The EVM's Universe and the Universe

Mariusz Nowostawski

The Information Science Discussion Paper Series

Number 2006/08
March 2006
ISSN 1177-455X

University of Otago

Department of Information Science

The Department of Information Science is one of seven departments that make up the School of Business at the University of Otago. The department offers courses of study leading to a major in Information Science within the BCom, BA and BSc degrees. In addition to undergraduate teaching, the department is also strongly involved in postgraduate research programmes leading to MCom, MA, MSc and PhD degrees. Research projects in spatial information processing, connectionist-based information systems, software engineering and software development, information engineering and database, software metrics, distributed information systems, multimedia information systems and information systems security are particularly well supported.

The views expressed in this paper are not necessarily those of the department as a whole. The accuracy of the information presented in this paper is the sole responsibility of the authors.

Copyright

Correspondence

This paper represents work to date and may not necessarily form the basis for the authors' final conclusions relating to this topic. It is likely, however, that the paper will appear in some form in a journal or in conference proceedings in the near future. The authors would be pleased to receive correspondence in connection with any of the issues raised in this paper, or for subsequent publication details. Please write directly to the authors at the address provided below. (Details of final journal/conference publication venues for these papers are also provided on the Department's publications web pages: http://www.otago.ac.nz/informationscience/pubs/). Any other correspondence concerning the Series should be sent to the DPS Coordinator.

Department of Information Science
University of Otago
P O Box 56
Dunedin
NEW ZEALAND

Fax: +64 3 479 8311
email: dps@infoscience.otago.ac.nz
www: http://www.otago.ac.nz/informationscience/

The EVM's universe and the Universe

Mariusz Nowostawski

Abstract

The notion that all (or in weaker sense, some) natural phenomena can be modelled as a computable process, some kind of algorithm is recently gaining scientific recognition, and more research is dedicated to the rigorous explorations of the mapping between natural phenomena and the formalised computational systems. There is some debate and controversy as to how much of *the natural* can be expressed in the models of *the artificial*, although due to formalised nature of mathematics and physics itself, it is generally accepted that computation is viable way to model physical reality. Contemporary developments in computer science and in physics not only do no refute computationalism – they provide more data and evidence in support of the basic theses. In this article we discuss some of the aspects of contemporary computationalist efforts based on the traditional notions of Turning Machine computation. Then we present an extended notion of computation, that goes beyond the traditional Turing limit. We propose a new interactive computation model called Evolvable Virtual Machines (EVMs). The EVM model uses the notion of many independently asynchronously executing processes, that communicate between each other and with the outside environment. We present some of the pitfalls of traditional computationalism, and compare it to our new, extended computationalist model, based on the notion of massively concurrent interactive computation (hypercomputation). We argue, that hypercomputationalism based on the collection of asynchronously concurrently communicating computational machines is a more compact and more appropriate way of representing natural phenomena (or the Universe in general). It is theoretically sound, and does not violate any of the current state-of-the-art physical theories. We discuss the details of our computational architecture, and present some of the implications of the hypercomputationalism on contemporary physical, life sciences, and computer science.

1 Introduction

The traditional algorithmic computation is derived from the notion of a person with a pencil and paper, carrying on a step-by-step mechanical operations until the final answer is obtained. This formalisation of computation was proposed by

Alan Turing [23], and refined by others, contemporary to Turing, into equivalent computational models [18]. It is important to remember, however, that contrary to common belief, the term *computation* from the beginning was not restricted exclusively to Turing Machine computation.

We refer to the philosophical notion of all processes being computations, as computationalism[1]. And it refers to the traditional Turing machine equivalent computations, i.e. computation at or below Turing machine level [23]. The idea that our universe is in its core equivalent to computation was introduced by Konrad Zuse [26, 27]. Zuse initiated the field of digital physics, and his thesis lately gained wider acceptance and was popularized most notably by work of Edward Fredkin [7] and digital physics programme. Digital physics postulates that all natural phenomena are equivalent to Turing-level computations. This we refer as (traditional) computationalism.

Traditionally the term *computationalism* is overloaded with alternative formulations; because of its cognitive science roots, philosophical implications, digital physics, and considerable work on narrower computational scope: the postulate that all natural phenomena are equivalent to Turing-level computation. Recently, hypercomputation research is adding new dimensions to the field. We will discuss in Section 2 an extended concept of computation – hypercomputation, i.e. computation above the Turing limit. In Section 4 we present our own computational framework, based on the notion of the Evolvable Virtual Machines (EVMs) introduced in section 3. Our aim is to extend the notion of computationalism, based on computation above Turing-limit, and we propose new notion, *hypercomputationalism*.

We try to use the term *computationalism* with its broader philosophical context, denoting the notion of all naturally occuring phenomena being equivalent to Turing-level computation. We use the term *hypercomputationalism* to denote the notion of all naturally occuring phenomena being computations, below or above the Turing limit.

Hypercomputationalism can be treated as an alternative model of traditional computationalism, with a physical implementation on traditional Turing Machine-like computers, and with all properties equivalent to a Universal Turing Machine (UTM). The potential hypercomputing capabilities although mathematically sound may or may not be physically realisable [1]. However, we argue that appropriate realisable hypercomputational models must be proposed, and experimental data must be collected to make a progress within the field. Our model is based on the notion of non-computable asynchronous timing of the independent processes. Establishing, whether uncomputable synchronisation between real physical processes[2] is possible in our Universe or not, is beyond the point of this article[3]. The main objective for our EVM model is to provide

[1]Note that in its weaker form *computationalism*, in the context of cognitive science, is often referred exclusively to the thesis that mental processes in a human brain are Turing-computable.

[2]For detailed discussion on physicality of mathematical models of physics see also [20].

[3]In general, the argument is not settled. Some physicists postulate a continuous universe, that would make hypercomputation plausible. However most argue discrete universe, equiv-

a broader, more expressive model of computation that can be used effectively to solve computational problems, and to provide more efficient computing architectures. We also aim at providing necessary framework for asynchronous timing investigations, and, possibly, help when faced with uncomputable problems. The actual hypercomputing capabilities of our computational model rely on the physical properties of our universe itself, and so far we have not demonstrated experimentally that uncomputable asynchronous timing can be exhibited.

The philosophical and physical implication of our model are of secondary nature and may or may not be rendered relevant to physics, life sciences and other empirical research areas. The EVM model provides appropriate framework for further experimental studies on computing architectures, and hypercomputation.

2 Hypercomputation

Driven by the mechanical processing of human-computers, Alan Turing proposed a device, that can simulate their behaviour. This device is called a Turing machine, and it consists of a tape, divided into discrete squares, and the read-write head, that can move left and right on the tape, one unit at a time. The tape can be initially annotated by marking some squares black, and some white. The colour of the square directly under is recognized by the head, and the head can change the colour of the square, and move left or right. The initial marking of the tape is referred as program, and the behaviour of the head, represented by a state-transition mapping, is the description of the machine itself. Although a given machine is fixed, the overall behaviour of the head and the tape depends on the initial marking of the tape. For detailed description of the Turing machine refer to [23] or any contemporary textbook on theory of computation, e.g. [10]. It has been proved that many different models of computation are in fact equivalent to this simple computing device, and the whole class of computations equivalent to Turing machine is referred as Turing-level computation.

It is important to remember, that from the inception Turing model was not the only model of computation and the only formalised model of human thinking. In 1945 Vannevar Bush published an article “As We May Think” in Atlantic Monthly [5]. In the article, Bush theorised that people do not think in linear structures. This, interestingly enough, contrasts profoundly with what his contemporary, Alan Turing, assumed for his models of computation. Bush proposed a visionary, at the time, model of a computing machine: Memex. Memex was designed for information retrival and cross-referencing based on high-resolution microfilms coupled to multiple screen viewers, cameras and electromechanical controls. On the design diagrams it looked like a big desk with a camera recording what users wrote and then linking it to other pieces of information indexed in the machine storage space. Bush described Memex as a ”device in which an

alent to a giant cellular computer [21, 22]. Even with discrete universe, it is not clear if the individual elements are synchronised in computable or uncomputable fashion.

individual stores his books, records and communications and which is **mechanised so it may be consulted with exceeding speed and flexibility**. It is an enlarged intimate supplement to his memory." (Emphasis added.)

The important point about the Memex machine is that it represents a belief that the way humans think goes beyond Turing computability models. The Memex description, which was written years before the first digital computers had been successfully built and utilised, is a clear reference to what these days would be called *hypercomputation*. Bush's beliefs are now shared by contemporary researchers from different fields working in the area of hypercomputing [4]. Our asynchronous model of the EVM computing architecture in some way is similar to the principles of the Memex architecture. The collection of independently computing and asynchronously communicating agents is believed to be a more powerful model of computation, and some believe that it closely mimic the way human cognitive processes work.

Any computation that goes beyond that defined by the Universal Turing Machine (UTM) is called *hypercomputation*. Such computation is also known as super-Turing, non-standard or non-recursive computation [11]. Hypercomputing is a relatively new, multi-disciplinary research area, spanning a wide variety of fields: computer science, mathematics, philosophy, physics, biology and others.

Hypercomputation provides a sound and consistent framework within the theory of computation, and is as old as the basic Turing model of computation itself. The first conceptual formalisation of hypercomputing machines has been done by Alan Turing, himself [24]. The original formalism based on the notion of oracles, are equivalent to trial-and-error machines, and other forms of hypercomputation [4].

Our EVM model, discussed in the following section, uses the combination of trial-and-error model with asynchronously communicating virtual machines. This allow us to incorporate both, the weak and the strong notion of hypercomputing within our framework.

Computing the uncomputable. This may sound like an oxymoron, but this phrase is used surprisingly often and has three sound interpretations within the context of hypercomputation.

1. a Starting with Turing-machine computation, the goal is to calculate with arbitrary precision an estimated answer to something that is effectively uncomputable by a UTM. This we call a *weak notion of hypercomputation*.

2. b Provide an accurate answer to Turing-uncomputable problem, by utilising oracles[24] or other means of hypercomputation. This we consider *strong hypercomputation*.

3. c In the context of resource bound computation, computing something larger than allowed by the upper resource bound.

Both, strong and weak notions of hypercomputation are mathematically sound and consistent with contemporary theories of computation. The weak notion does not represent any practical difficulty and can be physically built and empirically demonstrated. For examples, see Chaitin's work on computing the bounds of the definable but uncomputable number Ω [6], and a theoretical trial-and-error machine solving the halting problem [4]. The strong notion however has not been demonstrated empirically[4].

3 Evolvable Virtual Machines (EVM)

There is an increasing amount of work conducted independently within traditional computationalism, within digital physics, cellular automata, artificial life and evolutionary computation. Certain properties investigated in those diverse settings are invariant and are shared between different complex systems. Our original desire was to integrate some of the recent advances for those diverse fields onto a single coherent theoretical model, together with an experimental framework which could be used for some practical investigations.

The Evolvable Virtual Machine architecture (EVM) is a model for the autonomous building complex hierarchically organised software systems. Originally designed as a artificial evolution modelling system, the EVM stems from recent advances in evolutionary biology, and utilises notions such as specialisation, symbiogenesis, exaptation and computational reflection. Proponents of symbiogenesis argue that symbiosis and cooperation are a primary sources of biological variation, and that acquisition and accumulation of random mutations alone are not sufficient to develop high levels of complexity [13, 14]. Other opponents of the traditional gradualism suggest, that evolutionary change may happen in different ways, most notably through exaptation [8]. Computational reflection and reification, on one hand, provides very compact and expressive way to deal with complex computations, and on the other, provides ways of expanding a computations on a given level via the meta-levels, and meta-computations.

The EVM architecture allows independent computing elements to engage in symbiotic relationships, specialise in specific tasks, evolve towards new tasks, and be used in different contexts than originally designed for. In addition to classical computational tasks, the EVM architecture can be treated as a new hypercomputational model that combines features of trial-and-error machine and the asynchronous communicating processes paradigm. The trial-and-error behaviour is achieved through continuous looping of different hypotheses and their re-evaluating until the desired precision of the hypothesis is achieved. The asynchronous communication aspect provides the (potential) ability of strong hypercomputation. If our Universe exhibits not computable properties, (or if it is continuous), then non-computable time differences and delays are viable. This can be utilised by the collection of asynchronously communicating machines, hence, providing the strong hypercomputation. In case that our Uni-

[4]There are arguments suggesting strong notion of hypercomputation to be physically possible. For example, see the discussion about cognition and mathematical thinking in [4]. However, some strong claims are made opposing hypercomputation, see for example review and discussion in [1].

verse is computable, EVM still offers weak hypercomputing capabilities, and provides a more compact and general notion within the hypercomputationalism programme.

4 EVM Implementation

Our current implementation of the EVM architecture is based on a stack-machine[5]. The basic data unit for processing in our current implementation is a 64-bit signed integer[6]. The basic input/output and argument-passing capabilities are provided by the operand stack, called *the data stack*. The data stack is a normal integer stack. All the operands for all the instructions are passed via the stack. The only exception is the instruction `push`, which takes its operand from the *program* itself. Unlike other virtual machines (such as the JVM), our virtual machine does not provide any operations for creating and manipulating arrays. Instead, the architecture facilitates operations on lists. There is a special stack, called *the list stack* for storing integer-based lists.

Execution frames are managed in a similar way to the JVM, via a special execution frames stack. There is a lower-level machine handle attached to each of the execution frames. This is a list of lists, where each individual list represents an implementation of a single instruction for the given machine. In other words, the machine is a list of lists of instructions, each of which implements a given machine instruction. Of course, if the given instruction is not one of the Base Machine units, ie. primitive instructions for that machine, then the sequence must be executed on another lower-level machine. The Base Machine implements all the primitive instructions that are not reified further into more primitive units.

Potentially, EVM programs can run indefinitely and therefore each thread of execution has a special limit to constrain the number of instructions each program in a multi-EVM environment can execute. Once the limit is reached, the program unconditionally halts.

The EVM offers unrestricted reflection and reification mechanisms. The computing model is relatively fixed at the lowest-level, but it does provide the user with multiple computing architectures to choose from. The model allows the programs to reify the virtual machine on the lowest level. For example, programs are free to modify, add, and remove instructions from or to the lowest level virtual machine. Also, programs can construct higher-level machines and execute themselves on these newly created levels. In addition, a running program can switch the context of the machine, to execute some commands on the lower-level, or on the higher-level machine. All together it provides near limitless flexibility and capabilities for reifying individual EVM execution.

[5] With small differences, the EVM implementation is comparable to an integer-based subset of the Java Virtual Machine (JVM). The implementation is written entirely in Java, and developers can obtain it from CVS `http://www.sf.net/projects/cirrus`

[6] This, somewhat arbitrary constraint is dedicated by efficient implementation on contemporary computing devices

Each individual EVM can reference any other machine in the multi-EVM environment (the EVM Universe). This is achieved by using the first 32 bytes of the instruction to address any computer in the Internet, and the second 32 bytes for the index of the instruction on that machine.

One possible way of extending current EVM implementation is by adapting bias-optimal search primitives [12], or the incremental search methods [19]. To narrow the search, one can combine several methods. For example, it is possible to construct a generator of problem solver generators, and employ multiple meta-learning strategies. A more detailed description of the abstract EVM architecture is given elsewhere [17], and the preliminary experimental results are described in [16].

5 Computational Universe

The notion of computational Universe emerges from different disciplines and for different reasons. The precursors of the idea can be traced back to antiquity [20]. In computer science the very first formulation of computational nature of the Universe has been formulated by German computer scientist, Konrad Zuse [27] and he is credited with the first precise formulation of the computable Universe. Zuse postulated a "computational space", in which our Universe is performing the computation. The actually physical space has been assumed to have an isomorphic relation to the computational space. There is an alternative view, in which the actual computational space has no direct relation to the physical space-time.

Many scientists state the inherent belief in the strong interpretation of the Church-Turing thesis, and one often sees claims, like this:

> Almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication.

[25], pp. 5 and 716-717.

Fredkin [7] with the digital physics programme, pursues the idea that all natural phenomena (including quantum physics) are inherently reducible to a Universal Turing Machine (UTM).

It is our belief that, in principle, all naturally occurring phenomena are in fact reducible to computations (or hypercomputations). We mean here computations in a broader sense, exceeding the notion of Turing machine computability. One of the motivations for this work is to propose a model capable of expressing rules of arbitrary natural phenomena in terms of hypercomputation.

In the time of Newton, the universe was depicted as a perfectly working mechanical machine, like a giant clock, with simple rules governing all the interactions of mechanical building blocks. This analogy stretches to contemporary ideas about the Universe. For computer scientists the Universe looks like a giant computer. However, we go one step forward[7] and propose that this is exactly

[7] To push the analogy we actually employ the meta-meta...-tower, hence we go a countably infinite number of steps forward.

how the science, and the building up of models must proceed. We cannot discover something out of nothing, we always have to base the next theory, or the next formulation of a phenomenon on existing data and existing models, by the process of mixing the information, and combining the information available at any given time instance. The notion of the Universe as a giant computer, was not feasible in Newton times, because all the necessary notions of computation where not available at that time. The same is happening now to a certain extent. The progress in the area of hypercomputation is enabling the notion of the Universe as a giant hypercomputer to emerge. This will inevitably lead to further advances and new ideas. This is exactly what motivates the research on the EVM architecture – the ability to allow the system to expand, adapt and grow beyond what it's original design was.

The main contribution of this work, is the reflected architecture that can be used to adapt, modify and self-organise itself. Architecture based on the massively concurrent and asynchronously communicating units, deterministic, yet unpredictable and uncomputable in a Turing computation sense. The model, that can potentially unify different aspects of science into a single framework, providing the language and representations for distributed computing, hypercomputing, biology, social sciences and physics alike.

6 The need for reflection

The biggest limitation of the digital physics is, that it specifies the upper limit on complexity of the Universe, and on the language (or framework) used to model the Universe. The Turing Machine-based models will never be able to compute, or model anything more than what a Turing machine can. Although it may seem to be an attractive idea, it is actually easier (and safer) to assume that the limit is somewhat higher. That it lies on the hypercomputing level. Because of the nature of the physically realisable virtual machines, this assumption is not actually changing any assumptions on the physical Universe. It enlarges however the language (the framework) in which the Universe is modelled. The EVM architecture realised on the global network system such as Internet, will behave as a complicated Turing-machine, if only the universe is in fact a Turing computable process. The formalism however, is capable of expressing more than Turing computation. Therefore, on one hand, it can be used as a way of describing the physical universe, but also, it provides a way of expressing meta structures that go beyond the original Turing level. On the other hand, if the Universe is not inherently digital and Turing-computable, then the EVM model is capable of exploiting the hypercomputation within the physical world. As pointed out earlier, the weak hypercomputation provided by the EVM model naturally addresses some of the computational issues, that require special treatment within more traditional Turing-based models [4].

7 Physics and new computationalism

Most physicists claim that the universe, by virtue of its atomicity, is fundamentally discrete. The first observation of Planck confirmed that, when he observed that light is not continuous but exists in quanta. The discrete notion of the universe does not produce any controversy. It has been argued that continuous physical equations are actually useful approximation of the discrete physics. The discrete nature of reality is the first fundamental pre-requisite for Turing computationalism.

Traditional theories make a clear distinction between "things" and "processes". There are possible, alternative, process-only perspective on reality. The main push in this direction has been made through advances in computer science, and also in physics (process physics), in life sciences [15] and in evolutionary biology. The process-centred view of the reality is second a fundamental pre-requisite for computationalism.

There is an important ontological difference in modelling reality by processes. The actual substrate becomes irrelevant. Computationalism by its nature is process-oriented. All entities, machines, and data structures within the computational perspective are inherently processes.

Traditional physicists have generally difficulty in changing their mindset. For example Vic Stenger, a physicist, argues[8]:

> However, I cannot think of a single working physical scientist who is a relativist. That handful who are even aware of Kuhn's work (most would not even recognize his name) scoff at the idea that scientific truth is an arbitrary social convention. They all can produce examples that belie the notion. One of my favorite examples is the magnetic moment of the electron, which is both calculated and measured to one part in ten billion with the two results in perfect agreement. To characterize this spectacular achievement as nothing more than social convention is absurd. The magnetic moment of the electron (and thus the electron itself) is as objectively real as any concept that humans can bring to mind, including the chair you are sitting on.

To a computer scientist's perspective this however is too simple. Presented with an unknown computational module, a black box computational process, a scientist will investigate the properties of the module, and, by repeatable experiments establish certain invariant properties of a given computational processes of the black box. However, the scientist will never assume that his own model of the computational process is equivalent to the actual process hidden inside the black box. The reason is that there are infinitely many equivalent computational processes that can render the same experimental results all the time. There are even more equivalent models, that would render the same results given a finite amount of test cases. It is important to remember also that some computations

[8] Private communication, 2005, mailing list: avoid.

can change their behaviour after a particular stimulus is presented. Jumping to the conclusion that the model is the reality, is very unlikely in these settings, and I cannot think of any working computer scientist who would claim that.

Electrons cannot become something real because a physicist says so. What is real is only the measurement. The model that explains why the measuring device behaves in a particular way, is just a model, and will remain a model. It is very likely that we will continue refining our models into more and more detailed models, further and further detached from the common sense and our day-to-day experiences, and they will always remain models.

The analogy of the electron and the chair is a good illustration of the origins of the physical bias and (peculiar) ontological assumptions some scientists have. The belief that models are real stems from the history of scientific endeavours and the very social and historical conventions.

One of the good illustrations of this phenomena in physics is the quantum mechanics. There are a number of different, sometimes unrelated and counter-intuitive interpretations of quantum mechanics. The observations, predictions and equations are in each instance the same – no physical laws or empirical data are violated. However, the models, the story, the narrative and the interpretation of the observed phenomena are totally different.

The most interesting for us, from the EVM computationalism perspective, is the interpretation of quantum behaviour provided by David Bohm [3]. Bohm offered causal sub-quantum theory, where still-unknown forces act as the agents to produce quantum behaviour. This is similar on some level ot the way once-invisible atoms produce thermal behaviour[9]. In Bohm's model no signals are transferred faster than the speed of light, and thus no violation of relativity is implied. Bohm's theory restores traditional determinism, and does not require (unexplainable) quantum randomness. He answers ontological questions about the source of quantum phenomena in a way that is compatible with EVM computationalism. In fact, EVM computationalism is originally inspired by the Bohm's interpretation of sub-quantum physics. Unlike Bohm's original formulation, the EVM architecture does not postulate holistic universe (holistic in a sense that potentially everything can be linked directly with anything else). Due to the nature of the computational processes involved, holistic universe is mathematically possible, although highly unlikely to be physically realisable. The EVM computationalism connects certain parts of the universe and provides only limited account for non-locality.

Although the models of Bohm and de Broglie [2] have not been yet empirically confirmed, they represent the most general and promising models to date. They maintain the deterministic nature of the Universe, and they are compatible with general notions of computationalism (or hypercomputationalism).

[9]This analogy can be pushed further, comparing traditional randomness-based quantum theories to statistical thermodynamics, and Bohm theory to particle physics. However, this analogy is not quite right, as we are dealing here with some more fundamental properties of reality, and the simple reductionist approach into sub-sub-atomic particles may not make sense anymore.

8 Randomness

The main problem of the traditional Copenhagen and similar interpretations of quantum physics is the requirement for inherent, fundamental randomness. At first glance, claims that in our Universe that is fundamentally discrete, finite, and not continuously divisible, randomness on the quantum level does seem consistent and does not cause any eyebrow-raising. Again, for computer scientist, claims like that are highly speculative, because of the infinity inherently hidden in the requirement for truly random process. It has been shown, that to have randomness in a discrete system, one would require an infinitely long (infinitely complex) computational process [6]. This is in a direct contradiction to the claim of the discrete and finite universe. In finite and discrete universe a true randomness is not possible. Arguments then, about fundamental randomness on quantum level in discrete and finite Universe is inconsistent. To postulate true inherent randomness on quantum level one would require continuous universe, or at least discrete infinite one.

9 Observations

The way to make empirical experiments and create models of the reality (to conduct science) is to provide some data, and observe the results of the computation (the black box model). This is the simple premise of computationalism. What may not be obvious, is that it opens doors for certain behaviour that is non intuitive. Because the computational process can modify itself, it follows that each measurement in literally creating new reality. This idea is not new, and has been proposed by scientists and engineers alike [9].

Our EVM architecture can take advantage of this property, and this is where the "evolvable" part is actually exploited. The computational processes can adapt and change, depending on the input that is being fed into the machine. Detailed discussion of this and similar properties is beyond the scope of this article.

10 Summary

Many researchers from different fields follow Einstein in the desire for a deterministic universe. Traditional computationalism is one way to progress the deterministic reductionist programme. However, traditional computationalism, based on Turing computation comes up short in explaining some of the existing phenomena, such as possibility of physical hypercomputation, cognition [4] and others. To address those issues, we propose the new research programme: hypercomputationalism. Our approach goes beyond the notion of Turing-computation, and is based on the notion of multiple asynchronously communicating, and self-referencing machines. The new model is based on the trial-and-error hypercomputing model, and on the multiple communicating machines. If the Universe turns out to be non-computable in nature, our model, unlike traditional computationalism, will be able to cope with that by means of uncomputable delays between massively concurrent interacting computational systems.

The EVM computational model has been implemented, and we are planning some large scale experiments on the Internet. If the Universe is not computable in nature, it might be possible to exhibit hypercomputation behaviour. The main relevance of the new computationalism is within the field of computer science, and its relevance to physics and biology needs to be establish and investigated further.

References

[1] Scott Aaronson. Np-complete problems and physical reality. *ACM SIGACT News*, March 2005. http://www.arxiv.org/pdf/quant-ph/0502072.

[2] David Bohm. *Causality and chance in modern physics.* Princeton, N.J., Van Nostrand, 1957. Foreword by Louis de Broglie.

[3] David Bohm. *The Undivided Universe.* Routledge, 1st edition, March 28 1995. ISBN: 041512185X.

[4] Selmer Bringsjord and Michael Zenzen. *Superminds: People Harness Hypercomputation, and More.* Studies in Cognitive Systems Volume 29. Kluwer Academic Publishers, 2003. Cen BF 311 B 4867.

[5] Vannevar Bush. As we may think. *The Atlantic Monthly*, July 1945. http://www.theatlantic.com/doc/194507/bush.

[6] Gregory J. Chaitin. *Meta Math!: The Quest for Omega.* Pantheon, October 4 2005. ISBN: 0375423133.

[7] Edward Fredkin. A new cosmogony: On the origin of the universe. In *PhysComp'92: Proceedings of the Workshop on Physics and Computation.* IEEE Press, 1992.

[8] Stephen J. Gould and Elizabeth Vrba. Exaptation – a missing term in the science of form. *Paleobiology*, 8:4–15, 1982.

[9] Steve Grand. *Creation: life and how to make it.* Weiendfeld & Nicolson, London, Great Britain, 2000. Sci QH 324.2 GR12.

[10] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation.* Addison-Wesley Publishing Company, USA, 1979.

[11] H.T.Siegelmann. Computation beyond the turing limit. *Science*, 268, 1995.

[12] Leonid A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

[13] Lynn Margulis. *Origin of Eukaryotic Cells.* University Press, New Haven, 1970.

[14] Lynn Margulis. *Symbiosis in Cell Evolution.* Freeman & Co., San Francisco, 1981.

[15] Humberto R. Maturana and Francisco J. Varela. Autopoiesis: The organization of the living. In Robert S. Cohen and Marx W. Wartofsky, editors, *Autopoiesis and Cognition: The Realization of the Living*, volume 42 of *Boston Studies in the Philosophy of Science.* D. Reidel Publishing Company, Dordrech, Holland, 1980. With a preface to 'Autopoiesis' by Sir Stafford Beer. Originally published in Chile in 1972 under the title De maquinas y Seres Vivos, by Editorial Univesitaria S.A.

[16] Mariusz Nowostawski, Lucien Epiney, and Martin Purvis. Self-Adaptation and Dynamic Environment Experiments with Evolvable Virtual Machines. In S.Brueckner, G.Di Marzo Serugendo, D.Hales, and F.Zambonelli, editors, *Proceedings of the Third International Workshop on Engineering Self-Organizing Applications (ESOA 2005)*, pages 46–60. Springer Verlag, 2005.

[17] Mariusz Nowostawski, Martin Purvis, and Stephen Cranefield. An architecture for self-organising evolvable virtual machines. In Sven Brueckner, Giovanna Di Marzo Serugendo, Anthony Karageorgos, and Radhika Nagpal, editors, *Engineering Self Organising Sytems: Methodologies and Applications*, number 3464 in Lecture Notes in Artificial Intelligence. Springer Verlag, 2004.

[18] R.Gandy. The confluence of ideas in 1936. In R.Herken, editor, *The Universal Turing Machine. A Half-Century Survey*, pages 55–112. Oxford University Press, 1988.

[19] Juergen Schmidhuber. Optimal ordered problem solver. *Machine Learning*, 54:211–254, 2004.

[20] Karl Svozil. Computational universes. *Chaos, Solitons & Fractals*, 12May 2003. arXiv:physics/ 0305048 v1.

[21] Gerard 't Hooft. Quantum gravity as a dissipative deterministic system. *Class.Quant.Grav.*, 16:3263 – 3279, 1999. TechReport: THU-99/07.

[22] Gerard 't Hooft. *Quo Vadis Quantum Mechanics, The Frontiers Collection*, chapter Determinism beneath Quantum Mechanics, pages 99 – 111. Springer, 2005. TechReport:quant-ph/0212095, ITP-UU-02/69, SPIN-02/45.

[23] A. M. Turing. On computable numbers with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936–7. also 43, pp. 544-546, 1937.

[24] Alan Mathison Turing. *Systems of Logic Based on Ordinals.* PhD thesis, Princeton University, 1938.

[25] Stephen Wolfram. *A New Kind of Science.* Wolfram Media, Inc., first edition, May 2002.

[26] Konrad Zuse. Rechnender raum. *Elektronische Datenverarbeitung*, 8:336–344, 1967. ftp://ftp.idsia.ch/pub/juergen/zuse67scan.pdf.

[27] Konrad Zuse. Calculating space. MIT Technical Translation AZT-70-164-GEMIT, MIT (Project MAC), Cambridge, MA, 1970. Original: Rechnender Raum, Friedrich Vieweg & Sohn, Braunschweig, 1969.