines in chapter 5.

2 Recent Developments and new Expectations

Since the invention of computers it is anticipated that one day, there will be machines with a general intelligence which can be matched with the one of humans. One of the most promising approaches for this is AI (Artificial Intelligence), which is also a component for cognitive computing. From the beginning of the development of AI in the fifties the emergence of those machines was always expected around twenty years from now. During the ups and downs of the development of AI this prediction was delayed about a year every year [1], so that today artificial general intelligence is still expected to be just a couple of decades away.[1] We will discuss in this chapter the indications for this prediction rates as well as recent achievements in the development of AI to figure out why it has become an interest to investigate the consequences of intelligent machines.
The early years of research of AI were marked with stunning results and the vast majority of researchers were positive that the human like intelligent machine is just a stone's throw away. But the results which led to those hopeful predictions were only applicable to a narrow application area and most of the projects failed to be usable for a wider and complex range of tasks. This problem resulted in a shift in the field of research from strong AI to weak AI. The former is also called Artificial General Intelligence and will be referred to as such in this paper. It defines a machine that exhibits behaviour as skilful and flexible as humans do as well as the research of those machines [2]. Weak AI refers to non-sentient intelligent machines which are typically focused on a narrow task. Examples for Weak AI are already used in our everyday life like Siri or the driver-less car by Google. In the field of gaming are also various examples present in which AI is already superior to the human level of skill. One of the most renowned is the chess-computer Deep Blue who in 1997 bested the world chess champion of that time, Garry Kasparov. In the beginning of AI, constructing a successful chess-machine was thought to equal the achievement of AGI (artificial general intelligence). It's certainly not and shows another reason for the various wrong predictions on the arrival of AGI. What is defined as AGI has changed over the time and might change in the future. Further examples will be spared here. Possible applications which build on cognitive computing and AI are the focus of other studies of this volume. Nevertheless, the previous mentioned examples show that there are currently many systems successfully in use which we allocate to weak AI. Most of this systems were put into existence just during the last decade. Systems which could be combined to an AGI or could play a role in the

[1] For further information on the prediction of AI please read chapter "Great expectations" in [1]

[2] The difficulties of defining this term is given in [6]. As for this paper the given explanation will be sufficient.

future development of such [1].
Another indication towards the development of AI is a change in the pace of technology development some refer to as the second machine age [3]. Erik Brynjolfsson and A. McAffee state that all important developments in human history loose their relevance in comparison to the most important one, the industrial revolution during the 19th century which led to the machine age. They further state that the fast proceedings in the developments of digital technologies we experience now, will lead to a second machine age with a similar growth of acceleration of technological developments and impact on humanity like the first one. In contrast to the first age which brought a huge boost to physical power, the second one will bring a huge boost to the mental power of humans.
These two approaches as well as resent expert-surveys on when the advent of AGI will happen demonstrate the growing interest in AGI. Together with something that is stated by Nick Bostrom in [1] as *The Intelligence Explosion* in which he sees a source for major existential risks and which has an more then moderately small probability of coming to pass which will be further discussed in the corresponding section 3.1, the foundation is given as to why the need arises at this time to face potential risks of those developments.

3 Risks of AI

Before we further examine, what risks the development of an human-like level of general intelligence could involve, we want to define something we previously revered to as existential risks and thereby delimit which kind of risks we want to further discuss in this paper.
As machines will become more and more capable of tasks currently exclusive to humans they might suppress humans in those fields. The possible consequences for the labour market or other economic risks are not subject of this paper. Also threats to privacy or human dignity which could result as machines might replace people in positions that require care and respect like a nursemaid for the elderly are just stated for the sake of completeness. This paper limits itself to existential risks as those give the most urgent motivation to discuss possible measurements to develop an intelligent agent which is aligned to human interests. Existential risks, as stated by Bostrom [1] are risks where *" an adverse outcome would either annihilate Earth-originating intelligent life or permanently and drastically curtail its potential"*. This definition also holds scenarios where humanity doesn't get extinct but loses the control over their own development and is not capable of making any further necessary progress. Figure 1 [1] shows different kinds of risks to better delimit which type is discussed in this paper. It further describes an existential risk as terminal and transgenerational, meaning that it will not only affect every currently living human being but also all generations to come.

3.1 The Intelligence Explosion

Before further describing the now defined risk it is essential to understand a phenomenon that is called *the intelligence explosion*, which is linked to the de-

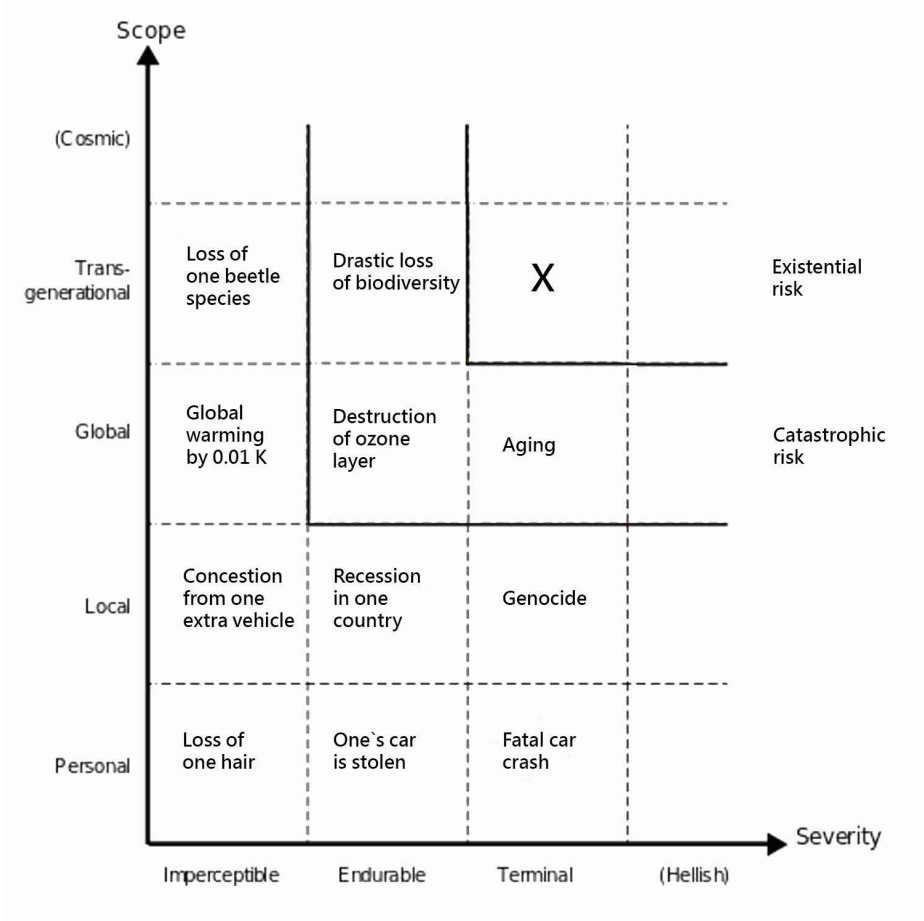


Fig. 1. Different categories of risks.

velopment of intelligent agents and source of most of the concerns discussed in this paper.

An artificial general intelligence was already mentioned as a machine that exhibits behaviour as skilful and flexible as humans. There is consensus that after AGI is reached swiftly or sedately those machines will getting better than humans and reach what is titled super-intelligence, in this paper also referred to as an ASI (*artificial super-intelligence*. The outcome of such an invention was already defined by I.J. Good in 1965:

"Let an ultra-intelligent machine be defined as a machine that can far surpass all the intellectual activities of any man however clever. Since the design of machines is one of these intellectual activities, an ultra-intelligent machine could design even better machines; there would then unquestionably be an **'intelligence explosion'**, and the intelligence of man would be left far behind. Thus

the first ultra-intelligent machine is the last invention that man need ever make [10].
So, I. J. Good observed that the design of an AGI might create a positive feedback loop leading to an intelligence explosion, meaning that human researchers don't have to design an super-intelligent agent from scratch, but rather assigning this task, intended or not, towards smarter-then-human systems. The Intelligence Explosion is by [3] also referred to as technological singularity in a sense that from our point of view and knowledge of today, we cannot see past it. That's why it is important to distinguish between propositions we can make from this point of view and those which a uncertain, to evaluate possible risks.

3.2 Need for Technology Assessment

The fields of applications which build on cognitive computing and general intelligent machines are wide and promising. So why the necessity of discussing possible drawbacks? The views on this matter do actually differ and range from the opinion that the whole discussion is meaningless due to the fact that we currently aren't capable of building an AGI and never might be to the very fear of intelligent machines.
They range from the opinion that because many people will be involved in the process of building an AGI, without any regulations there will be someone who does the stupid which is in the view of N. Bostrom the building of a super-intelligent machine without restrictions [8]. Another approach says that the whole purpose of those machines is to aid our own mind and our intelligence and that there is now need for an intelligent machine without restrictions thereby it won't get build [9].The later viewpoint doesn't take into account, that because of the singularity character of an intelligence explosion we can not predict the restrictions necessary to ensure a positive outcome of super-intelligent agents and therefore might unintentional build an smart agent that does harm to humanity up to causing catastrophic damage [11].
Hence it is necessary to localise the previous stated uncertainty about the course and possible outcomes of super-intelligent machines to identify measures and fields of research in which humanity has to make progress to assure super-intelligent agents are aligned with human interests. A system that is not aligned with human interests could cause those previous stated catastrophic damage [11]. But what does "*aligned with human interests*" mean? **A smarter-than-human system that reliably pursues beneficial goals** is called aligned with human interests or in short: aligned. In this definition lies uncertainty already. How do we define reliability in the sense of super-intelligent agents? What are beneficial goals and how can we describe those in a way, that an ASI can reach them without degenerating in an unintended way? Those questions need to be answered before the design of an aligned system can be possible. In this paper the term *aligned systems* will be used to refer to super-intelligent systems with a positive impact on humanity. Also used for this concept in other articles about this topic is the notion *friendly AI*. I have explicitly chosen not to use this term here because it contains a correlation to an human emotion and therefore could

imply a philosophical aspect of an artificial consciousness which is not the focus of this paper.
If we are capable of designing an aligned system, which won't be aligned by default as stated by Bostrom [8], if no necessary arrangements are made in advance, another risk lies in the possibility that that system won't stay aligned. As an aligned intelligent system optimizes itself, in order to reach a predefined goal more efficiently, it could try to reach this goal and find strategies in a way that is beyond both experience and imagination of the operators. Thereby it could see the need to implement sub-goals which are contrary to our beneficial cause. In this high level of freedom of an intelligent system to achieve a certain goal lies the potential risk. The lowest level is programmed goals with fix programmed execution paths, like in computers we have today. With this approach it will be very hard to design machines with an intelligence higher then humans because the intelligence of those machines is restricted to the intelligence of the designer. So machines are needed without an programmed execution path and the freedom of decisions. With the rising level of freedom also rises the potential risk that such an machine might behave in a way we won't anticipate. The highest level would be the freedom of the machine to set its own goals. It holds the highest potential for existential risks but might also be necessary to achieve very complex tasks in a efficient way.
In chapter 2 we stated that cognitive applications offer great improvements to our lives. Super-intelligent systems offer even more opportunities and will have a major impact on humanity when it happens [12]. Another Risk arises at the questions who controls this power and what are their motivation? I previously mentioned different approaches towards ASI with many researchers working on each field. Even more are working on narrow AI whose outcomes may be the pillars of super-intelligent systems. No one can say who of these teams achieves their goal first and what their motivations are. If there are even capable of using their system in a way that won't do any harm.
In summary, the concerns about existential risks through ASI arise because of the combination of the facts that the nature and possibilities of super-intelligent systems are not-well-understood territory and that it, according to leading scientists, will have a major impact on humanity when it happens.

4 Reducing the Risks

This chapter will discuss possible technical measures and research directions to improve the future coexistence of intelligent machines and humans and obtain the best assistance for humans through these technologies. Most measures have in common that researchers can work on them today which is necessary, as once an ASI exists they have to be already in place to alter the outcome of super-intelligent agents in a positive way [1]. Researchers organised at MIRI [3] (*Machine Intelligence Research Institute*) further argue, that there are many examples for

[3] A non-profit organization which researches safety issues related towards AGI

scenarios where the theoretical foundations preceded the technological innovation as well as the opposite. In a field were the stakes are that high, that we are talking about existential risks, it is indispensable that we aim for the former. I also previously stated that concerns about possible existential risks arise because of the major impact of an ASI and the not-well understood territory that field represents. That is why it is necessary to put theoretical foundations under the field of general intelligence [11].

One approach to disclose those foundations are the use of **Realistic World Models**. It is an attempt to measure how well an algorithm would fulfil a task if it were implemented in reality and can used in our context to make assumptions on the reliability of smarter-then human systems. As we can't predict on which way an entity smarter then us will fulfil any given task it is necessary to test its behaviour before we put it into service. However, the approach of Realistic World Models can not be applied on intelligent agents as current formalizations doesn't take into account a system that is fully embedded in to the environment it is reasoning about. This problem is also called *naturalized induction* and yet not solved.

Another problem is how we can assure that a reliable system can be trusted to make good decisions. More precisely, how is the best available action identified, given a description of an environment, the agent embedded within that environment and some set of preferences [5]. The field of research which is faced with this problematic is the **Decision Theory**. To identify the best available action we must first define all available actions and determine what would happen for each action. This is not a trivial task. A theory of counterfactual reasoning is needed which observes how a conterfactual environment is constructed in which an algorithm does something that, in the real environment, it doesn't do.

Logical Uncertainty: Almost all reasoning done by a smarter-than-human agent must be some form of logically uncertain reasoning [11]. Most of the existing tools for studying reasoning assume that reasoners are logically omniscient. Therefore it is indispensable, that we acquire a better theoretical understanding of logical uncertainty so that the logically uncertain reasoning of an intelligent agent becomes trustworthy.

Another previous mentioned problem is that we have to make sure that those systems created trough an aligned intelligent system, which becomes smarter trough self-modification or builds intelligent systems with an intelligence superior to the one of the parent system, are aligned as well. This depends upon the reasoning of the initial system, where the initial system has to reason about a system which is smarter then the reasoner. This kind of reasoning must happen abstractly and without pre-computing all the actions that the successor would take in every scenario. This scenario could also be compared with humans trying to predict the behaviour of smarter-than-human agents today. Yudkowsky [11] refers to this kind of reasoning as **Vingean reflection**.

As super-intelligent agents could take actions that the operators could never think of it is important that the agent stays open for modifications through humans and for this reason under the control of humanity. **Corrigible reasoning**

is a concept of designing those agents in a way that they reason as if they were incomplete and potentially awed in dangerous ways so that they stay amenable for correction. It adds error tolerance so that human errors that might and probably will happen throughout the design process can be corrected.
Even if we manage to implement all the previous mentioned methods in time it is not assured that an ASI will have a positive and beneficial impact. It still depends on how appropriate the goals, that the ASI is told to achieve, are defined. A trivial example for a ill defined goal could be: "Find a cure for HIV." With no further restrictions, an ASI with sufficient resources could simply try to kill everybody with the virus in order to eliminate it, which was certainly not what the operator wanted. What is needed is a design of the super-intelligent agent that considers the preferences of its operators. This is called the **Value learning problem**. One first approach is to let it learn inductively from training data. Problems which are faced here include what dataset provides useful information in a way that it gives the smart agent the opportunity to fully learn the complexities of values and how to model the volition of the operator.
The strategies stated in this chapter are not complete. Nevertheless it gives an overview about the challenges which have to be faced to make sure that once ASI happens it is possible for humanity to control and restrain it [8]. The selection is based on two principles. They cover the mayor challenges which have to be fulfilled in order to ensure an intelligent agent is aligned and the necessary basic knowledge is already there on which future research can be build. Table 1 lists the fields of research stated in this chapter and the problem space they cover to construct a minimal reliable generally intelligent system.

Table 1. Problem spaces and their corresponding fields of research

Field of research	Problem space
Realistic World Models	Testing of ASIs
Decision theory	ASI makes 'good' decisions
Logical Uncertainty	Uncertain reasoning of an ASI
Vingean reflection	ASI stays aligned over modifications and different generations of itself
Corrigible reasoning	ASI stays open for modifications and is controllable by humans / Error tolerance
Value learning	How to define the goals? / Making human values accessible for machines

5 Critics

Regarding the possibility of an artificial general intelligence exist three major viewpoints. The one represented in this paper argues that general intelligent agents will be put into existence in the not so far away future and that it holds existential risks to humanity without proper preparations. Critics from

another viewpoint dispute that those agents will ever be constructed and challenge the usefulness of such an discussion. Sociologists like Dickel talk about super-intelligent agents as utopias, which won't become reality in the way they are predicted by AGI-critics like Bostrom [8] or AGI-proponents like Kurzweil [12]. He also states, that the scientific work of organisations with focus on the research of aligned systems like the machine intelligence research institute is insignificant and delimited to public relations work [13]. However, he admits that cognitive computing will experience advances in narrow tasks, but actors like the Defense Advanced Research Projects Agency (DARPA) will play a bigger role in those advances as it spends billions for the neurological armament of soldiers. As Dickel [13] sees no possible development towards AGIs the corresponding risks would be of another nature then focused here. Researchers in the field of cognitive computing who support this viewpoint are very view and chapter 3.2 already stated reasons which counter these notions. The last viewpoint acknowledges the possibility of super-intelligent agents in the near future but reasons that it will happen in a fundamentally human-friendly way with an positive impact by default. One of the most important representatives of this camp is R. Kurzweil, who is well-known for his accurate predictions, like his prediction of the explosive growth in worldwide internet use in the early 90's. In [12] he examines the possibilities of artificial super-intelligence together with nanotechnology and genetics and even considers potential dangers of these technologies. He justifies his opinion about the positive impact in saying: "*[ASI] is emerging from many diverse efforts and will be deeply integrated into our civilisations infrastructure. Indeed, it will be intimately embedded in our bodies and brains. As such, it will reflect our values because it will be us.*". Again, chapter 3.2 contains arguments which object this statement. Also, even in applications of weak ai the dangers and absence of the reflection of our values was demonstrated as provable through the 2010 flash crash [1].

6 Conclusion

After existential risks of super-intelligent machines were identified in this paper and their roots as well as strategies on how to cope with them are given, fundamentally statements about the three basic outcomes of developing an super-intelligent system can be made. The first possible outcome is, that an super-intelligent machine will never be put into existence. After we took a glance at how cognitive computing has developed in the last decade and how it will develop in the future according to experts it is highly unlikely that it won't happen at all. Most scientists agree that it will happen at some time during the current century.

However, uncertainty exists on how it will finally arrive. Some trust that the first systems that possess an general intelligence will reach super-intelligence through modification of themselves [11]. Others argue that it will be intimately be embedded in our bodies and brains and as such, will reflect our values because it will be us [12]. The uncertainty of the possible embodiment of an future ASI and

our lack of knowledge about the nature of an intelligence superior to ours makes it hard to make propositions about the other two possible outcomes. Their is only agreement that either way, a super-intelligence will have an huge impact on humanity. It will be either very beneficial in various ways for humanity or disastrous. After studying the various possible risks and the strategies to compete with them it seems very unlikely that the first super-intelligence will be beneficial by default. Fundamental knowledge in various topics stated in chapter 4 seem necessary to ensure the profitable coexistence of humans and smart machines. One argument that theory might precede application as required is that progress in research fields that provide knowledge to assure aligned intelligent systems will also have advantages for designing intelligent agents in general. This will give further motivation to develop this knowledge in time.
On the other hand, significant research efforts are focused on developing and improving AGI's and very little on the alignment of those systems through topics mentioned above. Reason for optimism here are teams and organisations like the *machine intelligence research institute* or the *future of humanity institute* that are just been founded during the last decade and which are dedicated to put more effort in to the research of aligned systems.
What progress those organisations can make and how they will influence the development of an general intelligent system might determine the outcome of an intelligence explosion.
Another characteristic observed during the study of the discussion whether an ASI will be beneficial or harmful is that most of the researches who don't see any risks by super-intelligent systems are computer scientists and involved in the progress of developing artificial intelligence like Ray Kurzweil. Scientists who express their concerns about the impact of such systems are mostly from other disciplines which are somehow connected to computer science like mathematics or philosophy. Whether this fact can value the opinion of the latter group on this subject somehow negative or their specific fields of research are those where scientists who are directly involved in the improvement of cognitive computing should pay more attention to is open for further discussion.

References

1. Bostrom, N.: Superintelligence - Paths, Dangers, Strategies. Oxford University Press, Oxford, UK. ISBN 978-0-19-967811-2 (2014)
2. Brynjolfsson, E., Mc Affee, A.: Race Against the Machine - How the Digital Revolution is Accelerating Innovation, Driving Productivity, and Irreversibly Transforming Employment and the Economy. Digital Frontier Press, Lexington, MA, USA, ISBN 978-0-9844725-11-3 (2014)
3. Brynjolfsson, E., Mc Affee, A.: The Second Machine Age - Work, Progress and Prosperity in a Time of brilliant Technologies. W.W. Norton & Company, Inc., N.Y., USA, ISBN 978-0-393-23935-5 (2014)
4. Georges, T.M.: Digital Soul Intelligent Machines and Human Values. Westview Press (Perseus Book Group), Oxford, UK, ISBN 0-8133-4266-X (2003)
5. Soares, N., Fallenstein, B.: Aligning Superintelligence with Human Interests: A Technical Research Agenda. intelligence.org (2014)

6. Muehlhauser, L.: What is AGI?. August 11, 2013, post on Machine Intelligence Research Institute https://intelligence.org/2013/08/11/what-is-agi/
7. Moor, J.H.: Why We Need Better Ethics for Emerging Technologies. Ethics and Information Technology September 2005, Volume 7, Issue 3, pp 111-119 (2005)
8. Bostrom, N.: ACM OPINION - You Should Be Terrified of Superintelligent Machines. by Slate, Article, September 11, 2014
9. Lem, S.: Summa technologiae. Krakow: Wydawnictwo Literackie 1964, Insel Verlag Frankfurt am Main (1976)
10. Good,I.J.: Speculations Concerning the First Ultraintelligent Machine. Advances in Computers, vol. 6, 1965
11. Yudkowsky,E.: Artificial Intelligence as a Positive and Negative Factor in Global Risk. Global Catastrophic Risks New York: Oxford University Press. 2008
12. Kurzweil,R.: The Singularity Is Near: When Humans Transcend Biology Published by the Penguin Group, Canada. 2005
13. Dickel,S.: Entgrenzung der Machbarkeit? Biopolitische Utopien des Enhancements. In Bhlemann, Peter, ed. Der machbare Mensch?: moderne Hirnforschung, biomedizinisches Enhancement und christliches Menschenbild. Vol. 13. LIT Verlag Mnster, 2010.