AISB 2011

Computing & Philosophy

Editors:
Dimitar Kazakov &
George Tsoulas

THE UNIVERSITY of York

Foreword from the Convention Chairs

The AISB'11 call for symposium proposals particularly encouraged events drawing more strongly on the cognitive science aspect of the AISB remit. The result is a coherent programme with a very strong interdisciplinary character, which is also matched in the choice of plenary speakers. The three symposia looking at the interaction between Computing and Philosophy, the prospect of machine consciousness and the quest for a new, comprehensive intelligence test, form a coherent unit where the eternal questions of who we are and what makes us so are asked from a dual Human-Machine perspective. The Symposia on Active Vision, Computational Models of Cognitive Development and Human Memory for Artificial Agents demonstrate how better understanding of the nature and basis of cognitive processes can advance work on Artificial Intelligence and, inversely, how computational models of these processes can help better to understand them. The prominent multi-agent design and modelling paradigm links the Symposium on Social Networks and Multi-agent Systems with the one on AI and Games. Finally, the Symposium on Learning Language Models from Multilingual Corpora, which brings together some of the first attempts in this area, can also be seen through the prism of such a general notion in Philosophy and Linguistics as semiosis, and the dual role of sign and interpretant that text plays in translations.

We are delighted that after another ten successful years in its long history, the AISB convention is returning to the University of York. The 2011 convention takes place on the brand-new Heslington East campus, the result of a multi-million pound expansion that is now the new home of the Department of Computer Science, and hosts the Excellence Hub for Yorkshire and Humber, a new incubator for interdisciplinary research and interaction between academia and industry. The last few years have seen a strong involvement of the Computer Science Department in such interdisciplinary collaboration through the York Centre for Complex Systems Analysis (YCCSA), and we hope that this convention will provide a boost for more synergy between York departments, with other institutions conducting AI-related research in the region, and beyond. As the programme shows, we have also made an effort to promote cooperation with industry and use the convention to support school outreach. The convention format makes it perfect for establishing dialogue and collaboration in new areas of research, as well as across disciplines, and we hope that this year, it will play again this role to the full. We want to thank everyone who has contributed to it or otherwise made this event possible and wish all participants a fruitful and enjoyable time in York.

Dimitar Kazakov and George Tsoulas

Proceedings of AISB'11
Dimitar Kazakov and George Tsoulas (eds.)

ISBN: 978-1-908187-03-1

Published by the Society for the Study of Artificial Intelligence and the Simulation of Behaviour

Printed by the University of York, York, UK

Contents

The AISB'11 Symposium on Computing and Philosophy

As a subject for philosophical investigation computing has a long history (including the work of such figures as Leibniz and Turing). With the rapid technological progress of electronic computing since the mid-20th century we have seen the emergence of deeper and broader interactions between computing and philosophy, although the scope, and need, for such interactions has not yet been widely recognised. For example, computing is contributing to classical philosophical topics such as the nature of mind, intelligence, agency, the varieties of logic, and how representation 'works'. At the same time the phenomenon of computing itself calls out for sustained philosophical attention to new problems such as the nature of software, the nature of the 'science' of computing as a discipline, the relationship between personal worlds and virtual worlds, and the significance of computer-based communications for personal identity. Both philosophy and computing stand to benefit from this continuing dialogue, particularly where it leads to investigation and creative responses to traditional problems in each subject. In the case of areas such as cognitive science and ambient intelligence, for instance, the pace of contemporary technological innovation requires immediate philosophical analysis, if it is to take account of important broader issues.

The purpose of the symposium is to further strengthen communication between these disciplines, thereby to advance the philosophical study of computing in general in relation to a number of key issues. These include traditional philosophical problems and the philosophical issues surrounding computational modelling. To this end the group of topics selected for discussion in the symposium include: computational creativity; creativity and swarm intelligence; human/computational relationships; computational theory of mind; artificial embodiment; contextual affect modelling; information processing; autonomy and desire; singularity; quantum computing; hardware/software distinctions; hypercomputation.

On behalf of the Organising Committee of this third AISB Computing and Philosophy Symposium, we would like to thank all the members of the Programme Committee for their generous support, and for the excellent work in refereeing submissions. We hope that participants will find the event stimulating and enjoyable.

Yasemin J. Erden, St Mary's University College, UK
Mark Bishop, Goldsmiths, University of London, UK

Programme Chair:
Mark Bishop (Goldsmiths, University of London, UK)

Organising Committee:
Mark Bishop (Goldsmiths, University of London, UK)
Yasemin J. Erden (St Mary's University College, UK)
Kevin Magill (University of Wolverhampton, UK)
Steve Russ (University of Warwick, UK)

Programme Committee:
John Barnden (University of Birmingham, UK)
Mark Bishop (Goldsmiths, University of London, UK)
Barry Cooper (University of Leeds, UK)
Yasemin J. Erden (St Mary's University College, UK)
David Gamez (University of Essex, UK)
Kevin Magill (University of Wolverhampton, UK)
Slawomir Nasuto (University of Reading, UK)
John Preston (University of Reading, UK)
Steve Russ (University of Warwick, UK)
Murray Shanahan (Imperial College London, UK)
Ian Sillitoe (University of Wolverhampton, UK)
Mark Sprevak (University of Edinburgh, UK)
Susan Stuart (University of Glasgow, UK)
Steve Torrance (University of Sussex, UK)
Raymond Turner (University of Essex, UK)
Tillmann Vierkant (University of Edinburgh, UK)
Hector Zenil (Wolfram Research, UK)

Creativity and art: three roads to surprise[1]

Professor Margaret Boden [2]

The three roads to surprise are the three forms of creativity: combinational, exploratory, and transformational. In previous writings, I've illustrated these by examples drawn from many different specialist domains, and from everyday life too (Boden 2004). Here, I relate them to art (as understood in the post-mediaeval Western tradition), and especially to the visual arts. Besides many references to traditional fine art, I also discuss some less orthodox approaches–namely, conceptual art and several types of computer art. Craftworks of various kinds are also considered, in the context of a new account of the distinction between art and craft.

A brief statement of my threefold theory of creativity, which underlies my recent collection of the same name (Boden 2010), is given in the chapter 'Creativity in a Nutshell'. Creativity in general is the generation of novel, surprising, and valuable ideas. "Ideas", here, is a catch-all term covering not only concepts and theories but also (for example) music and literature, and artefacts such as architecture, sculpture, and paintings. The three types of creativity, which elicit differing forms of surprise, are defined by the different kinds of psychological process that generate the new structures.

As it happens, those processes have been greatly clarified by computer models of creativity. But that fact is not a core theme of this collection. If and when computers are mentioned here, it is in the context of computer art, not computer modeling. Computer modeling is a form of science, and most computer artists never engage in it. Like their fellow artists working in other genres, they have no particular interest in detailing the processes that may go on in human minds. (The leading exception is Harold Cohen, who in the late 1950s–when already a highly acclaimed abstract painter–embarked on his AARON suite of programs in order to throw light on his own creativity: Cohen 1981, 1995, 2002.) So the remarks in 'Creativity in a Nutshell' that note the scientific value of computer models of creativity are not followed up here (but see Boden 2004: chs. 5-8 and 12).

The first road to surprise that's identified in 'Creativity in a Nutshell' is combinational creativity: the generation of unfamiliar combinations of familiar ideas. Most discussions of creativity, even in the specialist psychological literature (e.g. Sternberg 1988, 1999), consider only this type. (Of the roughly sixty definitions of creativity that had been offered by experts twenty years ago, almost all boiled down to this: Taylor 1988.) The combinational way of generating surprise is indeed important. It underlies most spontaneous jokes and word-play, and is the key source of poetic/literary imagery and visual collage–and of conceptual art, too.

But my account allows also for two other sorts of creativity. These involve the exploration and transformation of familiar conceptual spaces–such as artistic styles. A style is a (culturally favoured) space of structural possibilities: not a painting, but a way of painting. Or a way of sculpting, or of composing fugues ... and so on.

It's partly because of these thinking-styles that creativity has an ambiguous relationship with freedom. On the one hand, it's commonly thought of as being the very opposite of disciplined, or rule-governed, behaviour. Creative ideas are surprising because they are unpredictable (for several reasons–Boden 2004: ch. 9). Some are so surprising that they strike us as outrageous (think of the unfamiliar combinations within the literary conceits in Finnegan's Wake). Indeed, some are deliberately intended to be outrageous (conceptual art, again). And transformational creativity, by definition, amends/ignores some normally respected constraints. So there's a tempting launchpad, here, for neo-Romantic stories of the divine spark of creative freedom.

On the other hand, all three types of creativity exploit stylistic and/or conceptual constraints. In exploratory creativity it's especially clear that "artistic discipline" is not a contradiction in terms. But even the transformational variety preserves much of the preceding style. And even the most surprising visual juxtapositions (Surrealism, perhaps), and the most challenging poetic imagery, have a connecting thread of intelligibility. (This is lacking, for instance, in the undisciplined outpourings of a schizophrenic's 'word-salad'–which, despite being unpredictable and occasionally suggestive to eavesdroppers, is not in itself an exercise of creative thinking.) That intelligibility is grounded in the rich network of conceptual structures in an adult's mind (Boden 2004: chs. 5-6)–which structures are recognized by combinational creativity, not ignored by it.

These remarks imply that creativity also has an ambiguous position with respect to education, and in particular to autodidacts. For the "one hand" cited above suggests that a lack of conventional education need be no barrier to creativity–indeed, it's commonly held that education may actually inhibit it. But the "other hand" stresses the importance of stylistic constraints that may require many years, and the help of dutiful tutors, to learn. So perhaps autodidacts, contrary to common belief, are actually at a disadvantage when it comes to creative thinking?

In 'Are Autodidacts Creative?' (Boden 2010: ch.3) I outline the complex pattern of relationships between original thinking and self-education. As well as saying a little more about the three types of creativity, I distinguish three types of autodidact. The 3x3 matrix implied by these distinctions defines a variety of inter-relationships between creativity and auto-didacticism, sometimes mutually supportive and sometimes not.

Most of those inter-relationships apply irrespective of the domain concerned. In other words, art and science here are in much the same boat. Indeed, the threefold account of creativity that I have sketched applies to both, and some scientific examples are mentioned there.

---

[1] This work first appeared as the introduction to 'Creativity and Art: Three Roads to Surprise', Boden (2010) and is reprinted by kind permission of the author. It describes the individual papers in turn - although explicit references to chapters have, in the main, been removed from this text.

[2] School of Informatics, University of Sussex, England; email: m.a.boden@sussex.ac.uk

Much of my new book, however, focuses on art.

In 'Crafts, Perception, and the Possibilities of the Body' I compare creativity in the fine arts with craftsmanship. I also take up the old problem of distinguishing "art" from "craft". In one sense, I avoid that problem, for I don't offer mutually exclusive definitions of these terms. Nor are satisfactory definitions available in the literature: quite apart from the hugely controversial question "What is art?", one historian has identified many competing interpretations of "craft", none of which is entirely apt (Harrod 1999: 10). Instead of offering definitions, I refer to many specific examples that are normally labelled as one or the other, and ask what are the differences between the two classes.

My answer distinguishes the practices of art and craft, but also explains why they aren't always clearly separable. That is, the difficulty of definition here goes beyond the fact that all everyday concepts are fuzzy, allowing for borderline cases and anomalies. There is a specific (psychological) reason why it is impossible to assign every relevant artefact to only one of these categories.

The crafts are grounded in biologically evolved human tendencies to respond to certain things in particular ways. The psychologist's jargon, here, is "affordances" (Gibson 1966). A pot or a textile, a chair or a sword, a box or a jewel, afford diverse opportunities for (fitness-related) action–which opportunities are readily, 'naturally', perceptible by Homo sapiens. The perception excites a disposition to act in a certain way: in general, to approach or to avoid the environmental feature concerned–so affordances embody basic values. (The play of affordances here is more complex than one might think: besides their more obvious functionalities, highly skilled craftworks–including the beautifully symmetrical, and unused, hand-axes fashioned 1,400,000 years ago–are counters in the game of sexual selection, signalling their makers' physical strength, muscular control, perceptual acuity, power of concentration, and endurance: Miller 2000.)

In the case of craftworks, then, the value-criterion of creativity is grounded in our evolutionary biology. That's why the aesthetics of craft are more stable, and its appreciation more crosscultural, than the aesthetics of fine art.

It's sometimes said that a pure craftsman, untainted by 'art-envy' (see below), isn't creative at all–merely skilled. I wouldn't go that far. To the contrary, I'd say that craftsmen rely on exploratory creativity. But this is of a relatively unadventurous kind. Even an internationally famous master potter (for instance) may be aiming to produce yet another–albeit more perfect–example of something located in a part of the possibility-space that has been visited many times before. The result can be an artefact that amazes us, in its (affordance-based) power to engage our attention/valuation. But if gaining our attention and positive valuation is part of the point of the exercise, producing amazement is not.

In other words, the crafts downplay the role of surprise in creative work. Even the best examples typically display only minor novelty, and our wonder is elicited less by what is done than by its being done supremely well. The crafts aren't dependent on highly imaginative combinational creativity, nor driven by increasingly adventurous exploratory creativity, nor sporadically progressed by transformational creativity–as fine art is.

That's not to say, however, that craftwork never involves journeying on these three roads to surprise. It's true that the key mental capacities that underlie the generation (and appreciation) of craftwork differ from those which underlie fine art. The former relies on basic affordances, whereas the latter relies on the high-level psychological processes involved in learning, retaining, exploring, challenging, and (sometimes) transforming culturally specific concepts and/or artistic styles. But both types of mental process can occur within someone's mind simultaneously, with respect to different properties of a single artefact.

This psychological fact makes it possible for a potter, jeweller, or tapestry-maker to exploit knowledge of specific fine-art styles in their work. Indeed, it makes it possible for them to do this in order to raise their social status in a culture that values fine art over 'mere' crafts–hence my reference, above, to art-envy. Similarly, a professional craftsman irritated by unsympathetic cultural values may deliberately, and atypically, aim for surprise–by following one or more of the three roads I've distinguished. This explains why there can never be a definition of crafts that can unambiguously distinguish every individual craftwork from a work of art.

In saying that craftworks don't involve adventurous creativity, I've assumed that the craftworker is using already-accepted techniques. But it is possible, of course, for someone to create (by combination, exploration, or transformation) a new way of firing pots, or of working metals. An intriguing recent example of the creative (combinational) use of novel techniques in craftwork is 'digital jewellery'. Here, rings and bracelets are not only attractive physical adornments but also digital devices. Considered as items of jewellery, they afford (sic) the social communications for which jewellery has evolved. But they can do more, for impersonal digital techniques can be put to highly personal uses. For instance, digital jewellery may be connected with some remote location of personal significance to the wearer, such as their birthplace–where their family may still reside (Wallace et al. 2007). Instead of a Victorian locket containing a lover's likeness, or a mourning brooch woven from a dead spouse's hair, these baubles may display images/sounds drawn from the wearer's own past, or from their far-flung relatives' situation at this very moment.

The crafts, then, typically underplay surprise–and are often dismissed as being uncreative, accordingly. But if surprise is a key criterion of creativity, the 1970s movement known as conceptual art shouldn't suffer the same fate. On the contrary, it should be seen as highly creative. For to say that conceptual art is surprising is an understatement. It is shocking, bizarre, outrageous, challenging ... so much so, that it's often said not to be art at all. In 'Creativity and Conceptual Art', I ask just which sort of creativity it involves. Which of the three roads leads to these artistic astonishments?

Prima facie, conceptual art may seem to be a case of transformational creativity. For its artworks are very different from traditional ones. Moreover, the conceptual artists were undoubtedly trying to effect a radical change in the public's ideas and expectations about "art". However, not every radical change counts as a "transformation", in the (stylistic) sense that I define this. Looked at more closely, conceptual art is an example–or rather, a set of highly various examples–of combinational creativity.

Each of the individual artworks generated by this movement involves some unfamiliar, often highly challenging, juxtaposition of ideas. And "ideas" should be interpreted literally, here. For a conceptual artist, as opposed to an orthodox fine artist, the physical artefact is not the main point. Indeed, there may be no physical artefact, merely (for instance) a verbal injunction to imagine one. Even if there is, the artist may have ensured that it remains utterly invisible to the 'audience'.

The key aim, then, is not to generate intrinsically beautiful objects, nor disturbing ones either. Admittedly, Eduardo Kac outraged people by producing a genetically engineered albino rabbit that turned a fluorescent green in ultraviolet light. But even this was as much an

idea as a thing: the notorious green image that sped around the world via the mass media was not an actual photograph, but a Photoshop-generated design; and the rabbit was never publicly exhibited, because fears about mad-cow disease prevented agricultural movements around the laboratory where it was born. Nor do conceptual artists aim to display hard-won painterly or sculptural skills (so they might be autodidacts). Rather, they aim to raise a host of questions in the minds of the audience. Many of these concern the nature of "art" itself–and its twentieth century social context, the art market.

Many of the conceptual artists were challenging the popular view of art wherein art is a highly personal matter: not merely effected by a person (not by a machine), but by some particular, unique, human individual. This view has its roots in the humanism of the Renaissance, but was strengthened by early-nineteenth-century Romanticism. Coincidentally, the historical circumstances of the mid-nineteenth century led to a need (on the part of collectors and curators) for better ways of attributing old paintings or sculptures to one artist rather than another. And this, in turn, led pioneering art connoisseurs to identify the characteristic marks, or personal signatures, of different artists.

'Personal Signatures in Art' not only describes how this notion arose, but also asks why personal signatures exist in the first place. What is it about the creative process which makes personal signatures near-inevitable?

Even exploratory creativity, which is the most highly constrained of all three types, leaves many points for individual choice. Were that not so, each artistic style would allow only one instantiation: a single painting, sculpture, fugue ... in that particular style. And because of certain general features of mental information-processing, necessitated by the finitude of our minds, it is highly probable that individual artists will develop idiosyncratic habits of working which distinguish their art from that of other artists–even those exploring the same style. Combinational creativity is more free, less predictable. But here too, general psychological features (affecting the perception of relevance, for example) will engender patterns of thought that are specific to the individual artist.

A further question raised here is whether personal signatures are not merely highly probable, but wholly inevitable. Why have those fine-artists who (in a twentieth-century reaction against Romanticism) have deliberately tried to avoid the personal signature been only partially successful at doing so? And why have they failed even when turning to impersonal machines for help?

In particular, is it possible for a human artist to lose his personal signature by engaging in a type of computer art wherein a robot is evolved (not specifically designed) to draw aesthetically acceptable marks that don't betray his authorship? One internationally famous artist is already trying to do precisely this–but it's by no means clear whether he will, or even could, succeed.

Using a robot, as opposed to a computer screen, is not mere gimmickry. For it makes it possible for serendipitous events to happen, wherein the robot interacts with some previously unconsidered aspect of its physical environment. In other words, the drawings that result needn't be wholly dependent on exploring the space of possibilities pre-defined by the program (including its mutation rules). In principle, some fundamental transformation could occur–comparable to evolving a first-time eye, not just an improved eye. (A first-time sensor has already been evolved in practice, by someone involved in the art-oriented robotics project being considered here: Bird and Layzell 2002.) In short, something deeply surprising could conceivably emerge from the evolutionary processes underlying the robot's behaviour (cf. Boden forthcoming). Such a result isn't guaranteed, and is even pretty improbable–but it is possible.

One might think that the desired loss of signature must be possible too. After all, the robot's final line-drawing behaviour will be the end-point of a process involving myriad random mutations. Indeed, one might think that this randomness makes it impossible for the evolving robot not to lose the artist's telltale sign. However this issue is not so easily decided.

The problem is that the signature-fleeing artist himself will have the final say in choosing the criteria of selection (the 'fitness function') that are used at each generation to pick the 'best' mutants for further evolution. His chances of success–that is, of enabling the robot to lose his personal mark–depend on the degree to which our (and his) aesthetic preferences rest on basic, culture-free, properties as opposed to culturally, or even personally, specific styles. These basic properties might include some of the affordances favoured in craftwork, but could also include other features that are fundamental to visual perception. Certain fractal properties, for instance, might be naturally attractive. In short, this project raises empirical psychological questions as well as philosophical ones.

Why "philosophical" ones? Well, references to robots making line-drawings may raise the hackles of some readers: "These papers are supposed to be about creativity and art" they may grumble, "Robots, in principle, can have nothing to do with either". In other words, they believe that there can be no such thing as computer creativity, and (a different, though related, point) that there can be no such thing as computer art.

With respect to the latter claim, some philosophers justify their refusal to admit the possibility of computer art by defining "art" in exclusively human terms. Anthony O'Hear (1995), for example, insists that art involves some form of communication between one human being and another. For this to be possible, he says, artist and audience must share human experience.

He would be willing to admit that computer art exists in the sense in which watercolour art, or marble art, do: that is, a computer can be used as an artist's medium. He'd probably allow, also, that a computer can be an artist's tool, or aid–perhaps even an artist's assistant (although that is more questionable). But if any 'artwork' is generated by the computer itself, by means of processes that are largely beyond the human artist's control, then–for O'Hear–it isn't really an artwork at all. It may happen to be visually/aurally arresting, decorative, or even beautiful. But to respond to it as an artwork is, he says, to be deceived. Even to see it as aesthetically valuable is to be largely misled. On discovering its provenance, our aesthetic satisfaction would–and should–decrease, even evaporate. (I have witnessed people making this sudden shift of evaluative attitude on several occasions.)

O'Hear is not alone in such views. For anyone who defines art in such a way that human experience and/or human creativity is essential to it must be sceptical about the notion of computer art. And the more the computer 'artwork' is generated by processes going on in the computer itself, the stronger their scepticism must be. At best, computer art will be seen as art at one remove, thanks entirely to its human instigation. Even if art is defined in terms of the natural, as opposed to the human, the notion of computer art will remain problematic. If, by contrast, art is defined in terms of properties of the art object that are not exclusively human and/or natural, talk of computer art might escape challenge.

However, I shan't offer any 'non-human' definition of art, designed to allow the inclusion of the computer-based varieties. Quite apart from the air of special pleading that would attend such a definition, it would require lengthy argument that would be out of place here. For the definition of art is a notoriously slippery matter, which

often threatens to exclude works that many people regard as art–such as conceptual art, and certain items of craftwork. I'll rely instead on the (undefined) common usage of "art", and on paradigm cases of it–from Fra Angelico's delicate murals to Mark Rothko's glowing colour constructions. The more problematic cases can be accepted as art to the extent that they show similarities to and continuities with the commonly accepted examples.

One more point must be made before talk of computer art can be specifically defended. Namely, our intuitive concepts of art, if not our explicit definitions, typically see it as creative. Indeed, the link is so close that people often fail to realize, or anyway forget, that science and mathematics involve creativity too. This leads to a further problem in speaking of computer art, since many people insist that no computer can really be creative. They may be willing to grant that a machine may generate novel, surprising, and even arguably valuable results: lifelike and/or beautiful images, for example. But, they say, the creativity involved can be attributed only to the human being/s who made it behave in that way.

This claim is usually grounded in arguments involving one or more key philosophical concepts that are (plausibly) assumed to be essential for creativity. These concern consciousness, intentionality, the role of 'brain-stuff' and/or embodiment, and membership of the human moral community. I've argued elsewhere that although the brain-stuff argument can be rejected, each of the others remains highly problematic (Boden 2004: 286-300). What's more, they are problematic primarily because of the disagreements concerning these philosophical concepts themselves. If we understood intentionality better, or consciousness, we'd be in a better position to pronounce on whether or not computers can "really" be creative.

Since these notoriously controversial problems remain unsolved, I nowhere claim that computers are "really" creative. If and when I mention creativity in computers I am asking what aesthetically interesting results can computers generate, and how? and Just what might lead someone to suggest that a particular computer system is creative, or that its functioning is somehow similar to creativity in human beings? In that sense, I'm content to leave the question of "real" computer creativity open. And if art necessarily involves creativity–a reasonable, if not a strictly provable, view–then (in that sense) I must leave the question of "real" computer art open too.

So I shan't try to prove that computer art can exist because it fits some favoured (and tendentious?) definition of art and/or of creativity. Instead, I'll rely on two strategies to persuade sceptical readers that this isn't an empty class. On the one hand, I'll point out many similarities and continuities between computer art and the more familiar varieties. On the other hand, I'll mention some examples where the work of computer artists is taken seriously as art by aesthetes of an orthodox kind. For instance a computer artwork was included in the Washington D.C. exhibition mounted in 2007 to celebrate the 60th anniversary of the ColorField painters: Rothko, Clyfford Still, Kenneth Noland, and the like (Edmonds 2007).

Despite the welcome imprimatur of the Washington gallery (and many others, including the Tate), computer art is still largely unknown even to art lovers and aestheticians. So 'What Is Generative Art?' (co-authored with Ernest Edmonds) offers a novel taxonomy of work in this genre–along with an indication of the philosophical issues that attend the various categories. As well as distinguishing significantly different types of computer art, this taxonomy displays several connections between computer art and more established forms.

In this context by different "types" of computer art I mean different techniques for producing computer artworks and/or different types of experience on encountering them. But one might also distinguish these artworks by the differences in their physical implementation–which cut across the generative distinctions used to draw up the taxonomy.

For instance, some computer artworks are framed 'pictures' hung on the wall. These may be unchanging images, both produced and printed by a computer program (e.g. Todd and Latham 1992). Or they may be ever-changing coloured patterns, the changes being prompted by the viewer's movements–thanks to a mini-camera and mini-computer hidden in the wooden frame (Edmonds 2007); a few involve physical robots. Whether attached to walls or ceiling, o ranging free over the floor, their movements have some intrinsic interest and/or produce results, such as sounds or line-drawings, that the audience finds intriguing. Others are interactive CD-Roms, which provide differing experiences as a result of the viewer's input (Leggett 1996). Yet others are static or (more usually) dynamic video-projections, perhaps presented on a computer monitor or perhaps filling an entire wall. And some of these are virtual reality environments, a millennial form of trompe l'oeil (a genre employed by artists since Roman times: Grau 2004) often projected onto all four walls, and maybe floor and ceiling too. Occasionally, it's not only the audience's eyes that are deceived, but their ears and (if special gloves are worn) their fingertips too.

Whereas all of those examples are located inside a building, whether an art gallery or someone's home, others are exhibited in bustling city squares. In that case, the installations are typically huge: much more than human-size. Being out-of-doors, their form may change as a result of weather conditions, as well as of the movements of the people passing by.

One must add, however, that some computer artworks aren't physically located at all. Rather, they exist on the Internet. (Thor Magnusson has suggested an additional entry for the taxonomy: N-Art, or Network art.) These works are accessed–and developed–by human beings located in physical space: staring at their PC screens, for instance, or using their mobile phones, or playing musical instruments while on-line. They may be altered, to some (highly variable) degree, by input coming from those individuals. But the artwork itself, even if there happens to be some physical installation at its core (which there may not be), isn't really located anywhere–except in cyberspace.

A prominent early case of Network art was Ken Goldberg's Telegarden. Developed at the University of Southern Californa in 1995, this was installed in the Ars Electronica Centre (now renamed the Museum of the Future) in Linz, Austria, a year later; it ran nonstop for nine years, until being switched off in 2004. Unlike many N-art works, it did have a physical core: a garden filled with living plants, which were planted and watered by means of a robot arm. The garden's progress could be monitored through images from an on-site camera. The movements of the robot arm were remotely directed by web-users all over the world: 9,000 people connected with it in its first year. Besides remarking on a wide range of ecological/environmental meditations prompted by this artwork (see http://goldberg.berkeley.edu/garden/Ars/), the users reported feelings of human community of a (distributed) type never experienced before (McLaughlin et al. 1997). The nearest analogy would be their prior experience, if any, of web-based 'games' involving huge numbers of players (Turkle 1995).

The Telegarden example shows us that tricky ontological questions arise with respect to some computer artworks. Is the garden the artwork?; or the community of human users that's been built up over the years?; or their comments and meditations, shared on the web alongside camera-images of the plants?; or ... ? Again, consider line-

drawing robots: are the robots the artwork, or is the artwork rather their drawings? Or perhaps both?

Conceptual art can engender similar conundrums (or conundra, if you prefer!). The work called "42nd Parallel" is said by its instigator–one can hardly say its "maker"–to consist in a geographically dispersed pattern of activity in the US postal system. As such, it isn't clearly located either. Of course, tricky ontological problems can arise with respect to much more familiar forms of art than this (a classic discussion is: Goodman 1968). So ontology is one of the various philosophical/aesthetic dimensions on which this new category of art is related to 'art as we know it'.

One difference noted in the taxonomy is that between computer-assisted or computer-aided art (CA-art) and computer-generated art (CG-art). (This distinction underlay my claim, above, that O'Hear might allow the possibility of computer-assisted art, though not of computer-generated art.) In CA-art, the human artist produces the artwork with some help from the computer (which is in principle non-essential). In CG-art, the artwork is produced by the computer itself, with minimal or zero interference from a human being.

The terms "computer-assisted" and "computer-aided" art are normally used interchangeably, and the category of CA-art covers both. But one might want to make a further distinction here. A tool (e.g. a paintbrush or chisel) that's wholly under the artist's control is more readily thought of as an aid than as an assistant. And indeed, the examples of CA-art given in Boden (2010: ch.7) involve off-the-shelf programs (Photoshop and video-editors) used by the artist as tools in the production of many different artworks. But 'Agents and Creativity' suggests that CA-art could also involve specially-written programs, containing AI "agents" for some particular style of art. As this label implies, these would be conceptualized–and experienced–by their human users less as mere tools than as semi-autonomous assistants, capable of cooperating (sic) in the task at hand.

AI agents in general have a significant degree of independence from the human being who is using the program. Indeed, they are often termed "autonomous" by AI researchers. There are two reasons for this. First, they are not deliberately called up by the human user, but are automatically triggered by specific cues: events occurring within the running of the program or in the environment (maybe including the user's actions). And second, they are not amenable to interference from the user once they have started to run.

Whereas some agents are relatively simple processes, comparable to a reflex knee-jerk, others are more like mini-minds. These can set and follow goals, and cooperate with partner-agents. For example, they can devise engagement schedules (avoiding conflicts with entries already in the user's diary), book hotel rooms, and arrange for flights and car-hire–perhaps without bothering the user, or perhaps making suggestions for human ratification (Norman 1994). Some can even learn how to do better in future by inferring, or being told, why their suggestion was rejected (Mitchell et al. 1994: 87). Where such 'mini-minds' are concerned, the user's illusion of having a quasi-intelligent assistant can be fairly strong.

Sometimes, the agent's action is to send a message to the user. This may be a warning, saying that he/she has made a mistake or that some danger-point is being approached. Or it may be a suggestion about what to do next: perhaps how to rectify the mistake, or avoid the danger. The user can then decide whether or not to heed the agent's advice.

The existing computer-art programs that are mentioned in Boden (2010: ch.8) do not contain agents: they exemplify CG-art, not CA-art. They include (exploratory) programs for designing Palladian villas or Frank Lloyd Wright's Prairie Houses, and for improvising jazz. Each of them, I suggest, could in principle be modified to form an agentive version. So too could large semantic networks–so as to help writers, whether in advertising studios or in garrets, to find and develop conceptual associations (alias combinations). In practice, however, agent-based computer art is thin on the ground.

The reason, I suspect, is that (with a caveat mentioned below) the more strongly the human user identifies him/herself as a creative artist, the less likely that they will want to rely on AI agents as design-crutches. They may be happy to bite the bullet–indeed, to swallow it whole–and go down the route of CG-art. But that's a different enterprise (one which could well involve agents working behind the scenes, like the diary-organizer mentioned above). In other words, CA-artists may feel that an agentive CA-system would compromise their own artistic autonomy, integrity, or authenticity: they want computer aids (tools), not computer assistants.

The caveat, here, is that some CA-artists might be perfectly happy to have the "A" mean "assistant", given that some non-computer artists rely heavily on human assistants in their work. Examples range from the Renaissance masters to conceptual artists such as Sol le Witt and Jeff Koons. The sixteenth-century masters would sometimes merely sketch the outlines of the picture and paint the faces of the key people depicted in it, leaving the drapery and/or background to be executed by their apprentices. And the conceptual artists all underplayed the role of personal art-making skills, if not of creativity, in their work; Koons, for instance, became notorious for employing others to paint 'his' canvases. So for a CA-artist who sympathizes with that general art movement, there's no reason to avoid using computer agents as assistants.

Nor is there reason to avoid this if the creative activity is thought of as practical design, as opposed to art. For instance, the computer-assisted design (CAD) programs used today by professional engineers and jobbing architects can monitor the provisional decisions of the user, identifying mistakes and sometimes offering suggestions. If a design for a building had a potential structural weakness, for instance, an engineering-wise CAD program could warn the architect of that fact; it might also be able to suggest how the fault could be put right. Or suppose that an architect's client had requested a building like a Prairie House, and that there were no CG-art program capable of designing one entire: in that case, the architect might find it helpful (sic) to have a set of Prairie-agents, to be consulted at particular choice-points during his design work. But the results wouldn't be presented to the world as "art". Moreover, no self-styled creative architect would be spending his/her time copying Lloyd Wright.

I said, above, that some computer artists may be loath to use computer assistants for fear of jeopardizing their own autonomy, integrity, or authenticity. And some readers will surely sympathize, feeling that concepts such as these can have no place in computer-based art–least of all, where the artwork is generated by wholly automatic processes. On their view, artists who adopt a CG-art methodology thereby abandon any claim to such epithets. The next two papers address these issues.

As for the first member of the problematic trio, some computer artists justify the value of their work in part by citing the "autonomy" of the computerised system concerned. They have inherited that terminology from the AI researchers whose methods they are exploiting. We've seen, for example, that agents are commonly termed autonomous within the AI community. Critics may object that the fact that computer scientists speak in this way merely shows that their field doesn't foster sensitivity to natural language. However, even if we ignore that objection and focus instead on the nature of the systems themselves, we must recognize that there are significantly dif-

ferent senses of "autonomy". If autonomy does have any aesthetic value, we need to know what type of autonomy is in question in a given case.

In 'Autonomy, Integrity, and Computer Art' (Boden 2010: ch.9) I show that the various senses of autonomy are distinguished not by mere nuances, but by differences that include some seemingly radical oppositions. So natural autonomy includes biological homeostasis, psycho-physiological reflexes, various kinds of animal behaviour, and human freedom. Likewise, the autonomy that characterizes (some) computer systems is comparable to those very different phenomena. (We've seen already that computerised agents are sometimes like reflexes, and sometimes like goal-seeking mini-minds.) Different types of computing methodology are best suited to achieve different types of autonomy.

It follows that to understand the vexed concept of autonomy we need to understand the differences that are involved. In the case of the natural autonomies, this requires biological and psychological knowledge. In the case of the seemingly paradoxical concept of computer autonomy, which some computer artists see as a key source of value in their work, it requires some knowledge about the details of the programs concerned. To ascribe (or to withhold) any particular sense of autonomy to/from a computer artwork therefore requires one to know something about just how the relevant program works. As remarked in Boden (2010: ch.9), the epistemology of art is thus more taxing in the case of computer art than for more familiar forms.

Admittedly, knowledge of the details of art-making can enrich art criticism in conventional areas too. For instance, someone who understands just how to place paint on a surface is in a better position to appreciate certain aspects paintings than someone who has never held a paintbrush. This is evident, for example, in a history of art written by the painter Julian Bell. Bell repeatedly points out virtues of artworks that result from the way in which the paint is applied, and often invites us to imagine the 'feel' that the artist would have experienced in making the work. For instance, he explains Rothko's colours that "pulse against one another–now drawing in, now glaring out" by referring to his ability to "coax big cloud-blocks of colour from the canvas with a soft glazing-brush" (Bell 2007: 415f.). And in writing about Giorgione, he says that "[his] brushwork, for the first time, exploits the unevenness of canvas, a surface for paint ... only recently adopted in Northern Italy. The definitiveness of brush-marks made on a perfectly flat wooden panel is gone, but something more alluring replaces it. A loaded brush quickly traversing canvas leaves traces on its 'teeth', not its valleys: the viewer, induced to complete the intended line in the imagination, also enjoys by proxy the sensation of the action that produced it" (2007: 189). Even the art connoisseur Giovanni Morelli, whose loving knowledge of Giorgione's paintings is mentioned in Boden (2010: ch.6), would not have picked up on that.

The second member of the trio, namely integrity, is a special case of autonomy–indeed, a special case of human freedom. It can arise only in adult human minds capable of holding, integrating, following, and also abandoning general principles of behaviour. These may be moral, political, religious ... or aesthetic. In general, to have integrity is not merely to show consistency and coherence in following one's principles: it also involves resisting–while also recognizing–the temptation to follow an easier path, in which those principles would be betrayed. So integrity is not simple innocence. Honesty is here joined with a refusal to compromise, where it's evident that compromising would in some ways be more comfortable.

An artist can be praised for the integrity of the content of their work, and/or for the integrity of its observable style. By the same token, they can be criticized for lacking integrity in these matters. Such judgments are common in the traditional arts.

Computer artists are subject to these sorts of evaluation too. And, again like their more orthodox fellows, they can be criticized for lack of either moral/political or logical/structural integrity. So all the familiar disputes about the relation between morality and art can arise in this relatively unfamiliar context: think of violent, sadistic, or pornographic video-installations (or computer games), for example. Similarly, a critic who praises Picasso for choosing to paint Guernica, or who condemns Jane Austen for her novels' silence on the Napoleonic wars, might complain of a lack of integrity in computer artworks with a socio-political content.

But computer artists might also be judged in terms of the technical integrity of the computational methods they use to achieve their art. In such cases, the art critic must understand how the various methods are distinguished, and how they may be integrated (sic) within a single artwork.

Chapter 9 of my recent book (Boden 2010) makes this point by reference to "hybrid" computer systems, in which two (or more) normally distinct approaches are combined. Methodologically, a hybrid system is less neat, less 'aesthetically' pure, than a single-method system. As such, it is comparable to a mixed-media work in art. If a hybrid computer artwork is not to be scorned as an inelegant ragbag of programming tricks, it needs to display a smooth switching between the various methods at appropriate points, and an overall result that is valuable in itself and which could not have been achieved in any other way. (I describe a psychological example: a hybrid system that models voluntary actions, and certain pathological disturbances of action caused by brain damage.)

Third, authenticity. This concept, too, is often used in critiques of the more familiar forms of art. 'Authenticity and Computer Art' distinguishes various senses of authenticity, and asks whether any can be satisfied by computer art.

Some critics argue that all computer art must be inauthentic–not even bad art: rather, not really art at all–because computers lack emotions. Others doubt its authenticity on the grounds that (non-interactive) computer artworks, even if they happen to be unique, are unlimited in quantity: in principle, they could be churned out for ever. Yet others refuse to respond to computer-generated works as anything but "computer output", not deserving the critical/appreciative thinking that greets traditional fine art. These negative responses are likely to be compounded if the computer can be seen as producing pastiches (even forgeries?) of the work of specific human individuals.

One renowned computer artist, the composer David Cope, has been so frustrated by these attitudes towards the results generated by his 'Emmy' program that he has recently destroyed the program's musical data-base, painstakingly built up over the last twenty-five years. For him, Emmy composes music, to be appreciated (or not, of course) as such. To consider it as mere computer output, on his view, is to miss the point.

People who refuse to treat Emmy's music as music (alias art) thereby threaten Cope's perceived status as an artist. This raises the question why anyone would want to do computer art in the first place. One of the three main reasons identified in Boden (2010: ch.!0) is to produce and exhibit works of art so to gain a public reputation as an artist. If the "public" adamantly deny the authenticity of any computer art, that hope will be frustrated. A would-be artist who does not already have a reputation as a painter (like the young Cohen), or as a composer (like the young Cope), might think twice before embarking on this unfamiliar path. For if this public attitude persists, they will be honoured as an artist only by a relatively small group of people.

My own bet would be that this public attitude will not persist. Although much computer art is still confined to a niche market, some examples have been exhibited in prominent public spaces–from the Millennium Dome in London (see below) to busy squares in Melbourne or Washington. The Internet, of course, ensures that some computer art is displayed–and, in the case of Network art, made freely open for worldwide creative participation–on the web. And sometimes, as remarked above, computer art is deliberately placed in the same aesthetic space (under the same gallery theme) as the work of more orthodox artists, such as Rothko.

Conventional galleries/museums face many difficulties in exhibiting computer art: new curatorial practices, and some dedicated facilities, will be needed (Leggett 1999). Increasingly, however, people won't have to decide to enter one of the (still rare) galleries that regularly feature these types of art. They won't even have to decide to visit one of the occasional exhibitions held at the Tate and comoarable venues. Rather, they will come across them willynilly–much as people today can't avoid seeing statuary in public places.

There's a caveat needed here, however. The continual invention of new types of interface doesn't merely present a practical problem for curators: some of the potential audience may become irritated and even bored by "the continually transitional character of the medium" (Kahn 1996: 30). Even the artists themselves may be discouraged by the amount of labour involved in adapting to yet another novel interface. (Not the young men, perhaps: "Meanwhile, computers are mobbed by adolescent males as if the mouse was a pimple-cream dispenser, and the whole scene is flanked by a smattering of art critics and journalists bent on asking what's next? or so what?" –Kahn 1996: 21.) In addition, computer artists face a problem similar to that facing librarians: computer technology fast becomes out-of-date, even unusable. Champions of art implemented on interactive CD-Roms point out that these are relatively permanent, like the bronzes of old (Leggett 1996). Even so, a suitable (out-of-date) CD-player must still be available.

'Aesthetics and Interactive Art', (Boden 2010: ch.11) turns to another category: interactive art. Here, the form/content of the artwork is significantly affected by the behaviour of the audience. Such art need not involve computers. Interactive theatre, for example, does not. Indeed, Marcel Duchamp (1957) maintained that every artwork, even the Mona Lisa, is part-created by the observer, because in interpreting it they "add [their] contribution to the creative act". Nevertheless, computers have hugely enriched the potential of this type of art, because they enable one to specify an indefinite variety of interactions between audience and artwork. Most of these would have been impossible, even inconceivable, without the new technology.

To give just one example, Richard Brown's Mimetic Starfish delighted visitors to London's Millennium Exhibition, and was even described in The Times as "the best thing in the Dome". This seemed to be a huge multi-coloured starfish, trapped inside a near-transparent table but managing to move nevertheless. Moreover, it moved in very lifelike ways. And it did so in response to the actions of the visitors. Mostly, these movements suggested mere cautious interest on its part, but if someone shouted at it or approached it very suddenly it would 'freeze' as though in fear. In fact, it was a coloured image projected down onto the table from the ceiling, whose changes were generated by a self-organizing neural network that was sensitive to various aspects of the visitors' behaviour.

Computerised interactive art has raised new aesthetic questions, for the aesthetic interest is focussed less on the images and/or sounds that are produced than on the (hugely diverse) nature of the interaction itself. The 'artwork', one might say, is the interaction at the heart of the entire human-computer system, not just the visible/audible results. And here, there is a good deal of disagreement.

For example, these artists differ over what degree of predictability and/or personal control will afford the greatest interest or satisfaction to the human participant. They even disagree over the extent to which, and the speed at which, the person should be able to realize that they are actually affecting what is going on.

As implied in the previous paragraph, the term "participant" seems more appropriate here than "audience". Indeed, many interactive artists, irrespective of whether they use computers, have made a point of stressing the creative role of the person who would normally be called the audience, not just of the person whom one would naturally identify as the artist. Sometimes, this attitude is explicitly justified in post-modernist terms (citing "the death of the author"), and also in terms of democracy: everyone an artist (Ascott 2003). This ideological justification invites a fairly high degree of participant-control as an aesthetic criterion, for if one cannot deliberately change the display in particular ways then it's not clear that one can be seen as creating it. (Causing it, yes; but that's not the same thing. Conscious monitoring, if not conscious planning, is usually involved: hence the common insistence on consciousness as a mark of "real" creativity–see above.)

The shift from audience to participant means that characterizing and attributing the creativity involved can be tricky. In general, the gallery-visitor explores (sic), more or less imaginatively, the space of possibilities implicitly defined by the system. But those possibilities can differ in kind, as we've seen.

For instance, suppose that a visual transformation occurs. The artist must have written the program so as to allow for this, but it would not have happened unless the audience/participant had behaved in a particular way. However, the gallery-visitor may have had no intention of causing a transformation, and may not even recognize it as such once it has happened. In such a case, they are a cause–but hardly a creator. Again, the system may have been set up to enable creative exploration of a particular conceptual space, but the participant may or may not actively explore it. (Stamping one's feet a couple of times hardly counts as exploration.)

Similarly, if combinational creativity is the name of the game there are at least two people effecting the combinations: the designer-artist and the participant-creator (of which there may be several). If the artist is highly imaginative in the combinations that he/she allows, the audience may not be. In other words, the creative potential in the artwork may be much greater than is evident from this particular audience's interactions with it.

Yet another major category is evolutionary art (Evo-art). Here, the artwork is evolved by processes of random variation and selective reproduction that affect the art-generating program itself.

Some computer artists employ evolutionary methods for purely pragmatic reasons. Perhaps they want to maximise unpredictability (within certain boundaries) or to try to lose their personal signature. Perhaps they simply want to save on physical effort, by switching from part-random art produced by means of paper, pencil, and dice to Evo-art generated by computer (Todd and Latham 1992: 2-5). But many artists who use evolutionary programming do so largely, even primarily, because of evolution's close connection with life. Much as a painted landscape may be intended as a celebration of the sublimity of Nature, so an Evo-artwork may be intended as a salutation to the wondrous phenomenon of life.

All the living organisms we know about have evolved. Indeed, evolution is often taken to be a defining feature of life. Moreover, many of the scientists who write evolutionary programs do so in the con-

text of artificial life, or A-Life. This area of research, using computer models and mathematics, aims to define the general principles of life in abstract, functionalist, terms: "life as it could be", not just "life as we know it" (Langton 1989/1996; cf. Boden 2006: ch. 15).

A-Life is not concerned only with evolution, nor reliant only on evolutionary methods of computing. For instance, it provided the simple algorithms that underlie the realistic animations in Jurassic Park, wherein each dinosaur in the flock follows its own idiosyncratic path but manages both to keep up with its fellows and to avoid bumping into them. A-Life has also provided models of cell formation; of the origin of naturalistic patterns (e.g. on the fur of dalmatians, leopards, cheetahs, and giraffes) from interacting waves of chemicals; of the ever- branching structures of plants; of the autonomous development of an initially random neural network into organized 'columns' of cells like those in visual cortex; of the emergence of cooperation within groups of minimally communicative robots ... and so on.

This scientific field has inspired a number of computer artists, including animators and designers of virtual reality installations. For example, A-Life's stress on autonomy has drawn a variety of artists towards the field. Again, many CG-artists produce images by using an A-Life methodology called "cellular automata", in which each unit in a large array behaves in a way that's determined by the current states of its close neighbours. And some of these artists choose that approach partly because of the strong analogies between cellular automata and multi-cellular organisms–analogies which they regard as adding to the value of their creations. In this, they resemble a jeweller who chooses to work with pearls not just because they are beautiful but also because they are natural products of living animals, or a wood-carver who favours wood not only because of its material properties (colour, vein-patterns, carveability ...) but also because of the fact that it was once a living thing. Such CG-artists see the value of their work, in part, as being an encomium to life itself.

Evo-artists, in particular, have been inspired by A-Life research on evolution (Whitelaw 2004). And some of them, in turn, have been especially excited by the claims of a small coterie of maverick A-Life scientists who hope to create life in computers.

In other words, whereas some Evo-artists simply wish to generate artworks that prompt the audience to meditate on the wonders of life, others go much further. They accept the claims of (some) A-Life researchers that life could be realised–not just simulated–in a computer. Believing that virtual life–life in cyberspace–would be genuine life (and therefore intrinsically valuable), they see their own art as, potentially, a step towards achieving it. This is evident in many interviews in which they describe their aesthetic motivations (Whitelaw 2004), and in claims that they require us "to consider the power of technology to create life, rather than simply represent it" and that their audiences are confronted by "the artificial creation of life and living systems" (Tofts 2005: 103).

To put this point in another way, these artists base their work on the assumption that "strong A-Life" is possible (compare "strong AI": Searle 1980). If they are wrong in that assumption, then their artworks–however visually/aurally engaging, and even intellectually stimulating, they may be–don't have the significance which they claim for them.

One couldn't say that their art is therefore fraudulent, for their belief in the possibility of virtual life is sincere. One couldn't even say that it is inauthentic, if authenticity is primarily a matter of honesty of intent. But one could call it inauthentic in the sense of being unrealistic, or even radically flawed–being based on a philosophical premise that is mistaken. Whether that premise is indeed mistaken is therefore a matter of interest to aesthetics, at least as regards this small corner of the art world. 'Is Metabolism Necessary?' (Boden 2010: ch.12) takes up that question.

At first sight, that piece may seem out of place. For it doesn't mention art, nor even "creativity" as defined above. However, it concerns something significantly akin to psychological creativity: namely, biological self-organization. Here, some higher level of structural order emerges spontaneously out of an origin that's ordered to a lesser degree. (Examples include cell formation, organ development, homeostasis, flocking, evolution ... and metabolism.) Such self-organization isn't the same as what's normally meant by creativity, for it doesn't generate ideas/artefacts and nor is it goal-driven–still less, consciously monitored. But it does generate phenomena that are new, surprising, and valuable. It even does so by way of biological versions of the three types of artistic creativity: genetic combination (crossover), exploration by mutation, and transformation by mutation and/or interaction with the environment (Boden forthcoming).

In discussing this close cousin of creativity, I also addresss the specific query raised above: Is virtual life possible? My argument that it isn't depends on my analysis of the concept of metabolism. (Autopoiesis, a notion often endorsed by artists influenced by A-Life, is a similar concept–but it is also interestingly different: Boden 2000.) Metabolism is a form of self-organization which is reasonably regarded as a defining feature of life. And it is irreducibly physical. It can't be understood in purely functionalist terms, but only by reference to physical energy.

Proponents of strong A-Life will rush to point out that computers consume energy, too. But merely consuming energy–as computers certainly do–isn't enough to count as metabolism in the biologist's sense. Nor is individual energy-budgeting, wherein the amount of energy held by the system is finite and must be continually replenished if functioning is to continue (again, something that can be true of computers). When the term is used to describe/explain processes in living organisms, it also denotes the self-creation and maintenance of a physical unity, by means of interlocking biochemical cycles of some necessary complexity.

Manufactured computers, 'feeding' on energy from a battery or a plug in the wall, don't metabolise in this sense. Since they don't metabolise, they aren't alive. The idea that Evo-art could be a first step on the road to virtual life is therefore mistaken.

It doesn't follow that all the Evo-art inspired by this fond hope is worthless. One can appreciate a Fra Angelico Annunciation, or a sculptor's Madonna or Pieta, without being a Christian, or even a theist. Possibly, one's appreciation is less full, less nuanced, and less deeply felt than if one were a believer. Certainly, an atheist can't admire these creative artworks as genuine intimations of the divine, but must seek other values in them. That's relatively easy, however, for there are many ways in which the content of these religious paintings (not to mention the skill of their execution) can evoke a response in non-believers. That's because most non-Christians are broadly familar with the Christian story, and all are intimately familiar with the general human themes of maternity and mourning.

By contrast, the notion of virtual life is highly unfamiliar. To understand what this sub-class of Evo-artists are up to, we need to know what 'faith' is driving them, much as we need to know the Scriptures to understand what Fra Angelico was up to. That is the third reason why my paper on metabolism has a place in this collection. And perhaps it is the most important reason: after all, it is the gospel stories themselves, not our decisions about their truth or falsehood, which are key to what the Renaissance painters were doing.

Reference to Renaissance painting brings us back full circle to the

more traditional, familiar, forms of art. As we've seen, to understand how creative artworks are possible, we need to understand the psychological differences between the three types of creativity. (In this project, computer modeling–as opposed to computer art–is helpful) And to see them as creative, we also need to appreciate their value.

This requires us to situate them within a wide–and highly controversial–range of issues. We can probably all agree that if something were a powerful expression of a true story about divinity then it would be valuable, even though we may differ about whether any such story is credible. And probably most of us can be persuaded to agree that a certain painter had a fine (and perhaps historically original) grasp of perspective, or colour, or light-and-shadows, or facial expressions ... etc. But even figurative painters can prompt very different evaluations, especially if they go out of their way to outrage their audience. Think of the disturbing oeuvres of Francis Bacon and Jean Dubuffet, for instance: the one saw painting as "purely the challenge of delivering an instant, visceral shock", and the other recommended his canvasses as "art brut", or "ugly art" (Bell 2007: 416, 419).

There's even less agreement about whether abstract art, or conceptual art, or interactive art, or computer-generated art is valuable. And in all those categories, there are distinct subclasses– which need not be valued equally. The works of the Color Field painters, for instance, are extreme examples of abstract art: someone could well be enamoured of much abstractionism, but draw the line at Rothko. Similarly, someone might appreciate some of the categories defined by my taxonomy Boden (2010: ch.7) more than others.

Our values change, of course. In part, they change because people gradually come to see–and to appreciate–the links and likenesses between the novel and the familiar. Indeed, the novel eventually becomes the familiar: Bacon's screaming Pope is now as acceptable, at least to art lovers, as its 17th-century model (Diego Velazquez' portrait of Innocent X). Even computer art won't remain shocking for ever, especially if it continues to invade our public spaces. Some aesthetic values may be enduring, because they are part of our biological heritage. But many are culturally based. Some of these are rooted in aspects of culture that are normally distinguished from art: religious beliefs, scientific theories, political ascendancies, economic conditions ... even high-street fashion and haut couture. And some are subject to sudden changes triggered by intrinsically trivial events in the cultural milieu (what a celebrity chooses to wear to a paparazzo-infested party, for example).

Values in general, whether enduring or not, can't be justified–even though they may sometimes be explained–in scientific terms. In that sense, and in that sense alone, both creativity and art lie for ever beyond the reach of science.

With respect to how it's possible for novel ideas to arise, however, science does have something to say. The wondrous idiosyncracies of works of art cannot be fully detailed, although they can–up to a point–be intuitively appreciated by art-lovers willing to familiarize themselves with the style, and the artist, concerned. Human minds (and cultures) are so rich that psychologists will never be able to map out every narrow track, still less every footstep, that led to an individual artist's creative idea. (That isn't science's aim, anyway–Boden 2006: 7.iii.d.) But we can now descry the main paths of possibility: the three roads to wonder and surprise.

1 References

Ascott, R. (2003), Telematic Embrace: Visionary Theories of Art, Technology, and Consciousness (London: University of California Press).

Bell, J. (2007), Mirror of the World: A New History of Art (London: Thames and Hudson).

Bird, J., and Layzell, P. (2002), 'The Evolved Radio and its Implications for Modelling the Evolution of Novel Sensors', Proceedings of Congress on Evolutionary Computation, CEC-2002, 1836-1841.

Boden, M. A. (2000), 'Autopoiesis and Life', Cognitive Science Quarterly, 1: 1-29.

Boden, M. A. (2004), The Creative Mind: Myths and Mechanisms (London: Routledge). 2nd edn. expanded/revised. (1st edn., London: Weidenfeld and Nicolson, 1990).

Boden, M. A. (2010), Creativity and Art: Three Roads to Surprise, (Oxford: OUP).

Boden, M. A. (forthcoming), 'Creativity and Artificial Evolution', in J. Copeland (ed.), Creativity, Mathematics, and Computers (provisional title), MIT/Templeton Press.

Cohen, H. (1981), On the Modelling of Creative Behavior. RAND Paper P-6681 (Santa Monica, Calif.: RAND Corporation).

Cohen, H. (1995), 'The Further Exploits of AARON Painter', in S. Franchi and G. Guzeldere (eds.), Constructions of the Mind: Artificial Intelligence and the Humanities. Special edn. of Stanford Humanities Review, 4: 2: 141-160.

Cohen, H. (2002), 'A Million Millennial Medicis', in L. Candy and E. Edmonds (eds), Explorations in Art and Technology (London: Springer), 91-104.

Duchamp, M. (1957), 'The Creative Act', in M. Sanouillet and E. Peterson (eds.), The Essential Writings of Marcel Duchamp (London: Thames and Hudson, 1975), 138-140.

Edmonds, E. (2007), Shaping Form (Sydney: Creativity and Cognition Press).

Gibson, J. J. (1966), The Senses Considered as Perceptual Systems (Boston: Houghton-Mifflin).

Goodman, N. (1968), Languages of Art: An Approach to a Theory of Symbols. John Locke Lectures, Oxford 1962 (Indianapolis: Bobbs-Merrill).

Grau, O. (2004), Virtual art: From Illusion to Immersion (Cambridge, Mass.: MIT Press).

Harrod, T. (1999), The Crafts in Britain in the Twentieth Century (London: Yale University Press).

Kahn, D. (1996), 'What Now the Promise?', in M. Leggett and L. Michael (eds.), Burning the Interface: International Artists' CD-Rom, 27 March-14 July (Sydney: Museum of Contemporary Art), pp. 21-30.

Koning, H., and Eizenberg, J. (1981), 'The Language of the Prairie: Frank Lloyd Wright's Prairie Houses', Environment and Planning, B, 8: 295-323.

Langton, C. G. (1989/1996), 'Artificial Life', in C. G. Langton (ed.), Artificial Life. The Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (held September 1987) (Redwood City, CA: Addison-Wesley), 1-47. Revised version in M. A. Boden (ed.), The Philosophy of Artificial Life (Oxford: Oxford University Press, 1996), 39-94.

Leggett, M. (1996), 'CD-ROM–The 21st Century Bronze?', in M. Leggett and L. Michael (eds.), Burning the Interface: International Artists' CD-Rom, 27 March-14 July (Sydney: Museum of Contemporary Art), pp. 31-42.

Leggett, M. (1999), 'Electronic Space and Public Space: Museums, Galleries, and Digital Media', Continuum: Journal of Media and Cultural Studies, 13(2): 175-186.

McLaughlin, M. L., Osborne, K. K., and Ellison, N. N. (1997), 'Virtual Community in a Telepresence Environment', in S. Jones

(ed.), Virtual Culture (London: Sage), pp. 146-168.
Miller, G. F. (2000), The Mating Mind: How Sexual Choice Shaped the Evolution of Human Nature (London: William Heinemann).
Mitchell, T. M., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. (1994), 'Experience with a Learning Personal Assistant', in D. Riecken (ed.), Agents (special issue of Communications of the Association for Computing Machinery, 37:7), pp. 81-91.
Norman, D. A. (1994), 'How Might People Interact with Agents', in D. Riecken (ed.), Agents (special issue of Communications of the Association for Computing Machinery, 37:7), pp. 68-71. O'Hear, A. (1995), 'Art and Technology: An Old Tension', in R. Fellows (ed.), Philosophy and Technology (Cambridge: Cambridge University Press), 143-158.
Searle, J. R. (1980), 'Minds, Brains, and Programs', Behavioral and Brain Sciences, 3: 417-457. Includes peer-commentaries, and reply.
Sternberg, R. J., ed. (1988), The Nature of Creativity: Contemporary Psychological Perspectives (Cambridge: Cambridge University Press).
Sternberg, R. J., ed. (1999), Handbook of Creativity (Cambridge: Cambridge University Press).
Taylor. C. W. (1988), 'Various Approaches to and Definitions of Creativity', in R. J. Sternberg (ed.), The Nature of Creativity: Contemporary Psychological Perspectives (Cambridge: Cambridge University Press), pp. 99-124.
Todd, S. C., and Latham, W. (1992), Evolutionary Art and Computers (London: Academic Press).
Tofts, D. (2005), 'Artificial Nature', in D. Tofts, Interzone: Media Arts in Australia (Fisherman's Bend, Victoria: Craftsman House), pp. 80-103.
Turkle, S. (1995), Life on the Screen: Identity in the Age of the Internet (New York: Simon & Schuster).
Wallace, J., Dearden, A., and Fisher, T. (2007), 'The Significant Other: The Value of Jewellery in the Conception, Design and Experience of Body Focussed Digital Devices', AI & Society, 22(1): 53-62.
Whitelaw, M. (2004), Metacreation: Art and Artificial Life (London: MIT Press).

ANONYMITY AND EVOLUTIONARY ART[1]

Professor Margaret Boden[2]

Abstract. Human artists typically have a personal signature, by which their individual authorship can be recognized. Modernist artists tried to avoid such idiosyncracies, focussing on abstract structure instead–and welcomed computers, accordingly. But even those computer artists who have deliberately tried to lose their signature have not managed to do so. Perhaps evolutionary methods might help? Reasons are discussed both for believing and for doubting that evolutionary art could be wholly free from personal signatures.

1 The Quest for Anonymity in Art

Artworks are typically attributable, by art historians and connoisseurs, to a particular person. Indeed, Romantic views of art value the fact that the individual artist's 'personal signature' enables one to recognize the authorship of the work. This personal signature is not literally a signature. Rather, it is a set of subtle features of the work, of which the actual artist may not even be consciously aware [4].

Modernist artists, reacting against Romanticism, down-played the role of the individual person in art. They stressed formal (often minimalist) structure, not perceptible idiosyncracies. Typically, the art-object was no longer celebrated as a unique artefact, nor the human artist as an individual person.

This attitude was epitomized in an influential statement by the modernist Sol LeWitt: "the idea becomes a machine that makes the art, [where] all of the planning and decisions are made beforehand and the execution is a perfunctory affair" [8]. Once the plan has been chosen, LeWitt said, "The artist's will is secondary to the [artmaking] process he initiates from idea to completion" [9]. Indeed, he produced many 'remote' artworks, where he faxed instructions intended to be followed by anonymous people who, by following these instructions, would make the work using standard off-the-shelf materials such as 2-inch by 2-inch wooden strips. The Romantic ideal, of art as the expression of human individuality, had been abandoned.

2 The Impersonality of Computers

It's not surprising, given the sentiments quoted above, that when computers appeared on the scene many artists with modernist sympathies welcomed them specifically for their impersonal, non-human, nature. (Romantics, by contrast, recoiled from them in horror.)

At base, the reason for the existence of personal signatures lies in factors concerning the economy of information processing in human minds [4]. Computers are only indirectly affected by such factors. And, of course, they are immune to the motor habits of the programmer, and normally cannot develop any motor habits of their own. (As we'll see in Section 3, certain sorts of robot may be exceptions to that.) The psychological basis for the personal signature therefore disappears. Or, more accurately, it is pushed into the background. The aims and imagination (and programming skills) of the computer artist will always have idiosyncratic features, which may or may not be reflected in the computer output. But for those mid-century artists who already wished to obscure, or even escape from, their human individuality, it seemed that the very impersonality of computers might help.

Today, that is still a very natural assumption. So much so, that three leading computer artists have recently felt the need to reassure newcomers to the genre that if they want to set their individual stamp on the computer's behaviour, then they can. As they put it: "As a designer working with generative processes [i.e. computer art/design] one may still wish to leave a recognizable mark on a creation. This may be achieved statically using fixed components with a trademark style. A more interesting way to achieve this is to ensure either that the organization of the artefact bears the stamp of its designer, or that its behaviour falls within the gamut of work typically produced by the designer. Of course the designer may not be interested in producing a recognizable style, however the utilization of generative techniques does not preclude this option" [10]. We'll return to the issue of "the organization of the artefact [bearing] the stamp of its designer" in Section 4.

One of the first artists to welcome computers for their very impersonality was the young Paul Brown. Visiting the "Cybernetic Serendipity" exhibition in London in 1969, he was inspired by the hope that this new methodology would enable him to do something he was already trying to do: namely, to lose his personal signature. Now, some forty years later, he is an internationally known computer artist. But his artworks are still recognizable, to those familiar with his oeuvre, as Brown's. Even his very earliest pieces [5] have an evident visual kinship with his recent/current work. In other words, it turned out that losing his individual artistic stamp, as his modernist sympathies inclined him to do, was easier said than done.

One reason is that Brown himself, after forty years as a professional artist, still cannot say just what what his personal signature is (i.e. just what needs to be avoided). In general, recognising a particular artist's signature and describing it explicitly are two very different things [4]). Whatever it is in Brown's case, it certainly is not a matter of a specific mark (such as a particular form of ear-lobe) recurring in his work. It is more a matter of an overall stylistic 'feel' that he cannot pin down in words.

He had hoped as a young man that the clarity with which artmaking has to be defined if computers are involved might help him both to identify his signature and (by changing the generative rules as a result) to lose it. Reasonable enough hopes, one might think. But no: his computer-generated work still betrays its human author's individual hand. And this, even though he has deliberately aimed for

[1] This paper is based on part of a longer paper on 'Personal Signatures in Art' [4]

[2] School of Informatics, University of Sussex, England; email: m.a.boden@sussex.ac.uk

aesthetic anonymity.

It appears, then, that if someone wishes to use computers so as to lose their personal signature, deliberate self-effacement in the hands-on practice of one's art is not the way to do it. Can some other way of achieving self-effacement be found?

3 Could Anonymity be Evolved?

Recently, Brown has been using computers in a new way in trying to achieve his long-standing artistic goal. An interdisciplinary team, with Brown as a leading member, has tried to evolve line-drawing robots whose products are of some aesthetic interest (no more than that!), but which do not carry the telltale traces of a work by Brown himself.

In evolutionary art in general, the selection at each generation can be done interactively, by a human being making the comparisons, or automatically by the program itself. In this particular case, interactive selection is best avoided, because it is likely to carry the personal mark of the human artist. Even automatic selection, however, requires that a 'fitness function' be defined, which the program can use to make its selections. (The fitness function itself may evolve, again either interactively or automatically.) As we'll see in Section 4, this fact is the Achilles' heel of Brown's current research.

The first obvious question to ask about this project–which is named Drawbots– is "Why evolve line-drawing gizmos, as opposed to simply designing (programming/building) them?" The second is "Why use robots, as opposed to computer graphics (i.e. programs for drawing images on paper or virtual images in cyberspace)?"

The answer to the first question is that if the line-drawing computer system has been evolved then, thanks to the many random mutations that will have taken place, it has not been prespecified in detail by the artist-programmer. Accordingly, there may (sic) be a chance of avoiding that individual's personal signature. Whether that "may" can, in practice or even in principle, be replaced by a "will" is the key point at issue here.

As for the second question, the answer is that a robot, being a material object functioning in the physical world, can be affected not only by its program and/or internal design but also by unexpected–and perhaps serendipitous–events in the physical environment. Again, this offers a means by which the programmer's personal signature may be bypassed, or anyway diluted. (An early example of this sort of thing occurred in the 1970s, when the moving 'legs' of a kinetic sculpture–alias a robot–happened to scratch the wooden floor of London's Royal Academy. Although the RA was doubtless incensed, the sculptor, Darrell Viner, was intrigued. He was so "fascinated by the structure of the repetitive scratches and their relationship to cross-hatching" that he went on to make artworks produced by comparable, though simulated, means–[6].)

The "serendipity" in the physical events involved can even include cases where a radically new feature appears in the robot's behaviour. In a previous experiment done by a member of the Drawbots team, a population of robots evolved a new sensory capacity–not merely an improved sensory capacity–as a result of contingent, and previously unremarked, facts about the physical environment [1]. That suggests the possibility that a fundamentally transformative change in the Drawbots' drawing-style might occur. If so, then presumably the new style would not bear Brown's individual mark, even if the previous style had done so.

The Drawbots themselves are small wheeled vehicles carrying a retractable pen. And the task in the team's minds is line-drawing. By that is meant not drawing pictures that represent real things (as both stick-men and Renaissance cartoons do), nor even drawing geometrical designs, but simply drawing lines ... which can curve, cross, stop, and approach each other in myriad ways–and which may sometimes change in thickness too. Brown's hope is that robots can be evolved which will draw aesthetically acceptable lines that do not exhibit his personal signature. In other words, the fitness function/s to be followed by the robot should guarantee aesthetic acceptability but should not be so 'rich' as to express his personal style.

In principle, that would not preclude there being a telltale identifier, or quasi-signature (one can hardly say a "personal" signature), produced by an evolved robot itself. This would be a pattern that distinguishes its drawings from those of its siblings and close cousins. The evolution of such patterns is in principle possible because new performance details will follow from random mutations, and these details can be perpetuated provided that they do not compromise fitness.

Such details could include drawn patterns or line-features discriminated by the gizmo's visual sensors. Indeed, a robot might even develop particular motor habits, driven by motor circuits conserved in its 'brain' (see Section 2). Suppose that a sudden movement, caused by a recently mutated motor circuit, led to a mark that was then selected (along with the rest of the drawing) by Brown. This might lead the motor circuit to endure, forming the basis of a future motor habit. That habit could be involved either in many different stylistic choices, or only in one (think of an overall stylistic 'feel' and of telltale ear-lobes, respectively). In short, the general style that is selected via the fitness function could allow for idiosyncratic expression (alias signatures) by different robots within the same generation or lineage.

If the fitness function were to include measures of computational economy, the different robots might even develop quasi-signatures for much the same (psychological) reasons that human beings do. However, it is hardly likely that such patterns would arise as a matter of course, as they do in the work of human artists. For the root of the personal signature, as remarked above (see also [4]), is the need for economy in information processing within a highly complex system–a criterion that does not apply in robots as simple as those being considered here.

Whether it is actually possible for the drawbots to lose the stamp of Brown's individual artistry depends on a number of things. One is the extent to which Brown, or anyone else, can say just what his personal signature consists in. If he knew that, he would be in a much better position to try to avoid it. However, as explained in Section 2, he does not.

Possibly, the Drawbots research may help him towards a better–if still incomplete–understanding of this. For in examining the various drawings made by the drawbots, he will have to ask himself two questions: Is it aesthetically acceptable? and Is it evidently a 'Brown'? In answering that second question over and over again, as the drawing style mutates across the generations, and in posing it to colleagues with an appropriately practised critical eye, he may achieve a more explicit understanding of just what his own style is. (Then again, he may not.) But that could happen without his ever answering No to the second question. In that case, he still would not have 'lost' his signature, despite understanding it more deeply. Whether the increased understanding would enable him to dilute it, if not to shed it, in his (non-evolutionary) future work is an interesting question.

Another factor that will affect the likelihood of success in the project is the extent to which aesthetic acceptability can rest on relatively primitive visual features. "Primitive", here, means both simple and naturally salient. For example, shininess (of satin, silver, pol-

ished ivory, lurex, chromium...) is relatively simple to discriminate, and naturally salient too. That's so for good evolutionary reasons, involving the fitness-enhancing nature of reflective expanses of water [3]. In other words, it's no accident that shininess is aesthetically appealing to a very wide range of individuals and cultures. Are there any features of line-drawings such as those the drawbots could produce which are naturally attractive (and easily discriminable) in a comparable way?

For example, if the drawbots were able to change pens, might they evolve a preference for the shiny lines left by a silver-ink pen? They could do so, if their visual apparatus could discriminate shininess. To be sure, the robotics team would have to build reflectance into the fitness function: no robot 'naturally' prefers it. But reflectance is such an easily discriminable property, and so near-universally liked by human beings, that the team could not be accused of cheating were they to do that. (Some cultural groups positively avoid shininess, regarding it as vulgar; but that is irrelevant here, since this discriminatory attitude has developed precisely because the liking for shininess is so very common.) Nor would putting silveriness into the fitness function result in drawings that display Brown's personal signature, for that (whatever it is) is not a matter of shininess.

It's easy to see that Brown's authorial mark does not involve shininess. What it does involve is less clear. Suppose it were to turn out that all the perceptible features favoured (via the fitness function) by 'aesthetically competent' drawbots were relatively high-level and/or complex, with no 'natural' attractiveness for human beings in general. In that case, their drawings would probably be more specific to Brown's personal style. His project would have failed. However, "success" and "failure" here admit of several levels. In the language used above, Brown's signature may become more or less diluted, even if it cannot be entirely lost.

Among the naturally discriminable features that are already being considered by the Drawbots team are holes, line-crossings, and fractals (of varying complexity or depth). But why should one expect any of these things to be 'naturally' attractive?

Well, consider fractals, for instance. These are ubiquitious in Nature, both in living things and in environmental features such as rocks and coastlines. According to the 'biophilia' hypothesis [**?**], Homo sapiens has evolved to respond favourably not only to conspecifics and other aspects of our original ecological niche (the African Savannah) but also to living things and natural environments in general. If that's so, then fractals might well have some natural attraction for us.

That's merely an argument for plausibility. But there is also some evidence that fractals of a certain kind are spontaneously favoured in art as in nature–and even, as William Congreve said of music, that they can soothe the savage breast. Richard Taylor claimed, in the late-1990s, that Jackson Pollock's canvasses, far from being random splashes of paint, have specific fractal properties to which most viewers respond in a positive way, and by which his paintings can be distinguished from fakes [13] [14]. Specifically, people prefer those Pollock paintings which have a fractal dimension of 1.5 (his later paintings reach 1.8+). By comparison, people asked to choose between natural images (or between simulated coastlines) prefer a fractal dimension of 1.3. Taylor's claim aroused huge interest [11], and was later followed by experiments showing that viewing Pollock's images can actually reduce stress [15].

Taylor's early remarks about how to discriminate genuine Pollocks from fakes, have recently been challenged [7]. One aspect of that challenge is especially intriguing here: Katherine Jones-Smith reported that a careless doodle done by her showed the same fractal properties as those found in Pollock's work. She didn't ask whether the doodle had any aesthetic value. To the contrary, she implied that, being a thoughtless scribble, it did not. But if she had asked people whether they "liked" it, or whether they preferred it to some other mark (maybe one produced accidentally), she might have found that people ascribed some–albeit small–degree of aesthetic merit to it. If that were so, it suggests that a suitably fractal-favouring drawbot might make aesthetically acceptable ('natural') drawings that don't show anyone's individual mark: not hers, not Pollock's, and not Brown's either.

4 The Likelihood of Success–and What it Would Mean

The discussion in Section 3 suggested that it is in principle possible for Brown's personal signature to be lost by evolved robots (even though it is also possible for those robots to develop individual 'signatures' of their own). But what of the likelihood of this happening in practice? Are there any specific reasons (beyond those mentioned in Section 3) to suspect that the Drawbots project will succeed, or fail? And if it succeeds, would it follow that the creativity exhibited in the drawings of the newly-evolved drawbots must be attributed to the drawbots themselves, rather than to Brown? 'No signature, no creative authorship', perhaps?

As remarked above, the Achilles' heel of the project lies in the fitness function. This is true in two related senses, one philosophical and one psychological.

First, if it is Brown who is continually deciding on the fitness function as the research proceeds then perhaps it is his aesthetic judgment, and also his artistic creativity, which is really responsible for the final drawings? (For shorthand purposes, let's ignore the creative role of the other human beings on the team.) Many philosophers would say that there is no "perhaps" about it, that of course Brown's creativity lies behind whatever aesthetic interest the drawbots' drawings happen to have. For they believe that it is in principle absurd to ascribe creativity, or aesthetic judgment, to any computer system–no matter how superficially impressive its performance may be.

Their belief typically rests on assumptions about one or more of four highly controversial issues, including intentionality and consciousness [2]. Accordingly, it can be challenged–though not definitively refuted. However, even if one were happy to reject their claim as a general philosophical position, that would not settle the question at issue here. For in the specific case of the Drawbots research, the largely human source of the fitness function is a distinct embarrassment for anyone wanting to grant all the creative credit to the computer.

This embarrassment would persist whether or not the project succeeded in its own terms–that is, irrespective of whether Brown's signature had been lost. For if the final fitness function were to exploit only what in Section 3 were called "primitive" aesthetic properties, so that Brown as an individual artist had become invisible in the final-stage drawings, it would still be true that the aesthetic decisions involved in developing the fitness function were such as are naturally made by human beings. Brown's hand (judgment) would still be there–but functioning as the hand of a generic human being, not of a particular individual. (In other words, the fitness function would describe the general style, without imposing any detailed 'authorial' implementation.)

That argument would apply even if the robots' drawing style had shown a truly fundamental change: a new style (presumably, a 'non-Brown' style), as opposed to an improved style. We saw in Section

3 that the physical 'embodiment' of the drawbots makes it in principle possible for such serendipitous change to occur. By definition, the stylistic change would have been caused by some unconsidered and/or contingent feature of the robots' physical environment. So Brown couldn't be credited with initiating it. But he could, perhaps, be credited with 'causing' it, since the incipient change will be maintained (and perhaps developed) only if it is approved/selected by his personal decision or by the fitness function already evolved under his direction. In such a case, Brown might be regarded as the creative spirit behind the final drawings even though he never foresaw them, and even though they are free of his personal mark.

What of the psychological question? Are there any psychological reasons to expect that Brown will not be able to decide on a fitness function that entirely avoids his personal signature?

One psychological consideration that is important in aesthetic judgments [2] is relevance–considered in terms of computational closeness and/or efficiency [12]. This issue is less obviously crucial here than it would be if Brown were trying to evolve robots capable of realistic representational drawings. If the drawbots were intended to draw human faces, for instance, they had better include depictions of eyes, mouth, and even (the relatively less relevant) ear-lobes. And they had better not add horns, or wings. But if a tinge of surrealism were to be favoured (by Brown), then a horn-like protuberance appearing in generation 1,000 might be selected and 'shaped' so that recognizable devilish/goatlike horns were visible at generation 9,000. The same might occur if Brown felt that familiar myths about the Devil were relevant to the 'topic' of the drawings. In either case, Brown's own judgments about relevance would be reflected in the robots' behaviour, and–to the extent that these are idiosyncratic–so would his personal mark.

In fact, Brown has always been an abstract artist, so is not aiming to evolve 'representational' robots. Even so, issues of relevance–or rather, issues of what he deems to be relevant–may arise.

Aesthetic acceptability depends in part on intelligibility. To be sure, intelligibility may be more or less easy to achieve in differing artistic styles. But utter chaos will satisfy nobody. In other words, one factor underlying judgments of aesthetic acceptability is the computational effort that is involved in comprehension. A 'messy' line-drawing (or doodle), for instance, may be unacceptable largely because its components do not appear to be mutually relevant. That is, they do not appear to be 'coherent', or to 'make sense'. (Perhaps there are no closed curves, suggesting bounded physical objects? And/or perhaps there are no T-junctions where one line stops as it meets another, suggesting occlusion of a line/edge by some other physical thing?) These judgments are not usually conscious–and it may not be possible to make them fully conscious. It follows that it may not be possible for Brown to avoid them deliberately.

A closely related issue is the extent to which Brown can banish his own preferred schemas from the fitness function. (Compare: evolving robots to draw faces without eyes.) If he cannot, because these schemas are so deeply entrenched in his mind and experience, they will inevitably be reflected in the fitness function and therefore in the final drawings.

At that point, we come full circle to the issue discussed in Section 3 in terms of "simplicity" and "naturalness". The more that the features favoured in the fitness function are complex, culture-based, and idiosyncratic to Brown, the less will the final-generation drawbots be free of his personal stamp.

If the Brown signature is preserved, despite all his efforts, that will be because he has found it necessary to build relatively 'rich' criteria into the fitness function. As we've seen, it is still an open question as to how rich the final criteria of aesthetic fitness will need to be. If they are all relatively simple, then Brown's creative inspiration may seem less important. At most, the fact that he is a human being will be relevant, not the fact that he is Paul Brown. (Any idiosyncratic 'signature' visible in the drawings might be attributable to the evolutionary vicissitudes of the robots themselves, as explained above.)

What if, contrary to all his hopes, Brown's personal signature remains still visible to experts (dare we say connoisseurs?) looking at the robots' drawings? In such a case, and even if one were willing in principle to grant creativity to some computer systems, it would seem bizarre to attribute creativity to the drawbot. For the concept of the personal signature arose specifically in order to attribute a given work of art to one creative source–normally, one human individual–rather than another [4]. The signature, in short, points to the person. This was recognized by the computer artists (quoted in Section 2) who spoke of "the organization of the artefact [bearing] the stamp of its designer". Whether that telltale organization were deliberately designed, as they were assuming, or gradually evolved, as in the Drawbots project ('failure' here being supposed), it would point to one person: Brown.

REFERENCES

[1] Bird, J., and Layzell, P. (2002), 'The Evolved Radio and its Implications for Modelling the Evolution of Novel Sensors', Proceedings of Congress on Evolutionary Computation, CEC-2002, 1836-1841.

[2] Boden, M. A. (2004). The Creative Mind: Myths and Mechanisms, 2nd edn., expanded/revised (London: Routledge).

[3] Boden, M. A. (2006). Mind as Machine: A History of Cognitive Science (Oxford: Oxford University Press).

[4] Boden, M. A. (2010) 'Personal Signatures in Art', in M. A. Boden, Creativity and Art: Three Roads to Surprise (Oxford: Oxford University Press), 92-124.

[5] Brown, P. (1977), 'The CBI North West Export Award'. First published in Page Sixty Two–Special Terminate CACHe Issue: Bulletin of the Computer Arts Society, Northern Hemisphere, Autumn 2005: 12-13.

[6] Brown, P. (2008), 'From Systems Art to Artificial Life: Early Generative Art at the Slade School of Fine Art', in C. Gere, P. Brown, N. Lambert, and C. Mason (eds.), White Heat and Cold Logic: British Computer Arts 1960-1980, An Historical and Critical Analysis (Cambridge, Mass.: MIT Press): 275-289.

[7] Jones-Smith, K., and Mathur, H. (2006), 'Revisiting Pollock's Drip Paintings', Nature, 444 (Nov. 30) :E9-E10 (published online 29 Nov.).

[8] LeWitt, S. (1967), 'Paragraphs on Conceptual Art', Artforum, 5(10): 79-83. Reprinted in K. Stiles and P. Selz (eds.), Theories and Documents of Contemporary Art: A Sourcebook of Artists' Writings (London: University of California Press), 822-826.

[9] LeWitt, S. (1969), 'Sentences on Conceptual Art', Art-Language, 1: 11-13. Reprinted in K. Stiles and P. Selz (eds.), Theories and Documents of Contemporary Art: A Sourcebook of Artists' Writings (London: University of California Press), 826-827.

[10] McCormack, J., Dorin, A., and Innocent, T. (2004). 'Generative Design: A Paradigm for Design Research', in J. Redmond, D. Durling, and A. de Bono (eds.), Futureground, vol. 1 (Melbourne: Design Research Society).

[11] Spehar, B., Clifford, C., Newell, B., and Taylor, R. (2003), 'Universal Aesthetic of Fractals', Computers and Graphics, 27: 813-820.

[12] Sperber, D., and Wilson, D. (1986). Relevance: Communication and Cognition (Oxford: Blackwell).

[13] Taylor, R., Micolich, A. P., and Jonas, D. (1999a), 'Fractal Analysis of Pollock's Dripped Paintings', Nature, 399: 422 (one page only).

[14] Taylor, R., Micolich, A. P., and Jonas, D. (1999b), 'Fractal Expressionism', Physics World, October.

[15] Taylor, R. P., Spehar, B., Wise, J. A., Clifford, C. W. G., Newell, B. R., Hagerhall, C. M., Purcell, T., and Martin, T. P. (2005), 'Perceptual and Physiological Responses to the Visual Complexity of Pollock's Dripped Fractal Patterns', Journal of Non-Linear Dynamics, Psychology and Life Sciences, 9: 89-114.

[16] Wilson, E. O. (1984), Biophilia (London: Harvard University Press).

On impact and evaluation in Computational Creativity: A discussion of the Turing Test and an alternative proposal

Alison Pease[1] and **Simon Colton**[2]

Abstract. Computational Creativity is the AI subfield in which we study how to build computational models of creative thought in science and the arts. From an engineering perspective, it is desirable to have concrete measures for assessing the progress made from one version of a program to another, or for comparing and contrasting different software systems for the same creative task. We describe the Turing Test and versions of it which have been used in order to measure progress in Computational Creativity. We show that the versions proposed thus far lack the important aspect of interaction, without which much of the power of the Turing Test is lost. We argue that the Turing Test is largely inappropriate for the purposes of evaluation in Computational Creativity, since it attempts to homogenise creativity into a single (human) style, does not take into account the importance of background and contextual information for a creative act, encourages superficial, uninteresting advances in front-ends, and rewards creativity which adheres to a certain style over that which creates something which is genuinely novel. We further argue that although there may be some place for Turing-style tests for Computational Creativity at some point in the future, it is currently untenable to apply any defensible version of the Turing Test.

As an alternative to Turing-style tests, we introduce two descriptive models for evaluating creative software, the FACE model which describes creative acts performed by software in terms of tuples of generative acts, and the IDEA model which describes how such creative acts can have an impact upon an ideal audience, given ideal information about background knowledge and the software development process. While these models require further study and elaboration, we believe that they can be usefully applied to current systems as well as guiding further development of creative systems.

1 The Turing Test and Computational Creativity

The Turing Test (TT), in which a computer and human are interrogated, with the computer considered intelligent if the human interrogator is unable to distinguish between them, is principally a philosophical construct proposed by Alan Turing as a way of determining whether AI has achieved its goal of simulating intelligence [1]. The TT has provoked much discussion, both historical and contemporary, however this has principally been within the philosophy of AI: most AI researchers see it as a distraction from their goals, encouraging a mere trickery of intelligence and ever more sophisticated natural language front ends, as opposed to focussing on real problems. Despite the appeal of the (as yet unawarded) Loebner Prize, most subfields of AI have developed and follow their own evaluation criteria and methodologies, which have little to do with the TT.

[1] School of Informatics, University of Edinburgh, UK
[2] Department of Computing, Imperial College, London, UK

Computational Creativity (CC) is a subfield of AI, in which researchers aim to model creative thought by building programs which can produce ideas and artefacts which are novel, surprising and valuable, either autonomously or in conjunction with humans. There are three main motivations for the study of Computational Creativity:

- to provide a computational perspective on human creativity, in order to help us to understand it (cognitive science);
- to enable machines to be creative, in order to enhance our lives in some way (engineering); and
- to produce tools which enhance human creativity (aids for creative individuals).

Creativity can be subdivided into everyday problem-solving, and the sort of creativity reserved for the truly great, in which a problem is solved or an object created that has a major impact on other people. These are respectively known as "little-c" (mundane) and "big-C" (eminent) creativity [2]. Boden [3] draws a similar distinction in her view of creativity as search within a conceptual space, where "exploratory creativity" searches within the space, and "transformational creativity" involves expanding the space by breaking one or more of the defining characteristics and creating a new conceptual space. Boden sees transformational creativity as more surprising, since, according to the defining rules of the conceptual space, ideas within this space could not have been found before.

There are two notions of evaluation in CC: (*i*) judgements which determine whether an idea or artefact is valuable or not (an essential criterion for creativity) – these judgements may be made internally by whoever produced the idea, or externally, by someone else and (*ii*) judgements to determine whether a system is acting creatively or not. In the following discussion, by evaluation, we mean the latter judgement. Finding measures of evaluation of CC is an active area of research, both influenced by, and influencing, practical and theoretical aspects of CC. It is a particularly important area, since such measures suggest ways of defining progress in the field,[3] as well as strongly guiding program design. While tests of creativity in humans are important for our understanding of creativity, they do not usually *cause* humans to be creative (creativity training programs, which train people to do well at such tests, notwithstanding). Ways in which CC is evaluated, on the other hand, will have a deep influence on future development of potentially creative programs. Clearly, different modes of evaluation will be appropriate for the different motivations listed above.

[3] The necessity for good measures of evaluation in CC is somewhat paralleled in the psychology of creativity: "Creativity is becoming a popular topic in educational, economic and political circles throughout the world – whether this popularity is just a passing fad or a lasting change in interest in creativity and innovation will probably depend, in large part, on whether creativity assessment keeps pace with the rest of the field." [4, p. 64]

The Turing Test is of particular interest to CC for two reasons. Firstly, unlike the general situation in AI, the TT, or variations of it, *are* currently being used to evaluate candidate programs in CC. Thus, the TT is having a major influence on the development of CC. This influence is usually neither noted nor questioned. Secondly, there are huge philosophical problems with using a test based on imitation to evaluate competence in an area of thought which is based on originality. While there are varying definitions of creativity, the majority consider some interpretation of novelty and utility to be essential criteria. For instance, one of the commonalities found by Rothenberg in a collection of international perspectives on creativity is that "creativity involves thinking that is aimed at producing ideas or products that are relatively novel" [5, p.2], and in CC the combination of novelty and usefulness is accepted as key (for instance, see [6] or [3]). In [4], Plucker and Makel list "similar, overlapping and possibly synonymous terms for creativity: imagination, ingenuity, innovation, inspiration, inventiveness, muse, novelty, originality, serendipity, talent and unique". The term 'imitation' is simply antipodal to many of these terms.

In the following sections, we firstly describe and discuss some attempts to evaluate Computational Creativity using the Turing Test or versions of it (§2), concluding that these attempts all omit the important aspect of interaction, and suggest the sort of direction that a TT for a creative computer art system might follow. We then present a series of arguments that the TT is inappropriate for measuring creativity in computers (or humans) in §3, and suggest that although there may be some place for Turing-style tests for Computational Creativity at some point in the future, it is currently untenable and impractical. As an alternative to Turing-style tests, in §4, we introduce two descriptive models for evaluating creative software, the FACE model which describes creative acts performed by software in terms of tuples of generative acts, and the IDEA model which describes how such creative acts can have an impact upon an ideal audience, given ideal information about background knowledge and the software development process. We conclude our discussion in §5.

2 Attempts to evaluate Computational Creativity using the Turing Test or versions of it

There have been several attempts to evaluate Computational Creativity using the Turing Test or versions of it. While these are useful in terms of advancing our understanding of CC, they do not go far enough. In this section we discuss two such advances (§2.1 and §2.2), and two further suggestions on using human creative behaviour as a guide for evaluating Computational Creativity (§2.3). We highlight the importance of interaction in §2.4.

2.1 Discrimination tests

Pearce and Wiggins [7] assert for the need for objective, falsifiable measures of evaluation in cognitive musicology. They propose the 'discrimination test', which is analogous to the TT, in which subjects are played segments of both machine and human-generated music and asked to distinguish between them. This might be in a particular style, such as Bach's music, or might be more general. They also present one of the most considered analyses of whether Turing-style tests such as the framework they propose might be appropriate for evaluating Computational Creativity [7, §7]. While they do not directly refer to Boden's exploratory creativity [3], instead referring to Boden's distinction between psychological (P-creativity, concerning ideas which are novel with resepct to a particular mind) and historical creativity (H-creativity, concerning ideas which are novel with respect to the whole of human history[4]), they do argue that much creative work is carried out within a particular style. They cite Garnham's response [8] to Boden's ideas, in which he emphasizes the importance of exploratory as compared to transformational creativity: "the origins of the symphony are lost in history and its major triumphs are the work of composers who did not invent the basic symphonic form." (Bundy argues along similar lines in [9]). Thus, Pearce and Wiggins suggest that their test rewards an appropriate level of novelty, since they found in their experiments that subjects could identify machine-generated compositions which were either too strange (too far away from well-explored areas) or too predictable (conforming too much to the well-explored areas). In anticipation of the objection that the process by which something has been created is important to judgements of creativity and thus a behaviour-based test is insufficient, Pearce and Wiggins refer to Hofstadter's argument that interaction with a system at an arbitrarily deep level can shed great insight into the processes it uses to generate its output [10]. While seeing the evaluation of the creativity of machine composers as an extension of their framework rather than a fully developed aspect, Pearce and Wiggins suggest that this type of evaluation is relevant for musical creativity within a specific style (that is, exploratory creativity). They also suggest that it may generalise to other creative domains such as art or story generation.

2.2 A Turing Test for artistic creativity

In [11], Boden discusses the Turing Test and artistic creativity. She provides an interpretation of the Turing Test which is specifically designed for computer art systems:

> "I will take it that for an 'artistic' program to pass the TT would be for it to produce artwork which was:
>
> 1. indistinguishable from one produced by a human being; and/or
> 2. was seen as having as much aesthetic value as one produced by a human being." [11, p. 409]

Boden describes several systems which produce art or music, which she considers to be either non-interactive or unpredictably interactive (such as a piece of art which responds to audience members or participants in ways they do not understand). She discusses comparisons with both mediocre human art, in this case pastiches of given styles (perhaps comparable to work by an art student exploring a given style), as well as examples which match world class human art, of interest as an artwork in itself (comparable to work done by a practising artist). She argues that the following systems all pass (her version of) the TT:

- Richard Brown's Starfish[5] – a computer generated starfish which appeared to be trapped inside a glass table, which interacted with audience members by responding to their movements and sounds. This featured in the Millennium Dome;
- AARON, a software program written by the artist Harold Cohen that creates original artistic images which are exhibited in art galleries around the world (described by McCorduck in [12]);

[4] Note that these two types of creativity are *not* analogous to the little-c/big-C distinction, since Boden talks of P-creativity being a subset of H-creativity [3, pp. 32-33].

[5] For further details, see http://www.mimetics.com/vur/mindzone.html.

- Computer art by Boden and Edmunds [13] which was exhibited in honour of world famous artists. This was composed of vertical stripes of colour which were continually changing, where the colours were partially determined by audience participation in an unpredictable manner, with constraints on certain colour combinations;
- Cope's system Emmy (Experiments in Musical Intelligence) [14, 15] which generated music in particular styles, such as that of Mozart, which was indistinguishable from human-composed Mozart pastiches, and was performed in concert halls.

Boden argues that these systems satisfy the second criterion: their aesthetic value has been proven by the degree of interest in their work (presumably, from members of the public, artists and musicians, rather than solely AI researchers). These all model exploratory creativity, where a style is explored. For examples of transformational creativity, Boden refers to systems by Todd and Latham [16] and Sims [17]. However, since these are much more interactive, she does not (yet) consider them to be candidates for the TT. Regarding the first criterion, Boden mentions anecdotally some occasions on which critics have admired a piece of art and then retracted the view when the art was discovered to be machine-generated. This suggests that, in some cases at least, systems have satisfied her first criterion.

We have a number of objections to Boden's usage of the term 'Turing Test' for the above evaluation criteria. Firstly, Boden reinterprets the TT and presents her own version, which differs substantially from Turing's proposal in at least two ways: (*i*) there is no interaction with the system, and (*ii*) by using a disjunctive rather than conjunctive relationship between the two criteria, she allows that all systems which produce output with "as much aesthetic value as produced by a human being" passes the TT. Systems which produce output of sufficient interest to be exhibited are therefore evaluated to have passed the TT. In particular, Boden argues that "If being exhibited alongside Rothko, in a 'diamond jubilee' celebration of these famous artists, does not count as passing the Turing Test, then I do not know what would." [11, p. 410]. This lack of emphasis either on interaction, or on discrimination between human and computer-produced artefacts seems to be rather missing the point of the TT. In particular, Boden seems to have expanded the term 'Turing Test' from being just one way of testing that intelligence might have been exhibited, to being a way of testing whether software has done something (or produced something) culturally significant. Our second objection is that the evidence for the second criterion, which is closest to the TT, is never explicitly addressed, and only implicitly in an anecdotal fashion. In fact, we see Boden's argument as supporting the idea that computer-created art may very well be distinguishable from human-created art, yet still have great aesthetic and cultural value, (see §3.1 for further argument on this point); that is, that the TT is inappropriate in this context. Clearly, art generation software could fail the originally conceived Turing Test, yet pass Boden's version of it.

Despite our objections to using a misleading naming based on the Turing Test, Boden's criteria can certainly be valuable for evaluating creative systems. However, we would caution that software which exhibits very little behaviour that would normally be considered (in computing or human circles) as creative can be evaluated positively using Boden's criteria. In particular, Brown's Starfish project, while a beautiful demonstration of neural net technology, and an exciting piece of human-computer interaction, certainly cannot be described as an example of software acting creatively. It is an example of kinetic art which was conceived, designed, produced, programmed and evaluated by humans (Richard Brown, Jonathan Mackenzie and Gavin Baily). While the software is generative, and to some extent unpredictable, it exhibits no higher level cognitive functioning such as the generation and/or application of aesthetic considerations or any behaviour which might be deemed remotely imaginative.

While Boden's criteria for the assessment of art-generating software are valid, we argue that calling it a Turing Test confuses the assessment of intelligence and creativity with the assessment of cultural impact, and that software which wouldn't ordinarily be considered creative can pass the test, hence the criteria have limited value for the assessment of software developed in a Computational Creativity context.

2.3 Using human creative behaviour as a guide for evaluating Computational Creativity

Wiggins proposes the following working definition of Computational Creativity:

> "The performance of tasks [by a computer] which, if performed by a human, would be deemed creative." [18, p. 451]

This type of behavioural test, in which output from a computer is compared to that from humans, has much in common with the Turing Test. In addition, Colton [19] has argued that creativity in software is often marked negatively, i.e., while there may be no obvious set of behaviours that software must exhibit in order to be regarded as creative, there are some common ways in which software can be immediately disregarded as being uncreative. In particular, Colton proposes that the criticisms levelled at software can largely be grouped into three categories: the software doesn't exhibit enough (or the right kind of) *skill*; the software has no *appreciation* of what it is doing, what it produces or what other people/machines do; the software exhibits no *imagination* in its processing. Hence, he suggests that Computational Creativity researchers should aim to build software which exhibits behaviour that might be deemed as skilful, appreciative and imaginative.

2.4 The importance of interaction

All of the versions of the TT which we have discussed here have one obvious similarity; there is no interaction with the program. This leaves out what is, arguably, the main strength of the TT. We have already introduced Hofstadter's argument that interaction with a system at an arbitrarily deep level can shed great insight into the processes it uses to generate its output (see §2.1). Hofstadter goes on to say:

> "In the spirit of much of the best science of our century, the Turing Test blurs the supposedly sharp line between probing of behavior and probing of mechanisms, as well as the supposedly sharp line between "direct" and "indirect" observation, and thus reminds us of the artificiality of such distinctions. Any computer model of mind that passes a truly deep Turing Test - one that probes for the fundamental mechanisms of thought will agree with "brain structures" all the way down to the level where the essence of thinking really takes place." [10, pp. 490-491]

The key word here is 'probe': interaction must form a necessary part of any test based on the TT, for it to hold any relevance to CC. For example, a Turing Test for artistic creativity which consisted of requests to draw something specific might be informative. A human

interrogator might attempt to distinguish between a computer art system and a human artist by making requests, such as:

- Draw something in the style of Picasso.
- Can you break/change/enhance the rules of the Impressionist style and draw something within the new style you've just created?
- Draw something which reflects your feelings towards the war in Afghanistan.
- Draw something warm.
- Show me your best painting and explain to me why you think it's good.
- Who or what has influenced your work?
- How does your work fit into the wider artistic community?

In order to avoid pitfalls of the current TT and focus on the important issues, the test could be conducted without the need for natural language,[6] timing issues, and so on.

3 Arguments that the Turing Test is inappropriate for measuring creativity in computers (or humans)

In this section, we argue that the Turing Test is largely inappropriate in the context of CC. Attempts to pass the Turing Test may result in losing differing, and valuable, styles of creativity (§3.1); might fail to take into account the importance of background and contextual information for a creative act (§3.2); encourage superficial, uninteresting advances in front-ends (§3.3); and result in rewarding creativity which adheres to a certain style over that which creates something which is genuinely novel (§3.4). We suggest that although there may be some place for Turing-style tests for Computational Creativity at some point in the future, it is currently impractical (§3.5).

3.1 The Turing Test penalises different styles of creativity

Creativity is a cultural notion, and people around the world understand, study and assess human creativity in many different ways, as detailed in [20]. There are also many different categories of creative humans: for instance, people with cognitive disorders such as autism, people with mental health problems, different nationalities and tribes, different genders, and what mathematician Alexander Borovik calls "that forgotten tribe of humanity, children".[7] We can often distinguish creative work performed by one of these groups; developmental psychologists can determine approximate age of a creator during childhood, people can often determine gender or nationality of an author, and so on. We do not discriminate against any of these categories purely because they are identifiable, rather we relish their differences. A writer with autism might tend to write more literally than one without, who might employ devices such as metaphor and imagery in their work. An artist with synaesthesia who can taste colour may well use colour differently to an asynaesthete. A poet under the influence of drugs might have different sorts of insights than when they were sober. A Chinese percussionist will compose music which is different to that of an African drummer. We can extend this to include animal creativity: the (plain looking) male Vogelkop Bowerbird will decorate the lawn in front of its bower in order to attract female Bowerbirds – we doubtless could distinguish a lawn which has been decorated by a human to one decorated by a Bowerbird [21] (who, for instance, has been known to consider litter such as Snickers wrappers to be highly decorative). In all of these, and countless more examples, it would be absurd to suggest that a member of one group is less creative than a member of another *simply on the grounds that we can distinguish which category they fall into.*[8] From here it is a natural step to argue that we should not discriminate against computers, even if their brand of creativity turns out to be distinguishable from human creativity (clearly this argument depends on one's motivation for studying CC).

Negrotti [23] suggests that instead of continuing to judge the computer's capabilities directly against those of the human mind, the potentials of the computer as an 'alternative intelligence' can be explored. Re-conceiving the nature of our interaction with the computer leads to a less impoverished appreciation of the human-computer as a creative assemblage. Just as it may be productive to think of the A in AI as standing for a respectable "alternative", rather than the rather derogatory "artificial", it may be productive in CC to aim to build systems which are creative in ways which are unique to machines. Humans and machines have different strengths, and rather than attempting to shoe-horn machines into a way of thinking which can be passed off as human, we should aim to develop computational systems which make the most of their strengths. It is simply carbon fascism to argue that only biological creativity is worth studying. Bedworth and Norwood [24] argue along such lines: instead of perceiving AI as recreating humans, they suggest that we should develop intelligent devices whose complexity could be used to complement human ability. Such devices would differ from the human mind in terms of nature and power, but be compatible with it. The TT forces us into the undesirable position, to paraphrase Hofstadter, of trying to make a machine act like it is not a machine.[9]

3.2 The Turing Test cannot take framing information into account

The context in which an idea or artefact has been created can affect how creative we judge the originator to be, and the value we ascribe to the idea/artefact. For example, an idea may be considered interesting if produced by a child or novice, yet dull if produced by an adult or expert, and similarly, the child/novice may be seen as more creative that the adult/expert. That is, the very thing that we are supposed to determine in a TT (who is responsible for a certain piece of work) is necessary information in the judgement of creativity. For that reason *interaction* is key, so the versions of the TT above which omit this, make the evaluation impossible. For instance, in the poetry magazine *Anon*, in which reviewers use the double blind review process to decide whether to accept or reject a poem, Askew [26] considers the difficulties of reviewing poetry without knowledge of the author. As an example, she cites a poem on childbirth, arguing that if it was written by a mother she would consider it rather mediocre, but if written by a man then she would consider it to be insightful and thoughtful. There is much work on the advantages and disadvantages

[6] These requests could be translated into a language which the program understands, without cheating, thus bypassing the need for verbal interaction.

[7] Personal communication.

[8] In psychology, inter-group comparisons have focussed on whether one group is more creative than another. For instance, work in developmental psychology such as [22] suggests that familiarity with a domain can be necessary for the flexibility required for creativity (Boden also subscribes to this view in her metaphor of exploration and transformation of conceptual spaces). Possible links between madness and creativity has been much explored, with proponents on either side (see [5]).

[9] The original quote is "... sometimes I think that all of AI has something of this playful, spoofing character. It is, after all, a delightful game to try and make a machine act like not a machine," ...[25, p. 475]

of blind peer review (for example [27]): while there are sometimes good arguments for double blind review, it is widely acknowledged to be difficult to fully evaluate a paper without the framing information of authorship and context.

3.3 The Turing Test rewards 'window dressing' and trickery

Many of the objections for using the TT to evaluate progress in AI carry over to CC. We shall not discuss most of them here: the most apt to creativity is a remark made by Lady Lovelace in her memoir on Babbage's Analytical Engine: "The Analytical Engine has no pretensions to originate anything. It can do whatever we know how to order it to perform." Turing considers this objection in [1]; both his response and Lady Lovelace's objection are explored by Boden [3] and Bringsjord, Bello and Ferrucci [28] and we do not expand them.

Hofstadter [10] addresses the issue we raised in §1 about encouraging developers of programs to focus on the wrong thing. He argues that in order to avoid the "race for flashier and flashier natural-language 'front ends' with little substance behind them", the person in the interrogator role must ask questions at the right sort of level, which will be difficult to achieve, and comments that "What is needed is a prize for advances in basic research, not a prize for window-dressing." [25, p. 491]. Techniques such as using random numbers to create what Hofstadter calls an "Artificial Wiggliness", in order to more closely resemble a hand-drawn figure could be seen in some situations as the equivalent in art programs of "flashy natural-language front ends". This is a technique used in the letterform-processing program MetaFont [29], as well as in AARON, and is hypothesised by Hofstadter to be key in our willingness to attribute AARON with artistic insight, despite being a simple, surface technique, of no real interest to CC researchers. Bringsjord *et al*. [28] argue that those in AI who do use the TT as a motivating goal know that they are competing in trickery; they are building programs which can *fool* a judge into believing that they are intelligent, rather than actually being intelligent. Thus, their goal is to create an agent which has a Chinese Room Argument-style rulebook comprehensive enough to be able to convince a judge: "In such scenarios it's really the human creators against the human judges; the intervening computation is in many ways simply along for the ride" [28, p. 2].

3.4 The Turing Test encourages pastiche

In §1 we argued that the motivation of the CC researcher will affect which evaluation criteria are appropriate. The problems with the TT and Computational Creativity are present, to different degrees, in different types of creativity, such as Boden's exploratory and transformational creativity, and other distinctions between everyday creativity and truly great creativity. In some circumstances, it may be appropriate for exploratory search to drive creative acts, but in others, this leads only to pastiche. As a particular example, while Photoshop image filters can produce images which look remarkably Impressionistic, it is very difficult to ascribe creativity to such processes as they do not innovate in either process or aesthetic evaluation. Given the value of such processes for graphic designers, etc., there is a danger that CC researchers will aim to write such pastiche generation software, missing the point of innovation and imagination in the creative process, and holding the study of creativity in software back, whatever the motivation of the CC researcher.

3.5 The Turing Test is simply too hard

We have seen that Boden argues that some systems have already passed her version of the TT. Similarly, Hofstadter argues that AARON's creations could "almost certainly be passed off as human art", and that they "look surprisingly like products of a sophisticated human artist" [10, p. 468]. Thus if we base a version of the TT on an inability to distinguish between human and computer-produced ideas, it appears that some systems may pass this test. However, in §2.4 we argue that tests based on the TT should include some form of interaction, and we suggested the sort of lines a TT for artistic creativity might follow. None of the systems so far discussed (nor any other in existence today) is anywhere close to passing this sort of test. Thus, even if the TT may at some point be a useful test of CC, it is not currently viable. While it may be useful to have a difficult (possibly unattainable) goal as an overall motivation, in practice CC needs pragmatic ways of measuring intermediate progress, which will enable us to objectively and falsifiably claim that program P_1 is more creative in ways X, Y and Z than program P_2 (where P_1 and P_2 may be different versions of the same program). Boden [3] suggests that it is more helpful to ask 'where does x lie in creativity space?' (assuming a continuous n-dimensional space for n criteria where we can measure each dimension), than 'is x creative?' (assuming a Boolean judgement), or even 'how creative is x?' (assuming a linear judgement). Turing-style tests do not allow for such subtleties. The recommendation of focusing on achievable goals in CC is echoed by Cardoso *et al*:

> To achieve human levels of Computational Creativity, we do not necessarily need to start big, at the level of whole poems, songs, stories or paintings; we are more likely to succeed if we are allowed to start small, at the level of simple but creative phrases, fragments and images [30, p. 17].

We take this to suggest that a measure of progress which covers the whole spectrum of possible achievement will be of greater practical use than one which only can only measure achievement of a grand vision.

4 Alternative suggestions: Two descriptive models

We have outlined problems with measures of CC that fail to value a type of creativity which may be specific to computers (§3.1), do not account for contextual information for a creative act (§3.2), or fail to reward genuine advances in CC (§3.3) or the genuinely novel over pastiche (§3.4). In particular, we argued for the need for workable measures which allow us to measure intermediate progree and make falsifiable claims about our programs (§3.5). These issues with Turing-style tests for CC help to motivate alternative measures of progress. In this section we describe our efforts to develop alternative measures which, we hope, avoid some of the pitfalls of the TT.

In [31, 32] we introduce and motivate two descriptive models, the FACE model and the IDEA model, which form a framework to aid us in the development and evaluation of creative software. These models are not intended to capture human creativity, nor even all of Computational Creativity. Our far more modest goal is to add another plank to the framework, begun by [33] and continued by [34], [35] and [19] to *provide a means of formalising some aspects of Computational Creativity*. At present, our discussion is limited to notions which could be used to describe creative software. While these notions are inspired by human creativity, we do not aim for a model of human creativity. Even within Computational Creativity, we merely

suggest that the FACE and IDEA models provide one possible way – by no means the only way – of describing software designed for creative purposes. The twin processes of generation and evaluation are considered fundamental within creativity studies (for instance, see [36, 33, 37, 38]). We maintain this distinction in our two complementary models; FACE, which proposes acts of creativity as the fundamental units to be assessed in creative systems, and IDEA, which describes ways of evaluating the acts.

4.1 The FACE model

The FACE model assumes eight kinds of generative acts, which produce the following kinds of results:

F^p:	a method for generating framing information
F^g:	an item of framing information
A^p:	a method for generating aesthetic measures
A^g:	an aesthetic measure
C^p:	a method for generating concepts
C^g:	a concept
E^p:	a method for generating expressions of a concept
E^g:	an expression of a concept

In order to cover as many creative acts as possible, we assume only that there must be something new created for the question of creativity to arise. This could be very small, a brush stroke of an artist, an inference step by a mathematician, a single note written in a piece of music. Our model, then, covers "merely generative" acts as well as "fundamentally generative" acts. Thus, by drawing our base line at the lowest level, our model can be used to describe the most basic "creative act" possible, and we avoid the thorny issue of where an act of creation starts. Important questions about where on the scale from basic to sophisticated an act must be to be judged creative, can be postponed.

In [30], Cardoso, Veale and Wiggins describe *The Upsidedowns of Gustav Verbeek*. These are panels which tell a story up to a half way point, the continuation of which then appears almost magically when one turns the panels upside down. Cardoso *et al*. celebrate the "artfulness" of Verbeek, while lamenting the "almost painful" gap between human and machine creativity: however they also show a simpler example of the same principle, which, they argue, *is* within reach of Computational Creativity. We show another example of this type in Figure 1. While the FACE model is designed for describing creative acts undertaken by computer, it is illustrative to describe (theoretically) how creative acts in human artistic endeavours might produce artwork such as the Verbeek piece described above. In particular, we could describe Verbeek as having undertaken a creative act of the form $\langle C^g, E^g \rangle$, which comprises an expression E^g of the concept C^g that the picture must make sense when upside down (and fit into the story). We could further describe this creative act as building on the results of multiple previous creative acts, for instance where the aesthetic A^g was invented as the notion of art having different meanings when viewed from multiple perspectives; and the generation of framing information F^g including contextual history of this genre of art, the artist's motivation, justification, etc.

Still using the Verbeek example as inspiration, at the process invention level, creative acts involving generative acts of the form F^g produce new methods for expressing the concept of art which have a different meaning when viewed upside down (for example, birds flying in the sky can double as waves in the sea, or a hat on one's head can double as a mouth on one's face). Moreover, creative acts involving generative acts of the form C^p produce methods for generating new perspectives from which the art might make sense (other examples would be rotating 90° rather than 180° - see Figure 2, or three-dimensional or moving images). Finally, methods for generating the aesthetic of art having multiple meanings when viewed from multiple perspectives would be denoted within creative acts involving generative acts A^p (another example would be the aesthetic of art having multiple meanings when viewed from a single perspective), and generative acts of the form F^p might include methods for generating new motivations, justifications etc.

Figure 1. A man coming out of the water – rotate 180° to see the same man drowning

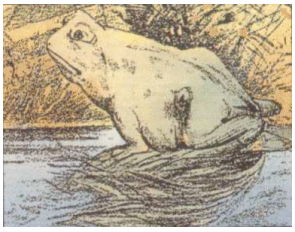


Figure 2. A frog – rotate 90° to see a horse

Clearly, not all of these generative aspects may be present in a single creative act, and they may be performed by different parties. While the model is not broad enough to cover all potentially creative software systems, we believe that it covers more than enough to guide and describe the first wave of creative systems. For example, a system which was able to perform creative acts involving generative acts of the form F^p would be more sophisticated than anything we have now: this is producing new ways to generate justifications and explanations of a creative act.

In [31], we use the FACE model to suggest ways in which different pieces of software for the same type of tasks – or indeed different versions of the same creative software – could be assessed. In particular, we suggest that a simple *quantitative* approach whereby a count of the number of creative acts produced in a given time period might be used. An alternative, or supplementary, approach might be *cumulative*, whereby software is assessed as more creative if it performs creative acts involving more types of generative acts, or a particular ordering of types of creative act could be put forward for individual domains of discourse. For instance, it could be argued that software is more creative if it invents and utilises an aesthetic measure rather than just employing a given one. We also suggest a various *qualitative* approaches where the value of the results of the creative acts of the form $\langle C^G, E^G \rangle$ are assessed against given (or invented) aesthetic measures. For instance, the average quality of the results of creative acts might be used, or an analysis of the worst ever, or best ever might be more appropriate. Finally, we suggest that the types of methods

employed within the individual generative acts might be used to differentiate creative software. For instance, a random method might be seen as less creative than one which uses induction, etc.

4.2 The IDEA model

Within the IDEA model, we begin to formalise notions of how creative acts can be measured, in terms of notions related to impact. We simplify matters by assuming an (I)terative (D)evelopment (E)xecution (A)ppreciation cycle within which software is engineered and its behaviour is exposed to an audience. We generalise past usual AI notions of correctness, soundness and value, because we are in a situation where software is meant to invent its own aesthetic or utilitarian criteria, rather than simply optimise solutions with respect to given value measures. To do this, we assume an *ideal audience* of individuals i, which is able to provide two indicators of the effect that an individual creative act, A, has had on them: (a) an indication of their change in well-being, $wb_i(A)$, between -1 and 1, with -1 indicating that they felt worse, +1 indicating that they felt better, and 0 indicating ambivalence, and (b) an indication between 0 and 1 of the cognitive effort they spent in trying to appreciate a creative act and the artefact(s) it produced, $ce_i(A)$. Denoting the mean value of the well-being rating over the n people as $m(A)$, we propose the following measures for use in impact assessment exercises:

$dis(A) = disgust(A) = \frac{1}{2n}\sum_{i=1}^{n}(1 - wb_i(A))$
$div(A) = divisiveness(A) = \frac{1}{n}\sum_{i=1}^{n}|wb_i(A) - m(A)|$
$ind(A) = indifference(A) = 1 - \frac{1}{n}\sum_{i=1}^{n}|wb_i(A)|$
$pop(A) = popularity(A) = \frac{1}{2n}\sum_{i=1}^{n}(1 + wb_i(A))$
$prov(A) = provocation(A) = \frac{1}{n}\sum_{i=1}^{n}(ce_i(A))$

By compounding the provocation measure with the others, we can attempt to capture some kinds of impact that creative acts might have:

$acquired\_taste(A) = (pop(A) + prov(A))/2$
$instant\_appeal(A) = (1 + pop(A) - prov(A))/2$
$opinion\_splitting(A) = (1 + div(A) - prov(A))/2$
$opinion\_forming(A) = (div(A) + prov(A))/2$
$shock(A) = (1 + dis(A) - prov(A))/2$
$subversion(A) = (dis(A) + prov(A))/2$

These all return a value between 0 and 1, and we argue that if A reaches towards 1 for any of these measures, it has had some impact, such as being shocking, or divisive.

In [31], we flesh out the models, by including notions of ideal background information and an ideal programming environment, and using these to suggest further ways to compare the creative acts performed by software and their impact. In particular, we suggest six stages for the development of software for creative purposes: (*i*) a developmental stage: where all the creative acts undertaken by the software are based on inspiring examples (using terminology from [35] (*ii*) a fine-tuning stage: where the creative acts performed are abstracted away from inspiring examples, but are still too close to have an impact as novel inventions (*iii*) a re-invention stage: where the software performs creative acts similar to ones which are known, but which were not explicitly provided by the programmer (*iv*) a discovery stage: where the software performs creative acts sufficiently dissimilar to known ones to have an impact due to novelty, but sufficiently similar to be assessed within current contexts (*v*) a disruption stage: where the software performs some creative acts which are too dissimilar to those known to the world to be assessed in current contexts, hence new contexts have to be invented, and (*vi*) a disorientation stage: where all the creative acts performed are too dissimilar to known ones for there to be any context within which to judge any of the activities of the software. We suggest that an analysis of the software with respect to which stage of development it is in, can be used to compare and contrast creative programs.

5 Conclusions and Further Work

We have described Computational Creativity as the AI subfield in which we study how to build software that models creative thought in science and the arts. In order to have a notion of progress, and to set an agenda for researchers who are modelling aspects of creative thought, it is essential to agree practical evaluation measures, based on sound theoretical foundations, which we can apply to our programs to help to identify aspects which are satisfactory and those which should be improved. We have discussed the use of the Turing Test, and different versions of it, for such purposes, and argued that it is largely inappropriate in this context. This is because attempts to pass the Turing Test may result in losing differing, and valuable, styles of creativity; might fail to take into account the importance of background and contextual information for a creative act; encourage superficial and uninteresting advances in front-ends; and result in rewarding creativity which adheres to a certain style over that which creates something which is genuinely novel. We suggest that although there may be some place for Turing-style tests for Computational Creativity at some point in the future, it is currently untenable and impractical.

As an alternative to Turing-style tests, we introduce two descriptive models for evaluating creative software, the FACE model which describes creative acts performed by software in terms of tuples of generative acts, and the IDEA model which describes how such creative acts can have an impact upon an ideal audience, given ideal information about background knowledge and the software development process. We believe that these alternative measures constitute a beginning in our efforts to avoid some of the pitfalls of the TT: they do not discriminate against a creativity which may be specific to computers, they take contextual information into account via the framing aspect of the FACE model, they reward genuine advances in CC and the genuinely novel over pastiche. Perhaps most importantly, we believe that they are workable measures which will enable us to measure intermediate progress and make falsifiable claims about our programs. We demonstrate the practicability of the descriptive models in [31], where we use them within comparison studies of existing software built for creative purposes. In particular, we compare and contrast mathematical invention software including the AM [39], HR [40] and HRL [41] programs. We similarly compare and contrast various pieces of generative art software, including the AARON program [12], The Painting Fool [42] and the NEvAr evolutionary art software [43]. Moreover, in [32], we further motivate the FACE and IDEA models by appealing to some of the authors mentioned above, and others like Sloman [44] and Thagard [45], who suggest criteria against which these descriptive models might be judged. We place the work in the context of existing approaches to the assessment of creativity in software, and in a wider context of creativity studies, in addition to providing a case study: the Basel problem from mathematics, described in [46] as the "best known problem of the time".

In [47], we suggest methods, methodologies and paradigms within which creative software might be written. In particular, we propose some ways in which to manage the public perception of creativity (or lack thereof) in computers. The descriptive models presented above are intended as a complement to these public perception guidelines, whereby AI practitioners can rely on concrete assessment methods

for the usually difficult topic of apportioning creativity to software. The FACE and IDEA descriptive models are not yet particularly acute tools for a full assessment of creativity in software, and we plan to develop sub-models for various notions which have been used to describe the creativity (or lack thereof) in computer systems in recent years. These terms include, but are not limited to, the following: affect, analogy, appreciation, audience, autonomy, blending, community, context, curiosity, exploration, framing, humanity, humour, idea formation, imagination, intentionality, interaction, interpretation, knowledge, metaphor, novelty, obfuscation, personality, physicality, playfulness, problem solving, process, programming, search, surprise, transformation and trust. Using the foundational terminology for creative acts and impact described above, we plan to expand each term into a formalism containing conceptual definitions and concrete calculations using those definitions which can be used for the assessment of creativity in software. In doing so, we hope to contribute a *Computational Creativity Theory* which will provide a strong foundation for objectively measured progress in our field.

Acknowledgements

We are very grateful to John Charnley for his thoughts on the FACE and IDEA descriptive models. We would also like to thank two anonymous reviewers for their helpful comments. This work is supported by EPSRC grants EP/F035594/1 and EP/F036647/1.

REFERENCES

[1] A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

[2] A. Kozbelt, R. A. Beghetto, and M. A. Runco. Theories of creativity. In J. C. Kaufman and R. J. Sternberg, editors, *The Cambridge Handbook of Creativity*, pages 20–47. Cambridge University Press, USA, 2010.

[3] M.A. Boden. *The Creative Mind: Myths and Mechanisms*. Weidenfield and Nicholson, London, 1990.

[4] J. A. Plucker and M. C. Makel. Assessment of creativity. In J. C. Kaufman and R. J. Sternberg, editors, *The Cambridge Handbook of Creativity*, pages 48–73. Cambridge University Press, USA, 2010.

[5] A. Rothenberg. *Creativity and Madness*. The John Hopkins University Press, Baltimore, USA, 1990.

[6] A. Newell, J. G. Shaw, and H. A. Simon. The process of creative thinking. In H. E. Gruber, G. Terrell, and M. Wertheimer, editors, *Contemporary Approaches to Creative Thinking*, page 63119. Atherton, New York, 1963.

[7] M. T. Pearce and G. A. Wiggins. Towards a framework for the evaluation of machine compositions. In G. A. Wiggins, editor, *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Science*, 2001.

[8] A. Garnham. Art for arts's sake. *Behavioural and Brain Sciences*, 17(3):543–544, 1994.

[9] A. Bundy. What is the difference between real creativity and mere novelty? *Behavioural and Brain Sciences*, 17(3):533–534, 1994. Open peer commentary on [3].

[10] D. Hofstadter. Epilogue: on computers, creativity, credit, brain mechanisms, and the Turing test. In D. Hofstadter and the Fluid Analogies Research Group, editors, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, pages 467–491. Basic Books, 1995. Epilogue in [25].

[11] M. A. Boden. The Turing test and artistic creativity. *Kybernetes*, 39(3):409–413, 2010.

[12] P. McCorduck. *AARON's Code: Meta-Art, Artificial Intelligence, and the Work of Harold Cohen*. Freeman, New York, 1991.

[13] M. A. Boden and E. A. Edmonds. What is generative art? *Digital Creativity*, 20(1-2):21–46, 2009.

[14] D. Cope. *Virtual Music: Computer Synthesis of Musical Style*. The MIT Press, Cambridge, Massachusetts, 2001.

[15] D. Cope. *Computer Models of Musical Creativity*. The MIT Press, Cambridge, Massachusetts, 2006.

[16] S. C. Todd and W. Latham. *Evolutionary Art and Computers*. Academic Press, London, 1992.

[17] K. Sims. Artificial evolution for computer graphics. *Computer Graphics*, 25(4):319–328, July 1991.

[18] G. A. Wiggins. A preliminary framework for description, analysis and comparison of creative systems. *Journal of Knowledge Based Systems*, 19(7):449–458, 2006.

[19] S. Colton. Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI Spring Symposium on Creative Systems*, 2008.

[20] J. C. Kaufman and R. J. Sternberg, editors. *The International Handbook of Creativity*. Cambridge University Press, Cambridge, New York, USA, 2006.

[21] J. Diamond. Animal art: Variation in bower decorating style among male bowerbirds *amblyornis inornatus*. *Proceedings of the National Academy of Sciences in the USA*, 83(9):3042–3046, May 1, 1986.

[22] A. Karmiloff-Smith. Constraints on representational change: Evidence from children's drawing. *Cognition*, (34):57–83, 1990.

[23] M. Negrotti. *Understanding the Artificial*. Springer-Verlag, London, 1991.

[24] J. Bedworth and J. Norwood. The Turing test is dead. In *Proceedings of the 3rd conference on creativity and cognition*, 1999.

[25] D. Hofstadter and the Fluid Analogies Research Group. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, NY, USA, 1995.

[26] C. Askew. Blind faith and anonymity. *Anon*, 7:pp. 55–58, 2010.

[27] F. Rowland. The peer-review process. *Learned Publishing*, 15(4), 2002.

[28] S. Bringsjord, P. Bello, and D. Ferrucci. Creativity, the Turing test, and the (better) Lovelace test. *Minds and Machines*, (11):3–27, 2001.

[29] D. E. Knuth. The concept of a meta-font. *Visible Language*, 16(1):3–27, 1982.

[30] A. Cardoso, T. Veale, and G. A. Wiggins. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine*, 30(3):15–22, 2009.

[31] S. Colton, A. Pease, and J. Charnley. Computational creativity theory: The FACE and IDEA descriptive models. In *2nd International Conference on Computational Creativity*. 2011, 2011.

[32] A. Pease and S. Colton. Computational creativity theory: Inspirations behind the FACE and the IDEA models. In *2nd International Conference on Computational Creativity*. 2011, 2011.

[33] M Boden. *The Creative Mind: Myths and Mechanisms (second edition)*. Routledge, 2003.

[34] G. A. Wiggins. Searching for computational creativity. *New Generation Computing*, 24(3):209–222, 2006.

[35] G. Ritchie. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, 17:67–99, 2007.

[36] G. Wallas. *The Art of Thought*. Harcourt Brace, NY,USA, 1926.

[37] R Finke, T Ward, and S. Smith. *Creative cognition: Theory, research and applications*. MIT press, Cambridge, 1992.

[38] G. Ritchie. Assessing creativity. In G. A. Wiggins, editor, *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Science*, pages 3–11. SSAISB, 2001.

[39] D. B. Lenat. Automated theory formation in mathematics. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 833–842, Cambridge, MA, 1977. Morgan Kaufmann.

[40] S. Colton. *Automated Theory Formation in Pure Mathematics*. Springer-Verlag, 2002.

[41] A. Pease. *A Computational Model of Lakatos-style Reasoning*. PhD thesis, University of Edinburgh, 2007.

[42] A Krzeczkowska, J El-Hage, S Colton, and S Clark. Automated collage generation - with intent. In *Proceedings of the 1st International Conference on Computational Creativity*, 2010.

[43] P Machado and A Cardoso. NEvAr – the assessment of an evolutionary art tool. In *Proceedings of the AISB00 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, 2000.

[44] A. Sloman. *The Computer Revolution in Philosophy*. The Harvester Press, Ltd., 1978.

[45] P. Thagard. *Computational Philosophy of Science*. MIT Press, Cambridge, Mass, 1993.

[46] C. E. Sandifer. *The early mathematics of Leonhard Euler*. The Mathematical Association of America, 2007.

[47] S Colton. Seven catchy phrases for computational creativity research. In *Proceedings of the Dagstuhl Seminar: Computational Creativity: An Interdisciplinary Approach*, 2009.

Creative or Not? Birds and Ants Draw with Muscles

Mohammad Majid al-Rifaie[1] and **Mark John Bishop**[2] and **Ahmed Aber** [3]

Abstract. In this work, a novel approach of merging two swarm intelligence algorithms is considered – one mimicking the behaviour of ants foraging (Stochastic Diffusion Search [5]) and the other algorithm simulating the behaviour of birds flocking (Particle Swarm Optimisation [17]). This hybrid algorithm is assisted by a mechanism inspired from the behaviour of skeletal muscles activated by motor neurons. The operation of the swarm intelligence algorithms is first introduced via metaphor before the new hybrid algorithm is defined. Next, the novel behaviour of the hybrid algorithm is reflected through a cooperative attempt to make a drawing, followed by a discussion about creativity in general and the 'computational creativity' of the swarm.

1 Introduction

In recent years, studies of the behaviour of social insects (e.g. ants and bees) and social animals (e.g. birds and fish) have proposed several new metaheuristics for use in collective intelligence. This paper explores an artistic application of this collective intelligence, which emerges through the interaction of simple agents (representing the social insects/animals) in two nature-inspired algorithms, namely, Particle Swarm Optimisation (PSO) [17] and Stochastic Diffusion Search (SDS) [5]. Additionally, the mechanism of muscle activation is utilised to introduce the drawing with another layer of detail.

Natural examples of swarm intelligence that exhibit a form of social interaction are fish schooling, birds flocking, ant colonies in nesting and foraging, bacterial growth, animal herding, brood sorting etc.

The parable of the *blind men and the elephant* suggests how social interactions can lead to more intelligent behaviour. This famous tale, set in verse by John Godfrey Saxe [30] in the 19th century, characterises six blind men approaching an elephant. They end up having six different ideas about the elephant, as each person has experienced only one aspect of the elephant's body: wall (elephant's side), spear (tusk), snake (trunk), tree (knee), fan (ear) and rope (tail). The moral of the story is to show how people build their beliefs by drawing them from incomplete information, derived from incomplete knowledge about the world [18]. If the blind men had been communicating about what they were experiencing, they would have possibly come up with the conclusion that they were exploring the heterogeneous qualities that make up an elephant.

Following other works in the field of swarm painting (e.g. [22, 3, 33, 34] and ant colony paintings [14, 21]), this work, in addition to exhibiting the cooperation of birds and ants as a new way in making a drawing, benefits from the mechanism used in skeletal muscles.

In this paper, each of the swarm intelligence algorithms used are first explained (Sections 2 and 3), and an approach to their possible integration highlighted (Section 4). Subsequently the simplified mechanism of muscle activation is described (Section 5), followed by an explanation of how the new hybrid algorithm produces a drawing; a process initially inspired by an input sketch and the role that muscle activation mechanism plays (Section 6). In Section 7 the similar individualistic approach of the swarm and their importance in making a drawing is highlighted, followed by future research in the field.

Lastly, despite the novelty of this hybrid approach, it is not the intention of the authors to use the results outlined in the work to make either strong epistemological claims of computational creativity or strong aesthetic claims of style.

2 Birds: Particle Swarm Optimisation!

Particle Swarm Optimisation (PSO), first developed in 1995 by Kennedy and Eberhart [17, 12], is a population-based, optimization technique which came about as a result of an attempt to graphically simulate the choreography of fish schooling or birds flying (e.g. pigeons, starlings, and shorebirds) in coordinated flocks that show strong synchronisation in turning, initiation of flights and landing. Despite the fact that members of the swarm neither have knowledge about the global behaviour of the swarm nor a global information about the environment, the local interactions of the swarms triggers a complex collective behaviour, such as flocking, herding, schooling, exploration and foraging behaviour [27, 19, 4, 16].

A high-level description of PSO is presented in form of a social metaphor – Lost Child in Jungle [4] – demonstrating the procedures through which the communication exchange is facilitated between members of the swarm in its simplest possible form (for detailed, formal explanation and mathematical equations, see [17, 12]).

2.1 The Lost Child in Jungle

A group of villagers realise that a child is lost in a jungle nearby and set off to find the child. Each one of the villagers is given a mobile phone equipped with GPS that can be used to communicate with the head of the village. Each villager is also provided with a diary to record some data, as explained below:

> The villagers should log the location where they find the best information so far about the child in their diaries (Personal Best, $pbest$ position) and inform the head of the village about it. Whenever they find something better that might lead to the location of the child (a location with a better fitness than their current $pbest$), they should provide the head of the village with the update.

[1] Goldsmiths, University of London, UK, email: m.majid@gold.ac.uk
[2] Goldsmiths, University of London, UK, email: m.bishop@gold.ac.uk
[3] Royal Free Hospital, London, UK, email: ahmed.aber@nhs.net

[4] Please note that this metaphor is presented here to give the reader an idea of how the algorithm works, without getting involved in detailed technical issues and mathematical equations.

The head of the village is responsible to contrast all the $pbest's$ he has received so far from all the villagers and pick the best one (Global Best, $gbest$ position). The resultant $gbest$ is communicated back to the villagers.
Each villager, on the other hand, should log the following three in his diary throughout the search:

- position
- speed (velocity) in walking
- $pbest$ position (which is also called $memory$)

Additionally, they should be able to access the $gbest$ position from the head of the village.
In the next step, when villagers decide about their next move from their current position, they need to consider their two bests ($pbest$ and $gbest$) and their current velocity.
Thus, while each villager does not neglect his personal findings, he has extra knowledge about its neighbourhood through $gbest$[5]; therefore, preserving a balance between exploration of the search space (e.g. jungle, in this case), and exploitation of potentially good areas around each villager's personal best.

In this example, villagers are analogous to particles in PSO, where optimisation is based on particles' individual experience ($pbest$) and their social interaction with the particle swarms (via $gbest$).

Algorithm 1 describes the metaphor chronologically.

At the convergence of the search process, villagers are most likely to congregate in the area of jungle where the child is most likely to be found; so hopefully, using this algorithm, the child is brought back to his family in the village.

Algorithm 1 The Lost Child in Jungle

```
Villagers spread in the jungle

While ( the child is not found )
   For all villagers
      Evaluate the fitness of the current location
          (how good the current location is
          to lead to the child)

      If (current location is better than pbest)
          pbest = current location
       If (pbest is better than gbest)
          gbest = pbest
       Villager decides about his next move
   End
End
```

3 Ants: Stochastic Diffusion Search!

This section briefly introduces a multi-agent global search and optimisation algorithm called Stochastic Diffusion Search (SDS) [5], whose behaviour is based on simple interaction of agents.

SDS introduced a new probabilistic approach for solving best-fit pattern recognition and matching problems. SDS, as a multi-agent population-based global search and optimisation algorithm, is a distributed mode of computation utilising interaction between simple agents [11].

Unlike many nature inspired search algorithms, SDS has a strong mathematical framework, which describes the behaviour of the algorithm by investigating its resource allocation [24], convergence to global optimum [25], robustness and minimal convergence criteria [23] and linear time complexity [26]. A social metaphor, *the Mining Game* [1], is used to describe the mechanism through which SDS allocates resources.

[5] The topology of the metaphor presented here is global neighbourhood.

3.1 The Mining Game

This metaphor provides a simple high-level description of the behaviour of agents in SDS, where a mountain range is divided into hills and each hill is divided into regions:

A group of miners learn that there is gold to be found on the hills of a mountain range but have no information regarding its distribution. To maximize their collective wealth, the maximum number of miners should dig at the hill which has the richest seams of gold (this information is not available a-priori). In order to solve this problem, the miners decide to employ a simple Stochastic Diffusion Search.

- At the start of the mining process each miner is randomly allocated a hill to mine (his hill hypothesis, h).
- Every day each miner is allocated a randomly selected region, on the hill to mine.

At the end of each day, the probability that a miner is happy is proportional to the amount of gold he has found. Every evening, the miners congregate and each miner who is not happy selects another miner at random for communication. If the chosen miner is happy, he shares the location of his hill and thus both now maintain it as their hypothesis, h; if not, the unhappy miner selects a new hill hypothesis to mine at random.

As this process is structurally similar to SDS, miners will naturally self-organise to congregate over hill(s) of the mountain with high concentration of gold.

In the context of SDS, agents take the role of miners; active agents being 'happy miners', inactive agents being 'unhappy miners and the agent's hypothesis being the miner's 'hill-hypothesis'.

Algorithm 2 The Mining Game

```
Initialisation phase
Allocate each miner (agent) to a random
  hill (hypothesis) to pick a region randomly

While (all miners congregate over the highest
   concentration of gold)

   Test phase
      Each miner evaluates the amount of gold
        they have mined (hypotheses evaluation)
      Miners are classified into happy (active)
        and unhappy (inactive) groups

   Diffusion phase
      Unhappy miners consider a new hill by
        either communicating with another miner
        or, if the selected miner is also
        unhappy, there will be no information
        flow between the miners; instead the
        selecting miner must consider another
        hill (new hypothesis) at random
End
```

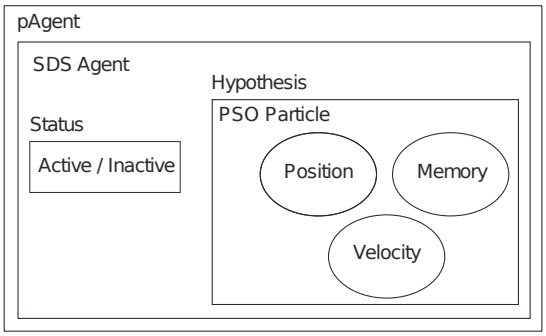
4 Cooperation: Birds and Ants!

In ongoing research [2], an initial set of experiments aimed to investigate if the information diffusion mechanism deployed in SDS ("ants") on its own improves PSO ("birds") behaviour. Early results demonstrate the high potential of this integration.

In the hybrid algorithm, each PSO particle (villager in the Lost Child metaphor) has a current position, a memory (personal best position) and a velocity; each SDS agent (miner, in the Mining Game metaphor), on the other hand, has hypothesis (hill) and status (happy or unhappy).

In the experiment reported here, every PSO particle is an SDS agent too – together termed *pAgents*. In pAgent, SDS-style hypotheses are defined by the PSO particle positions, and an additional boolean variable (status) determines whether the pAgent is active or inactive (see Figure 1).

Figure 1. pAgent



The behaviour of the hybrid algorithm in its simplest form is presented in Algorithm 3.

Algorithm 3 Hybrid Algorithm

```
Initialise pAgents

While ( stopping condition is not met )
   For all pAgents
      Evaluate fitness value of each particle

      If ( evaluation counter MOD n == 0 )
         // START SDS
         // TEST PHASE
         for pAg = 1 to No-of-pAgents
            r_pAg = pick-random-pAgent()
            if ( pAg.pbestFitness() <=
                 r_pAg.pbestFitness() )
               pAg.setActivity (true)
             else
               pAg.setActivity (false)
            end if
         end for

         // DIFFUSION PHASE
         for ag = 1 to No_of_pAgents
            if ( pAg.activity() == false )
               r_pAg = pick-random-pAgent()
               if ( r_pAg.activity() == true )
                  pAg.setHypo( r_pAg.getHypo() )
               else
                  pAg.setHypo( randomHypo() )
            end if
         end for
      end if
      // END SDS

     If (current fitness is better than pbest)
         pbest = current fitness
      If (pbest is better than gbest)
         gbest = pbest
      Particle decides about its next move
   End
End
```

5 The Simplified Mechanism of Muscle Activation

Motor neurons activate the skeletal muscle mainly through the neurotransmitter Acetylcholine (Ach) at the neuromuscular junction (NMJ). This junction is a synapse where the unmyelinated motor nerve terminals are separated from the postsynaptic membrane by a cleft that contains a basal lamina [28]. This cleft includes many proteins including acetylcholine esterase (AChE) which hydrolyse ACh. The postsynaptic membrane at the NMJ forms a series of deep folds. The acetylcholine receptors (AChRs) are found at the top one-third of these folds, whereas the voltage-gated sodium channels are anchored at the bottom of the folds [29, 15].

The nerve action potential from the motor neuron opens voltage-gated calcium channels that are located at the motor nerve terminal of the NMJ. The resulting influx of calcium leads to the release of acetylcholine (ACh) from the motor end of the junction into the synapse. Nearly 65% reaches the ACh receptors (AcHR) on the postsynaptic membrane. Binding of two ACh to each AChR leads to the opening of the AChR-associated ion channel, influx of cations (mainly sodium) and generation of an endplate potential (EPP) [31].

The EPP rapidly depolarises the postsynaptic membrane and, this depolarization should pass a certain threshold so that enough voltage-gated sodium channels are activated for the propagation of an action potential along the muscle fiber, once this happens the muscle contracts [10]. The extent to which the EPP exceeds that necessary threshold to initiate the action potential is usually called the safety factor for neuromuscular transmission [37]. The EPP is short-lived because the AChRs close spontaneously, ACh dissociates and escapes by diffusion or is hydrolysed by AChE.

In this paper, the effect of the activation of voltage-gated sodium channels on muscle contraction and the way motor neurons activate the skeletal muscle are used for an artistic purpose.

6 The Drawing Mechanism

In this section, first the drawing made with the hybrid swarm algorithm (PSO-SDS) is presented and then the influence of the muscle activation mechanism on the drawing is explored.

6.1 Birds and Ants Set off to Draw

Once the swarm (birds and ants) are presented with a sketch (see Figure 2), they use it as inspiration and begin making a drawing based on the sketch, but utilising their own 'style'.

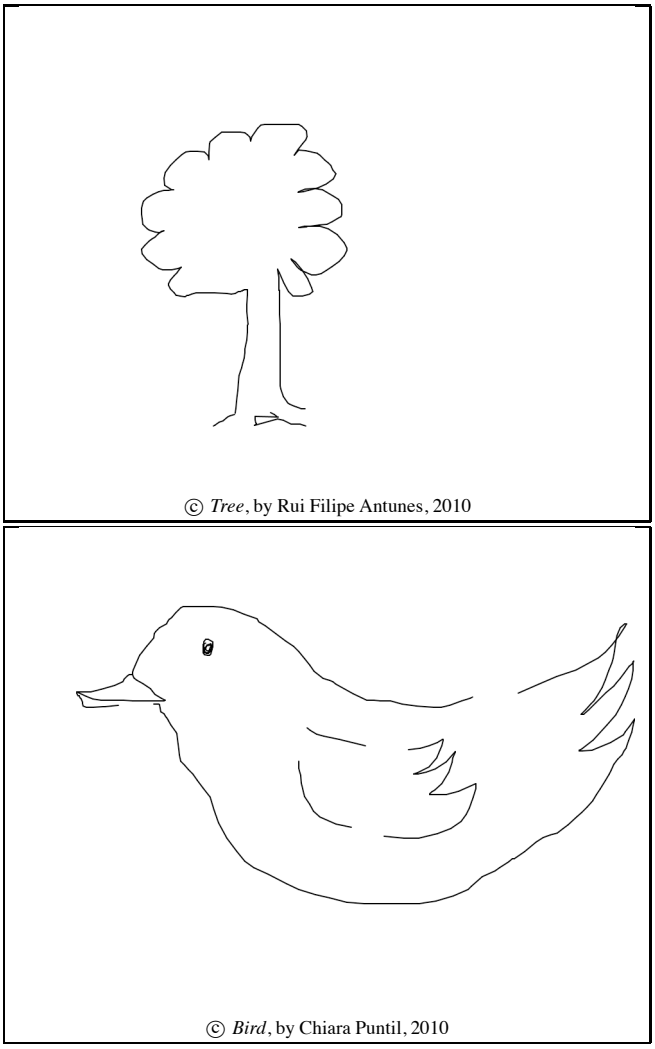
The goal of "birds" (PSO algorithm) is to trace the lines (series of points) in the sketch, and "ants" (SDS algorithm) help the birds in this process as explained in Section 4. The trace of the birds and the footprints of the ants stay on the canvas, creating a drawing inspired by the initial sketch, followed by a signature of the swarm at the corner of the canvas (see Figure 3).

6.2 How Muscle Contraction Shapes the Drawing

The simplified mechanism of muscle contraction is used in the drawing to reflect the relation between the time spent for drawing each part (e.g. each line) and the form (spikes' diameter) of the disks representing the contracted muscles, which are visible around each member of the swarm.

Here, in drawing, the concept of duration (for drawing a line), is reversely analogous to the idea of the activation of voltage-gated sodium channels in the mechanism of muscle contraction, which –

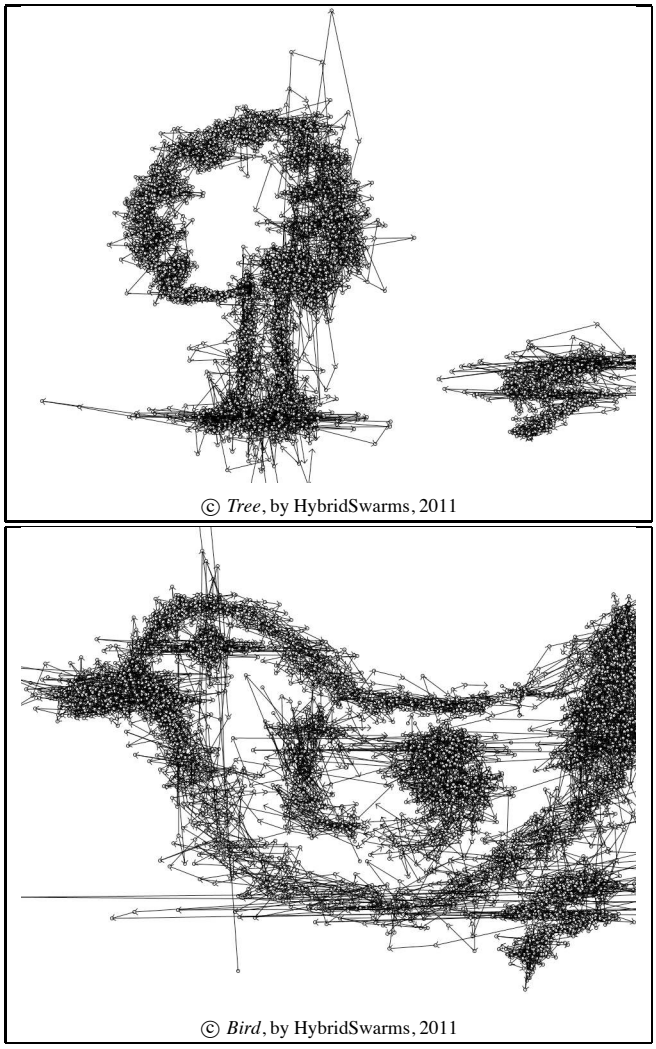
Figure 2. Sketches Provided to the Swarm



© *Tree*, by Rui Filipe Antunes, 2010

© *Bird*, by Chiara Puntil, 2010

Figure 3. The Drawings of the Hybrid Swarms



© *Tree*, by HybridSwarms, 2011

© *Bird*, by HybridSwarms, 2011

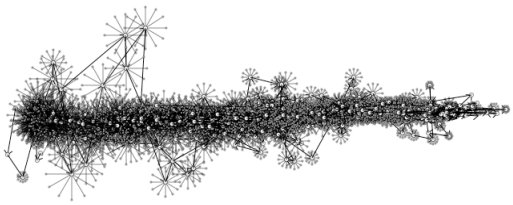
for this artistic purpose – indicates, the shorter the time, the higher the activation voltage-gated sodium channels, which in tum leads to a bigger contraction (or shock) in each member of the swarm.

When a line is drawn faster than the other in a drawing, the spikes formed around each member of the swarm (while drawing that line), is bigger (more spread on the canvas), but when a line is drawn slower (i.e. the pressure is higher), it will have smaller, more intense (concentrated on the canvas) disk around the member of the swarm. See Figure 4.

Having the concept of contraction or 'shock' derived from muscle activation, Figure 5 shows the sketches drawn by the swarm, using birds, ants and the mechanism of muscle contraction.

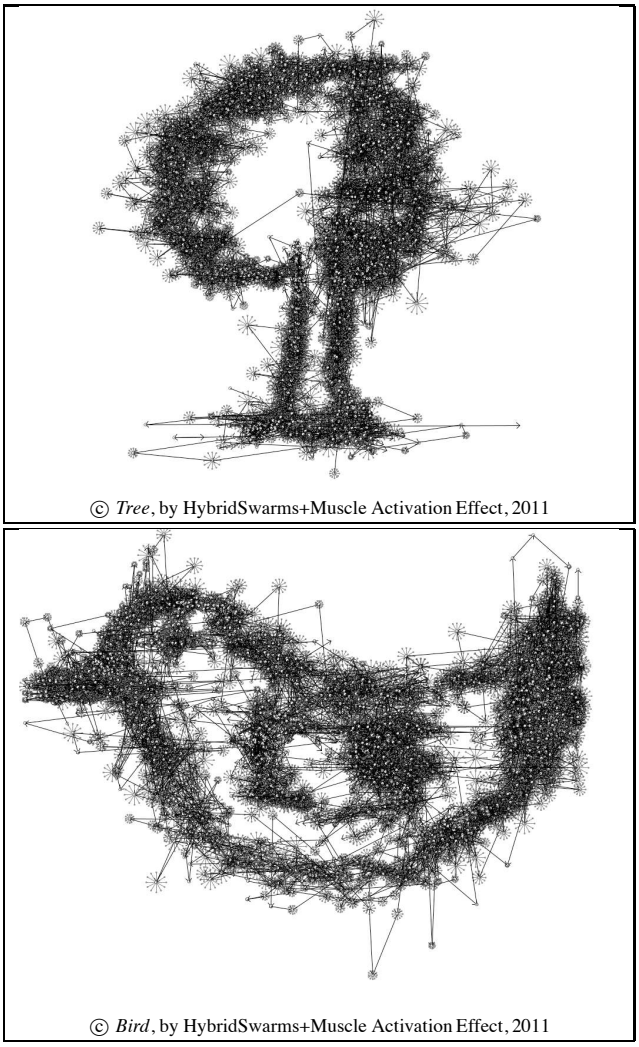
Although even if the hybrid swarm mechanism (of birds, ants and muscle) processes the same sketch several times it will not make two identical drawings; furthermore the outputs it produces are not merely randomised versions of the input. This can be demonstrated qualitatively by comparing the output of the hybrid swarm system with a simple randomised tracing algorithm (e.g. contrast Figures 6 with Figure 7). The reason why the hybrid swarm drawings are different from using random lines and spikes (shocked muscles) following the lines of a sketch, is that the underlying algorithms and mecha-

Figure 4. Muscle Contraction (shock) on Drawing



nism [used to coordinate the concentrations at any particular point on the canvas] employ proven swarm intelligence techniques; a method which is better (more 'loyal' to the original sketch) than a simple randomisation, but which still has enough 'freedom' to ensure originality in the resulting drawing (i.e. the swarm mechanisms ensure high-level fidelity to the input without making an exact low-level copy of the sketch). Thus, despite the fact that the swarm are constrained by the rules they follow (see Sections 2 and 3), the stochastic parts of the algorithms allow them to demonstrate a "regulated difference" rather than a simple "random difference".

Figure 5. The Drawings of the Hybrid Swarms with Muscle Activation



© *Tree*, by HybridSwarms+Muscle Activation Effect, 2011

© *Bird*, by HybridSwarms+Muscle Activation Effect, 2011

6.3 Regulated difference versus random difference

The drawings in Figure 6 (top and middle) show two outputs from the simple randomised algorithm when configured with limited 'artistic' freedom (i.e. there is a only small Gaussian random distance and direction from the lines of the original sketch); comparing the two drawings we note a lack of any significant difference between them. Furthermore, when more 'artistic freedom' is granted to the randomised algorithm (by further increasing the variance in the underlying Gaussian, which allows the technique to explore a wider areas of the canvas), the algorithm begins to deviate excessively from the original sketch. I.e. Excessive randomisation results in a poor - low fidelity - interpretation of the original sketch (Figure 6-bottom). In contrast although the agents in the hybrid 'bird, ant and muscle swarm' are free to access any part of the canvas they naturally maintain recognisable fidelity to the original input. Thus it can be seen that simply extending a basic swarm mechanism by giving it simply more randomised behaviour (giving it more 'artistic freedom') fails to demonstrate that more creative drawings would be produced.

Thus the 'controlled freedom' (or the *'tincture of madness'*) exhibited by the hybrid swarm algorithm (induced by the stochastic side of the algorithms) is crucial to the resultant work[6] and is the reason why having the same sketch does not result in the system producing identical drawings[7].

Figure 6. The Drawings of the Swarms with Random Behaviour

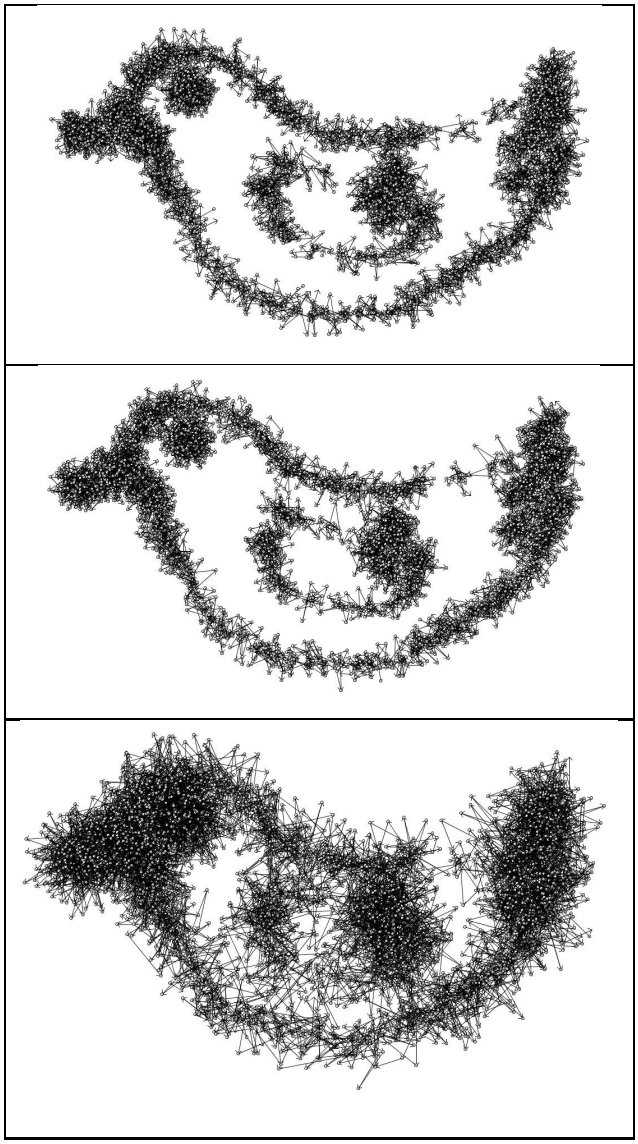


Figure 7 shows a few drawings made by the hybrid swarm system, inspired by a single input sketch. Interestingly, and irrespective of whether the hybrid swarm is 'genuinely creative' or not, its individ-

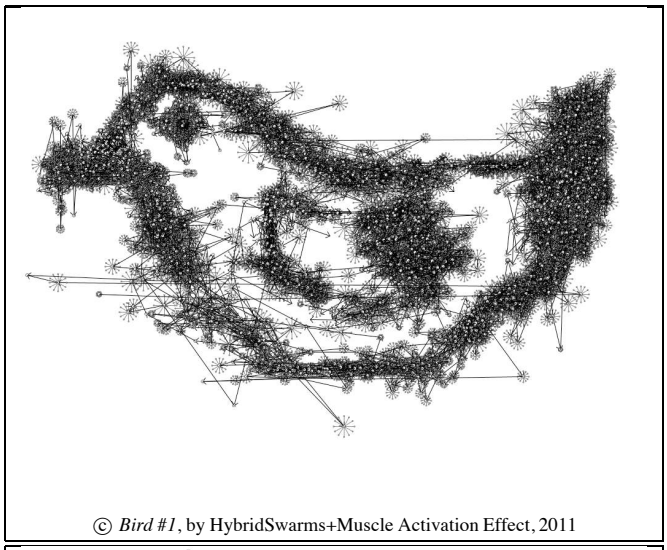
[6] This freedom emerges, among other things, from the the stochasticity of SDS algorithm in picking agents for communication, as well as choosing agents to diffuse information (see Algorithm 2); and the tincture of madness in PSO algorithm is induced via its strategy of spreading villagers in the jungle as well as the stochastic elements in deciding the next move of each villager (see Algorithm 1).

[7] Although the algorithms (PSO and SDS) and the mechanism (skeletal muscle activation) are biologically inspired we do not claim that the presented work is an accurate model of natural systems. Furthermore in designing the algorithm there was no explicit 'Hundertwasser-like' attempt - by which we mean stress on using curves instead of straight lines, as Hundertwasser considered straight lines not nature-like and 'godless' and tried not to use straight lines in his works - to bias the style of the system's drawings.

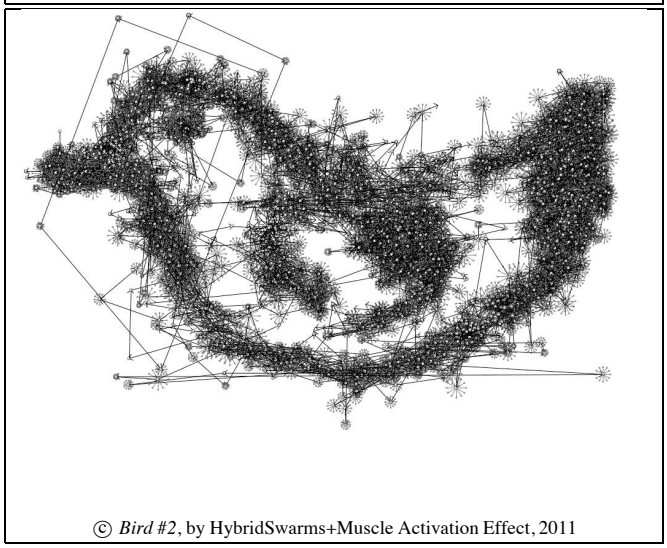
ualistic style is not totally dissimilar to those of the 'elephant artists' [36]):

"After I have handed the loaded paintbrush to [the elephants], they proceed to paint in their own distinctive style, with delicate strokes or broad ones, gently dabbing the bristles on the paper or with a sweeping flourish, vertical lines or arcs and loops, ponderously or rapidly and so on. No two artists have the same style."

Figure 7. Different Drawings of the Hybrid Swarms off a Single Sketch



© *Bird #1*, by HybridSwarms+Muscle Activation Effect, 2011



© *Bird #2*, by HybridSwarms+Muscle Activation Effect, 2011

7 Discussion on Creativity

In this section, the aim is to discuss whether the hybrid swarm algorithms can in some sense be 'computationally creative' in what they draw. In our discussion we emphasise the importance of: 'controlled freedom' (cf. unregulated randomness) and the combinatorial creativity of the hybrid swarm system and contrast it with examples of potential non-human assessment of aesthetic judgment and suggestions of creativity in natural distributed systems. In order to deflect the charge that computational systems cannot be sensitive to emotion we subsequently briefly discuss recent work from Simon Colton. Finally, we complete the section with a demonstration of the provenance of the use of [real-world] swarm-systems in successful exhibited artworks (e.g. by Julie Freeman). Our modest conclusion is that 'controlled freedom' (pace unconstrained randomness) - as for example exhibited in the hybrid bird, ant and muscle algorithm presented herein - can be useful in generating interesting and intelligible drawing outputs.

7.1 On Freedom and Art

For years, it has been argued that there is a relationship between art, creativity and freedom, among which is the famous German prose, by Ludwig Hevesi at the entrance of the Secession Building in Vienna:

"Der Zeit ihre Kunst

Der Kunst ihre Freiheit[8]"

Or a quote by Aristotle (384-322 BCE) [13], which emphasises on the link between creativity and freedom (here, having "a tincture of madness"):

"There was never a genius without a tincture of madness."

Boden, in [7], also argues that creativity has an ambiguous relationship with freedom. Among several definitions that have been given to creativity, around sixty of which (as stated by Taylor [32]) belong to combinational creativity, which is defined as *"the generation of unfamiliar combinations of familiar ideas"* [6]; a category that the presented work fits in. Considering the existence of many influencing factors in evaluating what is creative, raises questions about how humans evaluate artistic creativity. Galanter in [20] suggests that perhaps computational equivalent of a bird or an insect (e.g. in evaluating mate selection) is "all" that is required for computational aesthetic evaluation and furthermore states:

".. this provides some hope for those who would follow a psychological path to computational aesthetic evaluation, because creatures with simpler brains than man practice mate selection."

In this context Dorin and Korb [20] suggest that the tastes of the individual in male bowerbirds is visible when they gather collections of bones, glass, pebbles, shells, fruit, plastic and metal scraps from their environment, and arrange them to attract females [8]:

"They perform a mating dance within a specially prepared display court. The characteristics of an individual's dance or artefact display are specific to the species, but also to the capabilities and, apparently, the tastes of the individual."

However the underlying question - of whether 'mate selection behaviour in animals entails making a judgement analogous to aesthetic judgements in humans' - is perhaps (pace Nagel's famous discussion in Philosophical review (1974) of 'What it is like to be a bat?'), a question whose answer can never be known.

In contrast the role of education (or training) in recognising 'good' and 'bad', 'creative' and 'non-creative' has been more experimentally probed. A suggestive study investigating this topic by Watanabe [35], gathers a set of children's paintings which adult humans are asked to label 'good' or 'bad'. Pigeons are subsequently trained

[8] To time its art, to art its freedom.

through operant conditioning to only peck at good paintings. After the training, when pigeons are exposed to a novel set of [judged] children's paintings, they show their ability in the correct classification of the paintings; emphasising the role of training in aesthetic judgement and opening the door to computational (machine learning) explorations in this area[9].

A further area relating swarm intelligence and creativity is that of social, distributed and extended systems. For example Bown in [20] argues that our creative capabilities are contingent on the objects and infrastructure available to us, which help us achieve individual goals, in two ways:

> *"One way to look at this is, as Clark does [9], in terms of the mind being extended to a distributed system with an embodied brain at the centre, and surrounded by various other tools, from digits to digital computers. Another way is to step away from the centrality of human brains altogether and consider social complexes as distributed systems involving more or less cognitive elements."*

7.2 On the Emotional Sensitivity of Computer Artists

Can a computer program be sensitive to real emotion is directing its artistic output? Certainly Simon Colton's work at Imperial College suggests this may be so. Simon describes his 'Painting Fool' as follows, "Firstly, we used software developed by Maja Pantic, Michel Valstar and other members of the vision group at Imperial to take a video sequence of someone expressing an emotion (such as smiling, frowning, looking surprised, etc.). The software then: detected the emotion; determined where the features of the face were; and found the image in the video sequence where the emotion was being expressed the most. This information was then passed to the second piece of software in the combination, namely The Painting Fool, which proceeded to paint a portrait of the person in the video sequence. It based the portrait on the image provided from the emotional modeling software, and chose its art materials, colour palette and abstraction level according to the emotion being expressed. For instance, if it was told that the person was expressing happiness, it chose vibrant colours, and painted in simulated acrylic paints in a slapdash way. If, on the other hand, it was told that the person was sad, it chose to paint with pastels in muted colours." Such behaviour clearly suggests at least some sensitivity to [human] emotion is possible in computational systems.

7.3 Fish: Real-World Swarm Art!

An example of the use of real-word swarms in computer art come from the artist, Julie Freeman[10]. In 2005 Julie completed a site installation 'Swarm Intelligence' art work at Tingrith Fisheries (a 4000 square meter lake bordering the Woburn Abbey Estate). For the artwork - The Lake - Julie implanted 16 fish (four each of four species) with electronic transducers that could be tracked in real time 24/7 by 6 audio transponders and their real-time movements used to develop electronic soundscape and concomitant computer generated images; different behaviours were initiated by fish schooling (swarming) and by individual forays through the lake. This work is very successful and has been extensively installed and exhibited internationally[11]. The success of this work by Freeman clearly demonstrates that there is at least one niche for the [real-world] swarm aesthetic in art.

[9] This also raises the question of the degree to which humans are trained (or 'biased') to distinguish good and/or creative work.

[10] Artist in Residence at the Microsystems & Nanotechnology Centre, Cranfield University and Associate Artist, Goldsmiths Digital Studios

8 Conclusion

In this paper, we make no strong claim about the 'computational creativity' of the work presented, neither do we try to tackle the infamous question on whether computers can be creative at all or generate creative art. This specific work described herein merely emphasises the importance of 'controlled freedom' in the production of 'drawings' by computer. The computational artist so described is the outcome of a novel marriage between two classical swarm intelligence algorithms (PSO and SDS)[12] and a simplified mechanism of muscle activation. In an ongoing research, the application of the new hybrid algorithm to make a 'swarmic' drawing 'as though through a human's gaze' is currently being investigated.

ACKNOWLEDGEMENTS

We would like to thank the reviewers for their comments which helped improve this paper.

REFERENCES

[1] Mohammad Majid al-Rifaie and Mark Bishop, 'The mining game: a brief introduction to the stochastic diffusion search metaheuristic', *AISB Quarterly*, (2010).

[2] Mohammad Majid al-Rifaie, Mark Bishop, and Tim Blackwell, 'An investigation into the merger of stochastic diffusion search and particle swarm optimisation', (2011). In press.

[3] S. Aupetit, V. Bordeau, N. Monmarche, M. Slimane, and G. Venturini, 'Interactive evolution of ant paintings', in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 2, pp. 1376–1383, (2004).

[4] O. Burchan Bayazit, Jyh-Ming Lien, and Nancy M. Amato, 'Roadmap-based flocking for complex environments', in *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, p. 104, Washington, DC, USA, (2002). IEEE Computer Society.

[5] J.M. Bishop, 'Stochastic searching networks', pp. 329–331, London, UK, (1989). Proc. 1st IEE Conf. on Artificial Neural Networks.

[6] M.A. Boden, 'Creativity in a nutshell', *Think*, 5(15), 83–96, (2007).

[7] M.A. Boden, *Creativity and Art: Three Roads to Surprise*, Oxford University Press, 2010.

[8] Gerald Borgia, 'Complex male display and female choice in the spotted bowerbird: specialized functions for different bower decorations', *Animal Behaviour*, 49, 1291–1301, (1995).

[9] A. Clark, *Natural-born cyborgs: Minds, technologies, and the future of human intelligence*, Oxford University Press, 2003.

[10] S. Cohen-Cory, 'The developing synapse: Construction and', *Science*, 1075510(770), 298, (2002).

[11] K. de Meyer, J. M. Bishop, and S. J. Nasuto, 'Stochastic diffusion: Using recruitment for search', *Evolvability and interaction: evolutionary substrates of communication, signalling, and perception in the dynamics of social complexity (ed. P. McOwan, K. Dautenhahn & CL Nehaniv) Technical Report*, 393, 60–65, (2003).

[12] R.C. Eberhart and J. Kennedy, 'A new optimizer using particle swarm theory', in *Proceedings of the sixth international symposium on micro machine and human science*, volume 43. New York, NY, USA: IEEE, (1995).

[11] The work has also been shown in the Truman Brewery, London, UK; FILE, Sao Paulo, Brazil; FILE, Rio, Brazil; Arts Bioethics Network, Rijeka, Croatia; The National Centre for Contemporary Arts, Kaliningrad, Russia and in lastly in 2009 at MediaArtLab Center for Art and Culture, Moscow, Russia

[12] The scientific value of the new hybrid algorithm is currently being investigated via standard optimisation benchmarks [2].

[13] A. Etzioni, A. Ben-Barak, S. Peron, and A. Durandy, 'Ataxia-telangiectasia in twins presenting as autosomal recessive hyper-immunoglobulin m syndrome', *IMAJ-RAMAT GAN-*, 9(5), 406, (2007).
[14] G. Greenfield, 'Evolutionary methods for ant colony paintings', *APPLICATIONS OF EVOLUTIONARY COMPUTING, PROCEEDINGS*, 3449, 478–487, (2005).
[15] M. D Henry and K. P Campbell, 'Dystroglycan: an extracellular matrix receptor linked to the cytoskeleton', *Current opinion in cell biology*, 8(5), 625631, (1996).
[16] Charles H. Janson, 'Experimental evidence for spatial memory in foraging wild capuchin monkeys,cebus apella', *Animal Behaviour*, 55, 1229–1243, (1998).
[17] J. Kennedy and R. C. Eberhart, 'Particle swarm optimization', in *Proceedings of the IEEE International Conference on Neural Networks*, volume IV, pp. 1942–1948, Piscataway, NJ, (1995). IEEE Service Center.
[18] James F. Kennedy, Russell C. Eberhart, and Yuhui Shi, *Swarm intelligence*, Morgan Kaufmann Publishers, San Francisco ; London, 2001.
[19] M.J. Mataric, *Interaction and Intelligent Behavior*, Ph.D. dissertation, Department of Electrical, Electronics and Computer Engineering, MIT, USA, 1994.
[20] Jon McCormack and Mark d'Inverno (eds), *Computers and Creativity*, Springer, Berlin, 2011.
[21] N. Monmarch, S. Aupetit, V. Bordeau, M. Slimane, and G Venturini, 'Interactive evolution of ant paintings', in *2003 Congress on Evolutionary Computation*, ed., B. McKay et al, volume 2, pp. 1376–1383. IEEE Press, (2003).
[22] L. Moura and V. Ramos, 'Swarm paintings–nonhuman art', *ARCHITOPIA book, art, architecture and science*, 5–24, (2007).
[23] D. R. Myatt, J. M. Bishop, and S. J. Nasuto, 'Minimum stable convergence criteria for stochastic diffusion search', *Electronics Letters*, 40(2), 112–113, (2004).
[24] S. J. Nasuto, *Resource Allocation Analysis of the Stochastic Diffusion Search*, Ph.D. dissertation, PhD Thesis, University of Reading, Reading, UK, 1999.
[25] S. J. Nasuto and J. M. Bishop, 'Convergence analysis of stochastic diffusion search', *Parallel Algorithms and Applications*, 14(2), (1999).
[26] S. J. Nasuto, J. M. Bishop, and S. Lauria, 'Time complexity of stochastic diffusion search', *Neural Computation*, NC98, (1998).
[27] Craig W. Reynolds, 'Flocks, herds, and schools: A distributed behavioral model', *Computer Graphics*, 21(4), 25–34, (1987).
[28] S. Rigoard, M. Wager, K. Buffenoir, S. Bauche, J. P. Giot, J. M. Maixent, and P. Rigoard, 'Major mechanisms involved in the synaptic transmission of the neuromuscular apparatus', *Neuro-Chirurgie*, 55, S22, (2009).
[29] J. R Sanes and J. W Lichtman, 'Induction, assembly, maturation and maintenance of a postsynaptic apparatus', *Nature Reviews Neuroscience*, 2(11), 791–805, (2001).
[30] John Godfrey. Saxe, D.A. Lathen, and B. Chief, 'The Blind Man and the Elephant', *The Poems of John Godfrey Saxe*, (1882).
[31] T. C S\dhof, 'The synaptic vesicle cycle', *Neuroscience*, 27(1), 509, (2004).
[32] C.W. Taylor, '4 Various approaches to and definitions of creativity', *The nature of creativity: Contemporary psychological perspectives*, 99, (1988).
[33] P. Urbano, 'Playing in the pheromone playground: Experiences in swarm painting', *Applications on Evolutionary Computing*, 527–532, (2005).
[34] P. Urbano, 'Consensual paintings', *Applications of Evolutionary Computing*, 622–632, (2006).
[35] Shigeru Watanabe, 'Pigeons can discriminate good and bad paintings by children', *Animal Cognition*, 13(1), (2009).
[36] Aum-Mon Weesatchanam, *Are Paintings by Elephants Really Art?*, The Elephant Art Gallery, 31 July 2006.
[37] S. J Wood et al., 'Safety factor at the neuromuscular junction', *Progress in neurobiology*, 64(4), 393–429, (2001).

Object-oriented philosophy – the nature of the relations between humans and computational objects

Leighton Evans[1]

Abstract. I argue that the category of equipment denoted computational objects have, by virtue of the unique presence of those objects in the world as permanently withdrawn from full disclosure of operation due to their dependence on computational code, a unique manner of causal interaction with users that can only be described as vicarious. As computational devices become increasingly ubiquitous as tools for managing and navigation the human world, this vicarious relationship becomes important in understanding how this technology affects the phenomenological experience of being in the world as it is, alongside computational objects, and how orientation towards the world can be described as computational.

1 INTRODUCTION

This paper develops a philosophical approach to understanding how computational technologies co-exist with humans, and what affect these technologies have on human orientation in the world. Following from Heidegger, here the *world* is understood as a rooted through a phenomenological space consisting of semantic relations between entities in which Dasein dwells, and *technology* is understood as a factor which influences the manner in which people are oriented (and therefore understand) that space. In other words, the semantic relations are to some extent strengthened or weakened by the action of technology that serves to crystallise certain meanings.

In this paper I will firstly develop a working definition for *computational devices*, then consider specifically the concreteness of a computational device through the object-oriented philosophy of Graham Harman. In doing this I will consider the co-constituting structure of computational devices on being-in-the-world.

The underlying argument is that computational devices project a processual agency of computational code. However they are radically opaque, being *unready-to-hand* without being open to inspection [1]. This implies that the associations that they form with the world can potentially become a powerful force in constructing the world experienced by users. The way-of-being inaugurated by equipment (the computational devices) therefore alters the phenomenological experience of the world in which the user is coping through that particular computational device, and in the context of GPS and Location Based Services (LBS), particularly through a the continual computational mapping and will-to-map.

[1] Department of Political and Cultural Studies, Swansea University, SA2 8PP, email: l.evans@swansea.ac.uk

2 THE OBJECT ITSELF: THE ONTOLOGY OF COMPUTATIONAL OBJECTS

Computational objects or devices are quite clearly an advanced technology. Following Feenberg's [2] assertion that technology is a hermeneutic construction, being a mix of technological and social factors and developed contingently rather than deterministically, the view of technology expressed here will be that technology explicitly does not stand apart from people and society but is instead something that shapes and is shaped by the conditions in which it is embedded in the world. Technological design [3] is influenced and guided by social processes, and these processes are in turn about fulfilling social needs that are culturally defined. Following Heidegger's use of *world* in *Being and Time*, the world is the place of interaction and meaning for humans based on the associations that humans make with all other entities in that place, the social is the place where interaction takes place (*where Dasein dwells*). Therefore technology is both a product of being-in-the-world and shapes that being-in-the-world – as the discussion of Enframing in *The Question Concerning Technology* argued. Technology for Feenberg [4] provides the material framework of society – the ideals and ideologies of society are embedded in and are reinforced by the technologies which that society uses (see also Winner (1986)) and these technologies are also shaped by the society in which they emerge, forming a hermeneutic circle. Following this idea of technology as the genesis of a world [5], in the Heideggerian sense technological devices exist in the world and are interacted with. These interactions reveal aspects of the technology (while other aspects remain withdrawn) and the character of a world is built from the revealing of the technology within that dwelling. Technology is a mode of revealing [6], and it is this that is the social context of technology –but technology is also material, in-the-world, and something that is tangibly interacted with, rather than something that affects the world from outside or above the world itself.

The technological device itself deserves closer scrutiny. Sterling [7] argues that technological artefacts are objects made by hand and powered by muscle, but that machines (or devices) are complex, precise artefacts with a non-human power source. This, of course, introduces a new source of agency in technical devices. Sterling also identifies different ages, or techno culture of machines – in the gizmo techno culture (from 1989 to 2004) [8] the user is the end-user that must think and talk about the technology being used and the effects of that technology.

This technology is exemplified by second-generation mobile phones, non-networked PC's and televisions *broadcasting one way*. In the techno culture of the "spime" (from 2004, when RFID tags were first added to US defence equipment, another example of the military leading technological development, [9]),

exemplified by devices that have metrics that allow for monitoring and feedback loops within the device itself, the cognitive load and opportunity costs that came with the gizmo techno culture are lifted; everything is done for me by the device itself. The structuring form of these computational devices is expanded upon by Berry [10], who argues that as we live in the midst of technological devices, these devices become embedded in our way-of-being – that is, a computational mode of being emerges, which informs the understanding of the world that we have in this milieu.

> When viewing the world computationally, one is comported towards the world in a way in which it has already been mapped and classified through digital devices (Berry p.143)

Berry is arguing that the computational dimension is now a given – a part of the equipmental background to being where life experiences become procedural chains of digital information that are stored in databanks (p.149). The referential totality (which is the world) represented by entities around us is increasingly populated with actors enabled with computational techniques and abilities, and it is through this continual exposure to computation that a computational knowing-that (a presence-at-hand or *vorhandenheit)* emerges socially. The computational image is therefore a comportment towards the world that takes as subject-matter the manifest entities which it can transform through calculation and processing interventions (p.143), potentially resulting in a form of cognitive support which uses computational processes of reasoning. This, I argue, represents a new weltbilt that is emerging – and the remainder of this paper will develop an ontology of the computational device that will aim to extrapolate and expand on this foundation.

3 TOOL-BEING AND THE UN-ZUHANDENHEIT

Harman argues that the traditional ontological divide between humans and the world is mistaken – and that the key ontological divide that should be investigated is the one between an object and its relations [11]. This idea results in a new idea of substance that is irreducible to the physical, and instead concentrates on a key strife between the presence and being of an object. Harman is radical in his assertion that the key aspect of all things is readiness-to-hand (zuhandenheit), and that it is this readiness-to-hand that is the "tool being" which all things – objects, humans and all other things in the world – unavoidably are ontologically (p.3). In this ontological theory, the nature of all things is to recede – even the inanimate recedes, but in particular humans recede. This receding refers to how aspects of being are revealed and recede from appearance (p.4). This is a speculative form of philosophy that claims that through a radical attempt to think about the natural world. This has been christened the 'Speculative Turn' whereby philosophy returns to a pre-Kantian position of considering the thing-in-itself rejecting the claim of Kant that:

> Let us once try whether we do no get further with the problem of metaphysics by assuming that objects conform to our cognition, which would agree better with the requested possibility of an a priori cognition of them, which is to establish something about objects before they are given to us (Kant [12])

Harman argues, following Heidegger, that it is possible to separates the ontic and ontological by using the ontic to refer to what pertains to be present at hand, and not pertaining to the object that is the preserve of the ontological level of analysis (p.5). Harman's analysis of Heidegger's notion of *zeug* leads to the conclusion that equipment is always in reality – but withdrawn in order to be ready-to-hand (p.22). As was argued previously, all equipment is part of a totality i.e. in context with other tools in the world, creating a global tool empire – a bridge is not alone, it exists in a network with all the other tools that constitute the highway and traffic that gives that bridge its meaning (p.23). The totality means each tool occupies a space in the system of forces that makes up the world, and therefore the totality if equipment is the world (or at least the world of meaningful entities that we derive meaning from and in which we dwell). As Harman states "the world of tools is an invisible realm from which the visible structure of the world emerges" (p.23). This leaves a paradoxical state where equipment is always hidden, but is always there (through the totality, and this paradox can be resolved as referentiality (p.25), in that while equipment is always present it is only meaningful once revealed to us through use or reference.

For Heidegger, the primary towards which is the for-the-sake-of-which, which means that the collection of logs stacked together becomes a shelter only through the presence of humans to interpret and use that structure. Harman argues that human eyes are unnecessary for this – it is through the totality of *zeug* that the meaning of the structure emerges, not the interpretation of the arrangement by humans (p.29). This indicates a de-privileging of Dasein in Harman's philosophy; Dasein no longer being the sole arbiter of meaning in the world through its care and concern (p.40). Instead, Dasein is privileged by being able to formulate the question to being, not by being the provider of the answer. To be is not the same as to understand being – but you cannot understand being without being in the first instance (p42).

Zeug operates in its inconspicuous usefulness – when the tool fails to operate it appears to us as it is, as a broken tool. When broken, the tool is freed from the dimension of reference and allows space and the physical characteristics to be seen. This luminescence and the darkness of the withdrawal of the object are for Harman part of the same "as" structure of all things in the world – things exist as both withdrawn and revealing (p.225). Therefore, whether a tool is broken or not has nothing to do with tool being – the status of the tool is ontic, while the thing in itself existing as tool is the ontological truth of entities (p.225). This "as" structure sits above tool being – it is not a part of the tool, but is a means of describing things above that of the fundamental ontology. This harks back to the fundamental difference between objects and their relations – the object exists as a tool, the relations it has with other things is a function of the appearance and withdrawal of the object in the world.

For Harman then, all things in the world exist in a system or network which is the sum of relations that those things have with other things in the world as a function of the revealing and withdrawal of the being of that entity. As Heidegger argued, the nature of being is too complex to be revealed in its entirety at any time and is only grasped through its happening (*Erengis*)

which is a temporary event [13] and Harman takes this further, extending this from Dasein itself to all things in the world.

The root of Harman's object-oriented philosophy can therefore be summarised as a position that holds that objects exist with their qualities, accidents and relations, but are not reducible to these features of the object which are experienced by any other entity [14]. The object itself, as a thing-in-itself, remains beyond these ontic features, and beyond an all encompassing definition or even knowledge of that object itself. This definition has hence been grouped in the philosophical school of speculative realism, a collection of philosophical ideas that admit the reality of objects in the world but deny the possibility of an epistemological certainty about those objects [15, 16, 17]. This position grants objects "subterranean depths" [18] in the same manner that Heidegger afforded the being of Dasein such depth in *Being and Time*. Only some information about the object itself can be known at any given time, not a totality which would allow complete knowledge – and in this way Harman retains the scepticism of Heidegger about the ability of the physical sciences to offer a complete view of the world, preserving the world/universe distinction that Heidegger gave in *Being and Time* and reinforcing the importance of the relations that objects have in the world to other objects as the source of their meaning.

4 VICARIOUS CAUSATION

Harman's object-oriented philosophy may seem to hark back to problematic aspects of classic empiricist theories of epistemology. For Harman, contact between two things is always vicarious, as there is an indisputable 'autonomous reality of their components' [19]. So, to operate vicariously (and by extension with other things, which must be an unavoidable consequence of being-in-the-world) 'means that forms do not touch one another directly, but somehow melt, fuse, and decompress in a shared common space from which all are partly absent' [20]. There is no point of causal contact between two entities ever – due to withdrawal. As far as criticism goes, in Locke [21] for example, one cannot "know" the object as a thing in itself but only the primary and secondary qualities of this object, leading to a veil of perception between the mind and the world, and eventually to the radical scepticism of Hume on causation [22]. Another interpretation is to borrow from Leibniz' *Monadology* [23] where the phenomenal real is the ontic level of understanding and the object in itself takes on the status of Leibniz' monad or natural atom. Harman's insistence on the withdrawal of the object, and our inability to reach epistemological certainty about that object would appear to open itself to the criticisms used for the empiricist position, or an interpretation of the object as phenomenal and monadic following Leibniz. Especially given the duality in Harman's work between an objects and its relations, which would allow a line of argument that all we can know is the relations of an object and therefore, as an object is not its relations, we are left with only a knowledge of what the object is not, not what it is, forming another sceptical viewpoint. Harman certainly admits to the existence of sensual objects – is the same sense of the intentional object of Husserl [24] – but these sensual objects are not all that exists for Harman, they are just a consequence of the withdrawal of the object, which is a necessary feature of the object due to its ontology. The epistemological critique does not take into account firstly Harman's phenomenology – in that all things are in the world, necessarily so (as with Heidegger) and so the notion of the object "not being there" as we are only certain of its relations does not need to lead to the conclusion that we cannot have certainty of the object – we may not have epistemological certainty, but the philosophy demands an ontological certainty of both the relations and existence of the object in question.

Harman, to an extent, anticipates the problem of scepticism and offers a solution to that by offering a new method of describing the world and the objects that make up that world. Given that real objects in a sense withdraw behind sensual qualities, Harman [25] proposes a fourfold which will allow analysis of objects by examining the inherent tensions of the existence of the object as present and absent in the world. Initially, it should be noted that Harman argues that objects are not the sum of their qualities, nor are they an essential quality as that would be impossible to determine. The fourfold proposes that there are a number of ways of approaching an object in the world, based on what can be known about that object through the relations the object has to other things (or indeed the thing that is making the assessment) at that time. Therefore when real objects withdraw behind sensual qualities, there is a tension of time. Harman defines tensions of (i) *time* (real objects behind sensual objects), (ii) *space* (real objects behind real qualities) (iii) *essence* (sensual objects withdrawing behind real qualities) and (iv) *eidos* (sensual objects withdrawing behind sensual qualities). This fourfold acts to differentiate the real and the sensual and to provide a framework for the myriad of experience of objects that can be had when with them in the world.

Harman's difficulty of reconciling the ontology and ontic properties of objects in his philosophy harks back to Heidegger's fourfold of gods, mortals, earth, sky (of which Harman is particularly fond)[2] but requires a more fundamental analysis when considering the relationship that humans have with technical devices. In more broad strokes, his object-oriented philosophy admits that the contact with the object in the world can never be complete, as there is an aspect of that object's objectness which is always hidden or withdrawn. Harman's fourfold is a complex and philosophically somewhat confusing way of explaining the myriad relationships between objects in the world. By taking the basic tenet, a more parsimonious explanation of vicarious causation emerges as an explanation of how objects are with other objects.

The notion of vicarious causation is important here in explaining how the world is navigated when the world is made up of objects in a totality of reference, when these objects are only partially known to one another. The fourfold above illustrates the ways in which objects are always withdrawn – but humans and other things still operate in the world seemingly unaware or unconcerned about this withdrawal through the presence of the sensual element.[3] In our everydayness, we do not stop to consider the ontological status of the tool that we are using (except perhaps when broken and in deep contemplation) and accept that this tool is a tool that is being used or not used at that time. It is not my intention to jettison Harman's metaphysic, as the notion of withdrawal is useful for devices or tools that

[2] See Building, Dwelling, Thinking, in Basic Writings (2008) [26].
[3] This differs from Meillassoux's (2008) [17] explicit attempt to restore Locke's differentiation between primary and secondary qualities as a way of explaining the status of the object in the world – in turn this ironically restores the distinction of subject and object.

perform operations on the ontic level but operate through code at the ontological or withdrawn level. What must be made clear is how these tools come to exert influence despite the ontological facticity of withdrawal.

It should now be clear that speculative realism involves a tension between the real and the interpreted – an admission of the existence of the totality of an object with an addendum that experience does not give access to this totality. Berry (p.153) explains this as speculative realism having the manifest image being not fully correspondent to the scientific image, and while I hold this correct, I would add that the manifest image is not just incongruent with the scientific image, but also with the ontological truth of the object. Speculative realism therefore offers a position between the phenomenal or manifest and the level of totality or truth; a way of reconciling these differences. Thus, speculative realism gives a position of 'vicarious contact' with the objects that make up the world – and the world that is produced through objects and the relations with objects that make up the 'picture of the world' is also vicarious.

In order to clarify this position, the world as overflowing with computational devices is one where that device cannot be known with certainty due to the necessary withdrawal of that device – as illustrated through a computational device that operates through the execution of code without our circumspection – but is also one where meaning derives from the referential totality that these devices contribute towards and are an irrevocable part of creating. There emerges a tension between the fact that knowledge of the device is vicarious, mediated through that which is revealed through the interface, and how the device is integral in presenting the world to us as a part of the referential totality of that world. Hence, there is a vicarious causation (Berry, p.153) that encapsulates the way the world is presented through devices – the means of understanding the world (from the referential totality) is necessarily vicarious through the way that those devices are continually emerging and withdrawing into the clearing of understanding that is the referential totality. In the specific context of the computational device, which has an internal hidden state that is the code which operates in a withdrawn manner from the user, there is a clear indication that the relationship between the user and the device is a purely vicarious one as the inner operations (and the operations of the device per se) are necessarily and continually withdrawn from the user.

Harman's position is useful to understand this relationship in the way that Harman positions the human actor and other objects in the world. Humans are never directly in contact with the world that consists of computational devices (and following Harman, neither is any other object in the world) as all the translations of the world that are carries out by those devices are in themselves necessarily vicarious. The devices work beyond our control through the execution of code – and this is something necessarily withdrawn in our everyday interactions with the device, marking it as radically un-ready-to-hand. Berry (p.139) maintains that technological or computational devices are alone in that they have the feature of never fully withdrawing from the world – through their continuous operation and it is this feature that makes such devices as un-ready-to-hand, rather than present-at-hand or ready-to-hand.[4]

A question can be raised as to why are computational devices any different from other complex machinery, such as a car engine. It is perfectly perceivable, and in my view correct, that a car engine can withdraw in the same manner as a computational device – indeed I hold that this must be so; in the case of a fully operational engine it is withdrawn. When broken it is brought into circumspection by its conspicuousness. When one encounters a broken engine, we attempt to understand that engine and fix it – opening the bonnet, a cursory confused stare, possibly an adjustment to the machinery if we are skilled at this, and possible success. The broken engine, in its state of being a broken tool, invites understanding. Once it is working, it works – and we move away from considering that tool, as it withdraws from our circumspection.

I would add that even when broken, the engine is not revealed in its totality – for a start, it is not revealing its operational state. However, in an operational state we do not think about the tool – a well-engineered and operational tool is withdrawn from circumspection. When broken, our circumspection is drawn to the object. In comparing this with a computational device, whether operational or broken, they are always withdrawn. When operational, we do not consider the internal machinations of code – and when broken, we do not consider this either. We switch devices on and off, we ring tech support that offers more sophisticated methods of resetting programs and devices, but nevertheless offers the same solution. We uninstall and reinstall – the same method, a different process. We are prevented with engaging with the object in the same manner as a car engine, by the nature of the operation of the computational object. Accordingly, as car engines have become more dependent upon computation, the layperson's ability to interpret the problem on operation and offer solutions or remedies has been elided (as often has the mechanic's). There is not the same level of understanding of the computational device – if I was to try and fix an iPhone by opening it up a la car engine, I would both invalidate the warranty and be faced with chips and processors that offered no explanation of the computational code upon which the device is dependent. What is important here is how the world itself is presented back to the user through the device. The complex machine offers a view of technology as fixable – if it is broken, one can engage an expert or learn skills oneself that will remedy this state. The computational device, when broken, is beyond this comprehension – we hope resetting it will fix the problem, and if not we are stuck, as the operations of code are beyond our comprehension. As the world is translated back to us through these devices, always un-ready-to-hand and present-at hand simultaneously, a new way of being-towards the world from this comportment towards the world of the device emerges. We are not in contact with the computational device in the same way as the complex machine.

5 COMPUTATIONAL VICARIOUSNESS

To clarify this further, consider this example. Using a GPS navigation device such as Skobbler, a person can plot a route

[4] I would personally go further than Harman and Berry and suggest that all things, while in a state of withdrawal and revealing in the world, are unready to hand and hence experience is always vicarious.

from Hyde Park Corner in London to Princess Street in Edinburgh. The information is inputted into the device, and a route is calculated through the execution of the computational code that is the basis of the application. A series of instructions are then produced, which the application presents back to the user through auditory commands and a visual interface that guides the driver from the starting point to the destination from the information held in the databank of the application. What must be considered closely is what has been done to the process of navigation in this case. One can consider a time, pre-mechanisation, when using a series of instructions passed verbally, an intrepid individual could have ridden horseback between the two points using key landmarks to navigate the route successfully. The individual (or individual and horse) would have been in a situation where the route would have been a series of movements through relations between landmarks and the individual based on the mimesis of the physical environment provided by the instructions, traditionally what navigation was [27]. Consider now the GPS navigation; the physical world is translated into a database of instructions and distances, and interpreted by the application into a route to follow. The user is not complicit with the database, nor are they writing the code that determines the route to be followed – these computations are withdrawn within the device, hidden in operation and appearance from the user. The application then presents the world back to the user in a mediated form – a manifest image of the space being navigated that will not appreciably demonstrate equivalence with the actual space – which is used to navigate the route from London to Edinburgh.

This example of everyday navigation in the 21st century illustrates that the device gives us our worldview through its operations, but at the same time the operations of the device are detached, hidden and unobtainable to our conspicuousness – what would once be something dwelt upon (particularly in a case of equipment break-down) is now accepted and incorporated into our comportment towards the world due to the withdrawal of the device. Harman would argue that it was always thus – and it is not within the scope of this paper to question this further – but what is important here is how the computational device achieves this, to borrow from Latour a transformation from mimesis to navigation through computation. Berry (p.152) uses the term computational image to describe the process, with the computational image being the cultural technique used to select, store, process and produce the data for the process of computation from the world, and which is then presented back to the user. This computational image then becomes the comportment towards the world that takes as subject matter manifest entities that it can transform through calculations and processing interventions (Berry, p152). This process must necessarily involve a translation from the physical to the computational, and it is to how those translations can be explained and how translations create the important position of computational devices in the world that is the next logical step in this process. To conclude this discussion though, a reiteration of the main point is necessary. We do not experience or observe cartographic reason when using GPS computational devices to navigate the world, as cartographic reason is a form of instrumental reasoning – what is here is a new type of reasoning, computational reasoning, which has people and entities in the world linked vicariously or relationally rather than in cartographic or instrumental relationships.

6 CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

Appropriating Heidegger's concepts of world and worldview we can see that the world is not the physical space that is occupied by objects, but is instead the emergent phenomenological space created from understanding and interaction with those things found in the world alongside us – as we are necessarily in the world. The existential locales that make up the series of worlds of meaning that we find ourselves in are different in that they are characterised differently by the relations we have with objects in the world. These objects therefore give us the sense of the world that we are in, and in a locale with technological objects will give a sense of a technological world – a technological disclosure of being that Heidegger called *Enframing*[28].

Heidegger's problems with that technological mode of revealing have been covered extensively by many in the Philosophy of Technology, but what is most important from those discussions for this work is how the world disclosure shapes the interactions that people have with all things, not just technology. The technological mode of revealing primes the individual to treat all things in a particular way – it comports us towards the world. What Heidegger's approach lacks, and this is a criticism made of Heidegger's take on technology by many commentators (Haraway [29], Idhe [30], Rorty [31] and many others) – is that it seals with all technological (or modern technological) things in the world as a singular category. While there is much to be gained from this as Heidegger's writings show, there is also a need for a finer analysis of the technology itself, and in particular technology as objects and not just category. Harman's object-oriented philosophy is important in returning the object to the centre of analysis and reasserting the role of realism as a philosophical doctrine into the discussion of technology alongside phenomenology. Heidegger would not doubt that technological devices are real – but by dealing with all modern technology under a categorical definition rather than looking at the specifics of that device itself, a disservice is done, especially if one is proposing a new world disclosure due to technological developments. The contention for the philosophy of technology is therefore that computational objects, by being unready-to-hand (as argued by Berry, and as inferred from Harman's "subterranean depths" that he ascribes to objects) require an analysis of objects rather than the category of technology. Harman's object-oriented philosophy, with the notion of vicarious causation and the retention of the idea of objects framing the view of the world we have, fits this requirement, allied to Berry's argument that the computational device processes and re-represents the world to us in computational terms, therefore revealing a computational mode of being (a knowing-that) from the knowing-how of using computational devices.

What is left to consider, and to be researched, is how users themselves experience and consider their interactions with these permanently withdrawn objects, and how this shapes and influences their experiences of the world. This is the focus of my current work – suffice to say I predict the effects will be telling.

REFERENCES

[1] Berry, D. M. (2011). *The Philosophy of Software: Code and Mediation in the Digital Age*. London: Palgrave/Macmillan.
[2] Feenberg, A. (1999). *Questioning Technology* (1 ed.). New York: Routledge, p.5.
[3] Feenberg, A. (1999). *Questioning Technology* (1 ed.). New York: Routledge, p.83.
[4] Feenberg, A. (2002). *Transforming Technology: A Critical Theory Revisited* (2 ed.). New York: Oxford University Press, USA, p.18.
[5] Feenberg, A. (2002). *Transforming Technology: A Critical Theory Revisited* (2 ed.). New York: Oxford University Press, USA, p.18.
[6] Malpas, J., & Wrathall, M. (2000). *Heidegger, Authenticity, and Modernity/Heidegger, Coping, and Cognitive Science. Essays in Honor of Hubert L. Dreyfus, Volumes 1 and 2*. Michigan: MIT Press, p.206.
[7] Sterling, B. (2005). *Shaping Things (Mediaworks Pamphlets)*. London: The MIT Press, p.3
[8] Sterling, B. (2005). *Shaping Things (Mediaworks Pamphlets)*. London: The MIT Press, pp. 21-22.
[9] Sterling, B. (2005). *Shaping Things (Mediaworks Pamphlets)*. London: The MIT Press, p.12.
[10] Berry, D. M. (2011). *The Philosophy of Software: Code and Mediation in the Digital Age*. London: Palgrave/Macmillan, p.146.
[11] Harman, G. (2002). *Tool-Being: Heidegger and the Metaphysics of Objects*. London: Open Court, p.2.
[12] Kant, I. (1999). *The Critique of Pure Reason* (translated by Guyer, P.). Cambridge: Cambridge University Press, p.110
[13] Sacchi, M. E. (2002). *The Apocalypse of Being: The Esoteric Gnosis of Martin Heidegger (trans. G. X. Martinez)*. South Bend, Indiana: St. Augustine's Press, p.79.
[14] Harman, G. (2009). *Prince of Networks: Bruno Latour and Metaphysics (Anamnesis)*. Albany: Re.Press, p.156.
[15] Brassier, R. (2007). *Nihil Unbound: Enlightenment and Extinction*. London: Palgrave/Macmillan.
[16] Grant, I. H. (2006). *Philosophies of Nature after Schelling*. London and New York: Continuum.
[17] Meillassoux, Q. (2008). *After Finitude: An essay on the necessity of contingency (Trans. Brassier, R.)*. London: Continuum.
[18] Harman, G. (2009). *Prince of Networks: Bruno Latour and Metaphysics (Anamnesis)*. Albany: Re.Press, p.193.
[19] Harman, G. (2009). *Prince of Networks: Bruno Latour and Metaphysics (Anamnesis)*. Albany: Re.Press, p141.
[20] Harman, G. (2009). *Prince of Networks: Bruno Latour and Metaphysics (Anamnesis)*. Albany: Re.Press, p.142.
[21] Locke, J. (1979). *An Essay Concerning Human Understanding (Clarendon Edition of the Works of John Locke)* (New Ed.). New York: Oxford University Press, USA.
[22] Hume, D. (2010). *An Enquiry Concerning Human Understanding*. New York: General Books.
[23] Leibniz, G. W. (2007). *Leibniz: The Monadology And Other Philosophical Writings*. New York: Kessinger Publishing.
[24] Husserl, E. (2001). *Logical Investigations (International Library of Philosophy)*. New York: Routledge.
[25] Harman, G. (2009). *Prince of Networks: Bruno Latour and Metaphysics (Anamnesis)*. Albany: Re.Press, pp.216-221.
[26] Heidegger, M. (2008). *Basic Writings*. London: Routledge. Pp. 246-247.
[27] Latour, B., November, V., & Camacho-Hubner, E. (2010). Entering a risky territory: space in the age of digital navigation. *Environment and Planning*, *28*(4), 581-591, p.581.
[28] Heidegger, M. (1978). *The Question Concerning Technology, and Other Essays*. New York: Harper Perennial.
[29] Haraway, D. J. (1997). enlightenment@science_wars.com: A Personal Reflection of Love and War. *Social Wars 50*, *15*(1), 123-129.
[30] Ihde, D. (1991). *Instrumental Realism (The Indiana Series in the Philosophy of Technology)*. Bloomington: Indiana University Press.
[31] Rorty, R. (2005) in *Making Things Public: Atmospheres of Democracy*. London: The MIT Press.

Multiple Realization and the Computational Mind

Paul Schweizer[1]

Abstract. The paper examines some central issues concerning the Computational Theory of Mind (CTM) and the notion of instantiating a computational formalism in the physical world. I address a standard line of criticism of CTM, based on the claim that the notion of instantiating a computational formalism is overly liberal to the point of vacuity, and conclude that Searle's view that computation is not an intrinsic property of physical systems is ultimately correct. I argue that for interesting and powerful cases, realization is only ever a matter of approximation and degree, and interpreting a physical device as performing a computation is relative to our purposes and potential epistemic gains. However, while this may fatally undermine a computational explanation of conscious experience, I contend that, contra Putnam and Searle, it does not rule out the possibility of a scientifically justified account of propositional attitude states in computational terms.

1 FORMALISM AND ARTEFACT

From an abstract mathematical perspective, computation comprises an extremely well defined and stable phenomenon. Central to the theory of computation is the intuitive notion of an effective or 'mechanical' procedure, which is simply a finite set of instructions for syntactic manipulations that can be followed by a machine, or by a human being who is capable of carrying out only very elementary operations on symbols. A key constraint is that the machine or the human can follow the rules without knowing what the symbols *mean*. The notion of an effective procedure is obviously quite general – it doesn't specify what form the instructions should take, what the manipulated symbols should look like, nor precisely what manipulations are involved. The underlying restriction is simply that they are finitary and can proceed 'mindlessly' i.e. without any additional interpretation or understanding. So there are any number of different possible frameworks for filling in the details and making the notion rigorous and precise. Turing's 'automatic computing machines' [1] (TMs), supply a very intuitive and elegant rendition of the notion of an effective procedure. But there is a variety of alternative frameworks, including Church's Lambda Calculus, Gödel's Recursive Function Theory, Lambek's Infinite Abacus Machines, etc.

According to the widely accepted Church-Turing thesis, the class of computable functions is captured in an absolute sense by the notion of TM computability, and compelling 'inductive evidence' for the thesis is supplied by the fact that every alternative formalization so far given of the broad intuitive notion of an effective procedure has been demonstrated to be equivalently powerful, and hence to specify exactly the same class of functions [2]. Thus the idealized notion of in-principle computability, where all finite bounds on input size, storage capacity and length of running time are abstracted away, seems to constitute a fundamental category, a stable and highly pleasing 'mathematical kind'.

A related further question to ask is whether any sort of comparable feature carries over to computation as implemented or realized in the physical universe. Turing machines and other types of computational formalisms are *mathematical abstractions*. Like equations, sets, Euclid's perfectly straight lines, etc., TMs don't exist in real time or space, and they have no causal powers. In order to perform *actual* computations, an abstract Turing machine, thought of as a formal program of instructions, must be realized or instantiated by a suitable arrangement of matter and energy. And as Turing observed long ago [3], there is no privileged or unique way to do this. Like other abstract structures, such as chess games and isosceles triangles, Turing machines are *multiply realizable* - what unites different types of physical implementation of the same abstract TM is nothing that they have in common as physical systems, but rather a structural isomorphism in terms of a particular level of description. Hence it's possible to implement the very same computational formalism using modern electronic circuitry, a human being executing the instructions by hand with paper and pencil, a Victorian system of gears and levers, as well as more atypical arrangements of matter and energy including toilet paper and beer cans. Let us call this 'downward' multiple realizability, wherein, for any given formal procedure, this *same* abstract computational formalism can be implemented via an arbitrary number of *distinct* physical systems. And let us denote this type of downward multiple realizability as '↓MR'.

After the essential foundations of the mathematical theory of computation were laid, the vital issue then became one of engineering – how best to utilize state of the art technology to construct rapid and powerful physical implementations of the abstract mathematical blueprints, and hence perform actual high speed computations *automatically*. This is a clear and deliberate ↓MR endeavour, involving the intentional construction of artefacts, painstakingly designed to follow the algorithms that we have created. From this top-down perspective, there is an obvious and pragmatically indispensible sense in which the hardware that we have designed and built can be said to perform genuine computations in physical space-time.

2 COMPUTATION IN NATURE

In addition to these comparatively recent engineering achievements, but presumably still members of a single underlying category of phenomenon, various authors and disciplines propound the notion of 'natural computation' (NC), and invoke a host of indigenous processes as cases in point, including neural computation, DNA computing, biological evolution, molecular and membrane computing, slime mould

[1] School of Informatics, Univ. of Edinburgh, EH8 9AD, UK. Email: paul@inf.ed.ac.uk.

growth, ant swarm optimization, 'embedded and pervasive computation', etc. According to such views, computation in the physical world is not merely artificial – it is not restricted to the devices specifically designed and constructed by human beings. Instead, computation is a seemingly ubiquitous feature of the natural order, and the artefacts invented by us constitute only a very small subset of the class of computational systems in the physical world.

The disciplinary and terminological practices surrounding NC invite a more thorough and rigorous examination of the underlying assumptions involved. To what extent is computation a genuine *natural* kind – is there any intrinsic unity or core of traits systematically held in common by the myriad of purported examples of computation in nature? This question has deep and independent conceptual significance, in an attempt to gain clarity on whether and to what extent computation can be cogently and fruitfully seen as a natural occurrence. In what sense, if any, can computation be said to take place spontaneously, as a truly native, 'bottom-up' phenomenon? And of course, the issue has special philosophical interest with respect to positions on the conjectured computational nature of *mentality and cognition*. It is this particular domain that will comprise the primary focal point of the paper, within the broader context just outlined.

3 THE COMPUTATIONAL THEORY OF MIND (CTM)

According to the widely embraced 'computational paradigm', which underpins cognitive science, Strong AI and various allied positions in the philosophy of mind, computation (of one sort or another) is held to provide the scientific key to explaining and artificially reproducing mentality. The paradigm maintains that cognitive processes are essentially computational processes, and hence that intelligence in the physical world arises when a material system implements the appropriate kind of computational formalism. In terms of the classical model of computation as rule governed symbol manipulation, the relation between the abstract program level and its realization in physical hardware then yields an elegant solution to the traditional mind-body problem in philosophy: the *mind* is to the *brain* as a *program* is to the *hardware* of a digital computer.

It's an immediate corollary of CTM that the human brain counts as an exemplary instance of NC. However, CTM seems to require a more robust and literal stand on computation than that embraced by NC in general. It is crucial to recognize the distinction (as pointed out by, e.g. Gualtiero Piccinini [4]) between being a system/process that can be effectively *simulated* or *modelled* using a computational formalism and being a system/process that *literally instantiates* a computational procedure or executes an algorithm. Most purported cases of 'natural computation' in a scientific context are versions of the former and not the latter. It is clear that the brain *can* be viewed as a case of NC in this simulational or modelling sense. However, I take it that serious proponents of CTM would advocate a more substantive position, *viz*., that human mentality arises because the brain literally instantiates computational procedures and transforms symbol structures in a manner comparable to a computational artefact rather than a computer simulated thunderstorm.

According to CTM, mental states and properties are seen as complex internal processing states, which computationally interact within a system of internal state transitions, thereby mediating the inputs and outputs of intelligent behaviour. Hence any mental process leading to an action will have to be embodied as a physical brain process that realizes the underlying computational formalism. A perceived virtue of this approach is that it can potentially provide a *universal* theory of cognition, a theory which is not limited by the details and peculiarities of the human organism. Since mentality is explained in computational terms, and, as above, computational formalisms are multiply realizable, it follows that the mind-program analogy can be applied to any number of different types of creatures and agents. Combining CTM with ↓MR, it follows that a human, a Martian and a robot could all be in exactly the *same* mental state, where this sameness is captured in terms of implementing the same cognitive computation, albeit via radically different forms of physical hardware. So on this view, computation is seen as providing the scientific paradigm for explaining mentality in general – all cognition is to be literally described and understood in computational terms.

4 ANYTHING COMPUTES EVERYTHING

But rather than viewing ↓MR as a theoretical virtue promising a universal account of mentality, opponents of CTM target ↓MR as its Achilles heel. In *Representation and Reality*, Hilary Putnam [5] argues that implementing a computational formalism cannot serve as the theoretical criterion of mentality, because such a standard is overly liberal to the point of vacuity. As a case in point he offers a proof of the thesis that *every* open physical system can be interpreted as the realization of *every* finite state automaton. In a related vein, John Searle [6] argues that computation is not an intrinsic property of physical systems. Instead, it an observer relative interpretation that we project on to various physical systems according to our interests and goals.

Searle contends that this makes CTM vacuous, because virtually any physical system can be interpreted as following virtually any program. Thus hurricanes, our digestive system, the motion of the planets, even an apparently inert lecture stand, all possess a level of description at which they instantiate any number of different programs – but it is absurd to attribute mental states and intelligence to them on that basis. Even though the stomach has inputs, internal processing states and outputs, it isn't a cognitive system. Yet if one wanted to, one could interpret the inputs and outputs as code for any number of symbolic processes. And in his article 'Is the Brain a Digital Computer' [7] Searle attempts to illustrate the extreme conceptual looseness of the notion of implementing an abstract formalism by famously claiming that the molecules in his wall could be interpreted as running the word star program.

Let us label multiple realizability in this direction, wherein any given *physical system* can be interpreted as implementing an arbitrary number of different *computational formalisms* 'upward MR' and denote it as '↑MR'. The basic import of ↑MR is the *non-uniqueness* of computational ascriptions to particular physical systems. In the extreme versions suggested by Putnam and Searle, there are apparently no significant constraints whatever – it is possible in principle to interpret every open physical system as realizing every

computational procedure. Let us call this extreme version 'universal upward MR' and denote it as '↑MR*'. If every physical system can be construed as implementing every computational formalism, then clearly every computational formalism is realized by every physical system, and the corresponding position in the other direction, i.e. ↓MR*, is also true. So in this sense the two positions are equivalent and ↑MR* = ↓MR*.

But mere ↑MR is weaker than ↑MR*, since the former does not assert that there are no salient constraints, and hence ↑MR would be consistent with the denial that, e.g., the molecules in Searle's wall can in fact be interpreted as implementing the word star program, if we place the proper qualifications on the notion of implementation (although every physical system might still be interpretable as implementing some very large set of distinct computations). What ↑MR denies is simply that any particular computational description that can be legitimately applied is somehow privileged or unique.

5 SOME CONTRAINTS ARE IN ORDER

In response to the Putnam/Searle universal realizability objection, various defenders of CTM attempt to deny ↑MR* by (i) placing greater constraints on what counts as a legitimate physical realization and (ii) narrowing the set of computations relevant, since only very complex and advanced procedures will be of interest to CTM as candidates for mental architecture. Putnam's proof involves *inputless* finite state automata, and these are commonly dismissed as too primitive. Full input/output capabilities are required, as well as rich internal processing structure, which calls for something on a par with, say, Jerry Fodor's [8] Language of Thought (LOT) model of cognition.

In line with strategy (i) above, David Chalmers [9] advocates what he takes to be two essential constraints in distinguishing many of the 'false' cases of implementation assumed by Putnam's argument, from 'true' cases consistent with a non-trivial reading of CTM. The first is an appropriate *causal* structure relating the state transitions in the physical implementation of the computational formalism (this is also proposed by, e.g. Ronald Chrisley [10]) , and the second is the ability of the mapping to support *counterfactual* sequences of transitions on inputs not actually given (which is also considered by Tim Maudlin [11]). Both of these are quite significant features inviting extended analysis, which unfortunately is not possible within the confines of the current discussion. However, selected points regarding each of these proffered constraints will be touched on below.

Chalmers argues that it is a necessary condition that the pattern of abstract state transitions constituting a particular run of the abstract computation on a particular input must map to an appropriate transition of physical states of the machine, where the relation between succeeding states in this sequence is governed by proper causal regularities. However, I would argue that this constraint is too strong in the general case. For example, in the Chinese room scenario, or indeed *any* situation where a human being is following an abstract computational procedure, the transition from one state to the next is not causal in any straightforward physical or mechanical sense. When I take a machine table set of instructions specifying a particular TM and then perform a given computation with pencil and paper by sketching the configuration of the tape at each step in the computation, the transitions sketched on the piece of paper are *not* causally connected: one sketch in the sequence in no way causes the next. It is only through my understanding and intentional choice to execute the procedure that the next state appears on the paper. Physical causation comes in only very indirectly, as in light rays illuminating the page and allowing me to see the symbols, and at an elementary and extraneous level, as in the friction between the pencil lead and the paper's surface causing various marks to appear.

Yet this is a perfectly legitimate and indeed paradigmatic case of implementing a Turing machine. In the Chinese room, it is merely through Searle's *understanding* of English, his free choice to behave in a certain manner, and a number of highly disjointed physical processes (finding bits of paper in a certain location, turning the pages in the instruction manual, all mediated by the human agent) that the implementation takes place. In this case it counts as an implementation simply because what can be interpreted as the appropriate states in the procedure *occur* in the correct linear order. Questions regarding the mechanics of *how* they happen to occur are not relevant to answering the question of whether or not the procedure has been implemented. The physical how is a *different* question, and is not on the same level of analysis as that invoked when determining whether or not the desired mapping from formalism to physical configuration obtains. But this then critically loosens the requirements for counting a physical system as instantiating a program. As long as what can be described or interpreted as the correct sequence of states actually occurs, then the underlying mechanics of how this takes place are not strictly relevant.

The causal requirements advocated by Chalmers constitute a sufficient but not a necessary condition – in the general case we must still allow for chance and human agency to play a role. However, the right sort of causal regularities and connections are needed if the instantiation in question is to be *fully automatic*, and if we want to be able to rely on the automatic device to perform systematically correct computations yielding outputs with the potential to supply us with new information. And although this is the norm when constructing and interpreting computational artefacts, it does not exhaust the general space of possibilities.

In response to Chalmers' proposed *counterfactual* requirement, it is worth noting that for a physical system to realize a rich computational formalism with proper input and output capacities, such as an abstract TM, this will always be a matter of *approximation*. For example, any given physical device will have a finite upper bound on the size of input strings it is able to process, its storage capacities will likewise be severely limited, and so will its actual running time. In principle there are computations that formal TMs can perform which, even given the fastest and most powerful physical devices we could imagine, would take longer than the lifespan of our galaxy to execute. It will never be possible to construct a complete physical realization of an abstract TM – the extent to which the device can execute the full range of state transitions of which the abstraction is capable will always be a matter of *degree*. So in turn, the class of counterfactual cases on alternative inputs with which the realization can cope is by necessity limited – not all counterfactual cases will be supported by *any* physical device implementing a TM.

Consequently, there is no simple or principled cut off point demarking 'genuine' implementations from 'false' ones in terms of counterfactual considerations. Take a standard pocket calculator that can intake numbers up to, say, 6 digits in decimal notion. Is this a 'false' realization of the corresponding algorithm for addition, since it can't calculate $10^6 + 10^6$? It's an approximate instantiation which is nonetheless exceedingly useful for everyday sums. It will always be a matter of degree how many counterfactuals can be supported, where a single run on one inputV is the degenerate case. Where in principle can the line be drawn after that? It's a matter of our purposes and goals as interpreters and epistemic agents, and is not an objective question about the 'true' nature of the physical device as an implementation. In some cases we might only be interested in the answer for a single input, a single run

Hence for a physical device to successfully 'perform a computation' is distinct from 'fully instantiating a computational formalism'**.** Performing a computation is an occurrent event, an actual sequence of physical state transitions yielding an output value, whereas instantiating a complete computational formalism is much more stringent and hypothetical, requiring appeal to counterfactuals, and as above, this will only obtain as a matter of degree. In light of this distinction, it is clearly possible for a physical device to successfully perform a computation *without* instantiating a complete computational formalism.

6 OBSERVER RELATIVITY

One of Searle's basic claims is the allied tenet that computation is not an 'intrinsic' property of physical systems – instead it's an observer relative act of interpretation. This basic point has been objected to in different ways, and is itself in need of clarification. The latter part of Searle's claim may seem to suggest that it is a purely subjective matter, and Ned Block [12] objects by pointing out that it's simply not the case that anything goes. As an illustration, he notes that, although it's possible to reinterpret an inclusive OR gate as an AND gate by flipping our interpretations of the values of '0' and '1', it's simply not possible to reinterpret an *inclusive* OR gate as an *exclusive* OR gate. So although we have a great deal of latitude about how we interpret a device, there are also very important restrictions on this freedom, and according to Block, this makes it a substantive claim that, e.g., the human brain is a computer of a certain sort.

Block's position suggests that there are two important strands here that need to be separated. 'Observer relative' could mean that it's totally subjective and anything goes, which is the claim he wants to deny. But it could also mean something more curtailed, *viz*., that the attribution of computational activity requires an observer to supply the interpretation. This doesn't mean that the interpretation doesn't have to satisfy various objective constraints supplied by the given characterization of the system. It simply means that, as Searle also says, it's *not intrinsic* to the system itself, and must be provided by the observer as an outside ascription. Hence it's easy to reinterpret an inclusive OR gate as an AND gate – there is no objective fact to the matter as to which truth function is being computed, and this is in perfect accord with ↑MR. Some interpretations appear to be excluded (on the very pivotal assumption that the physical system itself is characterized as an 'inclusive OR gate' and not as something more fundamental), which seems to cast some doubt on ↑MR*. In the present discussion I will not argue for or against ↑MR* (see Mark Bishop [13], [14] for an interesting version of the claim) but instead confine my considerations to the more modest ↑MR.

In view of ↑MR, it's still never the case that any given computational interpretation of a physical system is privileged or unique, and this seems far more difficult to deny than ↑MR*. And the non-intrinsic nature of computation is a direct consequence of ↑MR. As long as there are at least two distinct interpretations, there is no objective fact of the matter regarding *which* computation is being performed, and it follows that the computation itself is not an intrinsic property of the physical device. Instead, it is an act of human interpretation, and is usually tethered to issues involving design and engineering, relative to our purposes and interests. Thus implementation is always a matter of both interpretation and degree of approximation, and its usefulness will depend on our interests and epistemic needs (e.g. as above - how big a counterfactual set of inputs we want it to be able to compute).

It's certainly true that there is no pragmatic value in most interpretive exercises compatible with ↑MR and ↑MR*, e.g. *post hoc* attributions of single runs, or any case where we know the outcomes in advance of the interpretation. Physically instantiated computation is *useful* to us only insofar as it supplies informative outputs, which in most cases will come down to new information acquired as a result of the implemented calculation. Interesting observer relative computation takes place when we can directly read-off something that *follows from* the formalism, but which we didn't already know in advance and explicitly incorporate into the mapping from the start. That's the incredible value of our computational artefacts, and it's the only *practical* motivation for playing the interpretation game in the first place

Of course, this doesn't mean that we cannot ascribe other interpretations to the same system – the difference is that in most cases the outputs will then be of no pragmatic or epistemic value to us. But this is still something relative to our human interests, practices and goals – the success of the strategy is based on objective features of the system (typically that we have designed and built), but this does not make computation itself intrinsic – it is still an interpretation, an *abstract* level of description, and as such is neither canonical nor unique. Indeed, computation is no more an intrinsic property of a physical systems than is 'being a sequence of inscriptions constituting a formal derivation of a theorem in first-order logic'.

In line with this logic/formal proof example, when I execute a particular TM computation by drawing the initial tape configuration on a piece of paper, then write down the tape configuration for each step in the computation according to the instructions in the machine table until I reach a halting configuration and stop, the physical states realizing the computation are a sequence of scratch marks on a two dimensional sheet of paper. There is nothing *physical* about these scratched in patterns that is intrinsically computational – indeed, the shapes could be interpreted in any manner one likes or not at all. The computational interpretation of the physical scratch mark is purely *extrinsic*. And this is the same for syntactic interpretations in general – e.g. being an instance of the spoken English sentence 'The cat is on the mat' is not an intrinsic property of the sound waves constituting an instantiating utterance.

Physical systems as such are intrinsically rule (i.e. physical law) *obeying* while formal systems are intrinsically rule

following. In the case of our computational artefacts, a rule obeying system must be deliberately engineered so that it can be interpreted as isomorphic in the relevant sense to a chosen rule following formal system. Rule obeying is an essentially *descriptive* matter and there is no sense in which mistakes or error can be involved – physical law cannot be broken, and the time evolution of natural systems is wholly determined (in the classical case at least) by the laws obeyed. Rule following on the other hand is an essentially *normative* matter and there is a vital sense in which error and malfunction can occur. If my desk top machine is dosed with petrol and set on fire while still in operation, the time evolution of the hardware will remain in perfect descriptive accord with natural law. However, it will very soon fail to comply with the normative requirements of implementing Microsoft Word, and serious computational malfunctions will ensue. Being an implementation of Microsoft Word is a normative and *provisional* interpretation of the hardware system, which can be withdrawn when something goes 'wrong' or when the system is disrupted by non-design intended forces - being an implementation of Microsoft Word is not intrinsic to the physical structure itself.

7 COMPUTATION AND CONSCIOUSNESS

Many versions of CTM focus solely on the functional analysis of propositional attitude states such as belief and desire, and simply ignore other aspects of the mind, most notably consciousness and qualitative experience – Fodor's LOT is a classic case in point. However others, such as William Lycan [15], try to extend the reach of Strong AI and the computational paradigm, and contend that *conscious states* arise via the implementation of the appropriate computational formalism. This then invites reapplication of the Putnam/Searle line in the ↓MR* direction, with the rejoinder that every open physical system implements the 'appropriate computational formalism', so that consciousness is everywhere. According to this polemical strategy, rampant panpsychism follows as a consequence of CTM extended to the explanation of consciousness (which will be dubbed 'CTM+'), and this is taken as a *reductio ad absurdum* refutation of such views.

A natural line of defense for CTM+ is to invoke the counterfactual constraint above in order to deny ↓MR*. Only highly sophisticated physical systems (such as brains, presumably) are able to support all the counterfactuals required to count as an implementation of the appropriate computational formalism, and hence the attempted *reductio* is blocked. But as Maudlin and Bishop have argued, this is a highly dubious strategy in the case of conscious states, sense these are essentially *occurrent* phenomena, and the invocation of non-occurrent process seems to verge on the occult. As Bishop rightly observes, the appeal to counterfactuals seems to require a non-physical link between non-entered states and the resulting conscious experiences of the system.

And I would agree that for conscious states counterfactuals don't matter – it's only the *actual* run that could have any bearing, so that the foregoing attempted defense of CTM+ is unsuccessful. Additionally, I would argue that the computational account of consciousness is fundamentally wrong in any case, and that even given the implementation of all purportedly relevant counterfactuals, this would still not constitute a sufficient condition for the presence of conscious experience. As above, computation is not an intrinsic property of physical systems, and so is inherently unsuited to serve as the foundation for conscious experience, which should be based on intrinsic properties of the brain as a physical system. As I've argued elsewhere ([16], [17]), propositional attitudes are potentially explainable in terms of functional/computational structure, which is abstract and multiply realizable (because non-intrinsic!). In contrast, conscious states, if they occur in a given implementation, should be explained in terms of the intrinsic physical properties of the medium of instantiation.

This is because, unlike computational formalisms, conscious states are inherently *non-abstract*; they are *actual*, occurrent phenomena extended in physical time. The computational camp makes a critical error by espousing ↓MR as a hallmark of their theory, while at the same time contending that qualitatively identical conscious states are maintained across wildly different kinds of physical realization. The latter is the claim that an actual, substantive and *invariant* phenomenon is preserved overly radically diverse real systems, while the former is the claim that *no* internal physical regularities need to be preserved. And this implies that there is no actual, internal property that serves as the causal substrate or supervenience base for the substantive, invariant phenomenon in question. The advocate of CTM+ cannot rejoin that it is *formal role* which supplies this basis, since formal role is abstract, and such abstract features can only be *instantiated* via actual properties, but they do not have the power to *produce* them. The only (possible) non-abstract effects that instantiated formalisms are required to preserve must be specified in terms of their input/output profiles, and thus *internal* experiences, qua actual events, are in principle omitted. Hence it would appear that the actual, occurrent nature of conscious states entails that they must depend upon intrinsic properties of the *physical* world.

8 OBSERVER RELATIVITY AND CTM

However, content laden propositional attitudes *are* highly dispositional in character, and for such abstract, dispositional states, the relevant counterfactuals pertaining to formal processing structure *do* matter. If we restrict CTM to the belief-desire framework commonly assumed to characterize intentional systems, and leave consciousness out of its purview, then it is possible to give an account of how this type of approach could, at least in principle, offer us an effective theoretical handle on the mind. If we take something like Fodor's LOT (as a starting point for the sake of illustration), this is at least the basic type of highly sophisticated and complex computational structure relevant to CTM. Propositional attitudes themselves are abstract, dispositional states, and their functional/computational rendition could in principle be interpreted as a computational level of description of the activities of the human brain.

In line with the foregoing discussion, even if, for the sake of argument, we grant that the brain implemented Fodor's LOT, still, this would *not* be an intrinsic property of the brain as a biochemical mechanism. Instead, it would be a scientifically fruitful and explanatorily powerful level of description, which could supply a unifying perspective that ties together actual brain function, seen as neurologically implementing relevant tokens of 'mentalese' symbols, and systematically manipulating these tokens in a manner consistent with the proffered computational formalism of LOT. This abstract level of description would then

have to mesh with the salient input and output capabilities that we want to explain via this attribution of internal cognitive structure. So from a purely physical perspective, the inputs and outputs are various forms of energy bombarding the organism's surface and emanating from it, and are not intrinsically computational either. But on the non-intrinsic cognitive level, these would be viewed as instances of written and spoken language, for example. And when interpreted as such, this non-intrinsic syntactic level will correspond to the internal processing activity triggered by the incoming energy pulse, interpreted as, say, a sentence in an English conversation.

There would be no scientific interest in a mere *a hoc* mapping from LOT onto the brain (though in principle this may be possible, *a la* ↓MR*). Instead, there would be a myriad of pre-existing and empirically intransigent 'wet-ware' constraints that the mapping would have to satisfy, in order to correspond to the salient causal structure of brain activity as discovered by neuroscience. And as above, this would have to conform with observed input and output patterns interpreted symbolically, to yield successful *predictions* of both new outputs given novel inputs, and predictions correctly describing new brain configurations entailed by the theory as realizations of the appropriate formal transformations required to produce the predicted output. This would be real science, with two primary levels of empirical constraint satisfaction and experimental testing and confirmation, to establish or refute the accuracy of the proposed theoretical mapping. Additionally, the linguistic interpretation of input and output signals would have to mesh with corresponding objects and states of affairs in the agent's environment, since in the human LOT case, we are studying and explaining an environmentally embedded system, and not a solipsistic syntax manipulator.

If this CTM project were to turn out successful, then the LOT would be as powerful and well confirmed as a scientific venture could hope to be, and the objection that computation is still not an 'intrinsic' property of the brain would fade into irrelevance. It is in virtue of all of these factors considered together that human cognition could be accounted for in computational terms, and not simply in virtue of the brain being (in-principle) interpretable as realizing the LOT, by appeal to a mapping that ignores these crucial factors.

9 CONCLUSION

In accord with Searle, computation should be viewed as an extrinsic, observer relative feature of physical systems. As such, it does not constitute a stable or independent natural kind. Various natural phenomena can be modelled or simulated using computational techniques, but this is to be distinguished from the notion that the system *itself* spontaneously instantiates and executes a formal procedure. Natural systems are essentially rule obeying, and computational modelling simulates this in a fundamentally descriptive manner. In contrast, formal procedures are essentially normative, rule following structures, and in principle this interpretation can be projected onto natural systems in an almost limitless variety of ways. However, *interesting and illuminating* cases of computation realized in the physical world will come down to a question of engineering, either artificial or perhaps biological (to attain a robust, informative, non-post-hoc, multiple constraint satisfying *degree* of fit as a level of description for a physical system).

It is conceivable that the human brain has been biologically engineered such that there exist interesting and informative levels of computational description in the above sense. Hence I would conclude that Searle's basic point against CTM is not well taken. Although CTM+ and a computational theory of consciousness are ruled out, in the case of propositional attitude states, the non-intrinsic status of computation does not trivialize predictively successful ascriptions of formal structure, and multiple realizability on its own does not render CTM empirically vacuous.

REFERENCES

[1] A. Turing, 'On Computable Numbers, with an Application to the Entscheidungsproblem, *Proceeding of the London Mathematical Society*, (series 2), 42, 230-265, (1936).

[2] Boolos, G., Burgess, J.P. and Jeffrey, R.C., *Computability and Logic*, 5th edition, Cambridge University Press, (2007).

[3] Turing, A., 'Computing Machinery and Intelligence', *Mind* 59: 433-460 (1950).

[4] Piccinini, C., 'Computational Modelling vs. Computational Explanation', *The Australasian Journal of Philosophy*, 85(1), 93-115, (2007).

[5] Putnam, H., *Representation and Reality*, MIT Press, (1988).

[6] Searle, J., 'Minds, Brains and Programs', *Behavioral and Brain Sciences* 3: 417-424, (1980).

[7] Searle, J., 'Is the Brain a Digital Computer?', *Proceedings of the American Philosophical Association*, 64, 21-37, (1990).

[8] Fodor, J., *The Language of Thought*, Harvester Press, (1975).

[9] Chalmers, D. J., 'Does a Rock Implement Every Finite-State Automaton?', *Synthese*, 108, 309-333, (1996).

[10] Chrisley, R. L., 'Why Everything Doesn't Realize Every Computation', *Minds and Machines*, 4, 403-420, (1994).

[11] Maudlin, T., 'Computation and Consciousness', *Journal of Philosophy*, 86, 407-432, (1989).

[12] Block, N., 'Searle's Arguments against Cognitive Science'. In J. Preston and J. M. Bishop *Views into the Chinese Room*, Oxford University Press, (2002).

[13] Bishop, J. M., 'Dancing with Pixies'. In J. Preston and J. M. Bishop *Views into the Chinese Room*, Oxford University Press, (2002).

[14] Bishop, J. M., 'Why Computers Can't Feel Pain', *Minds and Machines*, 19, 507-516, (2009).

[15] Lycan, W. G., *Consciousness*, MIT Press, (1987).

[16] Schweizer, P., 'Physicalism, Functionalism and Conscious Thought.' *Minds and Machines*, 6, 61-87 (1996).

[17] Schweizer, P., 'Consciousness and Computation.' *Minds and Machines*, 12, 143-144, (2002)

From Artificial Life to Artificial Embodiment: Using human-computer interfaces to investigate the embodied mind 'as-it-could-be' from the first-person perspective

Tom Froese[1], Keisuke Suzuki[2], Sohei Wakisaka[2], Yuta Ogai[1] and Takashi Ikegami[1]

Abstract. There is a growing community of cognitive scientists who are interested in developing a systematic understanding of the experiential or 'lived' aspects of the mind. We argue that this shift from *cognitive* science to *consciousness* science presents a novel challenge to the fields of AI, robotics and related synthetic approaches. AI has traditionally formed the central foundation of cognitive science, and progress in artificial life has helped to pioneer a new understanding of cognition as embodied, situated and dynamical. However, in the current experiential turn toward the phenomenological aspects of mind, the role of these fields still remains uncertain. We propose that one way of dealing with the challenge of phenomenology is to make use of artificial life principles in the design of systems that include human observers inside the technologically mediated sensorimotor loops. Human-computer interfaces enable us to artificially vary the embodiment of the participants, and can therefore be used as novel tools to systematically investigate the embodied mind 'as-it-could-be' from the first-person perspective. We illustrate this methodology of *artificial embodiment* by drawing on our research in sensory substitution, virtual reality, and interactive installation.

1 INTRODUCTION

The interdisciplinary field of cognitive science has undergone a number of conceptual and methodological transitions since its beginnings in the 1970s [1]. Cognition was first conceived as symbolic computation, then as sub-symbolic computation, and then as embodied, situated and dynamical. Most recently, there has been a growing interest to conceive of mind as rooted in the phenomenon of life, as proposed by the enactive approach [2, 3].

One particularly exciting aspect of this latest development is that life can provide a natural bridge over the mind-body gap. In brief, the idea is that our body can be investigated as a special kind of physical object, namely a living system, and yet at the same time it is also an essential part of how we subjectively experience ourselves in the world. In other words, the concept of the embodied mind means that we are embodied both as *living* (i.e. biological) and as *lived* (i.e. phenomenological) agents.

This radical conception of life-mind continuity is, to put it in a simplified manner, the very foundation of what has been called the 'enactive' approach to cognitive science [4]. Note that life-mind continuity has a double implication for cognitive science: on this view, the scientific study of mind becomes inseparable from the study of organismic existence and from the study of conscious experience. However, given the predominant focus on functionalism in cognitive science, neither of these topics (living and lived phenomena) has so far received much attention by the scientific mainstream. What about the synthetic approaches?

At least in the case of previous shifts in cognitive science, the new movements have always been strongly supported, if not even initiated, by concurrent developments in fields like artificial intelligence, robotics and artificial life. For instance, we now have an improved understanding of the importance of emergent sensorimotor dynamics for perception and cognition, and of the way in which these structures enable the agent to self-structure its perceptual affordances (see Figure 1). The acceptance of this insight in cognitive science is, to a large extent, based on work in situated robotics (e.g. [5-7]) and theoretical biology is getting a similar support from research in artificial life (e.g. [8, 9]).

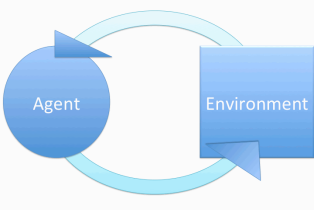


Figure 1. Recent synthetic approaches, such as situated robotics and artificial life, have made some substantial contributions to our understanding of the importance of sensorimotor coupling for perception and cognition.

Of course, the theoretical foundation of the enactive approach to cognitive science, namely the life-mind continuity thesis, has also been a central topic of interest in the field of artificial life for some time (e.g. [10]), and we can thus expect that there will be an enactive approach to synthetic modeling that pays closer attention to the organization of living processes (e.g. [11-13]).

However, this still leaves us unclear about how such synthetic approaches can deal with the most radical aspect of the enactive approach, namely the turn toward the experiential aspects of the human mind. In contrast to previous paradigm shifts in cognitive science, where breakthrough innovations in the artificial sciences were able to lead the way toward a new understanding of mind, the inspiration for this experiential turn has largely come from outside of cognitive science and synthetic modeling, in particular from the tradition of phenomenology [3, 4, 14]. And, since the focus is on how phenomena are experienced or 'lived' from the first-person perspective, the artificial sciences, if they want to stay abreast of this ongoing development in cognitive science, are confronted with a fundamental challenge.

[1] Ikegami Laboratory, Department of General Systems Studies, Univ. of Tokyo, Tokyo, Japan. E-mail: t.froese@gmail.com

[2] Laboratory for Adaptive Intelligence, Mind and Intelligence Research Core, RIKEN Brain Science Institute, Tokyo, Japan.

To be sure, even the recent surge of interest in the science of consciousness is only slowly coming to terms with the problem of how to best investigate experience as it is lived through from the first-person perspective [10, 15], so the artificial sciences are not alone in this dilemma. Moreover, while the artificial sciences are faced by a profound challenge, this is not to say that robotic agents and simulation models are completely irrelevant for the current experiential turn. They can offer important technological supplementation of traditional phenomenological methodology by helping us to explore some of the structures and dynamics of life and mind that are possible in principle [16]. Nevertheless, it is clear that since these artificial systems are by definition not actual human beings, they cannot help us to study the structures and qualities that are specific to human experience.

In order to illustrate this inherent limitation, let us consider a prominent debate in cognitive science related to the enactive or sensorimotor account of visual perception [17, 18]. Briefly, this account holds that the experiential quality of the visual modality is constituted by the subject's know-how of a particular set of sensorimotor skills, and that the deployment of these same skills through other means than the eyes will therefore also result in visual experience. But how should this hypothesis about visual consciousness be verified? In support of their claims proponents of the sensorimotor account are fond of citing Bach-y-Rita's [19] experiments with the tactile-visual sensory substitution system (TVSS), which translates images from a camera onto a tactile array placed on the user's body. Some subjects who are trained in using the TVSS report being able to 'see' objects in space and can indeed behave accordingly, for example by recognizing faces and avoiding obstacles.

Nevertheless, there is an ongoing debate in cognitive science about whether the experience of using the TVSS is the same as vision or at least vision-like, or another form of touch, or touch-based rational inference, or whether it may actually constitute a novel perceptual modality. How can we verify which of these explanations is the most valid if all we have is some fragments of verbal reports and external behavior? As Froese and Spiers [20] have pointed out, these questions about *what it is like* [21] to use the TVSS cannot be answered with any certainty without oneself actually having tried out using the device and lived through the experience of its usage from the first-person[3]. Or, at the very least, we need to elicit more detailed reports from those subjects who have had first-person access to the experiences in question, for example through interviews [22].

What this little detour should have made clear is that, even though active perception has been a hot topic of research in the artificial sciences for at least a couple of decades (e.g. [5, 6]), the results, although of great interest to cognitive science, can tell us nothing about what it is actually like to experience the various sensorimotor loops it has investigated. And this is a limitation in principle. While the synthetic methodologies can study in detail how the structural dynamics of brain, body and world give rise to certain kinds of behavior, by definition its research remains silent about what kind of first-person experiential quality those dynamics may have for a human subject. At least in this first-person phenomenological respect, therefore, the traditional role of the artificial sciences in pioneering the development of new conceptions of mind will be marginalized, and a focus on actual human subjects will have to take its place.

At the same time we propose that this development presents a new opportunity for the artificial sciences. If we need to know what kind of experiential qualities are entailed by what kind of sensorimotor dynamics, then all we need to do is to place a human observer inside the sensorimotor loop. More specifically, the lessons of artificial life and situated robotics can be adapted in the design of human-computer interfaces that are intended to systematically transform our embodied mind. For instance, how would the experience of using the TVSS change if we increased the spatial resolution of the tactile array? Would it result in a quantitative or qualitative shift? What if we transformed distance information rather than luminance information? What if the whole environment was responsive rather than having a device attached to one's body? What is it like to substitute the actual visual system with a virtual one? In order to answer these kinds of questions we need to design artificial systems much like we would have designed them in the case of robots, but in this case they are intended to be used by humans subjects rather than by artificial 'agents'. These technological devices can then be used as scientific tools to investigate the phenomenological mind 'as-it-could-be' from the first-person perspective, thereby giving rise to a novel methodology of *artificial embodiment* [23].

As we will argue in the rest of this paper, the methodology of artificial embodiment is structurally similar to the one already employed by some existing synthetic approaches, especially by the field of artificial life, except that in this case it is our own embodiment in the world which is systematically manipulated by means of artificial systems. We will illustrate this methodology by drawing on our recent research in sensory substitution, virtual reality and large-scale interactive installations.

2 FROM ARTIFICIAL LIFE TO ARTIFICIAL EMBODIMENT

In order to support the idea that artificial embodiment is a logical extension of artificial life, given the current experiential turn in cognitive science, it is important to make their respective goals and methodology explicit. In the case of the field of artificial life we can quote one of its founders, Chris Langton, who describes the field's mission as follows:

"By extending the horizons of empirical research in biology beyond the territory currently circumscribed by life-as-we-know-it, the study of artificial life gives us access to the domain of life-as-it-could-be, and it is within this vastly larger domain that we must ground general theories of biology and in which we will discover practical and useful applications of biology in our engineering endeavors."[4]

Another way of putting this is to say that the methodology of artificial life consists of two essential aspects: (i) it is *synthetic* – i.e. the phenomena to be investigated must be brought into being by artificially creating the conditions for their emergence, and (ii) it is *analytic* – i.e. the phenomena, once they have emerged, are still in need of further analysis [24]. The synthetic aspect is usually implemented in terms of computer simulation or physical systems, while the analytic aspect is typically approached by

[3] Note that this response should be familiar to anyone working in the area of complex systems. We can try to predict the results of interactions on the basis of our intuitions and by comparison with known results from other similar systems, but there is no certainty that we are right. We must actually *instantiate* the system and let the effects emerge by themselves.

[4] C. G. Langton, http://www.biota.org/papers/cglalife.html

means of dynamical systems theory. It is important to emphasize that both aspects are indispensable: without (i) there is no novel phenomenon that can be described and explained, and without (ii) there can be no scientific explanation. The methodology of artificial life has been successful in many areas. It has created proofs of concept, thought experiments, illustrative models, as well as mathematical and technological advances, many of which have been influential in cognitive science [25].

And yet this kind of artificial life research is clearly limited to the study of the dynamical and physical aspects of living and cognitive systems, while excluding any first-person experiential considerations. Of course, this limitation is not a problem for the field of artificial life itself, but it does make its relationship with cognitive science rather one-sided, especially with respect to the phenomenological mind. Studies of human experience have been used to create [26] and to criticize [27] the field of AI from the beginning, but it is difficult to conceive the relationship the other way around (but see [16] for an attempt).

The methodology of artificial embodiment, on the other hand, tries to fill precisely this gap by bringing human subjects into the domain of the artificial sciences, and thereby bringing the first-person perspective along with them. First-person experience can then be studied scientifically by using artificial media, which modify the subject's embodiment. Since the mind is embodied, we can systematically change our experience by systematically changing our embodiment (see Figure 2).

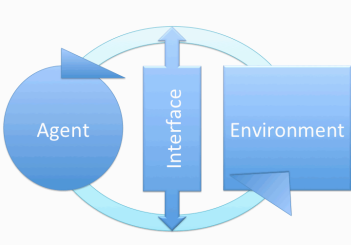


Figure 2. The structure and quality of our experience depends on our embodiment in the world, including on our sensorimotor capacities. It is therefore possible to systematically explore the domain of 'mind-as-it-could-be' by systematically varying our relationship with world, for instance by means of technological interfaces and immersive devices.

Paraphrasing Langton's description of artificial life, we can say that under the label of 'artificial embodiment' we understand a synthetic methodology that enables us to go beyond the study of *mind-as-we-know-it*, in order to access the domain of *mind-as-it-could-be*. The aim of artificial embodiment is to access a larger domain of mental phenomena, i.e. mind-as-it-could-be, so as to ground general theories of phenomenology and cognitive science, and to gain technological benefits along the way. The methodology of artificial embodiment is therefore similar to the one that is already familiar from artificial life. Indeed, it consists of the same two essential aspects:

(i) The methodology is *synthetic*: The phenomena of interest are typically not directly available to human experience and their conditions of emergence must first be artificially produced by technological means.

(ii) The methodology is *analytic*: The experiential phenomena generated with this technology are in need of detailed description and further analysis in order to become the basis for scientific explanations.

Of course, these two methodological aspects are not new in themselves. In terms of (i), it is possible to draw on research that is going on in virtual reality systems, tool-use, human-computer interfaces, interactive installations, sensory substitution, and so forth (e.g. [28]). And in terms of (ii), we can draw on the work in first- and second-person approaches to consciousness studies (e.g. [22, 29]). What we are suggesting is that it would be mutually beneficial for these two areas to work more closely together. This is the idea of artificial embodiment.

What could this collaboration look like in practice? While both of these methodological aspects are essential if artificial embodiment is to be informative for the experiential turn in cognitive science, we will here focus mainly on the first aspect (for a detailed discussion related to the second aspect, see [10, 15]). The synthetic aspect is a variation of the already familiar 'engineering for emergence' theme of the artificial sciences. But the medium for emergence is no longer the computer, the robot, or the chemical 'soup', but rather the human being. In other words, we want to design interfaces that couple with our bodies such that this body-technology interaction spontaneously gives rise to a lived experience that is of interest to the science of life, mind, or consciousness. The nature of this interaction can vary: augmentation, substitution, enaction, and deprivation are all possibilities. What matters most is that our bodily ways of engaging with the world can be systematically altered because this will allow systematic exploration of mind-as-it-could-be.

Other factors to consider include that the interface is cheap, non-intrusive, and requires little training time so that other researchers can easily replicate the experiments. Potential for replication is necessary for science in general, but it is especially important when the phenomenon that is to be explained has to be personally experienced. Otherwise, if the reported phenomena cannot be easily verified by other researchers from the first-person perspective, there is a danger to get caught up in debates that are not properly experientially grounded, e.g. the ongoing discussion about whether the perceptual experience of using TVSS is a form of vision, vision-like, touch, touch-based inference, or actually a novel perceptual modality [20].

This potential need for not only *experimental* but also *experiential* verification puts additional constraints on the analytic methods employed by artificial embodiment. To be sure, it is impossible to expect all researchers and their experimental participants to become experts at becoming aware and describing their lived experience. Becoming aware is a skill that requires sustained practice and depends upon a personal commitment to undergo a long and difficult process of training [30]. This makes so-called 'second-person' approaches [22], whereby the process of becoming aware is facilitated by a skilled interviewer, especially attractive for the practice of artificial embodiment.

Of course, there is already a tradition of qualitative research in human-computer interfaces and related areas, which could benefit from the perspective adopted here. Artificial embodiment has the potential to unify a variety of these synthetic approaches by placing them in an explicit relationship with the methodology of artificial life and the experiential turn in cognitive science.

3 CASE STUDIES

In the rest of this paper we discuss four kinds of technological interfaces that have been specifically designed to modify the

user's experience, namely two kinds of sensory substitution interfaces, a virtual reality setup, and an interactive installation.

3.1 Sensory substitution (I)

The value of using HCI technology to conduct psychological experiments on sensory substitution has long been recognized in cognitive science [28, 31]. Recently, there have been attempts to further generalize this approach beyond mere 'substitution' [32], a methodological shift which nicely complements the idea of artificial embodiment. For example, specialized interfaces have been used to investigate the minimal necessary conditions for the experience of depth, or spatiality [33] and the perception of other agents, or alterity [34]. This minimalism is reminiscent of the minimalism often advocated in artificial life research [35]. It not only facilitates the task of synthesis and analysis, but may also offer practical alternatives to the current commercial focus on producing the most high-resolution interfaces [36].

Let us begin with a case study of artificial embodiment in which the resemblance to artificial life is clearly visible. Ogai and his colleagues proposed an active tactile system, which uses a small tactile display and a 3D position sensor [37, 38]. A subject's hand movements are used as inputs for a Recurrent Neural Network (RNN), and the outputs from the RNN are fed back to the subject's finger by means of an Ionic Conducting Polymer gel Film (ICPF), which is attached to the tip of the finger. As a result, the subject feels a tactile sensation. The overall feedback system is illustrated in Figure 3, and the ICPF tactile feedback device and its placement on the tip of the subject's finger are shown in Figure 4 and Figure 5, respectively.

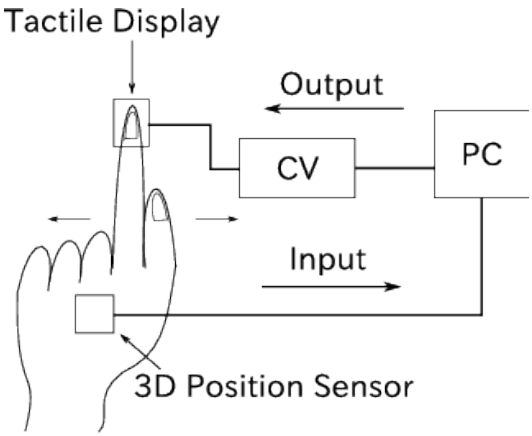


Figure 3. Illustration of the active tactile system by Ogain and colleagues. It uses basic artificial life principles to investigate the experience of tactile sensations referred to by onomatopoeias.

The task of the subjects is to train the RNN so that the inputs of their hand movements drive the response-profile of the ICPF on their finger in such a way that the feeling of certain tactile textures arises. This training is achieved by means of 'interactive evolutionary computation', an optimization method which is inspired by Dawkin's [39] approach to evolving 'biomorphs' but which has been generalized to other experiences than merely visual aesthetics [40]. In this case, two Japanese onomatopoeias, *uneune* and *zarazara*, are used as the evolutionary goals. *Uneune* means 'the tactile sensation of winding things', and *zarazara* means 'the tactile sensation of a coarse surface.' In other words, subjects were asked to optimize the RNNs so that the experience of the ICPF-mediated outputs became like the tactile sensation referred to by the words *uneune* and *zarazara*. When a subject chose the same RNN configuration 10 times continuously, the optimization of the RNN was regarded as completed.

After the RNN training, the subjects were asked to distinguish between the experience of sensations that were evolved by the subject himself and those evolved by others. The experimental results show that it is more difficult to make this distinction in the case of *zarazara* than it is in the case of *uneune*. Ogai and his colleagues suggest that this is because the experience of *uneune* involves a higher degree of active perception than *zarazara*.

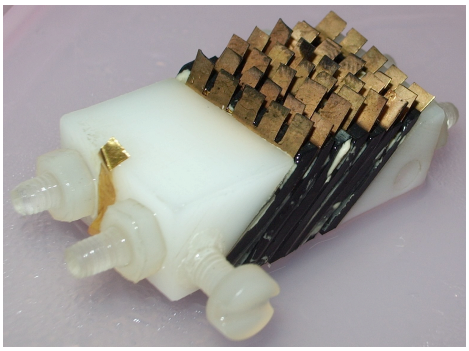


Figure 4. A small tactile display consisting of Ionic Conducting Polymer gel Film (ICPF) elements. Subjects wear it on the finger cushion of their index finger (see **Figure 5**).

Figure 5. The 3D position sensor is located on the back of the hand and the ICPF-based tactile display is placed on the finger.

3.2 Sensory substitution (II)

Another example of a minimalist interface that was conceived along the lines of artificial embodiment is the Enactive Torch (ET), a hand-held distal-to-tactile sensory augmentation device (see Figure 6). The ET was designed specifically for the purpose of allowing the study of enactive perception from the first-person perspective [20]. It consists of a single, continuous parameter of body-technology coupling, namely a distance measure (taken by means of ultrasonic or infrared sensors) that is translated into variations of vibro-tactile intensity in the user's hand or arm. This allows the user to feel distances by actively scanning the environment with the device. After a few minutes of practice, many blind-folded participants will spontaneously report that

they perceive obstacles located in front of them, rather than at the location of the tactile interface.

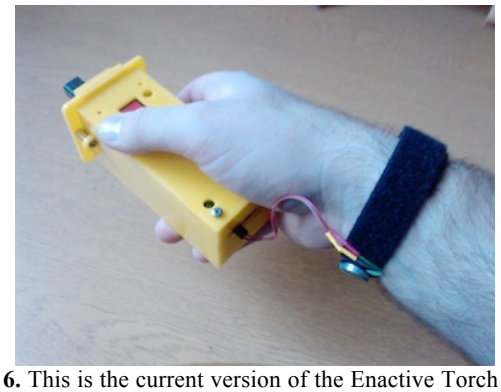


Figure 6. This is the current version of the Enactive Torch (ET), a distance-to-tactile sensory augmentation interface. This version uses an infrared sensor, which is visible at the top end of the device. The vibro-tactile motor is strapped to the arm.

Despite its simplicity, the ET therefore provides an intuitive platform to investigate the development of bodily skills and perceptual modalities, the exteriorization of stimuli, and as well as cross-modal influences. Interestingly, the latter influences were only discovered during a second-person interview, where it turned out that the motor sounds of the vibro-tactile interface modulated the appearance of the perceptual space afforded by the device. The discovery of this unforeseen effect indicates the need for a tight integration between the synthetic and analytic aspects of artificial embodiment.

3.3 Virtual reality

Virtual reality (VR) systems have been a hot topic of research and the technology has now progressed to a level that these systems can be used as a tool for consciousness science [41]. Indeed, much of the research in this area can be considered as a form of artificial embodiment. For example, the VR system can be used to systematically vary one's sense of embodiment, even to the extent of inducing out-of-body experiences [42].

One of the most studied phenomenological aspects of being embodied within a virtual reality environment is the feeling of presence [43], which has been found to be strongly related to the VR system's responsiveness to the user's actions [44]. Despite this felt presence, however, in most cases the users do not lose awareness of the fact that they are experiencing an artificial world. This lingering awareness of the virtuality somewhat limits the potential of these systems for artificial embodiment because it introduces the need for a suspension of disbelief that is not part of our normal being-in-the-world.

Accordingly, a novel VR system was designed by Suzuki and Wakisaka that avoids this problem (see Figure 7). Their system is developed specifically for users to have the unquestioned conviction that they are still present in the real world, even when they perceive a 'fake' world. In the system, the scene presented with a head-mount display can switch from a live view to a recorded view while keeping smooth visuo-motor coupling. People believe that they are in the 'here and now' even when they are in the prerecorded scenes because they can see wherever they want. This effect is achieved by using a panoramic camera to record the environment, and to present the head-oriented part of the panoramic movie on the basis of a motion tracker sensor.

At least under restricted conditions most subjects will fail to distinguish between the artificial and live visuo-motor coupling, and their felt presence is therefore of being in normal reality.

This new approach to VR nicely illustrates why the research program we are proposing is more appropriately called 'artificial embodiment' rather than 'artificial experience'. Even though the system, which is used to generate the experience by modifying our embodiment, is artificial (and may, in fact, be an example of artificial life), the experiences are not necessarily artificial.

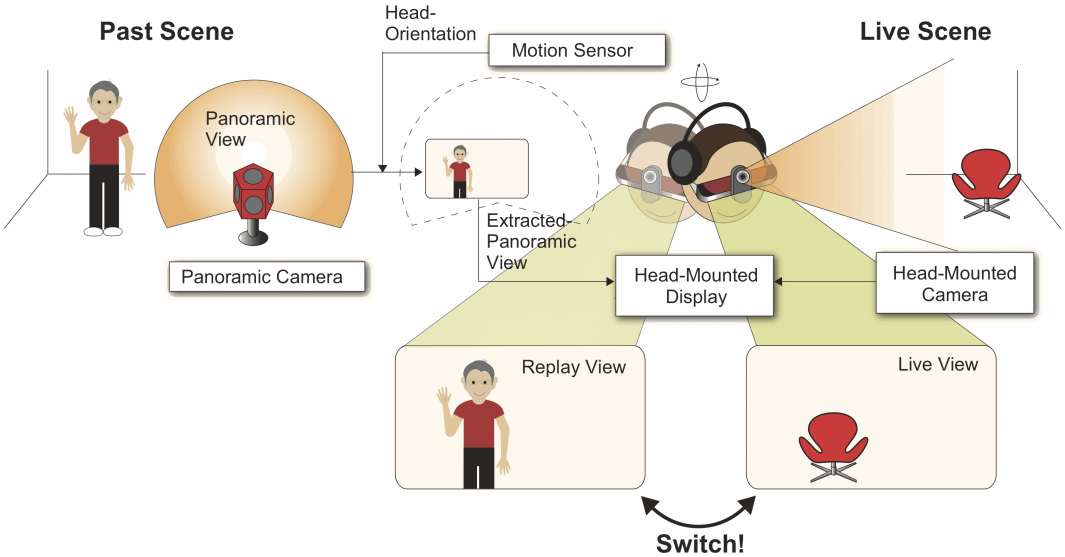


Figure 7. Illustration of the reality substitution system: First, a panoramic camera records the whole scene in advance. When the subject enters the scene, he is shown a live view captured with a head-mounted camera (right). Then, without warning, the video feed is switched to a replay view, which is created by trimming the previously recorded panoramic view by using the information provided by a head-mounted motion sensor (left). The apparent persistence of consistent visuo-motor coupling assures that the user really feels to be "here and now" even during the replay view.

3.4 Interactive installation

There is a long history of using technology to make interactive installations, and the field was greatly helped by the computer revolution during the last century. There has also been a close association between interactive art and the field of artificial life, dating back to the days of cybernetics [45]. A recent instance of this kind of work is the Mind Time Machine (MTM) created by Ikegami and colleagues [46].

The MTM is an artificial system designed to self-sustain its rich dynamics in an open-ended environment, typically a public exhibition venue. This machine consists of three screens (right, left, and above on the ceiling), which are displayed as faces of a cubic skeleton 5.4m in diameter (see Figure 8). Fifteen video cameras attached to each pole of the skeletal frame view things happening in the venue. The video recordings are decomposed into frames, and are processed by recurrent neural networks whose dynamics combine, reverse and superpose the frames to produce new frame sequences. The system itself is a completely deterministic system, using no random numbers, but it projects different images depending on the inherent instabilities of the neural dynamics that reflect environmental light conditions, movements of people coming to the venue, and the system's stored memory. The operating principle is to run the neural dynamics with plasticity and optical feedback to enable the emergence of autonomous self-organizing phenomena.

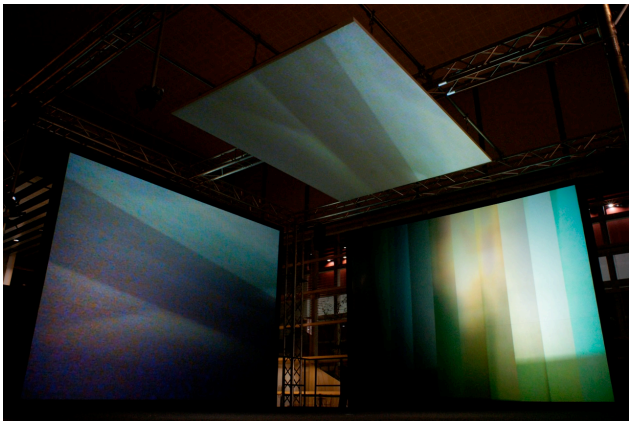


Figure 8. View of the three screens of the Mind Time Machine (MTM) during a display at the Yamaguchi Center for Arts and Media, Tokyo, in 2010. The MTM projects video output onto these screens and then records its own projection with video cameras, thus forming a video feedback loop. People can walk into this installation and can experience the effects their interference has on the dynamics of the feedback loop.

On the one hand, the MTM can be conceived as a peculiar example of artificial life research, because it uses the principles from that field and displays adaptive behavior. On the other hand, the MTM can also be viewed as an instance of artificial embodiment. This is because a crucial aspect of the MTM's environment is people. Visitors of the MTM can walk into the installation where they interact with its displays by casting shadows; in return their movements also appear in the machine's video recordings. We suggest that the MTM therefore presents us with an interesting hybrid between artificial embodiment and artificial life. The device is an instance of artificial life, which incorporates the bodies of human beings into its sensorimotor loop and thereby becomes an instance of artificial embodiment.

The advantage of this hybrid approach is that it now becomes possible to evaluate the status of the artificial life system in terms of the lived experience of the participants who interact with it. More precisely, a long-standing problem in artificial life, namely how to determine whether an artificial system has characteristics of agency or not, is addressed by determining whether human participants are experiencing the responses of the MTM as being informed by agency or not. This novel approach to the problem of agency detection is of interest to cognitive science, because it enables us to differentiate between those autonomous interaction dynamics that can give rise to a sense of the presence of others (alterity) and those that do not.

3.5 Other examples

We have chosen to discuss these four case studies because they are part of our ongoing research. However, there are many other examples, which may also be considered as instances of artificial embodiment. For example, there is ongoing work on experiential transformations during machine-mediated agency [47]; there is the work by Chrisley and colleagues to make use of experiences with HCI to engender conceptual change [48], and to use the states, interactions and capacities of an artificial agent for the purpose of specifying the contents of conscious experience [49]. It is beyond the scope of this paper to provide a comprehensive review of these and other areas of ongoing research. Future work will have to determine whether these various approaches share the same methodological structure with artificial embodiment.

4 GENERAL DISCUSSION

We have presented four case studies in order to illustrate the methodology of artificial embodiment. These examples share a common interest in evoking novel experiences in human subjects by modifying their normal sensorimotor coupling by means of artificial systems (see Figure 9).

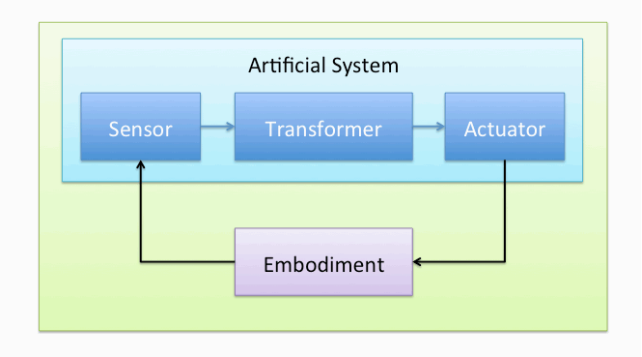


Figure 9. Schematic illustration of how artificial systems can be used to do artificial embodiment research.

The crucial step of moving the idea of artificial embodiment beyond technological wizardry and into a principled scientific research program is to link it to the rest of cognitive science in terms of hypothesis generation and verification. This integrative methodology should consist of four essential steps:

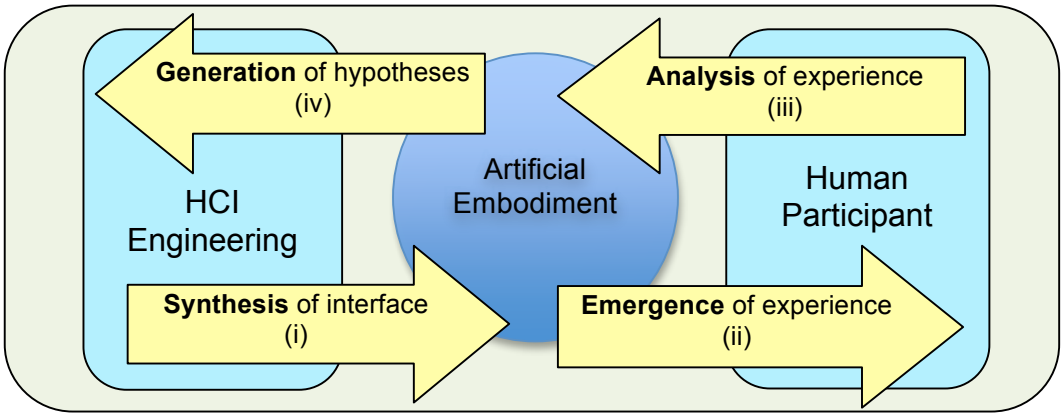


Figure 10. Illustration of the key steps involved when using artificial embodiment as a methodology for cognitive science.

(i) Synthesis of interface: The first step is generally the identification of an interesting experiential phenomenon whose systematic variation could be beneficial for cognitive science. This might be the case for a variety of reasons. For example, it could be a simple exploratory study because the scope of the phenomenon's variability is unknown, or perhaps existing explanations of the phenomenon posit conditions of necessity for its appearance that need to be investigated experimentally.

(ii) Emergence of experience: While the interface itself is designed by engineers, the experiential effects of its usage cannot be directly pre-specified. The experience of the user is an emergent phenomenon that depends on the particular history of agent-environment interactions that is realized by the device synthesized in step (i).

(iii) Analysis of experience: The experiential phenomena that emerge in step (ii) are essentially 'opaque' in that they require further analysis to be properly understood and in order to determine their essential structures and conditions of possibility. This analysis can be achieved by systematically varying the parameters of the interface to map out the range of experiential effects, and by using a combination of first- and second-person methods to obtain detailed verbal reports.

(iv) Generation of hypotheses: The insights gained in step (iii) form the basis for a theoretical response in relation to the study's original motivation. They also inform the process of generating novel hypotheses, which then become the basis for the design of novel interfaces for step (i).

The relationship between the *synthetic, emergent, analytic, and generative* aspects of the artificial embodiment methodology are illustrated in Figure 10. Note that steps (i)-(ii) already exist in all three of the case studies as well as in most HCI research more generally. But it is steps (iii)-(iv) which would really turn these studies into a proper scientific research program.

5 CONCLUSION

The artificial sciences are faced by the challenge of how to contribute to the experiential turn in cognitive science. We have proposed a research methodology of 'artificial embodiment', which draws on the insights developed by artificial life. It is focused on systematically altering the embodiment of human beings in order to investigate the embodied mind-as-it-could-be. We have presented some examples drawn from our research, which illustrate the potential of this new synthetic methodology. More work needs to be done in order to connect the resulting novel technological tools with the theoretical, experimental and phenomenological concerns of the rest of the cognitive science.

ACKNOWLEDGEMENTS

This paper was supported by a Grant-in-Aid awarded to Froese by the Japanese Society for the Promotion of Science. We thank Bill Bigge for his work on the latest version of the Enactive Torch. The photos in Figure 6 and Figure 8 were taken by Marek McGann and Kenshu Shintsubo, respectively.

REFERENCES

[1] A. Clark, *Mindware: An Introduction to the Philosophy of Cognitive Science*. New York, NY: Oxford University Press, 2001.

[2] J. Stewart, O. Gapenne, and E. A. Di Paolo, Eds., *Enaction: Toward a New Paradigm for Cognitive Science*. Cambridge, MA: MIT Press, 2010.

[3] E. Thompson, *Mind in Life: Biology, Phenomenology, and the Sciences of Mind*. Cambridge, MA: The Belknap Press of Harvard University Press, 2007.

[4] F. J. Varela, E. Thompson, and E. Rosch, *The Embodied Mind: Cognitive Science and Human Experience*. Cambridge, MA: MIT Press, 1991.

[5] R. D. Beer, "The dynamics of active categorial perception in an evolved model agent," *Adaptive Behavior,* vol. 11, pp. 209-243, 2003.

[6] R. A. Brooks, "Intelligence without representation," *Artificial Intelligence,* vol. 47, pp. 139-160, 1991.

[7] R. Pfeifer and J. C. Bongard, *How the Body Shapes the Way We Think: A New View of Intelligence*. Cambridge, MA: MIT Press, 2007.

[8] K. Suzuki and T. Ikegami, "Shapes and Self-Movement in Protocell Systems," *Artificial Life,* vol. 15, pp. 59-70, 2009.

[9] M. Egbert, X. Barandiaran, and E. A. Di Paolo, "A Minimal Model of Metabolism-Based Chemotaxis," *PLoS Computational Biology,* vol. 6, 2010.

[10] T. Froese, C. Gould, and A. K. Seth, "Validating and Calibrating First- And Second-person Methods in the Science of Consciousness," *Journal of Consciousness Studies,* vol. 18, pp. 38-64, 2011.

[11] T. Froese and T. Ziemke, "Enactive Artificial Intelligence: Investigating the systemic organization of life and mind," *Artificial Intelligence,* vol. 173, pp. 366-500, 2009.

[12] E. A. Di Paolo, "Organismically-inspired robotics: Homeostatic adaptation and teleology beyond the closed sensorimotor loop," in *Dynamical Systems Approach to Embodiment and Sociality*,

K. Murase and T. Asakura, Eds., ed Adelaide, Australia: Advanced Knowledge International, 2003, pp. 19-42.
[13] T. Ikegami and K. Suzuki, "From homeostatic to homeodynamic self," *BioSystems,* vol. 91, pp. 388-400, 2008.
[14] T. Froese, "On the role of AI in the ongoing paradigm shift within the cognitive sciences," in *50 Years of Artificial Intelligence: Essays Dedicated to the 50th Anniversary of Artificial Intelligence*, M. Lungarella, F. Iida, J. Bongard, and R. Pfeifer, Eds., ed Berlin, Germany: Springer, 2007, pp. 63-75.
[15] T. Froese, C. Gould, and A. Barrett, "Re-Viewing From Within: A Commentary on First- and Second-Person Methods in the Science of Consciousness," *Constructivist Foundations,* 2011.
[16] T. Froese and S. Gallagher, "Phenomenology and Artificial Life: Toward a Technological Supplementation of Phenomenological Methodology," *Husserl Studies,* vol. 26, pp. 83-106, 2010.
[17] J. K. O'Regan and A. Noë, "A sensorimotor account of vision and visual consciousness," *Behavioral and Brain Sciences,* vol. 24, pp. 939-1031, 2001.
[18] A. Noë, *Action in Perception*. Cambridge, MA: The MIT Press, 2004.
[19] P. Bach-y-Rita, C. C. Collins, F. A. Saunders, B. White, and L. Scadden, "Vision Substitution by Tactile Image Projection," *Nature,* vol. 221, pp. 963-964, 1969.
[20] T. Froese and A. Spiers, "Toward a Phenomenological Pragmatics of Enactive Perception," in *Enactive/07: Proceedings of the 4th International Conference on Enactive Interfaces*, ed Grenoble, France: Association ACROE, 2007, pp. 105-108.
[21] T. Nagel, "What is it like to be a bat?," *Philosophical Review,* vol. 83, pp. 435-350, 1974.
[22] C. Petitmengin, "Describing one's subjective experience in the second person: An interview method for the science of consciousness," *Phenomenology and the Cognitive Sciences,* vol. 5, pp. 229-269, 2006.
[23] T. Froese, "Exploring Mind-As-It-Could-Be: From Artificial Life to Artificial Embodiment," in *Workshop on Key Issues in Sensory Augmentation Research*, Brighton, UK, 2009, pp. 1-2.
[24] E. A. Di Paolo, J. Noble, and S. Bullock, "Simulation Models as Opaque Thought Experiments," in *Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life*, M. A. Bedau, J. S. McCaskill, N. H. Packard, and S. Rasmussen, Eds., Cambridge, MA: MIT Press, 2000, pp. 497-506.
[25] R. D. Beer, "Dynamical approaches to cognitive science," *Trends in Cognitive Sciences,* vol. 4, pp. 91-99, 2000.
[26] A. Newell and H. A. Simon, *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall, 1972.
[27] H. L. Dreyfus, *What Computers Can't Do: A Critique of Artificial Reason*. New York, NY: Harper and Row, 1972.
[28] A. Clark, *Natural-Born Cyborgs: Minds, Technologies, and the Future of Human Intelligence*. New York, NY: Oxford University Press, 2003.
[29] F. J. Varela and J. Shear, "First-person methodologies: What, Why, How?," *Journal of Consciousness Studies,* vol. 6, pp. 1-14, 1999.
[30] N. Depraz, F. J. Varela, and P. Vermersch, *On Becoming Aware: A Pragmatics of Experiencing*. Amsterdam, Netherlands: John Benjamins Publishing, 2003.
[31] P. Bach-y-Rita and S. W. Kercel, "Sensory substitution and the human-machine interface," *Trends in Cognitive Sciences,* vol. 7, pp. 541-546, 2003.
[32] C. Lenay, O. Gapenne, S. Hanneton, C. Marque, and C. Genouëlle, "Sensory Substitution: Limits and Perspectives," in *Touching for Knowing: Cognitive psychology of haptic manual perception*, Y. Hatwell, A. Streri, and E. Gentaz, Eds., Amsterdam, Netherlands: John Benjamins, 2003, pp. 275-292.
[33] M. Auvray, S. Hanneton, C. Lenay, and K. O'Regan, "There is something out there: Distal attribution in sensory substitution, twenty years later," *Journal of Integrative Neuroscience,* vol. 4, pp. 505-521, 2005.
[34] M. Auvray, S. Hanneton, C. Lenay, and J. K. O'Regan, "Perceptual interactions in a minimalist virtual environment," *New Ideas in Psychology,* vol. 27, pp. 32-47, 2009.
[35] M. Rohde, *Enaction, Embodiment, Evolutionary Robotics: Simulation Models for a Post-Cognitivist Science of Mind*. Amsterdam, Netherlands: Atlantis Press, 2010.
[36] A. Spiers, "A Tactile Navigational Aid for Visually Impaired People," B.Sc. Dissertation, Department of Cybernetics, University of Reading, Reading, 2004.
[37] Y. Ogai, "Constructive Research of Active Perception by Cognitive Experiment and Simulation Using Neural Networks," PhD Dissertation, Department of General Systems Studies, University of Tokyo, Tokyo, Japan, 2011.
[38] Y. Ogai, R. Uno, and T. Ikegami, "From Active Perception to Language: Analysis of Onomatopoeias Using a Tactile Display," presented at the The Third International Symposium on Mobiligence, Hyogo, Japan, 2009.
[39] R. Dawkins, *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe without Design*. New York, NY: W. W. Norton & Company, Inc., 1986.
[40] H. Takagi, "Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation," *Proceedings of the IEEE,* vol. 89, pp. 1275-1296, 2001.
[41] M. V. Sanchez-Vives and M. Slater, "From presence to consciousness through virtual reality," *Nature Reviews Neuroscience,* vol. 6, pp. 332-339, 2005.
[42] B. Lenggenhager, T. Tadi, T. Metzinger, and O. Blanke, "Video Ergo Sum: Manipulating Bodily Self-Consciousness," *Science,* vol. 317, pp. 1096-1099, 2007.
[43] C. Coelho, J. Tichon, T. J. Hine, G. Wallis, and G. Riva, "Media Presence and Inner Presence: The Sense of Presence in Virtual Reality Technologies," in *From Communication to Presence: Cognition, Emotions and Culture towards the Ultimate Communicative Experience. Festschrift in honor of Luigi Anolli*, G. Riva, M. T. Anguera, B. K. Wiederhold, and F. Mantovani, Eds., ed Amsterdam, Netherlands: IOS Press, 2006, pp. 25-45.
[44] C. Heeter, "Being there: the subjective experience of presence," *Presence: Teleoperators and Virtual Environments,* vol. 1, pp. 262-271, 1992.
[45] P. Brown, "The Mechanization of Art," in *The Mechanical Mind in History*, P. Husbands, O. Holland, and M. Wheeler, Eds., ed Cambridge, MA: MIT Press, 2008, pp. 259-282.
[46] T. Ikegami, "Studying a Self-Sustainable System by Making a Mind Time Machine," in *Workshop on Self-sustaining Systems*, Tokyo, Japan, 2010, pp. 1-8.
[47] A. B. Kocaballi, P. Gemeinboeck, and R. Saunders, "Enabling New Forms of Agency Using Wearable Environments," presented at the Designing Interactive Systems, Aarhus, Denmark, 2010.
[48] R. Chrisley, T. Froese, and A. Spiers, "Engeering conceptual change: The Enactive Torch," presented at the Workshop on Philosophy and Engineering, London, UK, 2008.
[49] R. Chrisley and J. Parthemore, "Synthetic Phenomenology: Exploiting Embodiment to Specify the Non-Conceptual Content of Visual Experience," *Journal of Consciousness Studies,* vol. 14, pp. 44-58, 2007.

Contextual Affect Modeling and Detection from Open-ended Text-based Dramatic Interaction

Li Zhang[1]

Abstract. Real-time contextual affect detection from open-ended text-based dialogue is challenging but essential for the building of effective intelligent user interfaces. In this paper, we focus on context-based affect detection using emotion modeling in personal and social communication context. Bayesian networks are used for the prediction of the improvisational mood of a particular character and supervised & unsupervised neural networks are employed respectively for the deduction of the emotional implication in the most related interaction context and emotional influence towards the current speaking character. Evaluation results of our contextual affect detection using the above approaches are provided. Generally our new developments outperform other previous attempts for contextual affect analysis. Our work contributes to the conference themes on sentiment analysis and machine understanding.

1 INTRODUCTION

Online interaction shows great potential to promote communication of people from different cultures and with physical barriers. It is even beneficial to (disadvantaged) young people to engage in such an online social interaction to have personalized learning/training experience. Thus our research has been focused on the production of intelligent agents with emotion and social intelligence. Since affect interpretation and detection play important roles in how effectively an intelligent agent is able to help users, we have made attempts in detecting affect from open-ended users' input previously and interpreting affect using context profiles has recently drawn our research attention.

Briefly, in our previous work, we developed online multi-user role-play software that could be used for education or entertainment. In this software young people could interact online in a 3D virtual drama stage with others under the guidance of a human director. In one session, up to five virtual characters are controlled on a virtual stage by human users ("actors"), with characters' (textual) "speeches" typed by the actors operating the characters. An intelligent conversational agent, EMMA, has been created to interact with the human characters, assist the human director to keep the general spirit of the scenarios for improvisation and stimulate the improvisation by detecting affect from the human characters' text input. The intelligent agent has been equipped with the capabilities of detecting a wide range of affect, including basic and complex emotions and recognizing affect from a few metaphorical language phenomena (e.g. affect as external entities metaphor ("Joy ran through me"), the food metaphor and the cooking metaphor ("She knew she was fried when the teacher handed back her paper")). The animation engine adopts the detected affect implied in users' text input to produce emotional gesture animation for the users' avatars. The conversational AI agent also provides appropriate responses based on the detected affect from users' input in order to stimulate the improvisation. In our application, we used several scenarios for testing including the Homophobic bullying[2] and Crohn's disease[3] scenarios.

Our previous affect detection has been performed solely based on the analysis of individual turn-taking user input. Thus the context information has been ignored. However, since open-ended natural language input could be ambiguous, sometimes contextual profiles are required in order to further justify the affect implied by the speaking character. Thus for affect interpretation in a comparatively simple scenario (e.g. where relationships between characters are fairly consistent throughout the improvisation), we previously used Markov chains for the modeling of the improvisational mood for individual characters by recommending a most related discussion context to the test situation. For affect analysis in comparatively complex scenarios (e.g. relationships between characters evolve throughout), we have also used a supervised neural network application with the assistance of fuzzy logic for the modeling of local emotional context for individual characters. However, both approaches for personal emotional context modeling are constrained to the scenarios used and cannot perform across different scenarios although the neural network approach provides a more flexible way for emotion prediction in a comparatively complex story context.

In order to make our contextual affect detection applicable across different scenarios or even to other interaction without any scenario constrictions, in this paper we present the modeling of personal emotional context (the improvisational mood of a particular character) using Bayesian networks and social

[1] School of Computing, Univ. of Teesside, TS1 3BA, UK. Email: {l.zhang}@tees.ac.uk.

[2] We briefly introduce this scenario in the following. The character Dean (16 years old), captain of the football team, is confused about his sexuality. He has ended a relationship with a girlfriend because he thinks he may be gay and has told her this in confidence. Tiffany (ex-girlfriend) has told the whole school and now Dean is being bullied and concerned that his team mates on the football team will react badly. He thinks he may have to leave the team. The other characters are: Rob (Dean's younger brother) who wants Dean to say he is not gay to stop the bullying, Lea (Dean's older sister) who wants Dean to be proud of who he is and ignore the bullying, and Mr Dhanda (PE Teacher) who needs to confront Tiffany and stop the bullying.

[3] In the Crohn's disease scenario, the sick leading character, Peter, needs to discuss pros and cons with friends and family about his life changing operation in order to make a decision. Janet (Mum) wants Peter to have the operation. Arnold (Dad) is not able to face the situation. Other characters are Dave (Peter's best friend) and Matthew (Peter's younger brother).

communication context (general emotional inclination in the recent interaction context) using an unsupervised neural network algorithm, Adaptive Resonance Theory. Moreover, we also employ a supervised learning neural network, backpropagation, to explore emotional influence from other characters to the current speaking character to further justify the affect derived from both personal and social emotion context modeling. Although training data is needed for backpropagation, we use emotional appraisal examples gathered both from transcripts across different scenarios and from common-sense knowledge as training data to enable the system to learn about emotional influence caused by other participants. The evaluation results indicate that the new context-based affect sensing with the integration of the above three components outperforms our previous alternative attempts and it is also capable of performing effective affect interpretation across different scenarios.

The paper is arranged as follows. We discuss related work in section 2, new development on contextual affect detection in section 3, and evaluation results and future directions in section 4.

2 RELATED WORK

Much research has been done on creating affective virtual characters in interactive systems. Emotion theories, particularly that of Ortony, Clore and Collins [1] (OCC), have been used widely therein. Picard's work [2] made great contributions to building affective virtual characters overall. Prendinger and Ishizuka [3] used the OCC model in part to reason about emotions and to produce believable emotional expressions. Mehdi et al. [4] combined a widely accepted five-factor model of personality, mood and OCC in their approach for the generation of emotional behaviour for a fireman training application. Gratch and Marsella [5] presented an integrated model of appraisal and coping, to reason about emotions and to provide emotional responses, facial expressions, and potential social intelligence for virtual agents. Egges et al. [6] provided virtual characters with conversational emotional responsiveness with the assistance of emotion and personality modeling. Aylett et al. [7] also focused on the agent development of affective behaviour planning.

Recently textual affect sensing has also drawn researchers' interests. ConceptNet [8] is a toolkit to provide practical textual reasoning for affect sensing for six basic emotions, text summarization and topic extraction. Shaikh et al. [9] provided sentence-level textual affect sensing to recognize evaluations (positive and negative). They adopted a rule-based domain-independent approach, but haven't made attempts to recognize different affective states from open-ended text input. Although Façade [10] included shallow natural language processing for characters' open-ended utterances, the detection of major emotions, rudeness and value judgements is not mentioned. Zhe and Boucouvalas [11] demonstrated an emotion extraction module embedded in an Internet chatting environment. It used a part-of-speech tagger and a syntactic chunker to detect the emotional words and to analyze emotion intensity. The detection focused only on emotional adjectives and first-person emotions, and did not address deep issues such as figurative expression of emotion. There is also work on general linguistic cues useful for affect detection (e.g. Craggs and Wood [12]).

Context-sensitive approaches have also been attempted to sense affect and emotion. Ptaszynski et al. [13] developed an affect detection component with the integration of a web-mining technique to detect affect from users' input and verify the contextual appropriateness of the detected emotions. The detected results made an AI agent either sympathize with the human player or disapprove the user's emotional experience by the provision of persuasion. However, their system targeted conversations only between an AI agent and one human user in non-role-playing situations, which greatly reduced the complexity of the modeling of the interaction context. Wallis et al. [14] also discussed different methodologies of conversation analysis to illustrate what they believed to be a major deficiency in many current approaches to human-machine dialogue. They also produced a theory about how language worked from applied linguistics and used it in an iterative process to improve conversations between a robot and human users.

Our work focuses on the following aspects: (1) real-time affect sensing for basic and complex emotions in improvisational role-play situations from literal and metaphorical expressions; (2) affect interpretation based on context profiles; and (3) affect detection across scenarios.

3 CONTEXTUAL AFFECT SENSING

Our original system has been developed for age 14-16 secondary school students to engage in role-play situations under loose scenarios in virtual social environments [15, 16]. Without pre-defined constrained scripts, the human users could be creative in their role-play within the highly emotionally charged scenarios. The language used by the secondary school students during their role-play is highly diverse with various online chatting features, such as abbreviations (e.g. 'den' (then), 'r' (are)), acronyms (e.g. 'lol' (laughing out loud)) and slang. Our previous work had pre-processing procedures to deal with abbreviations, acronyms, misspellings and slang [15]. Metaphorical language has also been used to convey emotions and feelings. In our previous work, we also detected affect from food metaphor ("u r a peach", "X is walking meat", "X has a pizza face") and cooking metaphor ("the lawyer grilled the witness on the stand", "I knew I was cooked when the teacher showed up at the door").

However, the affect detection processing we conducted previously only identifies emotions from the analysis of individual turn-taking input. Relevance theory suggested by Sperber & Wilson [17, 18] mentioned that "comprehension requires a common base of a cognitive environment that is shared by speaker and audience" and a lot of information needs to be inferenced by the audience to achieve the communication intention. Schnall [19] also further stated that the intention of communication is to achieve the greatest possible cognitive outcome with the smallest possible processing effort, i.e. "to communicate only what is relevant". From the above perspectives, emotion and interaction context in our application has great potential to create such a relevant cognitive environment to facilitate effective communication. Thus affect detection using contextual profiles draws our research attention.

We also gathered some linguistic indicators for contextual communication in the transcripts, including (i) imperatives, which are often used to imply negative or positive responses to the previous speaking characters (e.g. "shut up"), (ii) prepositional phrases (e.g. "by who?"), semi-coordinating conjunctions (e.g. "so we are good then"), subordinating conjunctions ("because Lisa is a dog") and coordinating

conjunctions ('and', 'or' and 'but'). These indicators are normally used by the current speaker to express further opinions or gain further confirmation from the previous speakers. Also other indicators include (iii) short phrases for questions (e.g. "where?", "who is Dave"), (iv) character names (e.g. "Mrs Parton, say something"); and finally (v) some other common contextual indicators shown in Internet relay chat (such as 'yeah/yes+ a sentence', "I think so", "thanks", etc). These indicators acted as signals for the activation of the contextual affect analysis in our application previously.

However, there are still cases ("ur a batty 2 then okay", "the rest dropped out cuz they didn't want to play with a gay", "I want to talk about it now") that contextual affect analysis fails to be activated due to the limitation of the above indicators. In order to deal with such difficulties, we have focused on inputs with structures of (vi) 'subjects + verb phrases + objects'. We notice that such statement structures with first person subjects tend to convey strong opinions ("I want to talk about it now", "I am the only thing this football team has", "I hate school"), while inputs with such structures and second person subjects are inclined to convey insulting or compliment ("u r an angle", "u aint needed here", "u know dean! go boy!", "u r not my dad/friend/mate", "u r a batty 2 then okay", "u r an idiot" etc). Moreover, for the Homophobic bullying scenario used in our application, there is other contextual communication with statement structures, which implies emotional implication, such as "BATTY MANZ CANT RUN", "the rest dropped out cah they didn't wanna play with a gay", "every1 iz avoiding u", "sexually and personality r 2 different things" etc. Thus we also consider inputs with such statement structures as signals for contextual communication.

At the test stage, first we detect affect for each input solely based on the analysis of the input itself. The contextual affect sensing presented in the following will be activated when an input conveys 'neutral' with at least one linguistic indicator.

Personal Emotion Context Modeling

Lopez et al. [20] has suggested that context profiles for affect detection included social, environmental and personal contexts. In our study, personal context may be regarded as one's own emotion inclination or improvisational mood in communication context. We believe that one's own emotional states have a chain reaction effect, i.e. the previous emotional status may influence later emotional experience. We make attempts to include such effects into emotion modeling. Bayesian networks are used to simulate such personal causal emotion context. In the Bayesian network example shown in Figure 1, we regard the first emotion experienced by a particular user as A, the second experienced emotion as B, and the third as C. We assume that the second emotional state B, is dependent on the first emotional state A. Further, we assume that the third emotional state C, is dependent on both the first and second emotional states A and B. In our application, given two or more most recent emotional states a user experiences, we may predict the most probable emotion this user implies in the current input using a Bayesian network.

Briefly, a Bayesian network employs a probabilistic graphical model to represent causality relationship and conditional (in)dependencies between domain variables. It allows combining prior knowledge about (in)dependencies among variables with observed training data via a directed acyclic graph. It has a set of directed arcs linking pairs of nodes: an arc from a node X to a node Y means that X (parent emotion) has a direct influence on Y (successive child emotion). Such causal modeling between variables reflects the chain effect of emotional experience. It uses the conditional probabilities (e.g. P[B|A], P[C|A,B]) to reflect such influence between prior emotional experiences to successive emotional expression. The following network topology has been used to model personal contextual emotional profiles in our application.

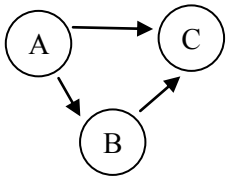


Figure 1. An emotion Bayesian network

In Figure 1, conditional probabilities are needed to be calculated for the emotional state C given any combination of the emotional states A and B. Theoretically, emotional states A and B could be any combination of potential emotional states, so does the successive emotional state C. In our application, we mainly consider the following 10 most frequently used emotional states for contextual affect analysis including 'neutral', 'happy', 'approval', 'grateful', 'caring', 'disapproval', 'sad', 'scared', 'threatening', and 'angry'. Any combination of the above emotional states could be used as prior emotional experience of the user thus we have overall 100 (10 * 10) combinations for the two prior emotions. Also each conditional probability for each potential emotional state given two prior emotional experiences (such as P[happy| A,B], P[approval| A,B] etc) will be calculated. The emotional state with the highest conditional probability is selected as the most probable emotion the user conveys in the current turn-taking.

Moreover, it is beneficial that Bayesian networks do not require us to gather training data from other sessions of the same scenarios beforehand. We can simply use the emotional states experienced by a particular character throughout one improvisation as the prior input to the Bayesian network so that our system may learn about this user's emotional trend gradually for future prediction without any constrains set by the training data or scenario related information.

Thus we take a frequency approach to determine the conditional probabilities. When an affect has been detected from the user's input, we increment a counter for that expressed emotion given the two prior implied emotional states. An example conditional probability table has been shown in Table 1.

		Probability of the predicted emotional state C being:			
Emotion A	Emotion B	Happy	Approval	...	Angry
Happy	Neutral	P00	P01	...	P09
Neutral	Angry	P10	P11	...	P19
Disapproval	Disapproval	P20	P21	...	P29
Angry	Angry	P30	P31	...	P39

Table 1. An example conditional probability table for emotions expressed for a particular character

For a prediction for an emotion state mostly likely implied by one particular character, the two prior recent emotional states are used to determine which row to consider in the conditional probability matrix, and select the column with the highest

conditional probability as the final output. Also the frequencies are sufficient to use to calculate probabilities when required thus no previous training needed. In our application, the frequencies of emotion combinations in a 100 * 10 ((A*B)*C) matrix are produced dynamically.

We extract the following example interaction from the Homophobic bullying scenario. Based on the affect detection purely from the analysis of each individual input, we assigned an emotional label for each input as the first step.

1. Tiffany Tanktop: sorry, all io could hear was I'M A BIG GAY [insulting/angry]
2. Mr. Dhanda: TIFFANY I WILL....GET YOU EXPENDED IF YOU DONT FOLLOW MY ORDERS! YOU HOMO-FOBIC [angry]
3. Rob Hfuhruhurr: tiffany is wierd lol y she spreadn rumors etc???? [disapproval]
4. Tiffany Tanktop: there not rumours...its the truth [disapproval]
5. Tiffany Tanktop: GGGGAAAYYYYY! [insulting/angry]
6. Mr. Dhanda: TIFFANY STOP IT NOW!!! [angry]
7. Mr. Dhanda: ILL BANG YOU [angry]
8. Rob Hfuhruhurr: god leav hm alone!!! [angry]
9. Tiffany Tanktop: ONCE A BATTY ALWAYS A BATTY [neutral] -> [angry]

Also we derive 'neutral' for the very last input without any contextual inference. Since the input is a simplified statement sentence (a linguistic contextual indicator), the context-based affect analysis will be activated to adjust the affect conveyed in the last input in the above example. The emotional profile of Tiffany (angry (1st), disapproval (4th), angry (5th)) is used to construct the Bayesian probability matrix. Then the conditional probability of P[C|angry, disapproval, angry] is calculated for each potential emotion C. Finally 'angry' is regarded as the most probable emotion implied in the input "ONCE A BATTY ALWAYS A BATTY".

In this way, we can produce emotion modeling for each individual character within the same and across scenarios. However, other social emotional contextual profiles haven't been considered yet. In the following section, we introduce emotion sensing in communication context and emotional influence of other characters to the current speaking character to justify the above reasoning.

Emotional Implication in Related Context

As discussed earlier, "comprehension requires a common base of a cognitive environment that is shared by speaker and audience" and a lot of information needs to be inferenced by the audience to achieve the communication intention [17, 18]. Thus an effective user input should be meaningful in its most related context environment. If the emotion implication (positive/negative/neutral) of such most relevant social context could be recognized during the interaction, it is very helpful to justify the affect inferenced by the personal emotional context. This motivates us to derive general emotional implication in the most recent communication context by employing an unsupervised neural network, i.e. Adaptive Resonance Theory -1 (ART-1).

Such unsupervised learning algorithms deal with object identification and recognition generally as a result of the interaction of 'top-down' observer expectations with 'bottom-up' sensory information. ART-1 in particular has the ability to maintain previously learned knowledge ('stability') while still being capable of learning new information ('plasticity'). Although it mainly accepts binary input vectors, this is sufficient enough in our application currently. In our application, it would be beneficial that the positive/negative context prediction modeling is capable of both retaining previously learned information (e.g. the sensing of positive or negative context in a particular scenario) and in the meantime integrating newly discovered knowledge (e.g. the sensing of such context across different scenarios). Such capability may allow the emotional social context modeling to perform across scenarios. Also, ART-1 has an advanced ability to create a new cluster when required with the assistance of a vigilance parameter. It may help to determine when to cluster an emotion feature vector to a 'close' cluster or when a new cluster is needed to accommodate this emotion vector.

In our application, we use the evaluation values (positive and negative) and neutralization of the most recent several turn-taking as the input to ART-1. In detail, for each user input, we convert its emotional implication into pure positive/negative and use three binary values (0/1) to represent the three emotional implications: neutral, positive and negative. For example, for the input from Arnold in the Crohn's disease scenario, "dont boss me about, wife [angry]" when the wife character, Janet, was too pushy towards the husband character, Arnold. We use '0 (neutral), 0 (positive), and 1 (negative)' to indicate the emotional inclination ('angry' -> 'negative') in the user input. In the previous example transcript from the bullying scenario shown in the above section, for the very last input (the 9th input from Tiffany), we previously only interpreted 'neutral' based on the analysis of the input itself. The personal emotional context prediction based on Bayesian networks is used and derives 'angry' as the most probable affect implied in it. However, we still need to resort to the inference of the general emotional trend in the most related interaction context to justify the previous prediction. In our application, we take the previous four inputs, from Tiffany (5th input), Mr Dhanda (6th and 7th input) and Rob (8th input), as the most related social context for prediction since there are up to 5 characters involved in each session normally. Since Tiffany implies 'angry' (binary value combination for neutral, positive and negative: 001) by saying "GGGGAAAYYYYY!", Mr Dhanda also indicating 'angry' (001) in both of his input: "TIFFANY STOP IT NOW!!!" and "ILL BANG YOU", followed by another 'angry' (001) input from Rob "god leav hm alone!!!", we have used the following emotion vector to represent this most related discussion context: '001 001 001 001 (Tiffany: 5th, Mr Dhanda: 6th & 7th and rob: 8th)'. This feature vector is used as the input to ART-1 to determine if the input context is 'positive/negative'. Similarly, we gather a set of such emotion vectors across scenarios. ART-1 classifies them into different groups based on their similarities and differences.

Briefly, we begin the algorithm with a set of unclustered emotional context feature vectors (emotional context) and some number of clusters (positive/negative/neutral categories). For each emotional feature vector, it makes attempts to find the cluster to which it's closest. A similarity test and a vigilance test calculate how close each emotional feature vector to the positive/negative/neutral cluster vectors. If an emotional feature

vector fails the similarity or vigilance test for all the available clusters, then a new cluster is created for this emotion vector. In our application, we gradually feed emotional context feature vectors to ART-1, which will not only remain the previous classification of positive or negative context in a particular scenario, but also indefinitely integrate new positive/negative context extracted from other interaction across scenarios. Suppose we have the following emotional contexts from the Crohn's disease scenario, classified previously by the algorithm into three categories:

Class 0 contains:

[1 0 0 0 0 1 0 0 1 0 0 1] negative1 (neutral, sad, disapproving and sad)

[1 0 0 0 1 0 0 0 1 0 0 1] negative2 (neutral, approving, disapproving and angry)

Class 1 contains:

[0 0 1 0 0 1 1 0 0 1 0 0] negative3 (angry, angry, neutral and neutral)

[1 0 0 0 1 0 1 0 0 0 0 1] positive2 (neutral, caring, neutral and disapproval)

[1 0 0 1 0 0 1 0 0 1 0 0] neutral1 (neutral, neutral, neutral and neutral)

Class 2 contains:

[0 1 0 0 1 0 0 1 0 1 0 0] positive1 (happy, grateful, happy and neutral)

Since ART-1 is not aware which label it should use to mark the above categorization although it classifies the emotional feature vectors based on their similarities and differences and achieves the above classification, a simple algorithm is used to assign labels (positive/negative/neutral context) to the above classification based on the majority vote of the evaluation values of all the emotional states shown in each emotional feature vector in each category. For example, Class 0 has assigned 2 emotional vectors and most of the emotional states in all the feature vectors in this category are 'negative', therefore it is labeled as 'negative context'. Similarly Class 1 is recognised as 'neutral context' with Class 2 identified as 'positive context'. If we add the above example context from the bullying scenario as a new feature vector, '001 001 001 001' (contributed by Tiffany, Mr Dhanda and Rob), to the algorithm, we have Class 0 updated to accommodate the newly arrived emotional vector as output. Thus the new emotion vector is 'classified' as a 'negative context'. Therefore, the last input from Tiffany ("ONCE A BATTY ALWAYS A BATTY") is more likely to contain 'anger' with a strong intensity indicated by the capitalization in a comparatively 'negative' context, which further justifies the inference of the personal context modeling. However, sometimes the unsupervised algorithm may classify a test emotion context as 'neutral'. Therefore we still need to resort to other social context modeling, e.g. the emotional influence of other characters to the current speaking character, to assist affect interpretation.

Emotional Influence of Other Characters

The simulation of one's own improvisational mood is important but the Bayesian approach still needs emotional profiles of each character as input in order to deduce affect conveyed in the current input. However when such personal emotional profile is not available (such as at the beginning of the improvisation) or the unsupervised neural nets fail to discover any emotional implication in the most related context (or 'neutral context'), we need to resort to the modeling of other characters' emotional influence to derive/adjust the affect implied by the current speaking character. E.g., the emotional context contributed by friend or enemy characters, may (dramatically) affect the speaking character's emotional expression. Therefore we also consider supervised learning neural networks, backpropagation, to model such effect to the current speaker and use two most recent emotions expressed by two other participant characters as input.

For example, in the above example transcript shown in the section of 'Personal Emotion Context Modeling', the most recent two other participant characters inputs, are from Mr. Dhanda ("ILL BANG YOU") and Rob Hfuhruhurr ("god leav hm alone!!!"). Their implied emotional states ('angry' (Mr Dhanda) and 'angry' (Rob)) are used as the input to Backpropagation. Since it is a supervised learning algorithm, we use emotional context gathered both from transcripts across different scenarios and from common-sense knowledge as training data. We obtain 'angry' as the predicted most probable affect conveyed in Tiffany's last input, which strengths our previous prediction performed by the personal and unsupervised social emotional context modeling. In the next section, we further discuss how the affect sensing component functions in real-time interaction using more examples.

Real-Time Contextual Affect Sensing

We discussed personal subjective emotion context modeling using Bayesian networks, the prediction of emotion implied in the discussion context using Adaptive Resonance Theory and emotional influence of other characters using Backpropagation. These three components integrate with one another linearly to sense affect from emotional ambiguous context in our application. In the following, we provide another example transcript followed the previous one from the bullying scenario to show how they work together to derive affect from dramatic improvisation by combining every weak affect indicator into a stronger interpretation.

9. Tiffany Tanktop: ONCE A BATTY ALWAYS A BATTY [neutral] -> [angry]
10. Rob Hfuhruhurr: HOMOSEXUAL NOT BATTY [disapproval]
11. Tiffany Tanktop: shut up man lea [angry]
12. Lea Hfuhruhurr: wat ur smellin tiffany, is ur mouth!lol [neutral] -> [insulting/angry]
13. Tiffany Tanktop: go get ya hair did [neutral] -> [angry]
14. Dean Hfuhruhurr: lol [happy]
15. Tiffany Tanktop: ur a batty 2 then okay [neutral] -> [insulting/angry]
16. Tiffany Tanktop: its sorted [neutral] -> [angry]
17. Lea Hfuhruhurr: dean, ur a gr8 football player. dnt let no1 stop u livin ur dream [approval]

First of all, affect is detected for each input without using any contextual inference. If an input contains any of the above discussed contextual linguistic indicators and is detected as non-emotional, then the contextual affect analysis is activated to further justify the affect implied in it.

Also based on our previous inference, the 9th input from Tiffany indicates 'angry' with high confidence. The affect interpretation based on the analysis of individual input has

detected ‘disapproval’ and ‘angry’ respectively for the 10th and the 11th input. The 12th input has been derived as ‘neutral’ and contains a linguistic contextual indicator, a statement sentence. Thus the contextual affect detection is activated. Since this is the very first input from Lea, we do not have any emotional profile of this character at this stage as input to the improvisational mood prediction using Bayes. However, we can still resort to the social emotional context prediction and emotional influence of other characters to further justify the affect conveyed in this ‘neutral’ input. ART is used to sense the current emotional context (‘angry (8th)’, ‘angry (9th)’, ‘disapproval (10th)’, ‘angry (11th)’). The input context is represented as follows:

001 001 001 001 [angry, angry, disapproval, and angry]

As discussed earlier, ART-1 has the ability to retain the previous knowledge and classification, learn & make the prediction for the newly arrived data and predicts the current input as another ‘negative’ context. Also the Backpropagation algorithm is used to determine which emotional state Lea is most likely to experience using the emotional profiles of other characters: ‘disapproval (Rob: 10th)’ and ‘angry (Tiffany: 11th)’ as input and emotional state ‘angry’ has achieved the highest probability as output. Thus Lea more likely implies ‘angry’ in a ‘negative’ interaction context in the 12th input. Similarly, we detect ‘neutral’ in the 13th input from Tiffany and also it carries one of the linguistic indictors for contextual communication (imperative), which indicates it may be caused by contextual interaction. Therefore contextual affect sensing is activated again. Briefly, the improvisational mood modeling, emotion sensing in most related context and emotional influence modeling of other characters have been employed to uncover the affect implied in the current input. Then we conclude the 13th input is again more likely to imply ‘anger’ rather than ‘neutral’ with a ‘negative’ interaction context.

Moreover, we sense ‘neutral’ in the 15th input without any context inference. Since it (“ur a batty 2 then okay”) is a statement sentence with a structure of ‘second person + copular form’, which has potential to indicate insulting or compliment as mentioned earlier, we activate the contextual affect analysis as well. The following analysis is applied.

1. Emotional input profile of Tiffany: ‘angry (1st)’, ‘disapproval (4th)’, ‘angry (5th)’, ‘angry (9th)’, ‘angry (11th)’ and ‘angry (13th)’ -> Bayesian networks -> ‘angry’ as output.
2. Emotion sensing in social context: ‘angry (11th)’, ‘angry (12th)’, ‘angry (13th)’ and ‘happy (14th)’, represented as ‘001 001 001 010’ -> ART -> ‘negative context’.
3. Emotional influence of other characters: ‘angry (Lea: 12th)’ and ‘happy (Dean: 14th)’ -> Backpropagation -> ‘neutral’ as output, which means that others’ most recent contribution affects little to the current speaking character.
4. However, due to the sensed ‘negative’ context, the input is more likely to convey ‘angry’.

In this way, by considering the potential improvisational mood one character is in, general emotional inclination in the closely related context and other characters’ emotion influence, our affect detection component has been able to inference emotion based on context in real-time interaction. After the description of various affect processing components, the overall affect detection model is shown in Figure 2.

4 EVALUTION AND CONCLUSION

We carried out user testing with 220 secondary school students in the UK schools. Generally, our previous statistical results based on the collected questionnaires indicate that the involvement of the AI character has not made any statistically significant difference to users’ engagement and enjoyment with the emphasis of users’ notice of the AI character’s contribution throughout. Briefly, the methodology of the testing is that we had each testing subject have an experience of both scenarios, one including the AI minor character, EMMA, only and the other including the human-controlled minor character only. After the testing sessions, we obtained users’ feedback via questionnaires and group debriefings. Improvisational transcripts were automatically recorded to allow further evaluation of the affect detection component.

We also produce a new set of results for the evaluation of the updated affect detection component with context-based interpretation based on the analysis of some recorded transcripts of Homophobic bullying scenario. Generally two human judges marked up the affect of 200 turn-taking user input from the recorded 4 transcripts of this scenario. In order to verify the efficiency of the new developments, we provide Cohen’s Kappa inter-agreements for EMMA’s performance with and without the new developments for the detection of the most commonly used 10 affective states. In the bullying scenario, EMMA played a minor bit-part character (the teacher: Mr Dhanda). The agreement for human judge A/B is 0.45. The inter-agreements between human judge A/B and EMMA with the new developments are respectively 0.43 and 0.35, while the results between judge A/B and EMMA without the new developments are only respectively 0.39 and 0.30.

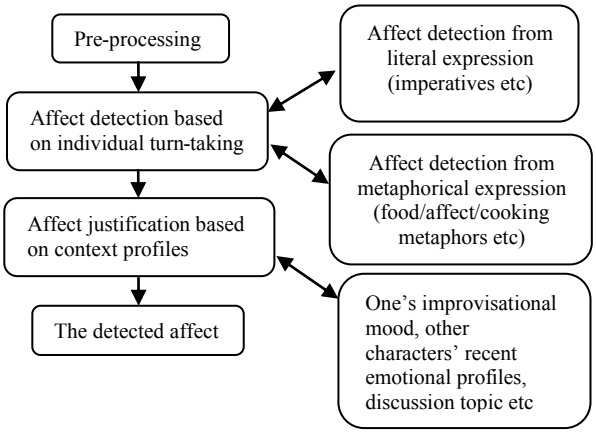


Figure 2. The overall affect detection model

Although future work is needed, the new developments on contextual affect sensing have improved EMMA’s performance comparing with the previous version. Moreover, we obtain the average accuracy rate 84.2% for the personal emotional context modeling using Bayesian networks, 68.5% for the emotion sensing in interaction context using unsupervised learning, and 58% for the emotional influence modeling of other characters using Backpropagation. Since our approach gathers every weak affect indicator to draw a stronger conclusion for affect

interpretation, the above three core results strengthen or reduce each other's effects for final affect annotation. Overall we obtain the average accuracy rate 88% for the contextual affect detection, while our previous contextual affect analysis using supervised neural nets & fuzzy logic achieved an average accuracy rate of 80% generally and the Markov chains' approach only obtained 69%.

Overall, we have made initial developments of an AI agent with emotion and social intelligence, which employs context profiles for affect interpretation using Bayesian networks, unsupervised and supervised neural network algorithms. Although the AI agent could be challenged by the rich diverse variations of the language phenomena used by the testing subjects and other improvisational complex context situations, we believe these areas are very crucial for development of effective intelligent user interfaces and our processing has made promising initial steps towards these areas. Also, the integration of these discussed approaches has great potential to derive affect in communication context which is closer to the user's real emotional experience. Another advantage of our implementation is that it has the potential to perform contextual affect sensing across different scenarios.

REFERENCES

[1] A. Ortony, G.L. Clore and A. Collins. *The Cognitive Structure of Emotions*. Cambridge, UK: Cambridge University Press. (1998)

[2] R.W. Picard. *Affective Computing*. The MIT Press. Cambridge MA. (2000).

[3] H. Prendinger and M. Ishizuka. Simulating affective communication with animated agents. In the *Proceedings of Eighth IFIP TC.13 Conference on Human–Computer Interaction*. Tokyo, Japan, pp.182–189. (2001).

[4] E.J. Mehdi, P. Nico, D. Julie and P. Bernard. Modeling character emotion in an interactive virtual environment. In *Proceedings of AISB 2004 Symposium: Motion, Emotion and Cognition*. Leeds, UK. (2004).

[5] J. Gratch And S. Marsella. A domain-independent framework for modeling emotion, *Journal of Cognitive Systems Research*, Vol. 5, pp.269–306. (2004).

[6] A. Egges, S. Kshirsagar & N. Magnenat-Thalman. A Model for Personality and Emotion Simulation, In *Proceedings of Knowledge-Based Intelligent Information & Engineering Systems* (KES2003), Lecture Notes in AI. Springer-Verlag: Berlin, 453-461. (2003).

[7] R. Aylett, S. Louchart, J. Dias, A. Paiva, M. Vala, S. Woods, L.E. Hall. Unscripted Narrative for Affectively Driven Characters. *IEEE Computer Graphics and Applications*. 26(3). 42-52. (2006).

[8] H. Liu & P. Singh. ConceptNet: A practical commonsense reasoning toolkit. *BT Technology Journal*, Volume 22, Kluwer Academic Publishers. (2004).

[9] M.A.M. Shaikh, H. Prendinger & I. Mitsuru. Assessing sentiment of text by semantic dependency and contextual valence analysis. In *Proceeding of ACII 2007*, 191-202. (2007)

[10] M. Mateas. Ph.D. Thesis. Interactive Drama, Art and Artificial Intelligence. School of Computer Science, Carnegie Mellon University. (2002).

[11] X. Zhe & A.C. Boucouvalas. Text-to-Emotion Engine for Real Time Internet Communication. In *Proceedings of International Symposium on Communication Systems, Networks and DSPs*, Staffordshire University, UK, 164-168. (2002).

[12] R. Craggs & M. Wood. A Two Dimensional Annotation Scheme for Emotion in Dialogue. In *Proceedings of AAAI Spring Symposium: Exploring Attitude and Affect in Text*. (2004).

[13] M. Ptaszynski, P. Dybala, W. Shi, R. Rzepka And K. Araki. Towards Context Aware Emotional Intelligence in Machines: Computing Contextual Appropriateness of Affective States. In *Proceeding of IJCAI 2009*. (2009).

[14] P. Wallis, V. Maier, S. Creer and S. Cuningham. Conversation in Context: what should a robot companion say? In *Proceedings of EMCSR*. (2010).

[15] L. Zhang, M. Gillies, K. Dhaliwal, A. Gower, D. Robertson & B. Crabtree. E-drama: Facilitating Online Role-play using an AI Actor and Emotionally Expressive Characters. *International Journal of Artificial Intelligence in Education*. Vol 19(1), pp.5-38. (2009).

[16] L. Zhang. Exploitation on Contextual Affect Sensing and Dynamic Relationship Interpretation. In *ACM Computer in Entertainment*. Vol.8, Issue 3. (2010).

[17] D. Sperber & D. Wilson. *Relevance: Communication and cognition* (2nd ed.). Oxford, UK: Blackwell. (1995).

[18] D. Wilson & D. Sperber, D. *Relevance Theory*. In G.Ward & L. Horn (Eds.), Handbook of Pragmatics (pp. 607–632). Oxford, UK: Blackwell. (2003).

[19] S. Schnall. The pragmatics of emotion language. *Psychological Inquiry*, 16, 28-31. (2005).

[20] J.M. Lopez, R. Gil, R. Garcia, I. Cearreta and N. Garay. Towards an Ontology for Describing Emotions. In *WSKS'08 Proceedings of the 1st world summit on The Knowledge Society: Emerging Technologies and Information Systems for the Knowledge Society*. (2008).

Digital Computation as Information Processing

Nir Fresco (student)[1]

Abstract. It is common in cognitive science to equate computation (in particular digital computation) with information processing. Yet, it is hard to find a comprehensive explicit account of concrete digital computation in information processing terms. An Information Processing account seems like a natural candidate to explain digital computation. After all, digital computers traffic in data. But when 'information' comes under scrutiny, this account becomes a less obvious candidate.

'Information' may be interpreted semantically or non-semantically, and its interpretation has direct implications for Information Processing as an objective account of digital computation. This paper deals with the implications of these interpretations for explaining concrete digital computation in terms of information processing. To begin with, I survey Shannon's classic theory of information, and then examine how 'information' is used in computer science. In the subsequent section, I evaluate the implications of how 'information' is interpreted for an Information Processing account. The key requirements for a physical system to compute are then fleshed out, as well as some of the limitations of such an account. Any Information Processing account must embrace an algorithm-theoretic apparatus to be a plausible candidate for explaining concrete digital computation.

1. INTRODUCTION

According to the Information Processing (hereafter, IP) account, for a system to be deemed a computing system it needs to process data, which carries information. It is often assumed, particularly in cognitive science discourse, that symbolic computation models can freely be described as information processing models. This is the motivation for this paper, which deals with the question whether concrete digital computation (i.e., digital computation as it is actualised in *physical systems*) can be adequately explained solely in information processing terms.

Furthermore, any resulting IP account hinges on the interpretation of 'information'. It can be interpreted semantically or non-semantically (and more specifically quantitatively as 'Shannon information' or Algorithmic Information). It is questionable whether an IP account of computation must presuppose semantic information. The important question is then whether computing systems traffic in semantic information inherently, or whether they traffic in data or non-semantic information, which in turn could be assigned some meaning by users[2].

A general account of Information *Processing* based on 'Shannon information' is outlined here in the context of concrete digital computation. I begin by surveying Shannon's classic theory of information in section 2, and then examine whether Algorithmic Information theory significantly changes the resulting IP account, in the third section. Subsequently, in section 4, I discuss the implications of how 'information' is interpreted for explaining concrete digital computation. The key requirements implied by the IP account for a physical system to perform digital computation are explicated in section 5. Eventually, in section 6, I examine the limitations of this account and argue that any IP account must embrace an algorithm-theoretic apparatus to be a plausible candidate for explaining concrete digital computation.

2. THE RECEIVED THEORY OF INFORMATION IN COMMUNICATION

The most influential theory of information in communication and engineering was introduced by Claude Shannon in 1948. He showed how information could be transmitted efficiently across communication channels by means of encoded messages. Shannon [1] attempted to solve the "fundamental problem of communication": finding the optimal manner by which messages from a source of information are exactly or approximately reproduced at their destination [2]. According to Norbert Wiener [3], one of the simplest unitary forms of information is the recording of a choice between two equiprobable basic alternatives. A sufficient condition for a physical system to be deemed a sender or receiver of information is the production of a sequence of symbols in a probabilistic manner.

Moreover, Shannon [1] and Wiener [3] analyse an information-generating system in terms of five essential components: an information source, a transmitter, a channel, a receiver and a destination. The information source produces a message to be communicated to the receiver. The transmitter operates on the message to produce a signal suitable for transmission over the channel, which is simply the medium of signal transmission. The receiver reconstructs the message from the signal. And the destination is the system for which the message is intended. So communication amounts to the source of information producing a sequence of symbols, which is then reproduced by the receiver to some degree of accuracy.

Nevertheless, 'Shannon information' does not entail any semantic content or meaning. Shannon's information theory approaches information syntactically as a physical

[1] School of History and Philosophy, University of New South Wales, Sydney, Australia. Email: *Fresco.Nir@Gmail.com*

[2] I do not mean to imply here (as John Searle would) that the idea of concrete computation requires something like a knower or an observer. A computing system will continue computing even in our absence, as long as it does not break and has a constant supply of energy to run.

phenomenon: whether and how much (not what) information is conveyed [4]. On the other hand, a stronger sense of 'information' (i.e., semantic information) entails that messages have specific meanings by representing how things are or could be.

According to Shannon's theory, 'information' is interpreted in the weaker sense. 'Shannon information' is different (but not distinct) from the ordinary usage of 'information'; it tells us nothing about the usefulness of or interest in a message. The basic idea is coding messages into a binary (or any other) system at the bare minimum of bits we need to send to get our message across. Even in this limited sense, the amount of 'information' conveyed is as much a property of our own knowledge as anything in the message. If we send the same message twice every time (a message and its copy), the information in the two messages is not the sum of that in each. Rather the information only comes from the first one (assuming it was successfully transmitted) [5].

The important aspect of 'Shannon information' is that the message is selected from a set of possible messages. A message, composed of symbols, is a physical structure discriminated by the probability of selecting it over other possible messages. So a nonsensical message composed of the sequence of symbols '%3-4Y7@*' could in essence generate more information than a meaningful message (in the ordinary use of 'information') such as 'daughter' in reply to some question. This could be the case, if the message '%3-4Y7@*' would be more "surprising" than 'daughter' [2]. Receiving a message containing the former string could change the recipient's circumstance from not knowing what *something was* to knowing what *it is*. The more possible messages a recipient could have otherwise received, the more "surprised" the recipient is when it gets that particular message [5].

3. INFORMATION IN COMPUTER SCIENCE

Shannon's theory of information is the reigning theory in communication and can adequately explain network communication between computers, encoding and decoding of messages, message transmission through data buses or network cables and so on. Algorithmic Information theory, which was introduced by Andrei Kolmogorov, Ray Solomonoff and Gregory Chaitin, deals with the complexity of data structures and can be described as the borderland where information and digital computation meet. It formally defines the *complexity* or the *informational content* of a data structure (e.g., a string) as the length of its shortest self-delimiting algorithm running on a Universal Turing Machine (henceforth, UTM) [6] [7]. The algorithmic information of any computable string is the length of the shortest algorithm that computes it on a UTM.

Moreover, Chaitin proposes to think of a computing system as a decoding device at the receiving end of a noiseless binary communications channel [6]. Its programs are thought of as code words, and the result of the computation (i.e., its output) as the decoded message. The programs then form what is called a "prefix-free" set so that successive messages (e.g., procedures) sent across the channel can be distinguished from one another. Still he acknowledges that Algorithmic Information has precisely the formal properties of Shannon's concept of information entropy.

Although Algorithmic Information is underpinned by classical computability theory, it too is non-semantic and quantitative as Shannon's theory. It interprets information and measures its quantities in terms of the computational resources that are needed to specify it [4]. Algorithmic Information measures the length of the shortest algorithm that computes a string. That algorithm may even be shorter than the output it produces. For example, representing π or e in a binary notation [7].

Algorithmic Information then may be deemed a competing notion of 'Shannon information' by allowing us to assign complexity values to individual strings and other data types [8]. Whilst Shannon's theory analyses the amount of information in a group of messages based on the probability of the messages, Algorithmic Information theory analyses the complexity of a string as a single message. The relative frequency of the message has no meaning, but there is some shortest program on a UTM that can produce this message. The length of this optimal program is an absolute measure for the amount of information in that message. Both Algorithmic Information and 'Shannon information' give rise to optimal compression codes for information. A bit string '01010101010101' can be compressed in Shannon's sense as *'01'=1;1111111*, or can be programmed in the Algorithmic Information sense as *for x = 1 to 7 write '01'* [8].

4. THE IMPLICATIONS OF INTERPRETING 'INFORMATION' FOR THE IP ACCOUNT

Most structural accounts of concrete computation assume that it is a type of information processing. These accounts consider the unique structural properties of computing systems (namely, their digital architectures) to be their distinctive feature. But in addition to the structural constraint, they also assume a semantic constraint on concrete computation: processing information [9]. Some connectionists, on the other hand, reject the structural constraint, and argue that information-processing properties of digital computation differentiate it from other causal and mechanical processes. On their view, concrete computation is explained in terms of the information transformed, represented, and stored in the process of computing [10].

Still, any current attempt to untangle concrete computation and IP must begin with a distinction between a weaker sense of 'information' (e.g., 'Shannon information' or Algorithmic Information) and semantic information. Using informational language in the stronger sense raises some problems in regard to concrete computation (e.g., does a computer process semantic information even in the absence of its user? is the meaning of the computer-processed information intrinsic? if some

information processing carries meaning and some does not, how are they distinguished? etc.).

Even if we interpreted the IP account in Shannon's sense, it would be hard to accept it as a satisfactory account of concrete digital computation. Brian Cantwell Smith [11] argues that since information theory is not a full analysis of information, it cannot be a solid basis for a comprehensive account of concrete computation. His argument relies on a semantic reading of both information [11] and computation [12], and if 'Shannon information' is not semantic, it cannot adequately explain computation. Moreover, Piccinini and Scarantino [2] maintain that it is not clear how 'Shannon information' is *processed*. Whether 'Shannon information' can be associated with a given vehicle does not depend on any specific physical properties. Instead, it is regarded as a selection of symbols from a given language according to the probability distribution of these symbols. Shannon information does not pertain to individual messages, and individual messages are those that may be created, and manipulated by digital computing systems.

Also, digital computation may be either deterministic or non-deterministic (e.g., probabilistic computation, random computation etc.). Still, most digital computing systems developed in the computer industry are deterministic, since their behaviour is repeatable and systematic. A dry run of a deterministic algorithm (using some test data) should systematically yield the same output when its input and initial state remain unchanged. The state-transitions of Shannon's communication model are probabilistic, whereas the transition probabilities of a Turing machine (hereafter, TM) are all set to 1. For every possible input, there is only one possible state into which the TM transitions [13].

Be that as it may, analysing digital computation using information-theoretic language could be very constructive. Smith [11] asserts that the IP account could indeed serve as the grounds for a plausibly comprehensive theory of concrete computation. But on his view several theoretical issues must first be addressed. Firstly, only a semantic theory of information stands a chance of doing justice to computation. Secondly, 'information' must be analysed in a manner that does not entail pan-informationalism. Otherwise, this could lead to a dangerous equivocation: if any object can be described in informational terms, then the nature of all objects is genuinely informational. At least prima facie, concrete digital computation being driven by the executed software seems most likely amenable to instructional information (e.g., do X if Y, otherwise halt) [14].

5. THE KEY REQUIREMENTS IMPLIED BY THE IP ACCOUNT

The key requirements for a physical system to perform digital computation implied by the IP account are fourfold: 1. having the capacity to send information, 2. having the capacity to receive information, 3. having the capacity to store and retrieve information, 4. having the capacity to process (or transform) information. The choice between a semantic and a non-semantic reading of 'information' affects both the characterisation of the resulting IP account and the key requirements it implies for computing systems, as will be shown below. The following discussion remains neutral on the semantic vs. non-semantic reading of 'information', unless specified otherwise.

The first key requirement implied by the IP account is the system having the capacity to send information. Whether 'information' is interpreted semantically or non-semantically, concrete computation requires a source of information to transmit the data. Regardless of the medium by which the data is transmitted (e.g., via data buses, network cables, etc.), the sender is responsible for the data transmission. The sender prepares the messages to be sent to the receiver and encodes them for transmission. To emphasise, in computing systems the sender and the source of information may be distinct entities. For instance, whilst the computer's main memory could be a source of information (e.g., a stored instruction), the memory controller is responsible for fetching the data from memory and transmitting it to the CPU. The memory controller acts as the sender, but not as the source.

Analogously, the second key requirement implied by the IP account is the system having the capacity to receive information. If the former requirement necessitated a sender to transmit the message, this requirement necessitates a receiver on the other end to accept it. The absence of a receiver on the other end means that the computation remains unexecuted (or in a suspend mode). For instance, an instruction, which was fetched by the main control unit from the memory, but not received by the receiver (the CPU in this case), will not be executed. Also, a computer program (the sender), which sends an input/output signal to the operating system (the receiver), will enter the suspend mode until its I/O request is acknowledged.

Furthermore, when construed as information processing, concrete digital computation is at best incomplete in the absence of either a receiver or a sender. Unlike a microphone acting as a sender of information even in the absence of a receiver (the audience), a computing system must have both a sender and a receiver that are well coordinated. The information contained in a message may indeed not depend on the receiver's learning something from it, or even being able to decode the message [15]. But if the receiver is absent or unable to decode the message in a computing system, the computation will be either incomplete or incorrect. Suppose that the CPU (the receiver) is unable to correctly decode the instruction from the main control unit (the sender), it will fail to execute the instruction hindering the overall computation.

The third key requirement implied by the IP account is the system having the capacity to store and retrieve information. Computing systems store and retrieve digital information, which can be thought of as a series of bits. The storage and retrieval of information in a computing system should be well synchronised, as one always presupposes the other. Without the system having the ability to retrieve the data, there is clearly very little sense to storing it in the first place.

Lastly, the fourth key requirement implied by the IP account is the system having the capacity to transform information. This requirement cannot be dismissed, as it the essence of processing of information. It is also the most problematic requirement, which becomes even more stringent when 'information' is interpreted semantically (as will be shown below). It is important to emphasise that transforming information does not amount to merely encoding and decoding information. Those are methods that typically preserve the information while converting it into a coded form and vice versa, and are useful in the communication of signals or messages. Transformation of information is more than that, it is characterised as the creation (e.g., a new database table containing salaries of employees), modification (e.g., giving some employees a pay rise) and destruction of information (e.g., deleting some records of employees, who left the company, from the system).

The transformation or processing of 'Shannon information' is problematic, because its focus is not on the content of individual messages. To restate Wiener's claim [3], a sufficient condition for a physical system to be deemed a sender or receiver of 'Shannon information' is the production of messages in a probabilistic manner. Processing 'Shannon information' can be the modification of the state or strings states that may result in changes of the conditional entropies among the states. It can also be the elimination of possibilities (reduction in uncertainty) represented by a signal or the introduction of redundancy to offset the impact of noise and equivocation. But again, sending the same message twice (to offset the impact of noise) does not yield information that is the sum of that in each. Similarly, elimination of redundancy does not reduce the underlying informational content that is conveyed.

Moreover, construing concrete digital computation as information processing requires more than merely communicating information in a non-deterministic manner. Telephones (not the *voice over internet protocol* systems) are information processing systems, but they are not digital computers [13]. They can be used to transmit information, but they certainly do not compute in any non-trivial sense. Computers do indeed encode, decode and transmit information, but they also perform tasks with inferential import (when I try to divide a number by 0, a good algorithm should yield an error message from the computing system). Yet, this requires a way of distinguishing the differences between the informational *contents* of the messages. Shannon's information provides the procedures for selecting messages, but it lacks this capacity [15].

Still, this ability to distinguish between different contents is necessary for modifying or adding new justified information. Shannon's information theory tells us about the probabilities associated with symbols from a given language, but it is indifferent to the content of the messages. For instance, the strings S_1 and S_2 have the same length (including that of their symbol constituents), but a different composition of symbol constituents. S_1= "All cars have four wheels"; S_2= "All cats have four ankles". Let us suppose that S_1 and S_2 are equiprobable (so in Shannon's sense, they are both potentially equally informative). Let S_3 be "Bumblebee is a car". By using Universal Instantiation, for example, one can infer some new justified information[3]: S_4= "Bumblebee has four wheels" from S_1 and S_3. This new information must also be true, if S_1 and S_3 are true (here enters semantic information again). It tells us something else about Bumblebee (that Bumblebee has four wheels).

On the other hand, S_2 and S_3 do not yield new justified information using Universal Instantiation (similar to S_4). One cannot validly infer any new singular statement about Bumblebee from the universal statement S_2, as Universal Instantiation does not apply to S_3 (for Bumblebee is not a cat). In order to apply rules of logic as a means of generating new true information, the symbolic constituents of strings must be distinguishable. But according to Shannon's information theory we may encode and transmit S_2 (rather than S_1) and S_3 to the recipient (since S_1 and S_2 are equiprobable). Yet, the recipient will have learned nothing new from S_2 and S_3 in this case.

Furthermore, when 'information' is construed semantically (as proposed by some philosophers) its transformation requirement becomes even more stringent. The syntactical manipulation of messages must be done in a manner that always preserves their semantics. Typically, rules that are applied in the transformation process must be truth preserving[4]. At the very least, new justified information has to be consistent with prior "known" information. If conjunction, for instance, is applied to add new justified information, then the conjuncts C_1 and C_2 must be neither contradictories nor contraries. Otherwise, their conjunction 'C_1 and C_2' would be false.

Likewise, when syllogistic rules are applied in the process of transforming semantic information, syllogistic fallacies must be prevented[5]. For instance, when deductive inference is used to validly infer P_3 from the premises P_1 and P_2 (where $P_1 \neq P_3$ and $P_2 \neq P_3$), then P_3 must be true to be deemed new (or modified) semantic information. Not only that, but the error detection mechanism employed by the computing system must be such that it verifies that every single premise (P_1 and P_2, in the example above) is true[6], even if the deductive argument is valid. Thus, to extract new semantic information, sufficient scrutiny is required to ensure its truth and coherence. The same principle also applies to other types of

[3] There is an ongoing debate regarding information in deductive inferences. Some, including John S. Mill and the logical positivists, have argued that logical truths are tautologies, and so deductive reasoning does not add any new information.

[4] Induction, abduction and non-monotonic logic do not abide by the same principle, and their application does not guarantee the truth of any new information that they potentially create. Both abductive reasoning and non-monotonic logic play an important role in artificial intelligence and should not be discounted, but they exceed the scope of this paper.

[5] In particular, when interpreting semantic information as being necessarily true [4] [15] [16].

[6] Immediate inferences from categorical propositions, for instance, do not require the same error verification mechanism. From the categorical proposition 'no dogs are cats', we can immediately infer that 'no cats are dogs' by swapping the predicate term and subject term of the original proposition. The truth of one of them guarantees the truth of the other.

transformation rules like existential generalisations, universal instantiations, inductive inferences and so on.

Still, digital computation will proceed (or fail) regardless of the truth-value of the information processed by the computing system. Gricean non-natural meaning of signs (e.g., three dings of the bus bell indicating that the bus is full) does not require a correspondence to the state of affairs in question (e.g., whether the bus is actually full). In a similar manner, non-natural information may be processed by the computing system without any correspondence to an external state of affairs. There is always a possibility that a computing system will produce an incorrect output as a result of a miscomputation[7] (i.e., a mistake in the computation process due to an error in the executed algorithm or a hardware malfunction). In that case, the only viable option is that the miscomputation misrepresents the state of affairs in question[8]. But from the system's "point of view", this wrong output has no less (or more) meaning than the correct output (which might correctly correspond to some state of affairs). Whether a computation represents some state of affairs or not is a contingent fact.

6. SOME LIMITATIONS OF THE IP ACCOUNT

When 'information' is interpreted narrowly the resulting IP account cannot adequately explain *how* computation is executed and how it differs from *miscomputation*. Any plausible account of concrete digital computation must be able to explain Turing computability, for it lays the ground rules for all existing digital computers as well as for programming languages. Any account of Turing computability has to at least be able to explain the three key algorithmic notions of input, output, and procedures. But an IP account of concrete computation, which is based on 'Shannon Information' or Algorithmic Information, does not adequately explain those three key notions.

'Shannon information', for one, only makes sense in the context of a set of potential messages that are communicated between a sender and a receiver and a probability distribution over this set [8]. There is no room for a probabilistic selection of messages in describing deterministic procedures. There must be a *specific set* of messages that are selected, encoded and transmitted in the same order in accordance with the specific steps of the procedure, regardless of the probabilities associated with each message (or its symbol constituents).

Whilst Algorithmic Information is indeed based on TMs, it is still insufficient as an IP account of concrete computation. Algorithmic Information theory's interest in TMs is limited to finding the shortest program that runs on a UTM and generates a particular string as its output. But the purpose of Algorithmic Information theory is simply to measure the amount of information conveyed by that string or its complexity, rather than being *about that program.* So the best one could hope for in relation to Algorithmic Information explaining a particular procedure is either measuring the information conveyed by that procedure (as a string) or determining whether it is the shortest one for generating the output it produces. Yet, many computer programs running on multitudes of different systems are neither the shortest nor the most efficient for achieving their tasks.

Another challenge for an IP account of concrete computation, which is based on 'Shannon information' or Algorithmic Information, is identifying miscomputations. Miscomputations that are the result of a hardware malfunction could be explained by some breakdown of the communication channel, for example. But other miscomputations resulting from errors by design or a malformed algorithm cannot be easily explained, since neither Shannon's information theory nor Algorithmic Information distinguishes messages by their *contents*.

Arguably, when information is interpreted semantically it must yield knowledge [4] [7] [11] [15] and that implies a further requirement for a semantic IP account of concrete computation. This additional requirement is that by processing information the computing system has to yield knowledge, which is either derived from its user (or programmer or interpreter) or intrinsic to the system. Plato defined knowledge as a true justified belief (which was widely accepted in modern philosophy[9]). Semantic information must tell us something true about some state of affairs, that is, yield knowledge. One option then is that this knowledge is derivative and used by the knower, who uses the information produced by the computing system[10]. Another option is that this knowledge is intrinsic to the computing system that traffics in information.

The latter option has been challenged by many philosophers [18] [19] [20] [21] [22] [23] and it is not at all clear that there is compelling evidence to support it. There is only a limited sense in which a digital computing system "understands" or "knows" something. A digital computer understands the semantics of its machine language. This understanding can be attributed to structural properties of the machine's architecture and language as well as causal links between bit patterns, memory addresses, primitive operations etc. Computers manipulate information that they need not understand, although they copy it, compare it with other information and change it

[7] Besides these two types of syntactic miscomputation, there is also the possibility of a semantic miscomputation relative to some task domain (e.g., the Roomba indoor cleaning robot that may malfunction eventually when it operates under abnormal operating conditions such as an airfield). However, this semantic miscomputation can also be reduced to either one of the syntactic miscomputations above.

[8] There is always also the remote possibility of a double negation. Suppose that the computing system did not correctly represent some state of affairs when the computation was initiated. But then the system's miscomputation incidentally results in a correct representation of that state of affairs.

[9] Edmund Gettier [17] has challenged Plato's widely accepted view of knowledge as Justified True Belief. He argued that truth, belief, and justification are not sufficient conditions for knowledge. He showed that a true belief might be justified but fail to be knowledge, because the belief might be true by sheer accident.

[10] Indeed, this option is no more problematic than an encyclopaedia yielding knowledge for its readers.

[24]. This is the basis for an internal meaning of its information processing.

But that does not imply that the computer manifests any beliefs that are associated with these operations. Suppose we replace the doorbell with a digital computer that emits the sounds: “someone is at the door”, only when someone pushes the door button. When someone pushes the door button, the computer picks up the information about it, processes it and delivers an output. However, this output is not a belief in someone being at the door, anymore than the doorbell would have believed that [15]. Roy Sorensen [25] makes a further distinction between information conveyed by assertions and displays. When an answering machine utters the sounds: "Mr. Smith is not at home", it simply displays this message, rather than assert it. The machine does not believe that Mr. Smith is not at home (he may even be home). Similarly, when a computer weather program displays a rainy weather forecast for tomorrow, it does not believe that it will rain tomorrow, although this output may be based on a reliable source of information. There is no intrinsic belief or knowledge in these information-processing systems.

7. CONCLUSION

Although the IP account, on the face of it, seems like a natural and promising candidate for explaining concrete digital computation, it is less than obvious. This is to a large extent dependent on how information is interpreted and what the resulting IP account is. Some argue that an adequate theory of information must give an account of information as semantic content [15] [4]. Smith [11] asserts that even on a semantic interpretation of information, the IP account is still inadequate, because information, which depends on a counterfactual correlation with the world, is objective. But without further restriction this account leads to pan-informationalism. And indeed an IP account, which leads to pan-informationalism, is not falsifiable and incapable of non-trivially explaining concrete digital computation.

Nevertheless, the IP account must embrace an algorithm-theoretic apparatus to be deemed adequate for explaining computation. As I have argued above, it is the processing part of IP that is very problematic. Arguably, even Algorithmic Information in the form of Kolmogorov-Chaitin-Solomonoff complexity will not do the trick for the IP account. Though it interprets information in terms of the *computational resources* needed to specify that information, it is a measurement method (i.e., analysing complexities, probabilities and randomness), rather than a descriptive one (e.g., explaining whether a miscomputation has just occurred).

All the above suggests that to be a plausible candidate for explaining concrete digital computation the IP account needs improving. If we want to explain certain cognitive functions computationally in terms of information processing, we should first be clear on how concrete digital computation proper is explained non-trivially in information processing terms. An IP account of computation should explain how a computing system is different from other non computing IP systems such as telephones or radios.

ACKNOWLEDGEMENTS

Thanks to Gualtiero Piccinini and Oron Shagrir for useful comments on earlier drafts of this paper. I would also like to thank some anonymous referees for their comments that helped improve this paper. I am especially grateful to Phillip Staines for his detailed comments and ongoing support.

REFERENCES

[1] Shannon, C. E. (1948). A mathematical theory of communication. Mobile Computing and Communications Review, 5, pp. 1-55.

[2] Piccinini, G. and Scarantino, A. (2010). Information processing, computation, and cognition. Journal of Biological Physics, 37, pp. 1-38.

[3] Wiener, N. (1948). Cybernetics: or control and communication in the animal and the machine. Cambridge: MIT Press.

[4] Floridi, L. (2008). Trends in the philosophy of information. In P. Adriaans and J. van Benthem (Eds.). Handbook of the philosophy of science, volume 8: philosophy of information, pp. 113-131. Elsevier.

[5] Feynman, R. P. (1996). The Feynman lectures on computation. Reading, MA: Addison-Wesley.

[6] Chaitin, G. J. (2003), Algorithmic Information Theory. Third Printing. Cambridge University Press.

[7] Dunn, M. (2008). Information in computer science. In P. Adriaans and J. van Benthem (Eds.). Handbook of the philosophy of science, volume 8: philosophy of information, pp. 581-608. Elsevier.

[8] Adriaans, P. (2008). Learning and the cooperative computational universe. In P. Adriaans and J. van Benthem (Eds.). Handbook of the philosophy of science, volume 8: philosophy of information, pp. 133-167. Elsevier.

[9] Shagrir, O. (2010). Computation, San Diego style. Philosophy of Science, 77, pp. 862-874.

[10] Churchland, P.S., Koch, C., and Sejnowski T.J. (1990). What is computational neuroscience? In E.L. Schwartz (Ed.), Computational neuroscience. pp. 46-55. MIT Press.

[11] Smith, B. C. (unpublished). Construals. volume 1, chapter 2. To be published at http://www.ageofsignificance.org

[12] Smith, B. C. (2002). The foundations of computing. In M. Scheutz (ed.) Computationalism: new directions. Cambridge, MA: MIT Press, pp. 23–58.

[13] Broderick, P. B. (2004). On communication and computation. Minds and Machines, 14, pp. 1–19.

[14] Floridi, L. (2009). Philosophical conceptions of information. In G. Sommaruga (Ed.). Formal theories of information, pp. 13-53. Berlin: Springer-Verlag.

[15] Dretske, F. I. (1981). Knowledge and the flow of information. Cambridge, MA: MIT Press.

[16] Millikan, R. G. (2004). Varieties of meaning: the 2002 Jean Nicod lectures. Cambridge, MA: MIT Press.

[17] Gettier, E. L. (1963). Is justified true belief knowledge? Analysis, 23, pp. 121-123.

[18] Agassi, J. (1988). Winter 1988 Daedalus. SIGArt newsletter, 105, pp. 15–22.

[19] Agassi, J. (2003). Newell's list. Commentary on Anderson, J. & Lebiere, C.: The Newell Test for a theory of cognition. Behavioural and brain sciences, 26, pp. 601-602.
[20] Dretske, F. I. (1993). Can intelligence be artificial? Philosophical studies, 71, pp. 201-216.
[21] Dreyfus, H. L. (1979). What computers can't do: the limits of artificial intelligence. 2nd edition. NY: Harper & Row.
[22] Harnad, S. (1990). The symbol grounding problem. Physica, 42, pp. 335-346.
[23] Penrose, R. (1989). The emperor's new mind. London: Oxford university press.
[24] Sloman, A. (1999). Beyond Turing equivalence. In P. Millican and A. Clark (Eds.) Machines and Thought: the legacy of Alan Turing. Oxford University Press.
[25] Sorensen, R. (2007). Can the dead speak? In S. Nuccentelli and G. Seay (Eds.) Themes from G. E. Moore: new essays in epistemology and ethics. New York: Oxford University Press.

Autonomy and desire in machines and cognitive agent systems

Dr Kevin Magill[1] and Dr Yasemin J. Erden[2]

Abstract. The development of cognitive agent systems relies on theories of agency, within which the concept of desire is key. Indeed, in the quest to develop increasingly autonomous cognitive agent systems desire has had a significant role. Yet we argue that insufficient attention has been given to analysis and clarification of desire as a complex concept. Accordingly, in this paper we will discuss some key philosophical accounts of the nature of desire, including what distinguishes it from other mental and motivational states. We will then draw on these theories in order to investigate the role, definition and adequacy of concepts of desire within applied theoretical models of agency and agent systems.

1 INTRODUCTION

In this paper we consider some philosophical approaches to the nature of desire, and then apply these to selected work already being undertaken in computing and engineering, particularly work that reflects specific views of agency with regard to desire. We begin by outlining a complex concept of desire that is applicable to animals as well as infant human agents, and then move on to critiquing particular understandings of desire within computing and engineering, drawing on examples of what is meant by *desire* in relation to adult human agents. This includes, for instance, approaches based on the belief-desire-intention (BDI) model of agency, as well as those models that view agents according to criteria of utility maximisation.[3] We draw attention to some flaws associated with these sorts of approaches, and highlight where they are either incomplete or contain over simplifications. As a result we claim that while making use of simple subsets of desire could in principle result in basic autonomy for cognitive agent systems, we also suggest that for this autonomy to be equal to our own, for instance, the account would require a deeper level of complexity. Approaches to the production of autonomous agent systems (machines and/or software) need to take this into account.

The dispositionality of desire (according to which it can be realised by a range of actions, depending on circumstances, or none at all), the variability of informed belief and judgement, as well as varied circumstances, means that there can be no definite and defining relationship between any desire or set of desires and any objective or set of objectives. One feature of approaches which assume a causal link includes distinguishing desire as defined or explained in relation to discrete subsections: want/need, pleasure/displeasure, reward/punishment, gain/loss. We argue that these categories are not fully encompassing, despite there being connections between them. We further claim that these sorts of dichotomies will not explain, nor fully contain the complex range of both cooperating and completing factors which constitute elements of what we mean by desire when considered in relation to cognitive agents. This, we maintain, will affect concepts of desire that underpin key features of those theoretical models applied to the development of cognitive agent systems, particularly with a view to a level of autonomy. We claim this is partly because, for example, the relationship to pleasure/displeasure, or pleasing/displeasing, isn't sufficient to distinguish desires from a range of other states, like wishing. In this paper we consider some alternative approaches which take into account the relationship to actions. Yet even here, though this makes the account of desires more complex, it does so in a specific and thereby limited way.

2 TURING MACHINES AND DESIRE

In Turing's celebrated 'imitation game' [1] thought experiment the role of the machine in the game is to deceive an interrogator into believing that it, rather than the other unseen respondent, is a woman (alternatively, that it is human). Turing's suggestion is that whether the machine can think could be answered by comparing its success rate in deceiving the interrogator with that of the man in the original version of the game. If one were impressed with the machine's success rate in the game enough to be convinced that it does indeed think[4], one could also infer from this that its deceptions manifest related mental attributes such as, for example, *understanding* (of the interrogator's thought processes, sufficient to judge what kinds of answers would prove most persuasive to him) and *belief* (for example, that the intention of the interrogator, having grasped and accepted the point of the game, is to accurately determine who is the deceiver). We would not, however, be in a position to infer from the machine's apparently thoughtful deceptions that it would be *pleased with* or *satisfied by* its successes, or indeed that it could be said to *want* or to *desire* them. For all we could tell from its intelligent capacity to mislead, the machine's deceptions might be carried out automatically. The machine might, by its successful deceptions, exhibit a high level of thought, but not any obviously genuine capacity for desire.

To have desires and to want things is clearly central to what it is to be human: an indispensible part of our animal just as much as of our intellectual nature. But what is desire? What is it to want? In Turing's thought experiment, the machine might

[1] Dept. Philosophy, Univ. of Wolverhampton, WV1 1LY, UK. Email: k.magill@wlv.ac.uk

[2] CBET/Dept. of Philosophy, St Mary's University College, TW1 4SX, UK. Email: erdenyj@smuc.ac.uk

[3] There are of course other models, for example the Soar (cognitive architecture) project, which offers more of a cognitive psychology approach, in contrast to the philosophical grounding of BDI. Nevertheless, since desires are primarily equated with goals even here, we shall restrict our analysis to BDI-based models.

[4] Note: the interrogator has no judgment to make about whether the machine thinks: he or she takes it for granted that both respondents are thinking humans.

deceive the interrogator about what it is, and what it thinks or believes, but not in the end – if it is successful enough in its deceptions – *that it thinks:* that would be no deception. On the other hand, there is nothing in the machine's imagined successes that would compel us to judge that it has desires. So could such a machine have genuine desires (a genuine desire to deceive at any rate)? And what would that call for or involve? We will consider this question in relation to two well known and opposing philosophical accounts of the nature of desire and desiring: action-based accounts and pleasure-based accounts.

One response to our scepticism about whether Turing's machine can be thought of as being motivated by a desire to deceive, drawing on an influential philosophical account of *the passions* (more prosaically described nowadays as *pro-attitudes* [2]) would be to say that if the machine has repeatedly attempted to convince the interrogator of something, and has done so with such a rate of success that it would be manifestly unreasonable to deny that it has done so intelligently, then it would be equally unreasonable, or at any rate confused, not to accept that it has done so intentionally and, *a fortiori*, that in doing so it has *acted*. If the machine's deceptions were actions, moreover, then it must also follow that it desired them, since 'thought by itself moves nothing' [3] and only a passion 'can direct the will' [4]. The familiar contemporary development of this view is that any action is a product of a belief and desire pair, each with its own distinctive function in relation to the production of action. According to this view, desire is defined by its natural function in relation to action and likewise distinguished from the role of belief in relation to action. Thus, if you are thirsty and you go to the tap for a glass of water, which you then drink, your actions can be explained by citing a desire to satisfy your thirst, together with a belief that drinking water would achieve this, a belief that water could be obtained from the tap, and so on. This yields a standard philosophical account of the nature of desire known as the 'action-based' or 'action-directed' theory of desire', according to which desires are defined and understood not according to subjective experiences of feeling desire, but according to the natural role or function of desire in the production of actions.

Just as children must learn that every desire need not be acted upon, accounts of the action-based theory typically define desires as *dispositions* to act in the way most likely to realise what is desired, according to circumstance, assuming no contrary desires or judgements, etc.[5] Since the ways in which a given desire can be realised will vary according to circumstance (e.g. ones thirst may be quenched from a bottle of water, a tap, the village pump, a mountain spring and so on), there will be a range of actions that could satisfy it, depending on circumstances: as well as there being circumstances and judgements that will prevent a desire from being acted upon. [5] [6] [7]

In Turing's game the machine could likewise have a range of potential responses that would, depending on the interrogator's questions, serve most effectively to deceive. If it were reasonable to see in those responses a consistently intelligent attempt to deceive, then why not also a consistent disposition and, therefore, a *desire* to deceive? The reason given earlier for thinking that the machine might lack any desire to deceive was that although its deceptions would clearly have manifested thought and understanding, its responses, for all we can know from the outside, so to speak, might have been carried out automatically and lacking in any real commitment, motivation or attitude. The desire to deceive, it might be said, would not be the machine's, but that of its designer or programmer.

But which of our desires may truly be said to originate with us as individuals? Most desires can be traced to parenting, acculturation, education, advertising, persuasive argument, propaganda and so on. Even supposing that we do have desires that begin with ourselves alone (whatever that would involve), there would be no obvious reason for treating only those as 'true desires'.

3 WANTON DESIRE

It might instead be argued that if the machine were bound automatically to deliver its deceptions, then its responses, as opposed to the judgements they express, would be, to that extent, blind and slavish. The same could be said, however, of the desires of animals, human infants and even to some extent, regrettably, of adults. Harry Frankfurt has famously argued for a distinction between first- and second-order desires and of a capacity to act on the latter as the basis of free will [8].[6] Second-order desires are desires about desires, e.g. ones desire, having given up cigarettes, not to give in to the desire to smoke. Someone who always acts on their first-order desires and lacks the ability or the inclination to act on second-order desires is characterised by Frankfurt as a *wanton*. Animals are also wantons, according to Frankfurt, because, lacking higher-order desires, they always act on their immediate desires (or whichever immediate desire or aversion is currently strongest with them). The machine's responses might, therefore, be slavish, but that would be no reason to regard them as not genuinely issuing from desires.

In considering Frankfurtian wantons, however, we take it for granted that it is desires that they act on. For our deceptive machine to be thought of as a wanton, we need a reason to think that in its case also there is an action causing state, distinct from the beliefs, understanding and thinking we are imagining it to exhibit, which corresponds to the desires of wantons. In the imitation game, however, there is no obvious need for such a state antecedent to its responses to the interrogator. The process is one in which the interrogator types out a question, which is followed, after varying intervals, by an answer. There is no reason to suppose that the initiation of the machine's thinking is triggered by anything other than the interrogator's input or that its response is caused by anything other than the conclusion of its deliberations in a judgement about what response is most likely to deceive the interrogator. If the machine has been designed always to begin ruminating when asked a question and

[5] Indeed, when desires are acted upon, whether immediately or following deliberation and decision, our conception of rational action, in which beliefs and desires provide *reasons* for acting, is one in which action follows a judgement in favour of the reasons for undertaking it. One possible implication of this is that while desires are causally antecedent to the actions that realise them, they are not, so to speak, the *immediate* or *direct* causes of actions. The position – immediate or mediated – of desires in the causal ancestry of actions is clearly relevant to an understanding of agency and presumably also therefore of potential relevance to agent systems, but is not an issue we can pursue in this paper.

[6] Elsewhere discussed as the basis of a related, but arguably different, capacity for 'fully human agency'.[9] [10]

always to automatically output the answer it judges most deceptive (likewise to do so after whatever intervening period is indicated as most likely to seem human), there would be no place for an action-causing state corresponding to that of desire in the process. The difficulty is not that the machine is incapable of not responding in the way that it does, which we have already allowed is true of many wantons, but that desire has no work to do here.[7] In fact, we consider below whether it is really called for in BDI models of cognitive agent systems. Similarly, while we might be obliged to regard the machine's responses as intelligent, there would be no reason to view them as actions or as in any way intentional.

By contrast, there would be no difficulty in regarding the machine's responses as *goal-oriented*. The machine's ruminations clearly have an overall aim: that of getting the interrogator to believe, mistakenly, that it is the woman. It could therefore properly be thought of as exhibiting goal-oriented behaviour. Note also that it is not the singularity or simplicity of the machine's goals that prevents the concept of desire from having any work to do or application in relation to the machine's deceptions. Multiple goals could be built into a calculative or ratiocinative process such that what to do would be determined according to the likelihood of success of an outcome in a particular situation, adjusted according to its ranking relative to other possible outcomes. One can imagine a more complicated version of the imitation game in which the respondents have additional objectives to that of convincing the interrogator that she/it is the woman, such as causing the interrogator to display emotion (with extra points awarded, say, every time the interrogator laughs, cries, bangs the table and so on). Depending on how these additional objectives are weighted and how the machine judges the interrogator's susceptibility at any point, it might calculate that a less than optimally deceptive response is justified by the possibility of its producing laughter. It is possible to imagine further additions to the machine's objectives, bringing greater complexity to its ruminations, but none that would call for any alteration in the question and response mechanism required for the basic game. The machine's calculations could still be triggered unvaryingly by the interrogator's questions, with its responses following automatically on their culmination of those calculations in a judgement. Once again, even with multiple goals, there would be no need for its responses be caused by its *desiring* anything.

A wanton, according to Frankfurt, lacks free will, but it can certainly be thought of as having a certain level of autonomy. A mouse may lack the wit to reflect on its own behaviour, but it has a sense of itself, in contrast to other features of its environment, and of its own interests, as well as the ability to control its own movements in pursuit and protection of those interests. Its *musine* desires and fears are central to its sense of its own interests – what matters to it – and its ability to pursue them, and therefore also to its small mammal autonomy. If Turing's machine, although thoughtful, lacks the desire or autonomy even of a mouse – if we have no need of the concepts of desire or autonomy in order to describe adequately its behaviour – what attributes would it need to acquire in order for us properly to think of its behaviour as motivated by desire? What else would be needed to bring it up to the level of wanton autonomy of a mouse?

4 *ACTION* AND *PLEASURE* ACCOUNTS

A well known criticism of action-based theories of desire is that it is possible to desire things that could not be brought about by actions: either because they could not be brought about by the actions of the individual who desires them or because they could not be brought about by the actions of anyone at all. Some agnostics might, for example, desire that there should be life after death, while not believing that any action can make a difference to whether or not this is the case. One response to this challenge is to say that desires for things we cannot act to bring about or render more likely are in fact *wishes* rather than *desires*, properly so called[8]. Nevertheless, there is clearly a close affinity between desiring and wishing, which suggests that there is more to the concept of desire than simply its relationship to actions, however characterised. What is it that the two have in common that is not shared with other mental state-object *relata*? One suggestion about what is shown by examples of wishing/non-action directed desires as lacking in the action-based account of desire is the focus of one of its chief competitors, namely pleasure-based accounts.

As with action-based accounts of desire, there are different pleasure-based accounts, differing, for example, about the role pleasure is thought to play in the causation of actions.[9] For the purposes of this paper a pleasure-based account of desire is one that claims that to desire something requires, among other things, that its realisation is regarded as (broadly) *pleasing,* and its absence as *displeasing*.

This, however, presents us with an immediate puzzle about how desire involves regarding its realisation as pleasing. If we were to say, for example, that to desire something is simply to think of it as pleasurable, this would entail that desire is a type of belief, i.e. a belief that something is pleasing or displeasing. This would conflict with the idea, set out above, that desires have a distinctive and defining role and content in relation to beliefs.[10] Another way of expressing that distinctive role is through the idea of 'direction of fit' [5], [15] between mind and world. A belief is true if it represents the world correctly, false if it does not, and therefore if it *fits* the world as it is: belief has a *mind-to-world* direction of fit. With desire by contrast, the direction of fit is *world-to-mind*, in the sense that for a desire to be *satisfied,* the world must be brought to match its content.

If desire is not a special kind of belief that something is or would be pleasing, we can still say that desire has a defining logical or normative relationship to its object, such that its realisation – the *satisfaction* of the desire – must be thought of as pleasing. The content of that defining relationship is that of the *intentionality* or *directedness* of desire (*what it is about)*. As

[7] The notion of desire in this context, to borrow an idea from Wittgenstein, would be idle.

[8] See below on section to do with fantastical desires.

[9] Morillo, in [11], for example, argues that actions are caused, conjointly with beliefs (neurologically conceived), by pleasure events in the brain, from which it is said to follow, since this is the causal role of desires, that desires are such pleasure events. See Strawson [12] for a very different set of arguments for the pleasure-directedness of desire.

[10] Although some philosophers have argued that desire might be a special category of belief: see [13] and [14].

James Cheney has suggested[11], 'To say that a desire is intentional is to say that, in some sense, it involves a concept of the object of desire.' [16] This concept or idea must have at least two elements according to the account we are developing: an idea of the object of desire *and* that it is thought of as pleasing (or, again, of its absence as displeasing).

While the requirement that an intentional content involving a concept of the object of desire *thought of as pleasing* may be an essential constituent of desire, this still fails to distinguish desiring something from merely thinking or believing that it is pleasing, which can be thought of as having the same intentional content. Any idea, thought or concept can be an object of thought, or of belief. Thus, if the intentionality of a desire involves the thought or concept of its object considered as pleasing, why couldn't the same thought be itself an object of thought or of belief? Indeed discussions about the intentional content of beliefs and desires typically assume that they can have the same intentional content; thus '*A* may believe that *x'* or '*A* may desire that *x'*. Against this view, we want to suggest that intentionality with respect to desires and other passions should be expected, somehow, to embody their characteristic *direction of fit* and that the kind of shorthand description of intentional content given by '*A* believes/desires *x'* do not entail that the full intentional contents of beliefs and desires of which such shorthand descriptions may be true can likewise be exchanged. But how might the direction of fit characteristic of desires be reflected in their intentional contents?

To begin with, the *world-mind* direction of fit that is thought to distinguish desires from beliefs is shared with a range of other passions or motive states. We have suggested that direction of fit should be expected to be reflected in their intentional contents; therefore that those contents must be of a different kind to the representational, descriptive or fact-directed character of beliefs, thoughts, memories and so on. So what might distinguish the intentional content of desire from, for example, that of its close cousin *wishing*? Wishes are typically directed at states of affairs whose occurrence is unlikely, doubtful or impossible and, as noted already, at things that are beyond our capacity to bring about or make more likely (except perhaps by supernatural or transcendent means of doubtful efficacy). What distinguishes desiring from mere wishing is its relationship to action. Desires, according to the account of the action-based theory set out earlier, are dispositions to act in the way most likely to realise what is desired: that will result in actions in favourable circumstances, assuming no contrary desires, etc. We argued that the similarity between desiring and wishing suggests that there must be more to what constitutes desire than its relationship to actions. Since neither the action-based or pleasure-based accounts of action seem to offer a complete account of desire, but both direct us towards notable and important features of desire, perhaps each can complete what is lacking in the other. Can desires, then, be understood as states with the combined features of being dispositions to act and having an intentional content involving a concept of the object of desire considered as pleasing?

The suggestion is tempting – the combination is clearly not arbitrary, or the relationship between actions and pleasure contingent – but, to begin with, if desires are dispositions to act it would be odd if that were not also to be reflected in the intentional content of desire. The case for regarding desires as dispositions to act would appear to present problems here: one can desire to do something without doing, or intending to do, anything about it. However, as we have noted already, the idea that desires are dispositions to act is consistent with multiple qualifications: that a desire 'will result in actions in favourable circumstances, assuming no contrary desires or moral judgements, etc.'. How might this be reflected in the intentional content of a desire? One plausible suggestion would be that the intentional content of desire also includes the idea that one *could act* so as to satisfy the desire (or at any rate so as to make it more likely). Such a limited intentional content would be consistent with a range of possible realising actions and circumstances, as implied by the range of qualifications involved in the idea of a disposition to act. However, this is clearly the kind of intentional content not of desire but of the kind of belief that, paired with desire, can constitute a reason for acting. Once again, it would lack the world-to-mind direction of fit that is characteristic of desire. It can be added that although the idea that one can act in such a way as to satisfy ones desire is *consistent* with the range of qualifications implied by the dispositional account, it lacks any clear *connection* with the idea of a disposition.

A notion that *would* have both the appropriate direction of fit and give intentional expression to the dispositionality of desire would be that of *readiness* or *readiness to*. To be ready to do something involves no intention to do it, but rather a willingness to do it subject to conditions.[12] If I have no intention of doing anything to satisfy my desire, I may still be ready to do it should whatever my reasons for not doing it cease to apply. The idea of *readiness to* would therefore appear to express a relationship to action, with the appropriate direction of fit, even for cases of desiring without intending.

Can desire, then, be understood as a dispositional state whose intentional content includes an idea of the object of desire considered as pleasing and a readiness to act so as to realise (or help to realise) it subject to the agent's judging it as worth doing?[13] We have a range of uses, often nuanced and particular, for *desire* terms, which is further extended and complicated by their relationships to seeming synonyms such as *wanting* and close cousins such as *wishing*. We have not, to take one example of the range of uses of *desire*, discussed cases involving a desire for someone else to do something. The idea of *readiness to* would not straightforwardly express an attitude to the actions of another. Although we would not envisage any great difficulty in adapting the notion, to do so would not be strictly pertinent to the focus on autonomy and autonomous systems in this paper. Therefore, while there is a wider range of significant uses of *desire* and desiring, and related examples, against which the dispositional and intentional-content account of desire we have developed could be further evaluated and developed, it appears to fit the uses and examples we have considered so far and as it stands it would present a complex and challenging set of

[11] Admittedly in relation to quite a different notion of the intentional content of desire from what is discussed in this paper.

[12] If, as we have suggested, it is *readiness to* that confers the appropriate direction of fit on the intentional content of desires and if wishes are thought of as having that same direction of fit, we still face the question of what it is that distinguishes wishes and desires. Our suggestion would be that *wishing* is after all just *desiring* but with an accompanying belief that its satisfaction, is chancy, unlikely, impossible, etc.

[13] See [2] for discussion of a range of issues relating to such judgements.

constraints on the idea of a machine that could be thought of as sufficiently autonomous to pursue its own desires.[14]

We suggested earlier that in considering what would be needed to think of Turing's deceptive machine as having a desire to deceive, over and above what is pressed on us by the results of the game, is that its deceptions and their successfulness mattered to it, or were important to it and that this turn would require a capacity to relate its actions and their outcomes to itself, which would also suggest that it have a sense of its own interests. For the machine to have desires, therefore, would appear to require, unsurprisingly, that it has other potentially complex and challenging attributes: that the idea of a desiring machine calls for much more – a sense of self, an awareness of its own interests – than the complex set of attributes we have so far identified as constituting what it is to desire. But the scope or type of the additional complexity and challenge may not lie, as such, in some imagined self-architecture. We have confined ourselves thus far only to identify what is required for the desires of wantons and the sense or level of self-interest called for by that would be no more than that of many animals. If an organism is capable of pleasure or pain and can in that sense be pleased or displeased by its actions, as well as events and outcomes, how could we think of it at the same time as lacking a sense of its own interests? Sometimes what is seemingly simple can have complex and ramified consequences. Rather than a self with interests being something calling for complex self-architecture, perhaps it can be brought into being solely by the capacity to feel pleasure or pain, to be pleased or displeased.[15] The additional complexity and challenge would lie in determining what would be required for a machine to feel pleasure or pain, or to be pleased or displeased in some broadly corresponding fashion, which, we need scarcely add, is no small matter.[16]

There is perhaps a more immediate challenge regarding the purpose of a desiring machine. It would be pointless to attempt to conceive of desire, according to the account we have developed, as having any role to play in the machine's response mechanism in the imitation game. As noted already, a reasonable explanation of what the machine does in the game has no need of any states corresponding to desire. Perhaps the prior and more fundamental challenge in thinking about machines with a desire to deceive (or as having desires to or for anything else) is not, as such, how they would need to be designed in order to have, or to be imbued with, desires, or with the capacity to be pleased or displeased, but rather what tasks or scenarios would call for such attributes and exemplify them in the way that the imitation game would exemplify machine thought.

5 DESIRE IN APPLIED MODELS

As we explain above, to a certain extent, desire is not only relevant to intention; it also enables understanding about action by means of explaining reasons. As such, it can be applied when predicting behaviour for example. These aspects can be useful for the development of intelligent systems, as well as in developing intelligent systems that actively respond to users to serve particular needs. In this paper we focus exclusively on the former.[17]

In contrast to the complexity already identified with understanding the concept of desire, accounts of the term within computing and engineering tend to summarise it in simplified terms, and more often than not as the pursuing of particular goals, or as precursor to goal-driven behaviour. Yet as we have shown above, while it is true to say that desires can sometimes be described as *precursors to actions*, in the sense that they sometimes lead to actions, this does not mean that desires therefore *define actions* in their systems (as, for example, goal-driven behaviour).[18] On the process of autonomous action selection, Park *et al* [19, p. 832], for example, note that where an agent must decide between which actions to perform, and has competing desires, these desires are simply equated with multiple goals. As such, the 'Desire states of the agent can be defined as a product of the goal variables'. If, by goal variables, is meant 'the range of things that could be goals', then certain desires could be so construed from such goal variables, to some extent. Goal variables, on this sort of account (depending on how they are defined), would only count as *possible* rather than *actual* desires. Lang *et al* [20] summarise desire in terms of utility loss and gain (where the former incur penalties, the latter rewards), and Ishikawa *et al* [21] write about the need for internal reward in developing autonomous mobile robots.[19] These approaches too present an account of desire as in some way equated with goals.

Significant problems arise from the equating of desire with goals, however. In the first instance, this approach can ignore the fact that desires are sometimes more accurately described as *wishes*, because they are not action-directed (as noted above), or they can even be *fantastical* (something akin to *wishes*, but without an aim of satisfaction).[20] Further, while desire *may* be a driving component with regard to intention and/or action (such that it may result in action), then again it may not (see above section on desires as dispositions to act). Indeed certain desires might never be expressed, acknowledged, nor even understood by the human agent.[21] Yet, accounts that employ BDI models

[14] We would add that a possibly major additional constituent element of desire, not considered here, is that the object of desire is experienced as *lacking*. While it has not been possible for us to consider this idea here, the topic may be given further consideration in future versions of this paper.

[15] Recalling Bentham's famous judgement on what our grounds should be for the moral consideration of animals: The question is not, Can they reason? nor, Can they talk? but, Can they suffer? [17]

[16] The notion of a machine capable of *intentional action* might be thought to present considerable further challenges.

[17] For contemporary analysis of the latter, cf. Dong *et al* [18].

[18] Setting aside the issue of whether what we are talking about is really actions as opposed to behaviour, as discussed in previous sections.

[19] There are, of course, other approaches. Dastani and van der Torre [22], for example, note the problems engendered by the unification of desires and goals into a *single motivational attitude*, and instead offer an approach that distinguishes between them. This is not, however, the most common approach. Cf. Schroeder [23], who, though not specifically dealing with a goal/desire distinction, does present a linear materialist view of desire as determined by reward systems in the brain.

[20] The authors hold different positions on this matter, and as such how far this relates to wishing or action-based desire is yet to be agreed.

[21] The fact that we might not be aware of certain desires may or may not play a role in affecting our behaviour and intentions, or indeed our sense of what we find pleasant or unpleasant, but as we note above and below,

neglect the necessarily purpose-driven attitude, which accompanies something that we call a goal. By its very nature, a goal is something that we seek to achieve, and for which we make plans. A desire need contain no inherent planning, and though it may influence behaviour, there is nothing about a desire, which necessitates that it does (again, a causal connection is not *a priori*).

Bratman [24, p. 22] makes a similar point, when he distinguishes between desire and *intention*:

> For example, my desire to play basketball this afternoon is merely a potential influencer of my conduct this afternoon. It must vie with my other relevant desires—say, my desire to finish writing this paper—before it is settled what I will do. In contrast, once I intend to play basketball this afternoon, the matter is settled: I normally need not continue to weigh the relevant pros and cons. When the afternoon arrives, I will normally just proceed to execute my intention.

Nevertheless, as noted in the examples above, *desire* in the BDI model (among others) is frequently treated as synonymous with a *goal*. The BDI model has been influential in computational and engineering approaches to AI, and was originally proposed in Michael E. Bratman's [25] seminal text *Intentions, Plans, and Practical Reason*. In this work he offers a theory about the relation between intention and practical reasoning, whereby the former is claimed to play a central affective role in the latter. An example of which is shown in his claim [25, p. 17] that: 'Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes.' While BDI theories are not limited to Bratman's approach (cf. Pollack in Georgeff *et al*, [26]), those that expand on his ideas, by adding new elements, still repeat the same sort of equivalence (cf. BOID models in Broersen *et al*, [27]; or KBDI models in Su *et al* [28]). As Cholvy *et al*, [29, p. 1] explain: 'Based on the idea that social concepts like obligations or more generally norms are important to "glue" autonomous agents in a Multi-agent System[22], the BDI model has recently been extended in order to take into account obligations and norms.'

Still, the general approach to desire remains linked with goal-driven intention or behaviour. Yet, it is clear that goal direction is an attribute of actions, intentions and *tryings* – desire is a causal antecedent of such states, so ontologically distinct from anything that can be described as goal-directed (as opposed to, say, object-focussed). Within BDI models, it is typically held that *belief* indicates what information the agent holds about the world; *desire* represents what the agent would like to occur; and *intention* represents what an agent plans to do in order to realise certain desires through a process of reasoning,[23] yet none of this explains how it is that desire should equate to goals. Where desire is coupled with intention to act, or a commitment to act, in order to achieve a goal, then this might rightfully be considered a connection between the two; yet such connections should not result in the subjugation of desire within goal-driven action or intention.

6 UTILITY MODELS AND WEIGHTED DESIRE

Even where attention is paid to the complexity of the term *desire*, there nevertheless remain unanswered questions. Lang *et al* [20, p. 2], for instance cite 'strength and polarity parameters' in relation to desires, noting that 'stronger desires can override weaker desires…more specific desires override more general desires, and gain, loss and mixed desires can be distinguished'. Despite this analysis, it still remains unclear how we are to consider those desires about which we are, for example, uncertain or unaware (desires that conflict such that we may not even be clear about what we desire). In addition, it is not an *a priori* truth that more specific desires override more general desires. Setting aside the basic challenges identified earlier involved in incorporating even basic animal desires in cognitive agent systems, while it is true to say that these are only a feature of some, not all, human desires, they nevertheless contribute to the complex fecund of desires within autonomous human agents. Just because we are not clear about *how* these might impact on autonomy, this is not conversely to say that they do not. Lang *et al* [20, p. 37] talk of generating 'the preference relation from a set of desires', in order to find 'the optimal feasible worlds, and thus the optimal decision', yet it is clear that this remains contingent on the values we ascribe to individual desires and judgements.

One process to realise certain outcomes in cognitive agent systems is based on calculations that attribute value to an action based on 'value to be received immediately and in the future through continued rational action' (Park *et al*, [19] p. 832), known as Markov Decision Processes (MDP), but this too is limiting since 'if the agent has multiple goals to achieve, their achievement state must be represented as part of the decision problem'.[24] To get around this problem Park *et al* [19, p. 837] propose a system to 'separate the concepts of desires (achievement states) and domain states to enable reasoning at various abstraction levels'. This, they suggest, will lead to an approach that estimates 'the cost of pursuing each goal', for which 'reasoning is performed in the "desire space," which describes the expected value of pursuing a goal in the context of how selected actions facilitate the pursuit of other goals in the future'. Yet even despite this separation, it still relies on a system that values in advance certain given outcomes, and also sets out

this is not to say that the significance of such desires can be easily dismissed from our conception of human autonomy.

[22] Multi-agent systems (MAS) are systems of multiple interacting intelligent agents, which enable surveyability of, for example, large systems, and rely on the autonomy of each individual agent. See Brazier *et al* [30]. Notions of desire therefore play a significant role in ensuring not only the autonomy of each agent, but also to ensure that each agent works towards achieving different but overlapping or complementary ends.

[23] There are of course some issues with these definitions. For instance, intentions would more naturally be thought of as being produced by or following from a process of reasoning. We do not have the space to consider these objections here, but they should not be neglected in applying BDI, and related, models.

[24] See discussion in earlier sections on multiple goals in relation to the Turing Machine, where we show that the multiplicity of goals does not automatically create a role for desire.

a rather limited, and thereby incomplete, account of any given set of desires.

There is little doubt that unquantifiable factors impact on action. The existence of a necessary causal link, however, leading from desire to action is nowhere near as apparent, and therefore is not easily identifiable as one that could fit a neat model of autonomous cognitive agent behaviour (as per our remarks on the open-ended nature of dispositions in sections above). Furthermore, mapping desires according to a model of utility, and so determining the relative gain/loss of particular desires, results in a simplistic and static approach to something, which as already noted, is more complex than might first appear. In Lang [20, p. 8], for instance, there are issues with the examples given for both gain- and loss-desires when considered in relation to cognitive agents. The preference for an umbrella when it rains, for example, indicates only one aspect of what a person may desire in relation to rain. The summary of the claim "if it rains then I prefer to have an umbrella" as a loss-desire (the violation of which is seen as purely negative), would not include other desires in relation to both rain and umbrellas. This might include, as they note, the desire to not carry an umbrella [20, p. 16], but also the desire to stay indoors when it rains, or even the desire to stand in the rain and get wet because you have a desire to imitate Gene Kelly (see commentary above on desires and dispositions).

Conceptions of utility are almost certainly always biased. Indeed they seem necessarily tied to judgement. They are never value-neutral and as such, this impacts on attempts to develop autonomy within cognitive agent systems. Programmed options are therefore equally biased (that one option is in fact included when another is not is, in itself, based on a judgement). Accordingly the criteria for judgement begin from a non-value-neutral position. If it is a judgement of where we simplify these details we may be successful in achieving simple processes that present certain levels of autonomy. Nevertheless we will not be able to create or expect any desire-driven behaviour, which though apparently rational, yet appears to contradict expectations about desire in relation to supposedly normal situations. For example, choosing wine with a fish based meal (an example offered by Lang *et al*, [20, p. 8], in discussing gain-desires). One may desire white wine with fish (as per convention, taste, or culture), or one may desire to drink whatever wine is cheapest, already open, of a favourite brand, type, or grape (whether it be red or white). One might even prefer a beer. In fact, where Lang *et al* [20, p. 9] discuss 'mixed desires', they are perhaps closest to addressing the level of complexity involved. They note: 'It seems natural to more (hungry, but not starving) human agents that eating a cooked potato is better that nothing and that eating a raw potato is worse than nothing.' What is interesting about the example used here is that what it shows is the minutiae of detail, which can affect desire. Here it may only be a matter of pleasantness with regard to the taste of certain foods in particular conditions, but at other times desires may be affected by a series of multiple factors at any given moment [20, p. 29]. This list includes convention (social, cultural, political, legal), habit, expectations of oneself, expectations by others, or one's beliefs about the expectations of others and so on. The above also applies when we talk about making decisions where there are competing desires [20, p. 28]. The point here is that there is a distinction to be drawn between, on the one hand, what might be predicted from a given desire, and on the other, what desires could be inferred from an action. Yet this is not the end of the story as regards the meaning of *desire*, and these issues remain pertinent to those who would seek to use it as a concept from which to develop autonomous cognitive agent systems.

7 CONCLUSION

This paper examines the idea of desire in relation to machines and cognitive agent systems. Drawing on action-based and pleasure-based philosophical accounts of the nature of desire, we have demonstrated, to begin with, that the level of desire even of a mouse has a degree of complexity that presents several challenges for any attempt to develop the same sort of autonomy in agent systems. It would depend, in the first place, on the development of a capacity for pleasure and pain, or at any rate a corresponding capacity to be pleased or displeased; alongside the possibly more straightforward challenge of elaborating a state with the appropriate kind of action-related dispositionality. More fundamentally, it would call for a clear account of what kinds of tasks or scenarios would require a capacity for desire (and a correspondingly self-interested basic level of autonomy) - in a way that Turing's imitation game clearly would not - such that their performance would exemplify machine desire just as the imitation game would exemplify machine thought. In examining some contemporary models of cognitive agent systems, we have in addition identified several further challenges in relation to the equating of desires with goal-driven behaviour. While we acknowledge that desire may yet have a role to play in the development of autonomous agent systems, we can see no reason to accept that it has a recognisable and realistic role in contemporary models, or that what has been described as *desire* in such models is realistically informed by a robust understanding of the concept in relation to autonomous human agents, for instance.

REFERENCES

[1] A. M. Turing. Computing Machinery and Intelligence. *Mind*, 59: 433-460, (1950).

[2] D. Davidson. Actions, Reasons, and Causes. In *Essays on Actions and Events*, 2nd edition. Oxford University Press, Oxford. (1980).

[3] Aristotle. *Nichomachean Ethics.* Trans. *W. D. Ross.* Clarendon Press, Oxford. 1139a35-b4, (1908).

[4] D. Hume, *A Treatise on Human Nature*, second edition. L. A. Selby-Bigge, Ed. Clarendon, Oxford. Bk II, Section III, p. 413, (1978).

[5] E. Anscombe. *Intention*, 2nd edition. Harvard University Press, Cambridge, MA. (2000).

[6] G. Ryle. *The Concept of Mind*. New University of Chicago Press. (1949).

[7] M. Smith. 'The Humean Theory of Motivation'. *Mind*. 96, 36–61, (1987).

[8] H. Frankfurt. Freedom of the Will and the Concept of a Person. *Journal of Philosophy*. LXVIII, 1, 5–20, (1971).

[9] J. D. Velleman. What Happens When Someone Acts? *Mind*. 101, 2, 461–81, (1992).

[10] K. Magill. *Freedom and Experience*: *Self-Determination Without Illusions*. Macmillan, London. Chapter 2, (1997).

[11] C. Morillo. The reward event and motivation. *Journal of Philosophy*. 87, 169–86, (1990).

[12] G. Strawson. *Mental Reality*. MIT Press, Cambridge, MA. (1994).

[13] H. A. Costa, J. Collins, and I. Levi. Desire-as-Belief Implies Opinionation or Indifference. *Analysis*. 55.1, pp. 2–5, (January, 1995).

[14] R. Bradley and C. List. Desire-as-belief revisited. *Analysis*. 69 (1), (2009).
[15] G. Schueler. Pro-attitudes and direction of fit. *Mind*. 100, 277–81, (1991).
[16] J. E. Cheney. The Intentionality of Desire and the Intentions of People. *Mind*. LXXXVII, (4), 517-532, (1978).
[17] J. Bentham. *Introduction to the Principles of Morals and Legislation*, second edition. Clarendon, Oxford. Chapter 17, fn 122, (1823).
[18] J. Dong, H.-I. Yang, K. Oyama, and C. K. Chang. Human Desire Inference Process Based on Affective Computing. *2010 IEEE 34th Annual Computer Software and Applications Conference*, 347-350, (2010).
[19] J. Park, K. K. Fullam, D. C. Han, and K. S. Barber. Agent Technology for Coordinating UAV Target Tracking. In R. Khosla, R. J. Howlett, and L. C. Jain (eds.) *Knowledge Based Intelligent Information and Engineering Systems*. Berlin, Springer. (2005).
[20] J. Lang, L. van der Torre, and E. Weydert. Utilitarian Desires in *Autonomous Agents And Multi-Agent Systems* Volume 5, Number 3, 329-363, (2002) and online: http://www.irit.fr/PERSONNEL/RPDMP/JeromeLang/papers/aamas.ps [accessed 25/01/11].
[21] M. Ishikawa, T. Hagiwara, N. Yamamoto, and F. Kiriake. Brain-Inspired Emergence of Behaviors in Mobile Robots by Reinforcement Learning with Internal Rewards, *2008 Eighth International Conference on Hybrid Intelligent Systems*, 138-143. (2008).
[22] M. Dastani and L. van der Torre. Specifying the Merging of Desires into Goals in the Context of Beliefs in *EURASIA-ICT 2002: Information And Communication Technology: Lecture Notes in Computer Science*, Volume 2510/2002, 824-831. (2002).
[23] T. Schroeder. *Three Faces of Desire*. OUP, Oxford. (2004).
[24] M. E. Bratman. What Is Intention? In P. R. Cohen, J. L. Morgan, and M. E. Pollack, (eds.), *Intentions in communication*, chapter 2. MIT Press, Cambridge, MA. (1990).
[25] M. E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard Uni. Press, Cambridge, MA. (1987).
[26] M. P. Georgeff, B. Pell, M. E. Pollack, M. Tambe, and M. Wooldridge. The Belief-Desire-Intention Model of Agency. In *ATAL '98 Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages*, (1999), online: http://www.ppgia.pucpr.br/~fabricio/ftp/Aulas/Mestrado/AS/Artigos-Apresentacoes/BDI-Agents/georgeff.pdf [accessed 24/01/11].
[27] J. Broersen, M. Dastani, L. van der Torre. Beliefs, Obligations, Intentions, and Desires as Components in an Agent Architecture. *International Journal of Intelligent Systems*, 20:9, 893-919. (2005).
[28] K. Su, X. Luo, A. Sattar, and M. A Orgun. The interpreted system model of knowledge, belief, desire and intention. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS '06).*, NY, USA, ACM, 220-222. (2006).
[29] L. Cholvy and C. Garion. Desires, norms and constraints. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, 1-10 (2004), and online: http://oatao.univ-toulouse.fr/520/1/Garion_520.pdf [accessed 25/01/11].
[30] F. M. T. Brazier, B. M. Dunin-Keplicz, N. R. Jennings, and J. Treur. Desire: Modelling Multi-Agent Systems in a Compositional Formal Framework. In , M. Huhns and M. Singh (eds.). *International Journal of Cooperative Information Systems*, 6: 1, Special Issue on Formal Methods in Cooperative Information Systems: Multiagent Systems. (1997). http://dare.ubvu.vu.nl/bitstream/1871/11287/1/IJCIS97.DESIRE10.pdf [accessed 25/01/11].

The Singularity Might Indeed Be Near, But the Next Interesting Level of Intelligence Is Too Far

Jiří Wiedermann [1]

"It turns out that, yes, there are limits to computations based on the laws of physics. But these still allow for a continuation of exponential growth until nonbiological intelligence is trillion of trillions times more powerful than all of human civilization today, contemporary computers included."

Ray Kurzweil, in "The Singularity is Near: When Humans Transcend Biology"

Abstract. Using the contemporary view of computing exemplified by recent models and results from non-uniform complexity theory, we investigate the computational power of cognitive systems. We show that in accordance with the so-called Extended Turing Machine Paradigm such systems can be seen as non-uniform evolving interactive systems whose computational power surpasses that of the classical Turing machines. Our results show that there is an infinite hierarchy of cognitive systems. Within this hierarchy, there are systems achieving and trespassing the human intelligence level. We will argue that, formally, from a computation viewpoint the human level intelligence is upper-bounded by the Σ_2 class of the arithmetical hierarchy. Within this class, there are problems whose complexity grows faster than any computable function and, therefore, not even exponential growth of computational power can help in solving such problems.

1 INTRODUCTION

The introductory quotation by Ray Kurzweil evokes an idea of certain ordered "intelligence levels", one of which corresponds to human intelligence, with still some levels of "superhuman", non-biological intelligence above it. The point in time when the "power" of non-biological intelligence will reach and trespass the level of human intelligence has obtained a popular label: the Singularity (cf. [16]). According to the leading thinkers in the field of artificial intelligence this point is near, lying merely a few decades in the future, with profound consequences for the mankind, cf. [5, 9, 16]. Nevertheless, it seems that the AI literature has not been paying much explicit attention to the formal investigation of the "power" of artificial intelligent systems, and hence, to the question whether there are some limits to this power. Yet, there is one notable exception to this state of the matters: this is the recurring idea that the human mind might perhaps possess the ability to go beyond the level of classical computability as defined, e.g., by the classical Turing machines [23]. R. Penrose seems to be the best-known defender of this idea ([21]). If this conjecture is right then, undoubtedly, any level of artificial intelligence reached after the Singularity will obey super-Turing computing power. Could the latter statements concerning the super-Turing power of the human mind and the levels of intelligence beyond it be also supported in theory? It is the goal of this paper to give one possible answer to this question. In order to answer this question we will use a similar approach as Penrose did, when seeking the answer to his problem concerning the power of human thoughts, or, more generally, of cognition. Namely, we will identify a computational "machine" model of cognitive systems, and investigate its computational power. However, unlike Penrose who attempted to show the superiority of the human mind by comparing it with the model of the classical Turing machine we will make use of a different, demonstrably more computationally powerful model that captures the main features of cognitive systems. These features cannot be modelled by the classical Turing machine.

Usually, the term "cognition" denotes the activities by which the living organisms collect, process, store and utilize information. These activities especially include perception, learning, memorization, and decision making [22]. W.r.t. this definition intelligence can be seen as a part of cognition which is less interested in perception and focuses mainly on the quality of cognitive processes. Both cognition and intelligence are related to information processing. The belief that human or biological cognition and intelligence present a specific kind of computations (cf. [4]) is the basis of *computationalism*. The proponents of this school of thoughts believe that the computational modelling of cognitive abilities of living organisms, inclusively that of humans, is at least in principle possible, and that in this way one can achieve if not a genuine than at least an approximative capturing and understanding of all mental faculties attributed to intelligence (inclusively those of thinking and consciousness) and explanation of the underlying algorithmic principles.

Any computational model of cognitive systems must capture three important obvious properties of cognitive systems. Namely, any cognitive system must clearly be (i) *interactive* — in order to be able to communicate with their environment, to reflect its changes, to get the feedback, etc.; (ii) *evolutionary* — in order to develop over generations, and (iii) potentially *time-unbounded* — in order to allow for their open-ended development.

Therefore, no fully-fledged cognitive system can be modelled by the classical Turing machines — simply because such machines do not possess the above mentioned properties. This also confirms Penrose's conclusion that classical Turing machines do not adequately capture human intelligence, albeit by a different reasoning. The cognitive systems, therefore, must be modelled by theoretical computational models capturing interactivness, evolvability, and time-unbounded operation of the underlying systems.

Surprisingly, early computational models of cognitive systems did not reflect any of the three above-mentioned properties. It was the paradigm of classical Turing machines that has traditionally served

[1] Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Prague, email: jiri.wiedermann@cs.cas.cz

as a framework for thinking about computing and, more generally, about intelligence [23] (and, e.g., for Penrose also about thinking [21]). However, with the recent progress of information technologies and in the sciences where many natural systems are now viewed as information processing systems, our perception of what is computing has substantially broadened. Contrary to the traditional view of computing nowadays we are witnessing a shift from finite to potentially non-terminating interactive computations, a shift from rigid computing systems towards systems whose hardware and/or software can change over time, and a shift from once-for-all-times given architecture of computing systems to computing systems with their architecture and software evolving in an unpredictable, non-uniform way. The latter systems are known as *non-uniform interactive evolutionary systems*.

The efficiency of our models will be measured by the standard methods used in the computational complexity theory, i.e., by comparing their efficiency and computational power to that of the standard basic models known within this theory. We will look for the computational limits and hierarchies of our models. We will be especially interested in their efficiency in processing the data in order to solve cognitive problems and, last but not least, whether there are cognitive problems which, in principle, cannot be solved by these models. In the rest of this paper we will be simply speaking only about cognition which, in the framework of its previous informal definition, also seems to be a key notion for the definition of intelligence.

The goal of this paper is to to apply the recent theory of non-uniform interactive evolutionary systems in the domain of cognitive systems. This will be done by showing that non-uniform interactive evolutionary systems capture in a natural way the computational properties of cognitive systems and by interpreting the results from the theory of non-uniform interactive evolutionary systems within the framework of cognitive systems. Along these lines we show four results that might be of interest from the viewpoint of the general theory of cognitive systems.

First, we show that in general cognitive systems may possess a super-Turing computational power. This means that in principle such systems can solve more problems than the classical Turing machines.

Second, based on our modelling, we offer a plausible explanation why human thought appears to be uncomputable (cf. [21]).

Third, we show that there exists an infinite number of infinite proper hierarchies of cognitive systems each of which can solve strictly more problems than all its predecessors in the hierarchy at hand. Within some hierarchies there are classes of problems which correspond to the level of intelligence reached at Singularity point. Thus, if the ability to solve the problems is taken as a measure of cognitive systems then theoretically there is no upper limit to this measure.

Last but not least, we also try to estimate at what level of arithmetical hierarchy the cognitive systems corresponding to the human level intelligence are to be sought. We give arguments pointing strongly towards the Σ_2 level of this hierarchy (cf. [6]).

The last four claims represent the original contributions to the theory of cognitive systems.

The structure of the paper is as follows. In Section 2, we will generalize the notion of cognitive systems also to include "hybrid" systems consisting of a combination of natural (living) and artificial systems. In Section 3, we will argue that any finite cognitive system can be modelled by interactive finite transducers. Next, in Section 4, we show that the main essence of information processing by cognitive systems is captured by the models of non-uniform interactive evolutionary systems and we introduce two representatives of such systems: evolutionary automata and interactive Turing machines with advice. In Section 5, a super-Turing computational power of cognitive system is shown and an explanation of apparent uncomputability of human thought (cf. [21]) is offered, too. An infinite proper hierarchy of cognitive systems is presented in Section 6. In Section 7, we speculate on the computational power of human intelligence and present reasons why this power appears to be upper-bounded by the class Σ_2 of the arithmetical hierarchy. Section 8 contains the conclusions.

2 EXTENDING THE NOTION OF COGNITIVE SYSTEMS

By the end of the past century computationalism has obtained an unexpected support both from the theoretical physics and computer science. In 1985 a paper by the theoretical physicist D. Deutsch appeared [7], showing that any real (dissipative) finite physical system can be efficiently simulated by a quantum computer. Since a quantum computer can be simulated by a Turing machine (albeit, as it seems, quite inefficiently) we have a proof of the computationalistic claim that, e.g., man can be genuinely simulated, at least in principle, by a quantum computer. Another result from the computational complexity theory asserts that this simulation will be efficient, indeed (it will be of polynomial time complexity w.r.t. the size of the simulated physical system [2]).

In our approach we will further generalize the scope of computationalism by proceeding beyond the cognitive abilities of living organisms *per se*: our considerations will include any organisms (such as humans) equipped by whatever device which will "strengthen" their cognitive capabilities, or allow their new quality. The Hubble telescope mounted on a satellite encircling the Earth may serve as an example of such a device of which the control and computing center on the Earth is also a part. No doubts that such a machinery will strengthen the cognitive abilities of an observer using this device. Clearly, using this device, an observer gets an access to data inaccessible to him by his own senses. Moreover, these data are processed in a way which, without computers, is also beyond men's abilities. We will call the resulting system, i.e., an observer as well as his or her apparatus, the *cognitive system*. The resulting "hybrid" cognitive system is clearly endowed by a new quality of cognition which is unattainable for a man without the respective devices. Let us be broad–minded by not insisting on the cognitive device being really constructed and at one observer's disposal in his or her experiments. We will be happy just with the gedankenexperiments, i.e., with the situation when the assumed existence of a "cognitive amplifier" does not violate any natural law. Standing firmly on the ground of computationalism, any kind of devices just mentioned can eventually be thought of as a data processing system. In the end, using this rather general approach, everything reduces to the question about the limits of all "thinkable" cognitive (read: computational) systems based on whatever principles obeying natural laws.

Therefore, as our starting point we accept computationalism and in order to get insight into human-level artificial or non-biological intelligence we will model cognitive systems by suitable computational models and investigate their computational properties.

3 MODELLING COGNITIVE SYSTEMS BY FINITE SYSTEMS

The basic notion we will be using for a while is the notion of a *configuration* of some finite artefact, organism, or of a matter in a fixed

space volume. All these categories will be termed as *devices*. A configuration of a device results from a particular observable or measurable arrangement of parts or components of the device at hand. We will say that in two successive times a device is in the same configuration if in the respective times the arrangements defining the configurations are the same. The contemporary quantum physics sets a theoretical upper bound on the number of configurations a device of a given mass and volume can enter. S. Lloyd has shown [18], [19] that the number of bits which can be represented by one kg of a matter in a volume of one liter can be of the order at most 10^{31}. That is, such a volume can enter approximately $2^{10^{31}}$ configurations. The former number is a huge, but finite number which can be seen as an upper limit on the memory capacity of conceivably the most efficient memory of a given size. It follows that within the framework of the previous consideration any finite device can serve as a memory of a capacity given by observable or measurable physical parameters of this device.

A finite device can be seen as a computing device that computes in accordance with a set of transition rules (i.e., with a program) if and only if it fulfills the following rather general conditions:

1. it must be possible to set the device into a distinguished initial configuration;
2. the device in a given configuration, possibly interacting with its input data, will enter the next configuration; the dynamics of such a transition must correspond to the transition rules, i.e., the device must "all by itself" cause the transition from one configuration into the other in accordance with its program;
3. the input data need not all be present at the beginning of a computation; rather, the data can appear at unpredictable times.

We claim that the previous idea of a computing device is general enough also to cover any realizations of finite non-evolvable cognitive systems. The adjective "finite" is important and in this particular case it means a system that can enter but a finite number of configurations.

The property ensuring that the device causes something to happen "all by itself" means that there is a mechanism in the device working in the desired way: the device is "made" in this way. The transition rules need not be explicitly known—it is enough if they exist and are finitely describable. The classical real computer can serve as the prime example of such a device; here the transition rules are known, similarly as in the case of automatic teller machines, mobile phones, etc. The brain presents another example of a computing device with the unknown set of transitions, but the computationalists believe that it does exist. A rock, a picture, a memory card, a mathematical model of a Turing machine are examples of devices which do not compute in the sense defined above.

Note that we defined neither the result of the computation, nor its termination. This has been done intentionally—our computing device should realize potentially *never ending computations*. Stated differently, the device transforms a potentially infinite stream of input data (which are called stimuli in the case of cognitive systems) into a potentially infinite stream of output data called actions in the case of living organisms; the sequence of actions corresponds to the behavior. In this case it is possible and the definition admits that some input data can represent reaction of the environment to some actions. Hence, we can speak about *interactive computations*. Obviously, with the device just described we can also realize finite computations—simply by artificially restricting the input stream. E.g., from a certain position the input stream will consist but of empty symbols, and we will be interested only in terminating computations.

In the sequel, we will only deal with classical (i.e., with discrete, non–quantum) computational devices of a finite size. Formally, computations of such devices are equivalent to computations of so–called interactive finite–state automata with output, which are also called *interactive finite-state transducers* [27]. Note that other equivalent formalisms capturing interaction and non-termination are known, but we prefer the framework of transducers which, as we will later see, can be seen as evolutionary forerunners of Turing machines.

There is no need to define interactive finite transducers formally here. We only note that each interactive finite transducer defines a relation between the input and output (data) *stream*. In general, a cognitive system can have several input and output ports through which the data stream into the systems or out of it. However, in our modelling we see all entering data as encoded into a single input stream (by increasing the input alphabet of the corresponding interactive finite transducer) and similarly we also treat the output streams. Then we speak about the *translation* of the input stream to an output stream.

Thus, comparing the notion of an interactive finite transducer with that of the commonly known finite–state automaton with output (so–called Mealy automaton) we see that the input data for an interactive finite transducer are not given on an input tape before the start of a computation and there need not be a finite number of them. That is why the output of an interactive finite transducer can also be an infinite stream.

Thesis 1 *From a computational viewpoint, any finite cognitive system is equivalent to an interactive finite transducer.*

We have intentionally defined the previous claim as a thesis — since a finite cognitive system is not a formally defined notion, the previous equivalence cannot be formally proved. However, it can be rejected if someone will come with a design of a finite cognitive system that, from a computational viewpoint, would not be equivalent to an interactive finite transducer.

4 NON-UNIFORM EVOLUTIONARY INTERACTIVE SYSTEMS

Returning back to the problem of modelling general cognitive systems, we should stress that so far our modelling by interactive finite transducers has been restricted to the case of finite cognitive systems. Obviously, for finite cognitive systems there is no way to trespass a certain finite number of configurations due to their finite size. To put it differently, the evolution (or learning) of such systems can only happen within a bounded space of all reachable configurations. However, as we have stressed in the Introduction, in a general case of a cognitive system we would also like to allow an evolution of such a system beyond the limits imposed by the number of achievable configurations. Moreover, we would also like to include a possibility of deep "structural" changes of such systems that can happen over time, induced, e.g., by changes of their transition rules (this also covers the ability to work with a larger set of symbols). In "real" world, such changes may correspond to the (Darwinian) evolution of some species along a certain evolutionary path.

In order to model the evolution of a computational system over time we have in mind we consider the idea of (ordered) *sequences* of interactive finite transducers. The notion of sequence of finite computational devices has been used in computational complexity theory in different contexts, e.g., to capture the computational power of non-uniform families of circuits (cf. [1]). In the sequence of interactive

finite transducers, each interactive finite transducer corresponds to a "stable evolutionary period" in which evolution (if any) can be realized within the respective space of achievable configurations. Should this space be exhausted or should there be an evolutionary change that cannot be captured by the current design of the interactive finite transducer at hand (increasing the number of internal states, increasing the input or working alphabet, change of the transition rules), then a new interactive finite transducer must be considered. This gives rise to a potentially infinite sequence of interactive finite transducers in which the i-th automaton corresponds to the contents and computations of a cognitive system during its i-th stable period. In the course of this time, only the i-th automaton receives input and produces output.

We have arrived at the computational model called the *evolving automaton*, introduced in [27].

Definition 2 *The* evolving automaton *with a schedule is an infinite sequence of interactive finite transducers sharing the following property: each transducer in the sequence contains some subset of states of the previous transducer in that sequence. The schedule determines when each transducer has to stop processing its inputs and thus, when is the turn of the next transducer.*

The condition that a given interactive finite transducer has among its states a subset of states of a previous interactive finite transducer captures one important aspect: it is the persistence of data in the evolving automaton over time (cf. [10]). In the language of finite automata, this condition ensures that some information available to the current automaton is also available to its successor. This models passing of information over generations.

On an on–line delivered potentially infinite sequence of the input symbols, the schedule of an evolving automaton determines the *switching times* when the inputs to an automaton must be redirected to the next automaton. This feature models the (hardware) evolution.

An evolving automaton is an infinite object given by an explicit enumeration of all its elements. There may not exist an algorithm enumerating the individual automata. Similarly, the schedule may also be non-computable. Note that at each time a computation of an evolving automaton is performed by exactly one of its elements (one automaton), which is a finite object.

It follows that the evolving automata are *non-uniform*, interactive evolutionary systems just like families of circuits: their development over time cannot be described by an algorithm. The cognitive systems are also a case in point: e.g., the decision to change the architecture of a cognitive system may be the result of evolutionary pressure that has nothing to do with computability. Thus, a cognitive system may take its own action in order to get information that will update its knowledge base (e.g., by connecting to the Internet, or by "making a conversation" with another cognitive system), and this may happen at unpredictable times and in an unpredictable way. By the way, the Internet itself can be seen as an evolutionary automaton as shown in [33].

General computing devices (and as a special case, finite cognitive systems) modelled via interactive finite transducers were restricted by the finiteness condition that prevented any memory growth in such gadgets. However, some computing devices can have a specific ability to increase their memory capacity. This can be achieved either by an additional mechanism or by connecting several computing mechanisms together. This additional memory capacity enables these devices to create and exploit a potentially unbounded set of configurations. A so–called *interactive Turing machine* introduced in [27] can serve as an example of such a device.

An interactive Turing machine is basically a standard Turing machine which has no input tape. Instead, it reads the input symbols via the input port and sends the output symbols to its output port. An interactive Turing machine can be seen as an interactive finite transducer which in order to increase its memory capacity (depending on the cardinality of its set of states) makes use of a potentially infinite tape. This tape alone cannot compute but in a symbiosis with an interactive finite transducer which is endowed by the ability to move along the tape while reading and rewriting the symbols on the tape leads to a more powerful computational device than was the interactive finite transducer alone. An interactive Turing machine computes all that was computed by an interactive finite transducer, but also more than that. This is because it can enter more than a finite number of configurations.

From the viewpoint of its construction, an interactive Turing machine is the extension of a classical Turing machine for the case of infinite input streams; this is what enables an interactive Turing machine to compute "more" than the classical Turing machine. For instance, an interactive Turing machine can process an infinite sequence of finite data segments. Of course, each such segment can also be processed by a classical Turing machine. However, the latter machine has no means for "transferring" information obtained from processing a finite segment in one run into the next run. This is simply because the classical Turing machine, after terminating its computation, cannot be restarted from the configuration in which it has terminated its previous computation: according to its definition, the classical Turing machine must start a new computation from its initial state, with all its tapes empty. For instance, the classical Turing machine cannot realize the following translation: if a segment of a stream gets accepted (the machine produces 1), then the following segment will always be rejected (the machine produces 0). The computational abilities of interactive Turing machines are studied in [28].

Obviously, interactive Turing machines capture well the ability of computing devices to cope with the growing demand on the memory size. In order to further increase the computational power of interactive Turing machines we will proceed to a yet more powerful model of Turing machines: the so-called *interactive Turing machine with advice*. The model extends the well-known and well-studied model of (ordinary) Turing machines with advice in computational complexity theory (cf. [14]) to the case of interactive computations:

Definition 3 *An* interactive Turing machine with advice *is a Turing machine whose architecture is changed in two ways:*

- *instead of an input and output tape it has an* input port *and an* output port *allowing for reading or writing potentially infinite streams of symbols;*
- *the machine is enhanced by a special, so-called* advice tape *that, upon request, allows for insertion of a possibly non-computable external information that takes a form of a finite string of symbols. This string must not depend on the concrete stream of symbols read by the machine until that time; it can only depend on the number of those symbols.*

An advice is different from an oracle also considered in the computability theory: an oracle value can depend on the current input (cf. Turing, 1939). The interactive Turing machines with advice also represent a *non-uniform model of interactive, evolving, and time-unbounded computation*. Such machines capture well an interactive and time-unbounded software evolution of cognitive systems.

The mechanism of advice functions is very powerful and in fact it can provide an interactive Turing machine with any non-computable

"assistance". For theoretical and practical reasons it is useful to restrict the size of advice growth in interactive Turing machines with advice to polynomial functions. With advice functions that grow exponentially one could encode arbitrary oracles in advice. Van Leeuwen and Wiedermann proved a perhaps surprising result showing the computational equivalence of interactive Turing machines with advice with the evolving automata.

Proposition 4 *Evolving automata can simulate interactive Turing machines with advice and vice versa.*

Based on the previous two models, van Leeuwen & Wiedermann (2001) have formulated the following thesis:

Thesis 5 Extended Turing Machine Paradigm: *A computational process is any process whose evolution over time can be captured by evolving automata or, equivalently, by interactive Turing machines with advice.*

Interestingly, the paradigm also expresses the equivalence of software and hardware evolution.

In Wiedermann & van Leeuwen (2008) the authors have shown that the paradigm captures well the contemporary ideas on computing. The fact that it also covers cognitive systems adds a further support to this paradigm. For contemporary computing the extended Turing machine paradigm appears to play a role that is similar to that played for "classical" computing by the classical Turing machines (cf. [26, 33]).

Thesis 6 *From a computational point of view, cognitive systems are equivalent to either evolving automata or, equivalently, interactive Turing machines with advice.*

5 THE SUPER-TURING COMPUTING POWER OF COGNITIVE SYSTEMS

Recall that the power of cognitive systems is measured in terms of sizes of sets of different reactions (or behaviours) that those systems can produce in potentially infinite interactions with their environment.

The super-Turing power of cognitive systems is shown by referring to super-Turing computing power of interactive Turing machines with advice.

Namely, in van Leeuwen & Wiedermann (2001) it was shown that such machines can solve the halting problem. In order to do so they need an advice that for each input of size n allows to stop their computation once it runs beyond a certain maximum time. This time is defined as the maximum, over computations over all inputs of size n and over all machines of size n that halt on such inputs.

Proposition 7 *Cognitive systems have super-Turing computational power.*

Roger Penrose (1994) asked about the power of human thoughts: how to explain the fact that mathematicians are able to find proofs of some theorems in spite of the fact that in general (by virtue of Gödel's or Turing's results) there is no algorithm that would always lead to a proof or refutation of any theorem. In our setting the explanation could be that the mathematicians discover a "non-uniform proof", i.e., a way of proving a particular theorem at hand and probably nothing else. This proof is found using a kind of heuristic search over known results in mathematics. All these results form a certain kind of "blocks" or modules from which sometimes, after their proper adaptation, proofs of new theorems can be constructed. The whole procedure is not unlike the process of creating a complex program system (in accordance with its specification) from simpler modules with known properties. In the computability theory a process of systematic enumeration of an infinite number of candidates in order to find a candidate satisfying the required conditions (a solution) is known as Levin's search [17]. The above described heuristic search can be seen as an informal realization of Levin's search adapted to a restricted domain of mathematical proofs over building blocks of known partial related results. The final solution then emerges in the mind of a mathematician (i.e., in a cognitive system) who happens to be knowledgeable of the required facts. For this to happen mathematicians in general perform "searches" in the literature for related results, take part in generating such results and interact unpredictably among themselves in order to spread the required knowledge and speed-up the search process. In particular cases, such a "search" can last over generations of mathematicians. When the respective "knowledge blocks" are ready then it is only matter of time and chance, when and where the final solution emerges. This also explains why solutions of certain problems appear independently, at about the same time, at several places. Similar ideas on how mathematicians "create" their proofs have been recently presented by Blum [3].

6 HIERARCHIES OF COGNITIVE SYSTEMS

For interactive Turing machines with advice or for evolving automata one can prove that there exist infinite proper hierarchies of computational problems that can be solved on some level of the hierarchy but not on any of the lower levels (cf. [29, 30]). Roughly speaking, the bigger the advice, the more problems can be solved by the underlying machine.

Proposition 8 *There is an infinity of infinite proper hierarchies of cognitive systems of increasing computational power.*

Among the levels of the respective hierarchies there are many cognitive systems corresponding formally (and approximately) to the level of human intelligence (the Singularity level), and also infinitely many levels surpassing it in various ways.

The interpretation of the last results within the theory of cognitive systems is the following one. There exist infinite hierarchies of computations of cognitive systems dependent on the amount of non-computable information injected into such computations either via advice or via the design of the members of the respective evolving automaton. The bigger this amount, the more translations can be realized. Among the levels of those hierarchies there are many levels corresponding formally (and approximately) to the level of human intelligence (the Singularity level — cf. [16]) and also infinitely more levels surpassing it in various ways. The complexity classes defining individual levels in these hierarchies are partially ordered by the containment relation.

7 CHARACTERIZING THE COMPUTATIONAL POWER OF HUMAN INTELLIFNECE

The previous hierarchy result was good enough for proving the existence of a level in the complexity hierarchy of cognitive systems corresponding to the level of human intelligence. Can we characterize this level more precisely? There is increased theoretical evidence that

the computational power of human intelligence (aided by computers or not) is upper bounded by the Σ_2 level of the arithmetical hierarchy. This level contains computations which are recursive in the halting problem of the classical Turing machines. For instance, Penrose [21] argues that human mind might be able to decide predicates of form $\exists_x \forall_y P(x, y)$, i.e., the Σ_2 level. The computations within this class can answer question related to the halting of the arbitrary (classical) Turing machines for any input (*"Does there* exist *a Turing machine such that* for all *Turing machines and* for all *inputs decides whether they halt?"*). Similar conclusions have been reached during the last few decades by a number of logicians, philosophers and computer scientists looking at the computations as potentially unbounded processes (cf. [25]).

Recent model of van Leeuwen and Wiedermann [25] of such computations — so called *red-green Turing machines* — offers perhaps the simplest illustration of the main features of such computations. A red-green Turing machine is formally almost identical to the classical model of Turing machines. The only difference is that in red-green Turing machines the set of states is decomposed into two disjoint subsets: the set of green states, and the set of red states, respectively. There are no halting states. A computation of a red-green Turing machine proceeds as in the classical case, changing between green and red states in accordance with the transition function. The moment of state color changing is called *mind change*. A formal language is said to be recognized just in case when on the inputs from that language the machine computations "stabilize" in green states, i.e., from a certain time on, the machine keeps entering only green states. Similarly, a language is said to be accepted if and only if the inputs from that language are recognized, and the computations on the inputs outside that language eventually stabilize in red states. The model captures in a neat way the main features of the current thinking of computing: namely, viewing computations as potentially infinite processes. The non-uniform evolution is not included — the model concentrates merely on uniform models of unbounded computations. Van Leeuwen and Wiedermann have shown that the computational power of red-green Turing machines increases with the number of mind changes allowed (it climbs along the so-called Ershov hierarchy, cf. [6]) and for any finite number of mind changes red-green Turing machines recognize languages in Σ_2 and accept languages from Δ_2. In fact, computations of red-green Turing machines exactly characterize Σ_2 (or Δ_2 in case of acceptance). This, together with the similar results achieved with the help of other machine or logical models of unbounded computation, along with the expected exponential increase of computational power leading to the Singularity point, suggests the following thesis.

Thesis 9 *The computational power of cognitive systems with human-like intelligence is upper-bounded by the Σ_2 class of the arithmetical hierarchy.*

Note that the previous thesis does not claim that the cognitive systems can solve all problems from Σ_2. Nevertheless, example of the halting problem theorem shows that occasionally human mind can solve specific problems that in general belong to Σ_2.

Even such a simple model as red-green Turing machines can solve the classical halting problem: we take the classical Turing machine with the input for which the halting problem is to be solved. We colour the original halting state as green and add a loop in this state. All the other states are coloured red. Now we run the machine on the given input. Obviously, the computation converges to green states just in case when the original machine halts. Otherwise, it will compute forever in red states. Of course, the problem is that we never know whether, and when an arbitrary red-green machine stabilizes in green or red states. However,such questions can be answered for simple machines.

The so-called *busy beaver problem* asks, for the classical Turing machines with k-states working in binary alphabet, without any input, what is the largest number of steps that such a machine can do before halting. The respective numbers are known only for $k < 5$. It is known that machines with $k = 5$ have the run time of $47,176,870$ steps and for $k = 6$ more than 10^{2879} steps [20]. Even from these figures one can see that with the increased number of states, the running times tend to grow incredibly fast. In fact, no recursive function can express the growth of the respective values as a function of k. What is known is the lower bound on that growth. Green [11] has recursively constructed machines for any number of states and derived a recursive function that provides a lower bound for their running time. He has shown that the running time of busy beaver machines with $2k$ states grows faster than $3 \uparrow^{k-2} 3$, using Knuth's up-arrows notation [15]. For $k = 10$ one gets $3 \uparrow\uparrow\uparrow 3 = 3 \uparrow\uparrow 3^{3^3} = 3^{3^{3^{\cdots^3}}}$ with $3^{27} = 7,625,597,484,987$ terms in the exponential tower. Thus, not even an exponential (or, for that matter, any computable) increase of the computing power of non-biological intelligence, as Kurzweil expects, can match the complexity increase of a busy beaver problem. This seems to be a good argument for claiming that although it is conceivable that artificial general intelligence will soon or later reach the level of human intelligence, any substantial progress towards higher "interesting" levels (let's say in the arithmetical hierarchy) cannot be expected since not even a "super-intelligence" can cope with computationally infeasible tasks.

8 CONCLUSIONS

We have investigated cognitive systems in the framework of the Extended Turing Machine Thesis and presented the results concerning computational power and hierarchies of such systems. The good news is that the cognitive systems may possess a super-Turing computational power and that, theoretically, there is no upper limit as far as the the computational power of such systems is concerned. The bad news is that this power cannot be purposefully harnessed for solution of concrete uncomputable problems. This is because the proof of super-Turing power of such systems is existential (non-constructive). That is, it only makes use of the fact that non-computable information needed for such systems to solve any concrete undecidable problem for sure exists. The proof does not care of how such information can be gained. Unfortunately, there is currently no known realistic way of systematically retrieving such information. Occasionally a properly tuned heuristic might help, as shown at the end of Section 5. On the other hand, the case of busy beaver machines shows that there is no hope for solving certain problems related to the large instances of the halting problem of the classical Turing machines, not even for the future powerful computations invented by non-biological intelligence, as Ray Kurzweil hopes.

At present, perhaps the only theoretically promising way of computing the uncomputable information, and thus, to move the intelligence beyond that corresponding to the Singularity, is offered by so-called relativistic (or "black-holes") computing (cf. [8, 32]), but any progress along these lines seems to depend on the progress of relativistic physics.

ACKNOWLEDGEMENTS

This research was carried out within the institutional research plan AV0Z10300504 and partially supported by a GA ČR grant No. P202/10/1333

REFERENCES

[1] Balcázar, J. L., Diáz, J., Gábarró, J.: Structural complexity, Vol. I, 2nd Edition, Springer-Verlag, Berlin, 1995.

[2] Bernstain, E., Vazirani, U.: Quantum Complexity Theory. Proc. of the 25th Annual Symposium on the Theory of Computing, ACM, New York, 1993, pp. 11–20

[3] Blum, M.: Can (Theoretical Computer) Science Come to Grips with Consciousness? Invited talk at FOCS 2009, 2009

[4] Chalmers, D. J.: A Computational Foundation for the Study of Cognition. Minds and Machines, Vol. 4, No. 4, 1994

[5] Chalmers, D. J.: The Singularity: A Philosophical Analysis, 2009, cf. http://consc.net/papers/singularity.pdf

[6] Cooper, S. B.:. Computability Theory, Chapman&Hall/CRC, 2004.

[7] Deutsch, D.: Quantum Theory, the Church-Turing principle and the universal quantum computer. Proc. of the Royal Society of London A 400, pp. 97-117, 1985

[8] Étesi, G., Németi, I.: Non-Turing computations via Malament–Hogarth space-times, Int. J. Theor. Phys. 41, 2002, pp. 341-370

[9] Goertzel, B.: Artificial General Intelligence: Now Is the Time. Published on KurzweilAI.net, April 9, 2007

[10] Goldin, D.Q., Smolka,S.A., Attie, P. C., Sonderegger, E.: Turing machines, transition systems, and interaction, *Information and Computation* 194:2 (2004) 101 - 128.

[11] Green, M.W.: A Lower Bound on Rado's Sigma Function for Binary Turing Machines, in Preceedings of the IEEE Fifth Annual Symposium on Switching Circuits Theory and Logical Design, pp. 91-94,1964

[12] Hopcroft, J., Motwani, R., Ullman, J. D.: Introduction to automata theory, languages and computation, 2nd Edition, Addison-Wesley, Reading, MA, 2000

[13] Karp, R. M., Lipton, R. J.: Some connections between non-uniform and uniform complexity classes, in Proc. 12th Annual ACM Symposium on the Theory of Computing (STOC'80), 1980, pp. 302-309.

[14] Karp, R. M., Lipton, R.: Turing machines that take advice, *L'Enseignement Mathématique*, II^e Série, Tome XXVIII, 1982, pp. 191–209.

[15] Knuth's up-arrow notation at http://en.wikipedia.org/wiki/Knuth_up-arrow_notation

[16] Kurzweil, R.: The Singularity is Near. Viking Books, 652 pages, 2005

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

[18] Lloyd, S.: Ultimate physical limits to computation. Nature 406 (6799), pp. 10471054, 2000.

[19] Lloyd, S.: How Fast, How Small, and How Powerful: Moor's Law and the Ultimate Laptop. In: Rebooting Civilization, http://www.edge.org/3rd_culture/rebooting/rebooting. html, 2001

[20] Michel, P.: Historical survey of Busy Beavers, http://www.logique.jussieu.fr/ michel/ha.html#tm62

[21] R. Penrose: Shadows of the Mind (A Search for the Missing Science of Consciousness). Oxford University Press, Oxford, 1994, 457 p.

[22] Shettleworth, S.: Cognition, Evolution, and Behavior, Oxford University Press, 1998

[23] Turing, A. M.: On computable numbers, with an application to the Entscheidungsproblem, Proc. London Math. Soc., Vol. 42-2, pp. 230-265, 1936; A correction, ibid., Vol. 43, pp. 544-546, 1937

[24] Turing, A. M.: Systems of logic based on ordinals, Proc. London Math. Soc. Series 2, Vol. 45, pp. 161-228, 1939

[25] van Leeuwen, J.: Computation as Unbounded Process. Slides from the presentation at the Workshop 'Philosophy of the Information and Computing Sciences', Lorentz Center, Leiden 2010, cf. http://www.cs.uu.nl/people/jan/LC-Philosophy-2010.ppt

[26] van Leeuwen, J., Wiedermann, J: The Turing machine paradigm in contemporary computing, in: B. Enquist and W. Schmidt (Eds.), *Mathematics unlimited - 2001 and beyond*, Springer-Verlag, 2001, pp. 1139–1155.

[27] van Leeuwen, J., Wiedermann, J.: Beyond the Turing limit: evolving interactive systems, in: Proc. SOFSEM'01, LNCS Vol. 2234, Springer-Verlag, Berlin, 2001, pp. 90–109.

[28] van Leeuwen, J., Wiedermann, J.: A Theory of Interactive Computation. In D. Goldin, S.A. Smolka & P. Wegner (Eds.), *Interactive Computation: the New Paradigm* (pp. 119-142). Berlin: Springer-Verlag. 2006.

[29] Verbaan, P.R.A., van Leeuwen, J., Wiedermann, J.: Complexity of Evolving Interactive Systems. In J. Karhumäki et.al., Eds., *Theory Is Forever,* LNCS Vol. 3113, Springer-Verlag, Berlin, 2004, pp. 268–281

[30] Verbaan, P.R.A.: *The Computational Complexity of Evolving Systems*, Ph.D. Thesis, Dept. of Information and Computing Sciences, Utrecht University, 2006.

[31] Wiedermann, J., van Leeuwen, J.: The Emergent Computational Potential of Evolving Artificial Living Systems. Ai Communications 15, 4, 2002, pp. 205-216.

[32] Wiedermann, J., van Leeuwen, J.: Relativistic Computers and Non-Uniform Complexity Theory. In: Unconvential Models of Computation (UMC'2002), LNCS Vol. 2509, Berlin, Springer 2002, pp. 287-299

[33] Wiedermann, J, van Leeuwen, J.: How We Think of Computing Today. (Invited Talk) Proc. CiE 2008, LNCS 5028, Springer, Berlin, 2008, pp. 579-593

On the State of Superposition and the Parallel or not Parallel Nature of Quantum Computing: a controversy raising point of view

Miclael Nicolaidis[1]

Abstract: In this paper we use ideas coming from the field of computing to propose a model of quantum systems, which gets rid of the concept of superposition[1] while reproducing the observable behaviour of quantum systems as described by quantum mechanics. On the basis of this model we contest the interpretation of quantum mechanics based on the so-called quantum superposition and the related parallel-computing interpretation of quantum algorithms. The goal of this presentation is to bring for discussion in the conference ontological arguments against the so-called quantum parallelism, which are mutually supported with recent advances in complexity analysis of quantum algorithms.

Keywords: philosophy of science and computation, quantum superposition, stochastic computational model, computational interpretation of physics, quantum computing, parallel computing, the ontological argument against quantum parallelism.

1 Introduction

The search for new computational paradigms from Nature leads to strong interaction between physics and computing, which may have a stimulating impact on the foundations of both. Revisiting the interpretations of physics by using ideas coming from computers has been suggested by many authors including R. Feynman (Feynman 1982): "There are interesting philosophical questions about reasoning, and relationship, observation, and measurement and so on, which computers have stimulated us to think about anew, with new types of thinking. And all I was doing was hoping that the computer-type of thinking would give us some new ideas, if any are really needed."

In this paper we address this issue in the context of the interrelations between quantum mechanics and its interpretations on the one hand and quantum computing and its interpretations on the other hand. According to a widely accepted interpretation of quantum mechanics, during coherence each observable of a quantum system is simultaneously in a plurality of states (quantum superposition). The power of quantum computing is explained on the basis of this concept: a quantum algorithm manipulates at the same time the 2^n superposition states of n q-bits. So, *the paradigm of quantum computing seems to accredit the concept of superposition by attributing to it a factual status.* Nevertheless, the concept of superposition raises some important philosophical questions. The position, otherwise salvaging, of certain physicists, including R. Feynman, condensed in "shut-up and calculate", is just one manifestation of the difficulty that feel even the greatest minds of science for attributing to this kind of behaviour a satisfactory sense. So, although there is today a solid consensus around the superposition concept, it is still relevant to analyze it with fresh arguments. This is important of course for physicists and philosophers but also for computer scientists, as this concept is on the very basis of quantum computing. More importantly, ideas coming from the field of computing could be used to examine the relevance of the concept of quantum superposition[2] and the related quantum parallelism.

It is to note that, the parallel capabilities of quantum computing are also questioned in several recent publications. Some authors notice that none of the existing quantum algorithms has the capabilities of a veritable parallel computer (Aaronson 2008). A more recent argument (Lanzagorta; Uhlmann 2008), (Lanzagorta; Uhlmann 2009) is related to the fact that quantum algorithms use unitary operators to transform the state of superposition, imposing important limitations to the computing capabilities of a quantum computer as expressed in the following quote from (Lanzagorta; Uhlmann 2009) "its is entirely possible that, for certain quantum algorithms, the apparent computation savings obtained from the application of the operator to the states in a superposition are always precisely balanced by the increasing number of reversible gates required to implement the operator".

The present paper highlights a fundamental, reason that disables quantum computers from performing truly parallel computations: our arguments are not based on the properties of the operators acting on the state of quantum computers or on complexity analysis of quantum algorithms but on the very fundamental (i.e. ontological) nature of quantum systems and of the so called state of superposition. In particular we propose a stochastic "computational" model that eliminates the state of superposition, while producing the observable behaviour of quantum systems as described by quantum mechanics. Since quantum computing concerns information related to the observable behaviour of quantum systems, a

[1] TIMA Laboratory (CNRS, Grenoble INP, UJF) , France, email: michael.nicolaidis@tima.fr

[2] It is important to clarify that we do not contest the property of quantum systems according to which a linear combination of state vectors is also a state vector (often referred as superposition), but the property according to which, in the state of coherence, each observable of a quantum system is simultaneously in a plurality of states (also referred as superposition). The former is an inherent part of the mathematical basis of quantum mechanics. The later may seem to be an implication of the former, but it is an interpretation associated to the state of coherence.

consequence of this model is that quantum computers do not support the so-called quantum parallelism.

2. Quantum Superposition and Quantum Computing

In quantum mechanics the state of a quantum system is described by a function ψ (the wave function). This function is determined by solving Schrödinger's equation or one of its relativistic counterparts (Klein-Gordon or Dirac). Then, based to this function a plurality of values α_1, α_2, ... is determined for each physical observable of the system, (position, translational momentum, orbital angular momentum, spin, total angular momentum, energy, etc.), together with associated probabilities P_1, P_2, …, according to the following rules:

- At each observable A is associated a Hermitian operator Â (whose form has a certain relation with the expression of the observable in non-quantum physics, i.e. Newtonian or relativistic).
- The values α_1, α_2, ... related to an observable A are the eigenvalues of the operator Â of this observable.
- The associated probabilities are $P_1 = |c_1|^2$, $P_2 = |c_2|^2$, …, with $c_i = \langle\psi_i|\psi\rangle \ \forall\ i \in \{1, 2, \ldots\}$, the inner product of eigenfunction ψ_i of Â and the wave function ψ.
- If the observable A is measured, then, the result will be one of the eigenvalues α_1, α_2, … with a probability equal respectively to $P_1 = |c_1|^2$, $P_2 = |c_2|^2$, ….
- After the measurement, ψ collapses to the eigenfunction ψ_i corresponding to the eigenvalue α_i obtained as result of the measurement. Since ψ becomes equal to the eigenfunction ψ_i a subsequent measurement of the observable A will give as result the corresponding eigenvalue α_i with probability equal to 1 (since, in this case $c_i = \langle\psi_i|\psi\rangle = \langle\psi_i|\psi_i\rangle = 1$)

Before the measurement the system is said to be in coherence. The measurement destroys this state and the system is said to be in decoherence.

The above rules concern observables having discrete spectrum. Similar rules are used in the case of observables having continuous spectrum, such as position or momentum. But in this case the statistical distribution is described by a probability density function P(a) (e.g. $|\psi|^2$ gives the probability density function of position).

These are the strictly necessary concepts and mathematical formalism required to describe the observable behaviour of quantum systems. But the state of coherence is strange, since we cannot allocate to the observable a unique value as in the macroscopic world. To give it a sense, this state was *interpreted* by considering that the observable is in superposition on a plurality of values. Figure 1 illustrates the concept of superposition by showing the observable A to be simultaneously on several values α_1, α_2, ..., α_n, … on which correspond certain probabilities p_1, p_2, …, p_n, …

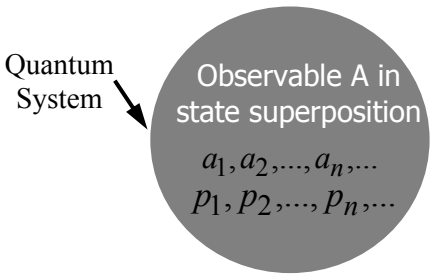


Figure 1. The state of superposition of an observable A.

Let us now remind in few words how quantum computing is supposed to process information by exploiting the superposition state. A q-bit corresponds to an observable A of a quantum system which can be in two possible states. We can associate the value 0 to the one of these states and the value 1 to the other. According to the interpretation of quantum mechanics based on the concept of superposition, in the state of coherence a q-bit may be at the same time at the state 0 and at the state 1. In a quantum computer comprising n q-bits, in the state of superposition these n q-bits can be simultaneously on 2^n states. A quantum algorithm comprises a certain number of steps which transform the state of the n q-bits. Then, according to the interpretation based on the superposition concept, the power of quantum computing is due to the fact that the quantum algorithm may manipulate at the same time all the 2^n values of the n q-bits (i.e. the computation is executed in parallel 2^n times), while a traditional algorithm manipulates only one of these values.

The claimed parallelism of quantum computing is also accredited if we consider the mathematical formalism used to describe the transformation of the state of the n q-bits performed by quantum algorithms. This formalism comprises:

- The description of the state of the n q-bits by a wave function $|\psi(t)\rangle = \sum_{i=0}^{2^n-1} c_i(t)\,|\psi_i\rangle$, where $|\psi_i\rangle$ ($i \in \{0, 1, \ldots, 2^n-1\}$) are the 2^n possible states of the n q-bits, and $c_i(t)$ is a complex number such that $c_i(t) = \langle\psi_i|\psi(t)\rangle$ and $|c_i(t)|^2$ is the probability of occurrence of state $|\psi_i\rangle$ during a measurement.
- A 2^n x2^n unitary matrix U(t,0) , which transforms the initial state $|\psi(0)\rangle$ of the n q-bits into their final state $|\psi(t)\rangle$ given by $|\psi(t)\rangle = U(t,0)|\psi(0)\rangle$.

Thus, the quantum algorithm would perform the operation $|\psi(t)\rangle = U(t,0)|\psi(0)\rangle$ which transforms the initial 2^n coefficients $c_i(0)$ into the final ones. That is, it performs simultaneously an operation over 2^n coefficients, revealing a parallel manipulation over 2^n values. Thus, it seems that we are obliged to accept the superposition as a factual state and the idea that quantum computers perform a truly parallel computation as valid. However, in the following we pretend that this may not be the case. But how it could be possible to support such a claim, when the above formalism clearly indicates the opposite? The answer is: that a formalism can be convenient for representing a

natural process in the human mind, but it does not necessarily describes the way nature proceeds.

3. Contesting the Existence of Quantum Superposition

The position, otherwise salvaging, of numerous physicists, including R. Feynman, condensed in "shut-up and calculate", is just one manifestation of the difficulty that feel even the greatest minds of science for attributing to the behaviour of quantum systems a satisfactory sense. One of the "paradoxes" of quantum mechanics concerns the concept of superposition. Below we raise some issues related to this concept, which lead to a model of quantum systems that eliminates this concept:

- It is difficult to give a clear sense to the concept of superposition. For instance, what exactly means that a particle is simultaneously in several positions in space?
- According to the superposition interpretation, during a measurement one of the values in superposition is randomly produced, following a well defined statistical distribution. Nevertheless, during each particular measurement, a particular value is selected among all the values in superposition. Thus, there is "something" that selects this particular value and not another one. But the superposition interpretation does not provide any means for realizing this selection. This issue is resolved in the Many Worlds interpretation (Everett 1957as it considers that during each measurement all values in superposition are realized in an equal number of parallel universes (so there is no need for selecting any particular value). However, this leads to an extraordinary proliferation of parallel universes!
- Since observables of continuum spectrum, like the position and momentum (but also observables of discrete spectrum like the energy), have infinite number of eigenvalues, a veritable superposition will result to the superposition of infinite number of values that correspond to *infinity* memory capacity and computing power!
- The concept of superposition describes a metaphysical state, since there are no means for observing directly this state (i.e. to observe that the system is indeed simultaneously in several states). In fact, any attempt for doing so (a measurement) leads to decoherence and provides as result a single value.
- The state of superposition is not among the strictly necessary concepts for describing the behavior of quantum systems. According to these concepts:
  - The state of the system is described by a wave function which is obtained as the solution of Schrödinger's equation or one of its relativistic counterparts (Dirac, or Klein-Gordon equation),
  - An operator is associated to each observable and the algebra of operators is used for determining the statistical distributions of the observables from the wave function.
  - The statistical distribution of an observable gives the values that can be obtained when the observable is measured (eigen-values of the operator associated to the observable), together with the corresponding probabilities.
  - After a measurement the wave function becomes equal to the eigen-function corresponding to the eigen-value obtained as the result of the measurement.
- Since the state of superposition is metaphysical and does not belong to the strictly necessary concepts that describe the behavior of quantum systems, we can conclude that it is a non-mandatory interpretation of the "strange" state of coherence. As it implies infinite computing power for quantum systems its validity is highly questionable.

4. Stochastic computation Model of Quantum Systems

In this section we propose a stochastic computational[3] model of quantum systems, which reproduces their observable behaviour as described by quantum mechanics. Such a model considers that the behaviour of a quantum system is the result of a computation-like process which produces the observable states of the system (the results of measurements). Any internal state involved in this computation is not observable[4]. Thus, we can consider that the computation is taking place in a meta-object, as nothing concerning this object is observable except the states it returns when a measurement is performed. We call such an object a computing meta-object (CMO). But, what kind of computation this object should perform?

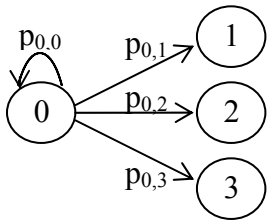


Figure 2. Probabilistic computation

Due to the stochastic behaviour of quantum systems we can think about probabilistic computation (Kaye et al 2007). Such a computation performs random state transitions as illustrated in figure 2. In this figure, the transition from state 0 to state i is performed with probability $p_{0,i}$. With this computation, the system state takes a precise value at each instant, but the choice of one or another value is done probabilistically. Thus, this kind of computation could realize the behaviour of quantum systems consisting in selecting probabilistically a precise value for an observable, when this observable is measured. But it fails to implement the behaviour of a quantum system when it is in the state of coherence, since in this state the observables of the system have no precise value.

[3] In this model, we are not interested about the complexity analysis of the one or the other quantum algorithm, but about the fundamental (i.e. ontological) nature of quantum processes, and more particularly about whether they have to be considered as parallel processes based on a veritable state of superposition, or something simpler.

[4] This is also the case for the other interpretations of quantum mechanics (e.g. the wave function and the states in superposition are not observable, only the results of measurements are).

In fact it misses the way a quantum system determines during coherence the statistical distributions of its observables, which provide the probabilities to obtain the one or the other value when "measuring" an observable.

So we need a different computational model. We are interested for a model which considers only the concepts that are strictly necessary for describing the behavior of quantum systems. The proposed stochastic computational model for the CMOs is illustrated in figure 3. But before detailing it, let us make clear that, from the physical point of view, a piece is missed in the interpretation of quantum mechanics based on the superposition concept. Indeed, the results of measurements of the physical observables of a quantum system are stochastic. Quantum mechanics provides a mathematical formalism allowing determining the statistical distributions of these measurements. The superposition interpretation says that during a measurement one of the values in superposition is randomly produced, following the probabilities (or the probability density function) describing the statistical distribution of the observable. Therefore, during each particular measurement, a particular value is selected among the values in superposition. Thus, there is some mechanism that selects this particular value and not another one. *Quantum mechanics do not provide any means realizing such a selection. That is, it does not provide a mechanism which realizes the observable values of the physical observables. Therefore, from the physical point of view, quantum mechanics is incomplete. Such a mechanism is external to this theory and has to be considered as meta-mechanism.*

Note however that the many-worlds interpretation (MWI) (Everett 1957; DeWitt and Wheeler 1968; DeWitt 1972; DeWitt and Graham 1973) of quantum mechanics eliminates the necessity of a meta-mechanism for selecting the value observed during measurements. According to MWI, there is no selection of a particular value during each measurement. Instead, all values in superposition are realized in an equal number of parallel universes. Thus, no meta-mechanism is needed for performing this selection. However, beyond the extraordinary proliferation of parallel universes introduced by this vision, the existence of entities that are not part of our observable universe is not eliminated, as all the universes parallel to ours are not part of our observable world.

In the stochastic computational model illustrated in figure 3, the CMO produces the observable behaviour of quantum systems by means of a computation-like process, which uses deterministic functions to transform a stochastic signal w_a into a signal that during each measurement of an observable provides the result of the measurement. As the stochastic signal w_a is not observable, it has to be considered as a meta-signal. Note that this signal can not be considered as a hidden variable. This is because, as w_a is stochastic, there is no cause-effect relation between on the one hand the values it will take in the future and on the other hand its present and past values and the present or past states of the system and its environment. Thus, the outcomes of future measurements can not be determined by the present and past states of signal w_a, of the quantum system and of its environment.

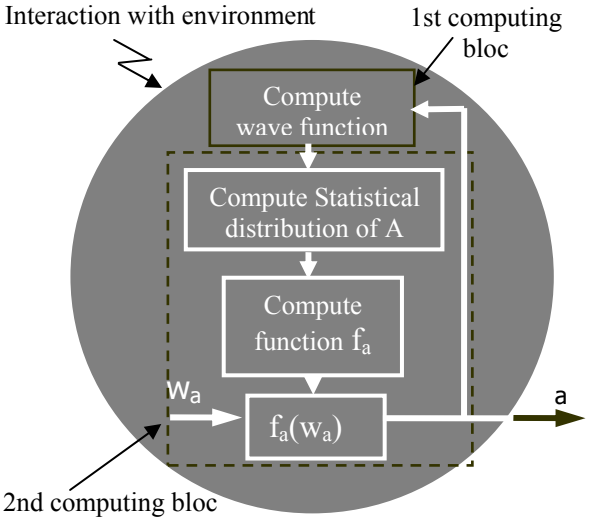


Figure 3. Computing meta-object (CMO)

As shown in Fig. 3, a CMO comprises 2 computing blocks:

The first bloc computes the wave function by resolving a differential equation (such as the equation of Schrödinger, Klein-Gordon, or Dirac) for a function of potential corresponding to the CMO environment (ultimately determined by the states of the other CMOs with which it interacts). The wave function represents at any time the state of the CMO.

The second bloc performs computation only during "measurements": each time an observable A is measured this block computes a deterministic function f_a and then uses this function to transform the stochastic meta-signal w_a into a stochastic signal α which provides the result of the measurement of observable A. As the function f_a transforms a unique value (the current value of w_a) to produce the current value of α, signal α will bring a unique value (during measurements), or no value (outside measurements).

The function f_a is computed in a manner that the statistical distribution (values α_1, α_2, … and the corresponding probabilities p_1, p_2, …) of signal α produced by the transformation $\alpha = f_a(w_a)$ is identical to the statistical distribution of the corresponding observable A determined by the rules of quantum mechanics. This computation will be possible if, for any statistical distribution $p(\alpha)$ of an observable A, it exists a deterministic function f_a which transforms the stochastic signal w_a to a stochastic signal α that has the statistical distribution $p(\alpha)$. In a research report (Nicolaidis, 2009) we show that for a signal w_a having any arbitrary but given statistical distribution of continuous spectrum $s(w_a)$ and for any statistical distribution $p(\alpha)$, it always exists a function f_a which produces a signal $\alpha = f_a(w_a)$ whose statistical distribution is equal to $p(\alpha)$. This result is valid for

statistical distributions $p(\alpha)$ of discrete spectrum as well as of continuous spectrum. This result is easily generalised to the case of vectorial observables (like for instance) the position $\bar{r}$ or the momentum $\bar{p}$).

The measurement induces decoherence: the observable A takes a precise value equal to the value of signal α (the result of the measurement). Accordingly, the first computing bloc computes a new wave function which is compatible with this particular state of observable A (that is the wave function becomes equal to the eigen-function associated to the eigen-value obtained on signal α during the measurement). This influence of the value of signal α on the computation of the wave function is represented in figure 3 by an arrow which brings the value of signal α to the input of the first computing bloc.

As an illustration, the computation performed by a CMO could comprise the following steps:

1st computing bloc (all the time). Computation of the wave function according to the rules of quantum mechanics.

2nd computing bloc (only during measurement of an observable A). Computation of the statistical distributions $p(\alpha)$ of observable A (according to the rules of quantum mechanics); computation of the function f_a which transforms the signal w_a to the signal α having the statistical distributions $p(\alpha)$ (according to the method described in the annexe of a research report (Nicolaidis, 2009)); Transformation of meta-signal w_a to signal α by means of function f_a: $\alpha = f_a(w_a)$. From the way we have derived function f_α in the above annexe, the statistical distribution of signal α will comply with quantum mechanics. Thus, the results of measurements of observable A will comply too.

1st computing bloc (only during measurement of an observable A). Computation of a new wave function compatible with the measured value α_i of observable A (according to the rules of quantum mechanics). That is, the 1st computing block sets the wave function ψ to be equal to the eigenfunction ψ_i of A that corresponds to the eigenvalue α_i of A obtained on signal α as the result of the transformation $\alpha = f_a(w_a)$. Since $\psi = \psi_i$, then, the statistical distribution computed by the 2nd computing block during a new measurement of A will give as result α_i with probability 1. Then, the computation of f_α will give $f_\alpha(w_\alpha) \equiv \alpha_i$, i.e. a function returning the value α_i for all values of w_α, (decoherence). Consequently, any new measurement of this observable will return the unique value α_i generated by the function $f_\alpha(w_\alpha) \equiv \alpha_i$ for any value of w_α.

The above behaviour corresponds to the observable behaviour of quantum systems as determined by quantum mechanics. Thus, the stochastic computational model of figure 3 reproduces this behaviour. In addition, this model eliminates the superposition state, as signal α that brings the value of observable A never takes a plurality of values: either no value or a single value is computed on this signal.

We can conclude that the concept of superposition can be eliminated by using a stochastic computational model that provides the same observable behaviour of quantum systems as the one described by quantum mechanics.

To further support the idea that the concept of superposition does not correspond to a veritable state, we also describe a computational model that is based on this concept and we compare it with the model shown in figure 3. The two models are shown in figure 4. The upper part of this figure illustrates the model related with the state of superposition, while the lower part illustrates the model that considers the strictly necessary concepts only (i.e. the model presented in figure 3). For simplicity, the former will be mentioned as the superposition model and the later as the stochastic computational model.

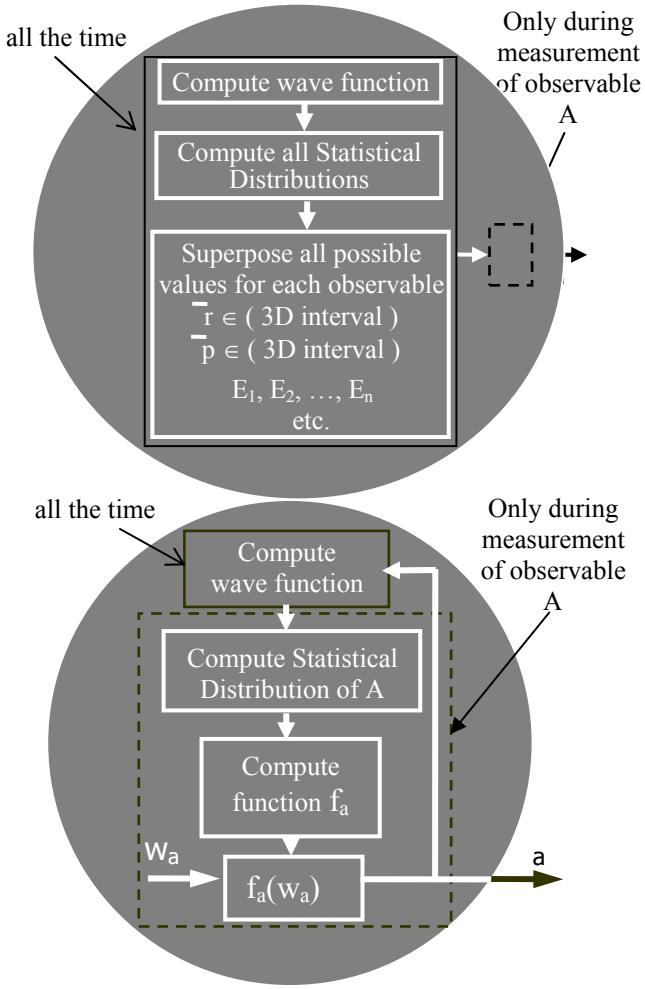


Figure 4. Computing means of superposition model versus stochastic computational model.

The computations performed by the superposition model (upper part of figure 4), include:

Step i (all the time).. Computation of the wave function.

Step ii (all the time). Computation of the statistical distribution for each observable by means of the algebra of operators. These distributions are continues (probability density function) for some observables like position and momentum, and discrete (discrete values with associated probabilities) for other observables like energy or spin.

Step iii (all the time). Creation of a superposition state for each observable A. It comprises all values that have no nil

probability (or no nil probability density) in the statistical distribution of the observable A, together with the associated probabilities (or probability densities). This superposition has to be done for each physical observable, but also for each observable corresponding to any arbitrary Hermitian operator (i.e. an infinity number of operators). This is because, theoretically, any such observable could be measured (for discrete Hermitian matrixes the experimental realisation of the measurement of the observable corresponding to any such matrix was shown as early as 1994 (Reck et al. 1994)), and all possible outcomes of a measurement are supposed to be in superposition before the measurement. Thus, the superposition interpretation requires the computation of the statistical distribution and the creation of a superposition for all such observables!!

Steps iv, v (only during measurement of an observable A). Generation of a value α for observable A each time this observable is measured: pick one of the values in superposition of A by means of a stochastic selection based on the probabilities associated to these values. As noted earlier, the existing interpretations of quantum mechanics does not describe how this happens and do not provide the required mechanism.

Computations i, ii, and iii (shown in a box drawn with black solid-lines in the top part of figure 4) are performed all the time. Computations iv, v (illustrated in the top part of figure 4 by a box drawn with black dashed-lines) is performed only when an observable A is measured. The last computation is numbered iv, v to show the correspondence with the stochastic computational model.

The lower part of figure 4 illustrates the computations performed by the stochastic computational model. As detailed earlier, these computations include:

Step i. Computation of the wave function. This computation is done all the time.

Step ii. Computation of the statistical distribution of the observable A (by means of the algebra of operators). This computation is performed only when an observable A is measured.

Step iii. Void

Step iv. Computation of function f_a corresponding to the statistical distribution of A (by means of the relationships described in the annexe of a research report (Nicolaidis, 2009)). This computation too is performed only when an observable A is measured.

Step v. Transformation of signal w_a to the signal $\alpha = f_a(w_a)$. Again only in case of measurement of A.

So, for the stochastic computational model only computation i (shown in a box drawn with black solid-lines in the lower part of figure 4) is performed all the time. The computations ii, iii, iv and v (shown in a box drawn with black dashed-lines in the bottom of figure 4) are performed only when an observable A is measured. The void step iii is added to ease the correspondence with the superposition model.

We can now compare the stochastic computational capabilities required by the two models.

Step i is identical in the two models, so the same computation are required. However, the computations required for the remaining steps are drastically more intensive in the case of the superposition model:

- First, *in the stochastic computational model, steps ii and iii are executed only at the instant of measurement of an observable and only for the particular observable measured at this instant*, while, *in the superposition-interpretation, they are executed at every instant of time and for all possible observables corresponding to a Hermitian operato*r. The reason is that, according to the superposition concept, the value obtained when an observable is measured has to be picked from a set of values, which are in superposition before the measurement is performed. Thus, the superposition state of any observable exists before and independently of an eventual future measurement of the one or the other observable. Therefore, in the superposition model the computation of the statistical distribution and the creation of the superposition state have to be done at every instant of time and for all observables corresponding to Hermitian operators!!
- Second, the step iii in *the superposition model involves computing and storing all the possible values for all physical and non-physical "observables"*. As certain observables (e.g. the ones with continues spectrum like the position or the momentum, but also some others with discrete spectrum, like the energy), can take an infinite number of values, their *superposition will require infinite computing power and infinite memory capabilities*!! No computations are required for step iii in the stochastic computational model (void step).

Attributing infinite computing power to the process that engenders the behaviour of a physical system does not seem reasonable. Also, as this behaviour can be engendered by a much simpler process (e.g. the one described by the stochastic computational model), considering that the observables of quantum systems can be in a veritable superposition over a plurality of values will mean that nature employs a very inefficient process, which wastes infinite amount of resources.

Concerning steps iv. and v., required for selecting a particular value during each particular measurement, such a selection does not exist in the superposition interpretation. But, as discussed earlier, this selection is realized during each particular measurement of a quantum system, so it is necessary for the physical completeness of any interpretation, including the superposition one. However, as such a mechanism and the way it acts is not described in the superposition interpretation, we have not a base for comparing the two models. Nevertheless, the operation of this mechanism is similar in the two cases, as in the computation model it has to select a particular value on the basis of a statistical distribution described by a probability density function, while in the superposition

model it has to select a particular value on the basis of a statistical distribution described by the enumeration of all possible values and their corresponding probabilities. Both descriptions of the statistical distributions are deterministic. So, to produce values obeying stochastic distributions, it is required to introduce a stochastic element. In the stochastic computational model this element is the stochastic signal w_a. We can expect that if the superposition interpretation is completed to include a similar mechanism, it will also introduce such a signal. In this case it will need to compute the function fa as in the stochastic computational model. So, we can reasonably expect that steps iv. and v will require similar computing power in the two models.

5 Where is coming from the computing power of quantum computers?

Let us now discuss the process taking place in a quantum computer according to the two models.

According to the superposition model, at each instant of time during the execution of the quantum algorithm, the quantum system determines its wave function. Then, from this function, it determines all the possible values and corresponding probabilities for each observable corresponding to a Hermitian operator and creates a superposition over these values. This includes the observable exploited by the quantum computer for performing computations and enables parallel computation over all possible values of this observable (the so-called quantum parallelism).

According to the stochastic computational model, during the execution of the quantum algorithm, the quantum system only computes its wave function As, it does not create any superposition this model precludes the so-called quantum parallelism (quantum systems could not exhibit higher computing power than a model reproducing their observable behaviour).

Since, from the previous section, both models produce the observable behaviour of quantum systems as described by quantum mechanics, the computational capabilities related to the creation of the superposition states do not have observable consequences (infinite memory capacity and computing power would be wasted), and the related quantum parallelism could not be used for computation purposes. As a consequence, the computing potential of quantum systems should be attributed to their ability to evolve their wave function (and therefore the related statistical distributions) in a very complex manner. To remind this complexity, let us use as example the Thomas Young's double-slit experiment. As this experiment is often referred as an evidence for the veritable existence of the state of superposition (a particle traverses simultaneously two paths), it is also relevant for illustrating that the stochastic computational model can handle it without employing any superposition. In the original experiment, light diffracts through two slits (referred also as hole 1 and hole 2), and creates wave-like interference patterns on a screen. These patterns could be attributed to the "wave" nature of light. But the same experiment can also be carried out by means of a bean firing a single particle at a time (e.g. a single electron). In this case each electron hits the screen at a given position, thus exhibiting corpuscular nature. This is expected since the particles were launched one by one, so they could not interfere with each other. The experimental evidence shows that the position on which each member of a series of electrons will hit the screen is completely unpredictable. However, after a large number of electrons hit the screen, we observe again wave-like interference fringes. The shape of the interference fringes is perfectly predictable. It reveals a perfectly predictable statistical distribution for the positions of the electrons which corresponds to a wave function $\psi = (\psi_1 + \psi_2)$, where ψ_1 corresponds to the case of an electron traversing hole 1 and ψ_2 corresponds to the case of an electron traversing hole 2. According to the superposition interpretation, each particle takes all possible trajectories (in the present case the trajectories traversing the two holes). The superposition of the wave functions ψ_1 and ψ_2 corresponding to these two trajectories gives as result the wave function $\psi = (\psi_1 + \psi_2)$. A particle taking simultaneously several paths is a paradox that we have to admit, unless we can explain this behaviour with a different interpretation. This can be done by means of the stochastic computational model of figure 3. According to this model, the electron (indeed the CMO that produces electron's behaviour) receives information through its interactions with the particles (indeed the corresponding CMOs) composing the wall on which are pierced the two slits. This information is used by the first computing bloc in figure 3 to compute the wave function $\psi = (\psi_1 + \psi_2)$. Then, the second computing bloc computes a function $f_{\bar{r}}$ that corresponds to this wave function. The function $f_{\bar{r}}$ is used to transform a stochastic signal $w_{\bar{r}}$ into a signal $\bar{r}$ whose value determines the position on which the particle hits the screen. From the manner the function $f_{\bar{r}}$ is determined in the annex of a research report (Nicolaidis, 2009), the stochastic computational model of figure 3 will give the same statistical distributions for the position $\bar{r}$ of the electron as for the first superposition. Thus, the two interpretations will give identical interference fringes. However, the superposition interpretation considers that the particle traverses both slits and then, at any time before hitting the screen, it takes all the possible positions allowed by the wave function $\psi = (\psi_1 + \psi_2)$. On the other hand, according to the stochastic computational interpretation, the particle takes a single position when it hits the screen, while before that instant it does not take any position but only computes its wave function.

We observe that *in both cases, the state of the electron is determined in an extremely complex manner:*

- *Determining its wave function by taking all the possible trajectories in the superposition interpretation,*
- *Receiving information from all the particles of the environment in the stochastic computational interpretation and using it to compute its wave function.*

Thus, according to the stochastic computational model, the power of quantum computing is not due to a hypothetical parallel process which manipulates simultaneously a plurality of values, but to the ability of quantum systems to evolve their wave function in a very complex manner. Thus, a quantum algorithm would consist in a judicious technique allowing to constraint quantum processes to produce pertinent results by manipulating what is deterministic in these processes, that is, their wave functions and the resulting statistical distributions of the observables exploited by the quantum computer for performing computations. Accordingly, in a quantum computer using q-bits, the quantum algorithm could transform the state of n q-bits from an initial state (wave function), corresponding to a certain statistical distribution where the solution of the problem appears with a low probability, into a new state (wave function) corresponding to a statistical distribution where the solution of the problem appears with a high probability. These steps, reveal and/or involve transformations of the statistical distributions of a physical observable, engendered by the evolution of the wave function and can be carried out by the stochastic computational model (i.e. do not require the creation of superposition over the 2^n values of n q-bits).

6. Is Quantum Computing Truly Parallel?

In the previous sections we question the existence of a quantum state of superposition arguing that:

- It is a metaphysical state (there are no means for observing it).
- It is not necessary for producing the observable behaviour of quantum systems.
- It implies that quantum systems possess infinite memory and computation resources and that nature employs a very inefficient process wasting infinite amount of resources.

We also argued that *the creation of a state of superposition for each observable of the quantum system is not necessary for supporting quantum computing. The evolution of the wave function is just enough.* We discus this claim in relation with the q-bit-based quantum computing paradigm.

Figure 5 illustrates the idea which attributes the power of quantum computing to the capability of quantum systems to transform in parallel a plurality of values, based on the concept of superposition. The left part of this figure shows a quantum system in which the observable A is in superposition of N different states. According to the superposition interpretation the evolution of the quantum system processes in parallel all these states, enabling quantum computers to perform parallel computations over them. Accordingly, the quantum computer will have the capabilities of a classical parallel computer, which processes in parallel N values stored in N registers (as illustrated in the right part of figure 5). However, quantum algorithms do not exhibit such capabilities. Indeed, performing parallel computations over a set of N values as do the parallel computer in the right part of figure 5, enables treating a problem by means of a black-box approach. That is, without considering the particular structure of the data of the problem. For instance, to find a particular value within a list of N values, a parallel computer can compare simultaneously this particular value with all the values in the list, and find the solution within one computation step. On the other hand, if we do not exploit the structure of the data in the list (e.g. the fact that the list could be ordered), traditional non-parallel computing needs on the average N/2 computation steps for the same search. We observe that truly parallel computing (in the sense that it simultaneously executes an algorithm over N values) achieves exponential acceleration. Also, thanks to these capabilities a truly parallel computing will solve NP-complete problems in polynomial time by examining in parallel all possible solutions.

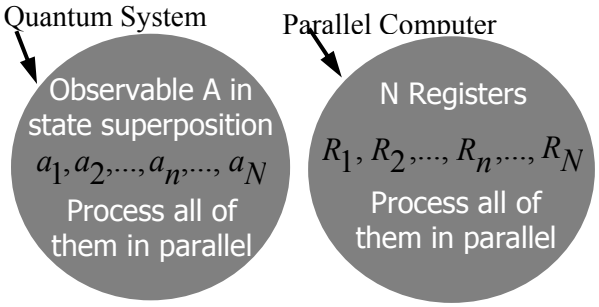


Figure 5. State of superposition and parallel computing

However, there is no known quantum algorithm able to achieve exponential acceleration for the black-box approach, or solve NP-complete problems in polynomial time (Aaronson 2008). Shor's factoring algorithm (Shor 1994) and Grover's search algorithm (Grover 1996) are good illustrations of the fact that known quantum algorithms do not deliver the computing power of a truly parallel computer. Shor's algorithm speeds exponentially the resolution of the factoring problem. But there is no evidence that factoring is a NP-complete problem. In fact, factoring has special properties that don't seem to be shared by NP-complete problems (Aaronson 2008). In particular the quantum part of Shor's algorithm uses the Quantum Fourier Transform algorithm. This algorithm transforms a quantum state which encodes a periodic sequence into a quantum state which encodes the period of that sequence. It works because it exploits specific properties of periodic sequences. Thus, besides the fact that there is no evidence that factoring is NP-complete, Shor's algorithm is not a black-box approach, since it relies on such special properties. Thus, this algorithm does not reveal computing power matching the power of truly parallel computing. Grover's algorithm treats a black-box problem, such as finding an element in an unordered list. It provides a significant acceleration (√N time for Grove's

algorithm instead of N/2 for a classical one). But this is still far from the exponential acceleration that a truly parallel algorithm could achieve. In fact, the $\sqrt{N}$ complexity of Grover's algorithm matches the best performance that can achieve quantum computers for solving black-box problems (Bennett 1997). So, quantum computing is not able reaching the computing power of a truly parallel computer that processes in parallel 2^n states (the number of superposition states of n q-bits).

A more recent argument questions the veracity of quantum parallelism by taking into consideration not only the time required for executing a quantum algorithm but also the complexity for implementing it (Lanzagorta; Uhlmann 2008), (Lanzagorta; Uhlmann 2009). Indeed, the complexity for implementing the unitary operators may overwhelm the gains in execution time as expressed in the following quotes from (Lanzagorta; Uhlmann 2009) "its is entirely possible that, for certain quantum algorithms, the apparent computation savings obtained from the application of the operator to the states in a superposition are always precisely balanced by the increasing number of reversible gates required to implement the operator", "if the same number of logic gates is used by Grover's algorithm and by the classical solution, Grover's algorithm has $O(N^{1/2})$ sequential time complexity and the classical solution has $O(\log N)$ complexity.".

Definitely, quantum computing is not reaching the computing power of a truly parallel computer.

Conclusions

The aim of this paper is to bring in discussion several arguments pleading for abandoning the idea that quantum systems create and manipulate a veritable superposition state for each of their observable. It suggests instead a stochastic computational interpretation of quantum mechanics, in which the behaviour of quantum systems is engendered by deterministic computations performed over stochastic signals, and attributes the computing power of quantum systems to the complex way in which they evolve their wave function and the associated statistical distributions. The paper also pleads for abandoning the interpretation of quantum computing as parallel computing supported by the state of quantum superposition. This interpretation does not correspond to the computing power of quantum processes and is misleading, not only for the general public but often for computing professionals too. Indeed, virtually every presentation of the principles of quantum computing starts by saying that quantum computers manipulate in parallel all the 2^n states of n q-bits. Even works questioning the computation power resulting from this interpretation still rely on it, like for instance (Aaronson, 2008) "There are $2^{1,000}$ possible outcomes, or about 10^{300}, … The technical terminology for this situation is that the 1,000 particles are in a superposition of those 10^{300} states. Put another way, we can store 10^{300} numbers on our 1,000 particles simultaneously. Then, by performing various operations … we can carry out an algorithm that transforms all 10^{300} numbers (each one a potential solution) at the same time."

This is an elegant presentation, easy to understand even for the non specialist, and sells well to the general public as well as to a large part of computing professionals. But it creates the false perception, associating quantum computers with truly parallel computers able to simultaneously manipulate 2^n states. It required more than one decade of thorough investigation of quantum algorithms in order to show that none of them has the capabilities of a truly parallel algorithm.

Is there any reason for selling the idea of 2^n parallel computations and then trying to explain with complex arguments that no quantum computer can deliver such a computation? The rigorous interpretation should be to attribute their computing power to the way quantum systems evolve their wave function and produce the statistical distribution of their observable. This is for instance what Shor's algorithm do as it relies on "certain mathematical properties of composite numbers and their factors that are particularly well suited to producing the kind of constructive and destructive interference that a quantum computer can thrive on." (Aaronson, 2008). This interference is related to the way wave functions evolve, not to the superposition of 2^n states.

References

Aaronson, S., (2008) The Limits of Quantum Computers", *Scientific American, March 2008*

Bennett, C.H., (1997) Strengths and Weaknesses of Quantum Computing, *SIAM J. Comput.* 26, 1510-1523

DeWitt, C.M., Wheeler, J.A., (Eds), (1968) The Everett-Wheeler Interpretation of Quantum Mechanics, Battelle Rencontres: 1967 Lectures in Mathematics and Physics.

DeWitt, S. B.; (1972) The Many-Universes Interpretation of Quantum Mechanics, Proceedings of the International School of Physics "Enrico Fermi" Course IL: Foundations of Quantum Mechanics, Academic Press

DeWitt, S. B., Graham, R. N., (Eds). (1973) The Many-Worlds Interpretation of Quantum Mechanics, Princeton Series in Physics, Princeton University Press, ISBN 0-691-08131-X.

Everett, H., (1957) Relative State Formulation of Quantum Mechanics, Reviews of Modern Physics, 29, 454-462.

Feynman, R., (1982) Simulating physics with computers, International Journal of Theoretical Physics, 21(6/7), 467-488

Grover, L., (1996) A Fast Quantum Mechanical Algorithm for Database Search, Proceedings 28th ACM Symposium on the Theory of Computing, 212-219.

Kaye, P., Laflamme, R., Mosca, M., (2007) An Introduction to Quantum Computing, Oxford University Press, New York.

Lanzagorta M., Uhlmann J. (2008) Is Quantum Parallelism Real?, Proc. Congress Quantum information and computation VI , Orlando, Florida

Lanzagorta M., Uhlmann J. (2009) Quantum Computer Science, Synthesis Lectures on Quantum Computing Series, Morgan & Claypool, Publishers

Reck, M., et al, (1994) Experimental Realization of Any Discrete Unitary Operator, Physical Review Letters, 73(1).

Shor, P., (1994) Algorithms for Quantum Computation:Discrete Logarithms and Factoring, Proceedings 35th Annual Symposium on Foundations of Computer Science,. 124-134

Nicolaidis, M. (2009). On the State of Superposition and the Parallel Nature of Quantum Computing, Tech. Report TIMA lab, ISRN TIMA-RR--09/09--01-FR

Using Ontological Dependence to Distinguish Between Hardware and Software

William Duncan*

Abstract. The distinction between hardware and software is an ongoing topic in philosophy of computer science. We often think of them as distinct entities, but upon examination it becomes unclear exactly what distinguishes the two. Furthermore, James Moor and Peter Suber have cast doubt on the idea that there is a worthwhile distinction. Moor has argued that the distinction should not be given much ontological significance. Suber has argued that hardware is software. I find both these positions implausible, for they ignore more general ontological distinctions that exist between hardware and software. In this paper, I examine the arguments of Moor and Suber, and show that, although their arguments may be valid, they draw implausible conclusions. The ontological perspectives on which their arguments are based are too narrow, and the ontological distinctions used to motivate their arguments are not applicable to reality in general. I then argue that distinctions do emerge between hardware and software when they are considered using ontological distinctions that have wider applicability: A piece of computational hardware is an ontologically independent entity, whereas a software program is an ontologically dependent entity.

1 INTRODUCTION

How do we distinguish hardware from software? Answering this question is more difficult than you might think. Perhaps, it may be suggested, that software is modifiable whereas hardware is not. But this cannot be right. With the right equipment and expertise, hardware is modifiable. Moreover, the further qualification that software is *intended* to be modified is also wrong, for this does not apply to many commercial software products. What about distinguishing the two by the criterion that software is portable whereas hardware is not? Again, we run into difficulties. Many of the components in a computer system can be removed and placed in another computer system.

Given the difficulty in distinguishing hardware and software, some have doubted that there is a distinction. James Moor has argued that the distinction between hardware and software should not be given much ontological significance [4]. Peter Suber has argued that hardware is software [6]. I find both positions implausible, for they ignore more general ontological distinctions that exist between hardware and software. Moor and Suber, in their arguments, do not consider the fundamental categories that describe reality in general. Rather, by focusing solely on certain aspects of hardware and software, they draw implausible conclusions. However, when we consider hardware and software from a general ontological perspective, distinctions do emerge between them. That is, when hardware and software are considered using ontological distinctions that have wide applicability to reality, they are very different things: A piece of computational hardware is an ontologically independent entity, whereas a software program is an ontologically dependent entity.

2 MOOR'S ARGUMENT

2.1 THE DISTINCTION IS PRAGMATIC.

In his article "Three Myths of Computer Science", James Moor (1978) argues that the distinction between hardware and software should not be given much ontological significance. Instead, he holds that we should take a "pragmatic view of the software/hardware distinction" [4]. What Moor means by this will be explained below.

His argument is based on two assertions. The first is that a "computer program is a set of instructions which a computer can follow (or at least there is an acknowledged effective procedure for putting them into a form which the computer can follow) to perform an activity" [4]. The second is that computer programs are to be understood on two different levels: the symbolic level and the physical level [4]. The symbolic level consists of the symbols used to represent some set of computer instructions. The physical level consists of the various media on which computer instructions are physically stored (such as floppy disks, CDs, magnetic tape, and so on). For example, suppose I write a computer program that sorts a list of integers. This program will consist both of the symbols that represent the various actions necessary to sort the list, and the medium in which the symbols are stored (or inscribed). It is important to bear in mind that whether a computer can read my program is an issue for Moor in determining whether it is a computer program. I can scribble my program on a napkin or carve it in a tree. As long as my program is written using symbols that can (in principle) be read and executed by a computer, it is a computer program.

When we distinguish between hardware and software, though, we often overlook either the physical level or symbolic level of computer programs. For instance, in a computer system, software is often thought of as the part of the system that contains the computer's programs [4], whereas hardware is often "characterized as the physical units making up a computer system" [1]. In other words, software is often associated with the symbolic level that represents the instructions of a computer program, and hardware is often associated with the physical components necessary to execute a computer program. This association of software with the symbolic level overlooks the physical level of computer programs, for it ignores two important facts about computer programs. First, many early computers were programmed by throwing switches or setting wires. The computer programs of these early computers were part of the hardware, not separate. Second, modern digital

*Department of Philosophy, University of Buffalo
135 Park Hall, Buffalo, NY 14260-4150, USA
Email: wdduncan@buffalo.edu

computers usually store computer programs internally, and these stored computer programs are part of the physical structure of the hardware [4].

The symbolic level of computer programs, on the other hand, is overlooked when we consider only the physical level of computer programs. At the physical level, software is taken to be the part of a computer system we can change. However, this distinction between hardware and software cannot be consistently maintained. Depending on the context, the part of a computer system that a person can modify may vary. For example, a person with expertise in circuit design may be able to modify a computer system's mother board, a component we normally consider to be hardware.

Given that computer programs are best understood as having both symbolic and physical levels, Moor advocates that we view the distinction between software and hardware as "pragmatic" [4]:

> [S]ince programming can occur on many levels, it is useful to understand the software/hardware dichotomy as a pragmatic distinction. For a given person and computer system the software will be those programs which can be run on the computer system and which contain instructions the person can change, and the hardware will be that part of the computer system which is not software. At one extreme if at the factory a person who replaces circuits in the computer understands the activity as giving instructions, then for him a considerable portion of the computer may be software. For the systems programmer who programs the computer in machine language much of the circuitry will be hardware. For the average user who programs in an applications language, such as Fortran, Basic, or Algol, the machine language programs become hardware. For the person running an applications program an even larger portion of the computer is hardware.

In stating that the software/hardware distinction should be understood as pragmatic, Moor is focusing on the practical activity of computer programming. The ways in which people perform this activity and the levels at which they perform it are not uniform. What is considered a computer instruction and which instructions can be modified is dependent upon the person doing the programming and the type of programming being performed. Hence, we cannot clearly distinguish between hardware and software, for what counts as hardware for one person may be considered software for another [4]. Thus, Moor concludes that we should not read too much into the software/hardware distinction. Rather, it is better to understand the hardware/software distinction as a pragmatic matter, for, "[t]his pragmatic view of the hardware/software distinction makes the distinction both understandable and useful" [4].

2.1 DISCUSSION OF MOOR'S ARGUMENT

Moor's argument consists of an analysis of computer programs at both the physical and symbolic level. My discussion of him will proceed in the same manner. First, let us consider the physical level. Moor's observation that, on the physical level, the ability to modify a computer program is dependent on the computer programmer and type of programming activity is insightful. For in certain contexts, the ability to modify something depends on the person who performs the activity. Consider two homeowners. One is skilled in the art of home construction. The second is not. For the first homeowner, certain aspects of the house may be modifiable, while for the second these aspects may be fixed. For example, the first homeowner may be able to build an addition on to the house. In this particular context, then, the distinction between what is and what is not modifiable depends on the homeowner who is performing the activity.

I grant that Moor is correct about the modifiability of computer programs. However, this does not necessitate that what counts on the physical level as hardware or software is solely dependent upon the person doing the programming. Although activities and their outcomes are intimately linked, the same general kind of activity can be related to different kinds of entities. Consider, again, my example of house construction. As shown above, there are elements to this activity that are dependent on the person performing it. However, this aspect of house construction does not necessitate that the distinction among all houses is dependent only on the skills of those involved in constructing houses. A wigwam is a very different dwelling from a castle, although both result from the activity of house construction (in the broad sense). The distinction between them is not solely based on the practical activities involved in their construction. Each dwelling has properties that are not shared by the other. Similarly, although the activity of computer programming (taken in Moor's broad sense) may be involved in creating hardware and software, this may not necessitate that the distinction between the two is based on the practical aspects of the activities that create them. There may be other properties germane to hardware and software that differentiate them.

Next, let us consider the symbolic level. Recall, Moore asserts that at the symbolic level of computer programs what counts as a computer instruction is also dependent upon the practical activity of computer programming. Thus, depending on the person involved in the activity, a computer program that is considered software by one person may be considered hardware by another. Again, Moor's reasoning does not hold in general. Houses, for example, also have a symbolic level of understanding in the form of building plans. Building plans, like computer programs, consist of a set of instructions, and these instructions, of course, specify different activities for each person involved in the construction of a house. For example, one section of a building plan will instruct the electricians how to install the electrical system, and another section of the building plan will give instructions to the brick masons on how to build the foundation. However, this does not mean that the distinction between electrical systems and foundations is based upon the practical activities of the electrician and brick mason. Similarly, although what counts as an instruction will be different for the factory worker installing circuit boards and the systems programmer, this may not necessitate that the distinction between a circuit board and some block of computer code is based solely on what counts as an instruction for each person.

There are a number of responses to my counterexample. First, it may be argued that I have been too broad in my examples of house construction. However, I reply that the same criticism can be applied to Moor's examples of computer programming. He, after all, includes both the activities of replacing circuits and typing lines of code under the umbrella of computer programming. Thus, Moor's description of computer programming suffers from the same deficiency as my example of house construction. In order to enquire further, a more detailed

description of computer programming is needed. However, Moor does not provide one.

Second, one can respond that my counterexample is wrong. By comparing house construction to computer programming, I am comparing apples to oranges. I agree that most if not all analogies break down at a certain point. However, Moor has provided a very general description of an activity and the relation of the activity to its instances. Likewise, I have provided a very general description of the activity of house construction. If my analogy breaks down, it does so in the details. But, again, Moor has not provided adequate details in order to determine where my analogy breaks down.

There are, of course, other responses, but it is not the purpose of this essay to address them all. Rather, I suggest that the problem with Moor's position is that his ontological perspective is too narrow. That is, Moor's position is motivated by his consideration of three main kinds of things: the symbolic level of computer programs, the physical level of computer programs, and the activities related to each level. Within this limited ontology, then, he is not able to find any significant ontological differences between software and hardware. This, for me, is an implausible result, and, in my counterexamples, I have argued that when Moor's reasoning is applied to other aspects of reality it does not necessitate the same conclusions. What is needed, then, is an investigation into the nature of hardware and software using an ontological framework that can describe these entities as well other aspects of reality.

3 SUBER'S ARGUMENT

3.1 HARDWARE IS SOFTWARE

One conclusion reached by Peter Suber in his (1988) article "What is Software?" is that hardware is software. His argument can be summarized as follows:

(P1) Software is pattern that can be read and executed.
(P2) A pattern can be read and executed if it can in principle satisfy the physical and grammatical conditions of readability and the requirement of executability.
(P3) All patterns can satisfy the physical and grammatical conditions of readability and the requirement of executability.
(P4) All concrete objects display patterns.
(C1) All concrete objects can satisfy the physical and grammatical conditions of readability and the requirement executability.
(C2) Therefore, all concrete objects are software.
(P5) Hardware is a concrete object.
(C3) Therefore, hardware is software.

Suber's first premise (P1) is based on his assertion that software, at its most basic level, must be represented as some type of pattern. In this assertion, it is important to understand that Suber is using pattern "in a broad sense to signify any definite structure, not in a narrow sense that requires some recurrence, regularity, or symmetry" [6]. Thus, whether in the form of magnetic oxide on a disk, pits and lands on a CD, or some other form, it is the pattern which represents the instructions contained in software.

This characterization of software, however, is not adequate for at least three reasons. First, it does not specify whether software must have a material expression; second, it does not distinguish software from noise; third, it does not distinguish software from data (p. 93).

To address these deficiencies, Suber adds (P2): the requirement that software "must in principle be capable of meeting the physical and grammatical conditions of readability and the requirement of executability" (p. 101). In adding this requirement, the first concern (whether software must have a material form) is addressed since the physical condition of readability requires that pattern must be in a form that a machine can read.

The second concern (how to distinguish software from noise) is addressed by the grammatical condition of readability and the requirement of executability. Together, these conditions specify that there must be "certain syntactic structures within the pattern" [6] that can act as instructions to the machine. This would seem to exclude patterns that have no discernable meaning. However, Suber is quick to point out that although a pattern may not, at present, have a meaningful interpretation, "[a] very clever person working backwards from an arbitrary series of bits could create language conventions that would make the string a meaningful program that did something interesting" [6]. A pattern is "noisy" relative to some set of language conventions that the pattern may or may not fit, and since it is always possible to give a pattern a meaningful interpretation, no pattern is noise from all perspectives [6]. All noise is, thus, capable of becoming software. Suber dubs this conclusion the "Noiseless Principle" [6].

Lastly, the question still remains of how to distinguish software from data. Software gives instructions to a machine, but, in some circumstances, the software itself can be treated as data. Depending on the circumstance, the same pattern may be software, data, or both. For example, a compiler treats another computer program as data, and the programming language LISP allows for a computer program to treat itself, or a copy of itself, as data [6]. The determination, then, of whether a pattern is software or data is not due some intrinsic quality of the pattern. Rather, it is determined by a pattern's role. In some cases, the role of a pattern may be as input to software. In other cases, the role of a pattern may be as software (i.e., as instructions to a machine). As long as the physical and grammatical requirements of readability are met, a pattern can perform either role. If a pattern can be read by a machine, the pattern may be passively treated as data or the pattern may actively read as one or more instructions, thus meeting the requirement of executability.

The third premise (P3) follows from two principles Suber calls the Sensible and Digital Principles. The first states that "any pattern can be physically embodied" [6]. As Suber puts it [6]:

> [P]atterns that can be imagined can be drawn. Patterns that are conceivable but not imaginable (like Descartes' chiliagon or 1000-sided polygon) can be described in a notation that provides a complete recipe for conception; and the notation can be drawn. If something cannot be conceived, it probably does not deserve the name of pattern. And what is drawn is thereby given a physical

representation that can be read or decoded by suitably designed machine.

Thus, since all patterns can be physically embodied, all patterns can satisfy the physical condition of readability.

The Digital Principle states that any pattern can be represented as a digital pattern [6]. For example, any analog pattern, such as a painting, can be digitized. Once digitized, the grammatical condition of readability and the requirement of executability are met. For within a digital pattern, the necessary distinctions are present for constructing syntactic structures, and these syntactic structures make it possible for the digital pattern to be read and executed. From the Sensible and Digital Principles, then, it follows that all patterns can satisfy the physical and grammatical conditions of readability and the requirement of executability.

Since Suber uses the term "pattern" in a broad sense, the fourth premise (P4) is rather uncontroversial. However, once we grant (P4) a number of things follow. First, from (P3) and (P4) we infer (C1): All concrete objects can satisfy the physical and grammatical conditions of readability and the requirement of executability. Second, from (C1) and (P2) we infer that concrete objects can be read and executed, which in conjunction with (P1) gets us the conclusion that concrete objects are software (C2). The last premise (P5) and final conclusion (C3) are trivial. As Suber states, "Hardware, in short, is also software, but only because everything is" [6].

3.2 DISCUSSION OF SUBER'S ARGUMENT

One reasonable inquiry to make at this point is whether Suber's argument is valid. However, I am not going to challenge the validity of his argument. Much of Suber's article is devoted to spelling out his reasoning, and in order to question the validity of Suber's argument much effort would have to be devoted to spelling out Suber's reasoning in more detail than I already have. This would be a time consuming and, I think, ultimately unnecessary project. Thus, I will simply grant that Suber's argument is valid.

We are now left with the question of the plausibility of the argument's premises. There are a number of questions one could raise concerning each premise individually. However, I will forego doing this. Rather, I will consider whether Suber's argument leads to an implausible conclusion. If it does, then, assuming the argument is valid, there must be a problem with at least one of the premises. For this purpose, consider the following additional premise and conclusion:

(P6) A peanut butter sandwich is a concrete object.
(C4) Therefore, a peanut butter sandwich is software.

Given that Suber's argument asserts that everything is software, (C4) is a valid inference. However, (C4) strikes me as implausible. Aside from being somewhat humorous, it ignores an important distinction between peanut butter sandwiches and software. Namely, peanut butter sandwiches are things we eat. I realize that that my distinguishing sandwiches from software on the basis of being edible does not necessarily show Suber's argument to be invalid. After all, we can also eat compact disks and circuit boards.[1] Rather, it is only meant to demonstrate a conclusion of Suber's argument that I (and I think many others) find implausible.

So, how might one respond to my criticism? First, of course, one could charge that I have not adequately represented Suber's argument. To this I respond that even if I have misrepresented some of the finer points of the argument, Suber's conclusion that "everything determinate is software" [6] is clear. Thus, (C4) is still a valid inference, for from the assertion that everything is software, it follows that a peanut butter sandwich is software.

Second, one could hold that although (C4) sounds implausible, the argument is still, in fact, sound. After all, I imagine that food in some sense can be considered as software for the body. However, if one really wishes to hold to the view that everything is software, the question must be raised as to how to distinguish the various types of software in the world. Suber's argument not only asserts that peanut butter sandwiches are software, but so are automobiles, shopping malls, and roller coasters. I find this implausible, but not because Suber's argument is invalid. Rather, I find it implausible because Suber's assertion that everything is software does not account for other distinctions we make. By focusing solely on the nature of pattern and its role in defining software, Suber, like Moor, presents an ontological perspective that is too narrow to make other distinctions. This lack of ontological perspective is not necessarily damning for Suber's argument. He may still be correct. However, it does present us with an option. We can hold to Suber's conclusion that everything is software, or we can consider software from a perspective that does allow us to make more plausible distinctions. I will now turn to this latter option.

4 WHAT IS ONTOLOGY?

Part of the shortcoming with Moor's and Suber's treatment of hardware and software is that their respective positions do not hold up well against more general considerations of reality. What is needed, then, is a more in-depth investigation that considers hardware and software as they exist in relation to categories and relations that account for all aspects of reality rather than a specific domain. That is, what is needed is an ontological investigation into the nature of hardware and software.

But what is ontology? Historically, ontology is a branch of philosophy concerned with the nature of what exists. This definition of ontology, I realize, is vague. Thus, to be more precise, I will adopt Barry Smith's definition of ontology as the "science of what is, of the kinds and structures of objects, properties, events, processes and relations in every area of reality" [5]. This emphasis on the kinds of entities that exist in every area of reality entails that ontology is primarily concerned with describing reality in its most general sense, and not necessarily concerned with properties that define a particular entity.

Some readers who are familiar with ontology from the perspective of artificial intelligence or geoinformatics may associate ontology with the creation of some taxonomy (or partonomy). Taxonomy creation, however, is not the main goal of ontology. Ontology, as stated, is concerned with describing

[1] I thank Neil Williams for pointing this out to me.

reality in general. Although taxonomies are useful for conveying information, the underlying importance of ontology is in understanding the reasoning that went into the creation of a given taxonomy.

Before continuing, it is important to note that I am using the term "entity" in a somewhat specialized way. Often, we use the term "entity" to refer to some kind of concrete object, and not to an object's properties. This is not how I am using the term "entity". Rather, I am using the term "entity" in a more general way to refer to anything that has spatiotemporal existence. Thus, my use of the term "entity" will also refer to things we may not normally call entities, such as relations, processes, functions, etc.

5. ONTOLOGICAL DEPENDENCE

In ontology, a key area of inquiry is the determination of the dependency relationships that hold for all entities in general. This is not a trivial task, for there are many different senses of what it means for an entity to depend on another entity. One sense of dependence may refer to the relationship between an entity and its environment. For example, a mammal depends on oxygen in order to exist. Another sense of dependence may refer to the composition of an entity's physical structure. For example, a human being depends on a properly functioning heart in order to exist. Although these senses of dependence are important within the domains of mammals and human beings, they are too specific for the level of generality needed in ontology, for there are a number of entities that depend neither on oxygen nor hearts for their existence. Rather, the kind of dependence I am concerned with is *ontological dependence*, or, in other words, what dependency conditions hold for entities in general.

In order to gain a better understanding of ontological dependence, let us consider two general ways we can distinguish entities. First, we can distinguish between the general features an entity shares with other entities of the same kind, and the particular entity under consideration.[2] To illustrate this, consider the color green. There is a distinction between the colored green in general and the greenness of a particular entity. For instance, we can talk about the greenness (in general) of trees, shamrocks, and frogs; and we can also talk about the greenness of a particular tree, the greenness of a particular shamrock, and the greenness of a particular frog.

This task of describing the distinction between the general features of an entity and the particular entity itself can be quite cumbersome. To be concise, I will use the term "universal" for entities of the former kind, and the term "instance" for entities of the latter kind.[3] So, for example, in relation to the color green, the general feature of this entity is a universal (the universal green), but the greenness of a particular shamrock is one instance of this universal. Furthermore, since I am using the term "universal" to refer to the general features shared by some group of particular entities, universals are dependent on the existence of one or more of its instances. In other words, a universal exists only if there exists some instance of it. Thus, we have arrived at our first example of ontological dependence: the relationship between a universal and its instances.

Second, we can distinguish between an object and that object's properties. Consider, again, an instance of the color green. We do not encounter free-floating instances of green. Rather, when we encounter an instance of the color green, it is the color of some object. From this, it follows that any instance of green cannot exist if the object that is colored green does not exist.

This line of reasoning allows us, in general, then to distinguish between two types of entities: those that do not depend on another entity for their existence, and those that do depend on another entity for their existence. The former I will refer to as "independent entities", and the latter I will refer to as "dependent entities". When a dependent entity *x* stands in a relationship to an independent entity *y* such that *x* cannot exist if *y* does not exist, I will express this relationship in terms of "*x* inhering in *y*", "*x* is borne by *y*", or "*y* is the bearer of *x*". So, for example in the case of a green shamrock, the relationship between the shamrock and the instance of the shamrock's color is expressed as "the instance of the color green inheres in the shamrock", "the instance of the color green is borne by the shamrock", or "the shamrock is the bearer of the instance of the color green". Thus, we have arrived at our second example of ontological dependence: the relationship between independent and dependent entities.

6. GENERICALLY AND SPECIFICALLY DEPENDENT ENTITIES

Among dependent entities, we can further distinguish between specifically dependent entities and generically dependent entities.[4] First, let us consider specifically dependent entities. These are entities that depend upon a specific particular bearer in order to exist. We have already seen an example of this in the above example of the color green. When an instance of green exists, it is specifically borne by (or inheres in) the particular independent entity that is colored green. If that particular independent entity ceases to exist, that particular instance of green also ceases to exist.

Generically dependent entities, in contrast, do not depend upon a specific bearer in order to exist, but exist as long they are borne by some entity. For example, consider your favorite book. The story represented by the book's printed words is a generically dependent entity. If you destroy your copy of the book, the story continues to exist as long as there is some other book (or other media) in which the story appears. [5] It is important to keep in mind that there is a distinction between the story and the particular qualities of the book such as the color of the book's pages. The latter are specifically dependent entities

[2] In speaking of the general feature of a kind, one may wonder whether I am referring to our mental representations of these entities, or to some set of mind-independent features that entities of a certain kind possess. In other words, one may wonder where I stand on the question of metaphysical realism. For purposes of this essay, I believe that the view I am advocating can accommodate both ontological realists and many types of nominalism. Thus, I will remain silent in regards to this issue.

[3] My use of the terms "universal" and "instance" follows that of Basic Formal Ontology (BFO). See http://www.ifomis.org/bfo for details.

[4] My use of the terms "specifically dependent" and "generically dependent" follows that of Basic Formal Ontology (BFO). See http://www.ifomis.org/bfo for details.

[5] I am ignoring cases in which you own the only manuscript of some book.

bound to the existence of your particular book, whereas the existence of the story, however, is not.

Given this description of specifically and generically dependent entities, there are two important clarifications needed. First, it is easy to confuse specifically dependent entities as being generically dependent upon their bearers. To illustrate, consider my example of the color green. I distinguished specifically and generically dependent entities by appealing to the fact that if you destroy a green colored object, that instance of green ceases to exist. However, one can object, the color green does not cease to exist. There are, after all, still many other green colored things in existence. Thus, an instance of the color green is really a generically dependent entity.

The problem with this confusion is that it fails to consider the distinction (discussed above) between universals and instances. Instead, it is only concerned with the universal color green. When the color green is considered only as a universal, this confusion does make a correct point: The universal color green does not cease to exist when some particular instance of it ceases to exist. However, this is not the case when we consider the nature of an instance of the color green. Under this consideration, we find that instances of the color green are specifically dependent upon their bearers, and it is this level (the level of instances) where the distinguishing criterion between specifically dependent and generically dependent entities is found.

A second, somewhat related, confusion is that generically dependent entities are specifically dependent upon their bearers. Consider the following argument by analogy: suppose I am holding a green shamrock in each hand. Each shamrock, of course, is the bearer of an instance of the color green. Next, suppose I burn the shamrock in my left hand. The instance of the color green borne by that shamrock perishes along with the shamrock, but an instance of the color green still inheres in the shamrock I am holding in my right hand. Now, let us repeat these actions, but instead of holding a shamrock in each hand, let it be a copy of the novel *Brave New World*. When the copy in my left hand is destroyed, the result is the same: the dependent entity in my left hand perishes, but the dependent entity in my right hand still exists. So, there is no difference (in terms of ontological dependence) between an instance of the color green and an instance of the novel *Brave New World*. Furthermore, since instances of the color green are specifically dependent entities, instances of *Brave New World* are also specifically dependent upon their bearers.

The problem with this second confusion is that it does not adequately consider the natures of the instances involved. It correctly asserts that instances of the color green are specifically dependent upon their bearers, but incorrectly asserts that the books are instances of the novel *Brave New World*. The books, themselves, are independent entities, and as independent entities they bear a number of dependent entities, *Brave New World* being one of these dependent entities. *Brave New World*, however, is a particular instance of a novel, and as an instance there are not further instances (or second level instances) of this instance. An instance, as I am using the term, denotes a relationship that obtains between universals and particulars, not between particulars and particulars. There may exist a copy (or clone) of an instance, but an instance is itself a particular entity. Thus, since *Brave New World* is a particular, it is incorrect to assert that some book is an instance of it.

7. DISTINGUISHING HARDWARE AND SOFTWARE

With these ontological categories in mind, let us now turn to the issue of how to distinguish hardware and software. Before continuing, though, it is necessary to address some ambiguities concerning the terms “hardware” and “software”. First, consider the term “hardware”. Does it refer to some specific piece of hardware or some disconnected aggregate of components that may form a computing system? Thus, rather than using the term “hardware”, I will use the terms “piece of computational hardware” (singular) and “computational hardware” (plural) to refer to physical entities contained in a computing system that are recognized as being a unified whole.

Next, consider the term “software”. Again, this term is ambiguous. Sometimes we use the term “software” to refer to specific physical objects, such as a CD, that we can load onto multiple computer systems. Other times we use the term “software” to refer to the computer programs that are encoded on these physical objects.[6] Thus, to be clear, I will use the term “software program” to refer to software in the second sense of the term.

7.1 A PIECE OF COMPUTATIONAL HARDWARE IS AN ONTOLOGICALLY INDEPENDENT ENTITY.

With the ambiguity of the term “hardware” addressed, let us consider what kind of entity a piece of computational hardware is. Earlier, I distinguished between independent and dependent entities. Based on this distinction, then, a piece of computational hardware is an independent entity. Instances of computational hardware (such as a CPU or hard drive) are physical objects that are not dependent upon other entities. Instead, these instances are the bearers of various dependent entities (qualities, functions, etc.).

This assertion that a piece of computational hardware is an independent entity, however, is not adequate for distinguishing an instance of computational hardware from other instances of independent entities. To this end, I submit that a piece of computational hardware bears certain qualities and is involved in the realization of certain functions that are necessary for computation. For example, a hard drive is designed to store information in a magnetic medium, and some set of logic gates on a CPU is designed to add integers. In each example, there is some quality or function (in the sense of purpose) that distinguishes the piece of computational hardware from other independent entities not involved in computation.

7.2 A COMOPUTATIONAL FUNCTION IS A SPECIFICALLY DEPENDENT ENTITY.

In the last section, I proposed that instances of computational hardware are distinguished from other instances of independent entities by being the bearer of certain qualities and functions that are necessary for computation. That is, since an instance of a piece of computational hardware is an independent entity, it is

[6] By encoded I mean only that the computer programs are represented using some system of representation such as the pits and lands on a CD or the holes in a punch card.

the bearer of a number of dependent entities. These dependent entities are the distinguishing characteristics (in Aristotelian terms, the *differentia*) that differentiate a piece of computational hardware from other independent entities (such as a bicycle or a lawn mower). So far we have considered two types of dependent entities: specifically dependent entities and generically dependent entities. Let us now consider how each may serve to differentiate computational hardware.

Regarding generically dependent entities, it seems plausible that these entities may play a role in characterizing an independent entity. For instance, supposed I am presented with two hard drives that are perceptually identical (same size, shape, color, etc.). One way I can differentiate them is by referring to the documents (generically dependent entities) stored on each. For example, if one hard drive contains a copy of *Brave New World* and the other contains a copy of *The Grapes of Wrath*, I can distinguish them by the novels contained on each.

This approach, however, is problematic in that it is too easy to generate cases in which generically dependent entities do not play a role in distinguishing independent entities. For example, suppose I am presented with the book *Brave New World* and a hard drive containing a copy of it. In this case, the generically dependent entity *Brave New World* plays no role in distinguishing the two independent entities. Rather, the book and the hard drive are distinguished by their physical properties, and these physical properties are specifically dependent entities. Even in cases where two objects appear to be identical, we can, at a minimum, distinguish them based on spatial location. For example, in the case of the hard drives (above), we can refer to the one on the right versus the one on the left. Thus, while generically dependent entities may still play a role in distinguishing a piece of hardware from other independent entities, specifically dependent entities seem to offer a more promising way of accomplishing this task.

Specifically dependent entities (recall from above) must inhere in some specific entity in order to exist. The most obvious examples of these entities are an object's physical properties. This, at first glance, would seem to be the best candidate for differentiating computational hardware. However, there is a problem: The physical properties of computational hardware are constantly changing. Computational hardware in the 1950s was very large compared to modern standards (compare the size of ENIAC to size of a typical laptop). The compositional material was also very different (vacuum tubes made of glass as compared to circuits made of silicon). In fifty years, what kinds of qualities will computational hardware possess? It is difficult, if not impossible, to predict. Since the physical properties necessary for computational hardware are difficult to specify, I will focus on the functional aspect of computational hardware. It is important to note that I am not using the term "function" to refer to an abstract mathematical entity. Instead, I am using the term "function" to refer to an entity's purpose. For example, it is the function of a scalpel to make precise incisions.

Under this consideration of functions, there are two important points to make. The first is that functions are specifically dependent entities. This may strike you as odd, but to illustrate this consider a claw hammer. One of its functions is to drive nails. This function is specifically dependent on the hammer, for if the hammer ceases to exist so does its function. Secondly, there is distinction to be made between a function and the process in which it is realized. For example, the hammer's function to drive nails is realized in the process of driving nails, but the hammer's function to drive nails does not cease to exist when the hammer is not engaged in the process of driving nails. Rather, as long the hammer continues to exist, its function to drive nails continues to exist.

In order to refer to functions that are realized through some computational process, I will use the term "computational function". I realize that this term has its draw backs. Although I have stipulated that am using the term "function" in the sense of purpose, readers with a background in computer science or mathematics may associate my use of term "computational function" with abstract entities. I considered using the term "computational procedure", but this seems, at least to me, to bring to mind the process of computation, rather than the particular kind of entity that is realized in the process of computing. I also considered using the term "computational purpose", but this seems to ascribe some kind of intentionality to computation which may make what I'm trying describe more confusing. Thus, for now, I will retain my use of the term "computational function". I remain open to suggestions.

Despite my arguments that functions are specifically dependent entities, it is tempting to classify a computational function as a generically dependent entity. For example, adding integers is a computational function, and instances of this computational function are found on multiple computers. Thus, instances of computational functions are instances of generically dependent entities.

This, however, commits the following mistake: it fails to consider the nature of an instance of a computational function. An instance of a computational function is realized in a process in which the bearer of the computation function participates, and since the computational function cannot exist without its bearer, it is a specifically dependent entity. To illustrate this distinction, consider this fragment of Java text the execution of which adds two integers (i.e., int i, int j):

```
public int addTwoIntegers(int i, int j) {
    return i + j;
}
```

The text is an instance of a generically dependent continuant. It exists as long as it borne by some independent entity. However, the computational function that is realized when the code is executed on a piece of computational hardware is not the same as the Java text, for it is possible to write code that adds integers in many different programming languages. Thus, the entity described in this fragment of Java code (i.e., the computational function of adding two integers) is *specifically* dependent, whereas the fragment of Java text is generically dependent.

With these points regarding computation functions in mind, there are a number of advantages to using computational functions to distinguish instances of computational hardware. First, a considerable amount of research has been done regarding the kinds of problems we can compute. For instance, we know that we can compute whether a string is a palindrome, and we know that we cannot compute, in general, whether a program will halt (i.e. The Halting Problem). Thus, we have some more definite guidelines for determining what kinds of functions can be realized through the process of computation.

Second, the functional aspect recognizes that a piece of computational hardware is often composed of a number of

components. For example, a CPU has many different components, such as the ALU (arithmetic logic unit) and the cache. The computations carried out by the CPU are the result of these components acting in concert. This is analogous to the example of the claw hammer. A claw hammer is composed of different components, such as the handle and the head, and its function of driving nails is realized when these components work together.

Finally, the functional aspect is not overly restrictive as to the types of independent entities in which a computational function may inhere. Does this mean that any independent entity can bear a computational function? The answer is 'perhaps', for when we consider the types of things which can be used to compute, there seems to be a large (maybe indefinite) range of possibilities. This may seem to be a drawback, but if we reflect on the functions associated with many common objects, we find that a wide variety of objects can be used to accomplish the same task. For example, a brick can be used to drive nails, a knife can be used as a screwdriver, and a laptop can be used as a doorstop. Thus, when we consider the functions of objects in general, it is not surprising that there may be a number of different kinds of independent entities in which computational functions can inhere.

This wide variety of potential bearers for a computational function, however, does not imply that we cannot distinguish computational hardware from other independent entities. We recognize that an entity has specific function because its design (or structure) is such that it is better suited for certain tasks. For example, shovels are better suited for digging holes than pitchforks because they have been designed specifically for the function of digging. Similarly, this holds for a piece of computational hardware. Hard drives are designed for the task of storing information in a magnetic medium. The complex arrangement of silicon circuits in a modern CPU is specifically designed for the execution of electronically encoded instructions. This recognition that a piece of computational hardware is designed to perform certain tasks as opposed to others helps to constrain the kinds of things that count as a piece of computational hardware. A piece of computational hardware has been designed to realize some computational function, whereas, while other independent entities may potentially realize some computational function, they have not been specifically designed to do so.

7.3 A SOFTWARE PROGRAM IS A GENERICALLY DEPENDENT ENTITY.

Finally, let us consider a software program. As previously discussed, I am using the term "software program" to refer to the computer programs that are encoded on various physical objects. More specifically, by software program, I mean some set of instructions written in some programming language. This definition of software program, I admit, is almost identical to Moor's definition of a computer program [4], and fits within Suber's conception of software as pattern. However, there are also important differences.

The first is that a software program (like a novel) is a generically dependent entity. The encoded computer program does not depend on a specific independent entity (such as a CD or floppy disk) in order to exist; only that it inhere in some entity. For example, if you destroy my CD of Microsoft Word, Microsoft Word does not cease to exist. Neither Moor nor Suber addresses this aspect of software programs.

The second difference lies in the emphasis that the instructions are in some programming language.[7] This entails that the instructions can in principle be executed. In order to execute these instructions, it is necessary for some piece of computational hardware to participate in some appropriate process. Thus, my stipulation that a software program is a set of instructions in some programming language entails not only that the instructions are in principle executable, but also that the instructions describe how to realize some computational function.

This description of a software program will probably raise some eyebrows (or cause some head scratching). As justification, consider again my "addTwoIntegers" above. The instructions represent a way (or method) for taking two integers as input and returning their sum as output. The particular programming language used to represent this (in this case Java) is accidental. However, what the program code describes how to do is realized in a process in which a piece of computational hardware participates.

This may be more clearly in a non-computational example. For this, let us consider a sheet of music written for guitar. The language used to represent what notes to play is accidental. However, when the instructions are followed and a string is plucked, a function of the guitar is realized. Analogously, when a piece of computational hardware executes a programming language instruction, the computational function described by the program code is realized.

8. CONCLUSION AND FURTHER QUESTIONS

The goal of this paper was to investigate the nature of software and hardware and to determine whether there is any clear criterion for distinguishing the two. In pursuit of this, I have offered a strong reason for holding that such a distinction does exist: Software programs are ontologically dependent upon some other entity, whereas a piece of computational hardware is ontologically independent. This fact alone draws a sharp distinction between software programs and computational hardware. Thus, in contrast to the positions of Moor and Suber, a piece of computational hardware and a software program are very different entities.

Furthermore, there are still a number of issues which need to be resolved:

1. My description of computational hardware, computational functions, and software programs is very general. Is a more detailed description needed? If so, what is the best way to go about providing this description?
2. An important aspect of computational functions is that their bearers participate in corresponding computational processes. However, I have not provided much detail as to what a computational process is. Can a more detailed description be provided?

[7] What counts as a programming language is beyond the scope of this paper. However, if you recall Suber's discussion on what kinds of patterns can be used as an instruction, you will be reminded that any concrete pattern can serve this purpose.

3. What role does specification have in defining computational hardware and software programs?
4. Programming languages play an important role in defining software programs. Will a more thorough investigation into the nature of programming languages yield better insight into the nature of software programs?

9. ACKNOWLEDGEMENTS

I would like to thank the following individuals for their helpful comments: Amnon Eden, Jeff Gower, Pierre Grenon, Mark Holliday, Werner Kuhn, William Rapaport, Barry Smith, and Raymond Turner. I would also like to thank the National Center for Geographic Information and Analysis Integrative Graduate Education and Research Training Program (NCGIA IGERT) at the University of Buffalo and the International Research Training Group (IRTG) at the University of Muenster for providing funding and accommodations while I was working on this paper.

REFERENCES

[1] Chandor, A., Graham, J., and Williamson, R. (1970). *A Dictionary of Computers.* Baltimore: Penguin Books.

[2] Moor, J. (1978). Three Myths of Computer Science. *British Journal for the Philosophy of Science*, 29(3), 213-222.

[3] Smith, B. (2004). Ontology. In L. Floridi (Ed.), *The Blackwell Guide to the Philosophy of Computing and Information* (pp. 155-166). Victoria, Australia: Blackwell Publishing, Ltd.

[4] Suber, P. (1988). What is Software? *The Journal of Speculative Philosophy*, 2(2), 89-119.

Why is it necessary to build a physical model of hypercomputation?

Florent Franchette [1]

Abstract. A model of hypercomputation can compute at least one function not computable by Turing Machine and its power comes from the absence of particular restrictions on the computation. Nowadays, some researchers claim that it is possible to build a physical model of hypercomputation called "accelerating Turing Machine". But for what purposes these researchers would try to build a physical model of hypercomputation when they already have mathematical models more powerful than the Turing Machine? In my opininon, the computational gain provided to the accelerating Turing Machine is not free. This model also lost the possibility for a human to access to the computation result. To define this feature, I'll propose a new constraint called the "access constraint" stating that a human can access to the computation result regardless of computation ressources. I'll show that the Turing Machine meets this constraint unlike the accelerating Turing Machine and I'll defend that build a physical model of the latter is the solution to meet the access constraint.

1 Introduction

The aim of the computability theory is to define mathematical functions computable by algorithms. An algorithm is a computation method which meets the following constraints [4] :

1. The algorithm must have a finite number of symbols and instructions.
2. It must include a finite number of steps.
3. It must be executed within a finite time.
4. A human being can follow the algorithm step by step, from initial data to result regardless of the resources of time and memory space.
5. The algorithm can be executed by a human without the aid of any physical machine such as a computer.
6. A human should be able to perform each step of the algorithm in an effective way, that is to say without ingenuity or intelligence.

Since the definition of an algorithm is an informal one, the computability theory needs for a mathematical definition of this notion. In order to formalize a predicate which means "can be computed by an algorithm", Alan Turing proposed in 1936 the formal predicate of "computed by Turing Machine" or "Turing-computable" [16]. According to Turing, the Turing Machine is a mathematical computation model with a power equivalent to an algorithm. Turing however, showed that the computing power of his Machine, that is to say the number of functions it could compute, depended on the type of constraints applied to the model [17]. Nowadays, numerous published papers propose models exceeding the computational power of the Turing Machine [3].

[1] University of Paris 1, France, email: florent.franchette@gmail.com

These models are called "hypercomputation models" or "hyperMachines" and their capacity to compute more functions than the Turing Machine comes from the absence of constraints on the computation. Recently, Jack Copeland has proposed a hyperMachine named "accelerating Turing Machine" which is based on the absence of the constraint that the computation must include a finite number of steps [2]. More importantly, some researchers defend the idea that it is physically possible to build an accelerating Turing Machine [14]. Notwhistanding, the physical construction of a computing model exceeds the initial framework of the mathematical computability theory. Therefore, for what purposes these researchers would try to build physical hyperMachines when they already have mathematical models more powerful than the Turing Machine?

In my opininon, although the absence of a constraint such as the finite number of computational steps allows the accelerating Turing Machine to compute more functions than the Turing Machine, the computational gain is not free. The hypercomputation model also lost a key feature : the possibility for a human to access to the computation result. To define this feature, I will propose a new constraint called the "access constraint" stating that a human can access to the computation result regardless of the resources of time and memory space. In the one hand, I will show that the Turing Machine meets this constraint unlike the accelerating Turing Machine and on the other hand, I will defend the idea that build a physical model of this hyperMachine is the solution to meet the access constraint.

2 Turing Machine as a formal definition of an algorithm

Since the notion of algorithm is an informal one, it is necessary to formalize it in order to show what kind of functions can be computed by algorithms. To this purpose, Alan Turing proposed in 1930 a mathematical model today named "Turing Machine" representing the way a mathematician would follow step-by-step an algorithm [16]. A Turing Machine (TM) can be viewed as a procedure that includes a single data structure: a string of symbols written in the squares of a potentially infinite tape. The operations available allow the program to move a read head to the left or right of the string, to write a symbol instead of the symbol read or do nothing. These operations are very simple and primitive but according to Turing they are sufficient to compute all the functions computable by algorithm.

Formally, a TM is a quadruple $M = (K, \Sigma, \delta, s)$ where:

- K is a finite set of states representing instructions of the Machine,
- $s \in K$ is the initial state,
- Σ is a finite set of symbols, the alphabet of M.
- δ is the transition function.

Function δ is the "program" of the Machine, that is to say δ dictates the behavior of M. It specifies, for each combination of current state $q \in K$ and current symbol $\sigma \in \Sigma$ a triple $\delta(q, \sigma) = (p, \rho, D)$ where p is the next state, ρ is the symbol overwritten on σ, and $D \in \{\leftarrow, \rightarrow, -\}$ is the direction in which the cursor will move. Initially, the program starts in state s. The string is initialized by a finitely long string. We say that x is the input of the TM. From the initial configuration the TM takes a step according to δ, changing its state, printing a symbol, and moving the cursor; then it takes another step *etc.* If a TM halts on input x, we define $M(x)$ as the output of the TM on x.

Definition 1 *Let f a function and M a TM. We say that M computes f if, for all string x, $M(x) = f(x)$. If M exists, we say that f is a Turing-computable function.*

It is possible to prove that the mathematical definition of the TM meets the intuitive definition of an algorithm and therefore that Turing-computable functions are computable by algorithms. However, we are unable to formally prove the converse called "The Church-Turing thesis": functions computable by an algorithm are Turing-computable. This thesis has never been invalidated and all the computational models devised to formalize the notion of algorithm are able to compute exactly the same functions[2]. The key point is that if we adopt the Church-Turing thesis, we can prove that some functions are not computable by algorithm because they are not Turing-computable. For exemple, consider the following function : let x a Diophantine equation[3] and let f a function such as $f(x) = 1$ if x has at least one solution and $f(x) = 0$ otherwise. Youri Matiiassevitch has proved that this function called the "Diophantine function" is not Turing-computable [11], hence according to the Church-Turing thesis there is no algorithm which can compute it. Intuitively, the Diophantine function is not Turing-computable because we are forced to use the "brute-force search" method if we want to know whether such an equation has a solution. This method consists to successively test all possible solutions of the equation until to find one. The problem arises if the equation has no solution because the TM will test the infinite set of integers numbers and since the computation of a TM is finite, it is unable to run the infinity of integers. It seems therefore that a TM can't compute the Diophantine function without violate the constraint of finiteness of the computation. Is it nevertheless possible to violate this constraint in order to compute such a function? More generally, is the removal of constraints on the TM computation would expand the set of its computable functions?

For this purpose, a reasonable choice would be to define an alternative concept of algorithm where the number of computational steps is infinite even though the computation is finite. I'm going to explain this choice. First, the constraint of a finite number of steps is not explicit in the Turing's work and it is not mentioned in the description of the TM. Turing lists the elementary actions executed by the TM (read a symbol of the tape, print a symbol, move the cursor to the right *etc.*) but tells us nothing about the number of primitive action. Secondly, the alternative definition of algorithm allows the computation model to perform an infinite number of steps in a finite time. It is this property which is used by Jack Copeland to develop models more powerful than the TM.

[2] We can cite the λ-definable functions model [1] or the recursive functions model [7].

[3] A Diophantine equation is an equation whose cœfficients and solutions are integers numbers. A exemple of Diophantine equation is $x^3+y^3+z^3 = 0$.

3 Accelerating machine as model of hypercomputation

The term "hypercomputation" was introduced by Jack Copeland and refers to the computation of functions which are not Turing-computable. A model of hypercomputation or a hyperMachine can compute at least one non Turing-computable function, and their power come from the absence of particular restrictions on the computation. Hypermachines tend to be of two general types: one uses non Turing-computable numbers[4] [17], and the other uses infinite computation in finite time. In 2002, Jack Copeland introduced a hyperMachine of the second type called "accelarating Turing Machine" (ATM) [2]. The ATM is similarly defined to the TM except concerning the number of computational steps it can perform. While the TM always stops after a finite number of steps, the ATM has the option to stop after an infinite number of steps on the following principle: the time of a step is two times less than the previous step. More precisely, the ATM is a quadruple $M = (K, \Sigma, \delta, s)$ which differs only from the TM by increasing its speed after each new computation step. Given a computation takes one time unit to execute its first iteration, the total time of the computation can be expressed by this geometric series [6]

$$\sum_{i=0}^{n} \frac{1}{2^i}$$

where i is the current iteration and n is the number of iterations of the computation. Therefore, when i tends to infinity, the total time approach 2

$$\sum_{i=0}^{\infty} \frac{1}{2^i} = 2$$

If the computation can be performed using a finite number of steps, the ATM computes exactly the same functions as the TM. However if the computation requires an infinite number of steps to achieve the result, the ATM can overcome the power of the TM. For example, here's how the ATM can compute the Diophantine function.

Given a specific Diophantine equation $x^n + y^n = z^n$ printed on the tape with x, y, z fixed, the ATM will cover all integers n in a finite time in order to find a integer satisfying this equation. In the case where the ATM finds an integer n satisfying this equation, the read head of the ATM returns to the beginning of the computation and prints 1, otherwise prints 0. Thus, given any Diophantine equation an ATM is able to know in a finite time whether this equation has a solution. This result shows that hypercomputation can be seen as a generalization of computation bringing a computational gain to a mathematical model in terms of computable functions. Nevertheless, the term "hypercomputation" also designates computational models physically built in theories and exceeding the TM's power [8]. This second definition of hypercomputation allows us to distinguish two types of model :

1. A "conceptual type" where the hyperMachine is not devised in order to be physically built. In this case, we do not need to ask ourselves whether the increased speed of the ATM could be physically achieved.
2. A "real type" where the hyperMachine is devised in order to be physically built. The ATM have to be compatible with the laws set by the physical theory in which it will be devised. These laws are additional constraints imposed on the model.

[4] A Turing computable number is one for which there is a TM which, given a number n on its initial tape, terminates with the nth digit of that number.

Nowadays, many real type models of a ATM have been proposed both by John Norton in quantum theory [12] and by Shagrir and Pitowski in the theory of general relativity [14]. It is however pertinent to ask why they consider hypercomputation as an area with two sides : one mathematical and the other physical. This question is justified by the fact that the physical construction of a computational model, whether equivalent to the TM or not, goes beyond the original framework of the computability theory. On the one hand, the Church-Turing thesis states nothing about the computing power of a TM physically built, it only states an equivalence between the intuitive concept of algorithm and the mathematical concept of Turing Machine. On the other hand, hypercomputation has the primary purpose of determining whether there are computational models which could overcome the TM if we modify the constraints imposed on the computation. Although this purpose also goes beyond the framework of computability theory, this excess is within the mathematical side unlike the issue of whether hyperMachines can be physically built. So why leave the mathematical framework of hypercomputation in order to turn to the physical sciences? I'm going to suggest in the next section a reason why it is necessary to build a real type hyperMachine.

4 Why is it necessary to build a hyperMachine?

In this section, I'll defend the following thesis : although the absence of a constraint such as the finite number of computational steps allows the ATM to compute more functions than the TM, the computational gain is not free. The ATM also lost a key feature: the possibility for a human being to access to the computation result. While the ATM computes the Diophantine function, its computing results are inaccessible from humans. Nevertheless, if we construct a physical model of the ATM which is able to compute the Diophantine function, it could be possible to get back this feature.

First, we must define what it means "to have access" to the computation result. Hence, I propose a distinction between "to access to the result" and "to compute the result".

Definition 2 • *We have access to the computation result when the result is available to us in principle. This result doesn't need to have a meaning, it can only be a string of symbols.*
• *We compute a result when we can follow in principle each computational step from input to output.*

From these definitions, we can set out two constraints: one asserting that we can compute results printed by a model and the other asserting that we can have access to these results. I'll call these two constraints the "computing constraint" and the "access constraint".

Definition 3 *Let a function f which is computable by a model.*

- *This model meets the computing constraint (CC) if for all input x, a human can follow step by step the computation, from input to output $f(x)$ regardless of computation resources.*
- *This model meets the access constraint (AC) if for all input x, a human can have access to the output $f(x)$ regardless of computation ressources.*

It's straightforward to show that these two constraints are set out in the informal definition of an algorithm. Therefore, the TM which is a mathematical formalization of algorithm meets these two constraints. But what about the ATM? Does it also meet these constraints? In the first part of my reasoning, I'll try to show that if we remove the constraint of finite number of steps, the ATM doesn't meet the CC and the AC.

My main point is to show that it is actually unlikely that a human can compute an infinite number of steps in a finite time. Indeed, if the ATM meets the CC it means that a human can compute the results of the Diophantine function in principle, specially when an equation has no solution. Thus in the same manner as the ATM, a human must use the brute force search method and execute an infinite number of steps in a finite time to test all possible solutions. Notwhistanding, there is a convincing argument against the fact that we are able to make such an infinite computation. This argument consist to say that the brain where computations are made, is a finite entity both in space (there is a finite number of neurons and synaptic connections) and time (the life of a human being is limited). This argument seems pertinent in order to show that we are not able to follow step by step an infinite computation. But it is not sufficient to prove that we can't have access to the result of an infinite computation because it could be possible that we have access to Diophantine function results without to follow each computational step. This possibility amounts to defend the idea that the human mind is more powerful than the TM. Some authors defend this claim in a philosophical way[5] but it is also possible to defend it in a mathematical way by devising a model of the human mind and by studying its computational power [15].

Such a model named "Artificial Neural Networks" (ARNNs) was inspired from their biological counterparts and by a simplistic vision of how messages are transferred between neurons. In an informal way, ARNNs are represented by a graph divided in layers whose vertices are artificial neurons. An artificial neuron is a computing unit which receives an input data directly from the environment if it is inside the first layer of neurons or from neurons otherwise. When the information comes from a neuron, we associate it a real number w called "weight" which is used to compute a weighted average Σ to determine the successive steps of the computation. There are two sources of the computational power of ARRNs :

1. One comes from the message between neurons involving pulses, actions potentials and timing [10].
2. The other comes from the complexity of the neural network weights [18].

As noticed by Zenil and Hernandez-Quiroz [18], weights and pulses are equivalent in terms that they can be replaced one for the other preserving the whole complexity of ARNNs. Therefore, we can build a simplified hierarchy of computational power regarding the two possible sources (weights and pulses). On the one hand, if ARNNs is allowing rational numbers as weights, they can only compute at most Turing-computable functions. On the other hand, if ARNNs is allowing non Turing-computable numbers as weights, they can compute non Turing-computable functions. Although it appears that ARRNs may exceed the power of the TM, this model has been strongly criticized by Martin Davis in his article entitled *The myth of hypercomputation* [5]. According to Davis, ARNNs can go beyong the Turing limit only if they already have non Turing-computable weights and thereby don't prove that the human brain can compute a non Turing-computable function.

From the two arguments outlined above, the one on the finite resources of the brain and the others about the criticisms regarding the brain's model, I shall make the assumption that human beings are not able to compute and to have access to the result of a non Turing-computable function computed by a conceptual type hyperMachine.

[5] We can cite Lucas' argument[9] and Searle's argument[13].

Nevertheless, can ATM meet these constraints? In my view, the CC concerns mainly limits and capabilities of the human brain. The issue of whether the ATM meets the CC, that is to say if human beings can follow step by step an infinite computation in a finite time, must be dealt with areas such as cognitive sciences and philosophy of mind. However, my view is different about the AC because the issue of whether it is possible to have access to the result of a non Turing-computable function can be performed by other areas such as the physical sciences which don't study the human brain. Indeed, it is necessary to distinguish two ways for a model to meet the AC.

On the one hand, a model can meet the AC in an "internal sense" if a human is able to have acces to the computation result without a physical realization of the model. On the other hand, a model can meet the AC in an "external sense" if a human is able to have acces to the computation result with a physical realization of the model. Although we can't access to the computation result of the Diophantine function in an internal sense, it could be possible to physically build a real type ATM in order to have acces to the computation result in an external sense. Indeed, suppose we have an ATM physically built. We enter a Diophantine equation on its tape and wait for a finite time interval and the ATM prints 1 or 0 whether the equation has a solution or not. We could then have access to the results of the Diophantine function without to follow each computational steps. This result shows that we can regain the AC with a physical realization of an ATM. This position is supported by advocates of hypercomputation such as Jack Copeland : "the computing power of a hyperMachine usually results from operations that humans can not accomplish without the help of a real machine" [3]. This result, characterized by the link between the computing power of a hypercomputation model and its physical realization has important consequences for the notion of computation. It shows that some features belonging to hypercomputation models not only depend on mathematics. Specifically, while the computing power of a hyperMachine comes from the absence of particular mathematical constraints (the finite number of steps for example) the possibility to access to results of non Turing-computable function computed by these hyperMachines is based on physical constraints. Therefore, physical sciences decide whether or not it is possible to have access to the results of the Diophantine function.

5 Conclusion

One reason for which advocates of hypercomputation want to build a physical model of an ATM lies on the possibility to have access to the computation result of a non Turing-computable function. More specifically, a real type hyperMachine could compute an infinite number of step in a finite time without that we need to follow each computation step. Computation results of a non Turing-computable function will be printed on the ATM's tape in a finite time. Moreover, since ATM real type is based on physical laws unlike conceptual type ATM, the physical realization of a hypercomputation model adds physical constraints on its computation. Therefore, hypercomputation is not an area of pure mathematics but build a bridge between mathematics and physical sciences.

ACKNOWLEDGEMENTS

I would like to thank the editors and referees for very helpful comments during the preparation of this paper.

REFERENCES

[1] A. Church, 'An Unsolvable Problem of Elementary Number Theory.', *American Journal of Mathematics*, **Vol. 58, No. 2, pp. 345-363**, (1936).
[2] J. Copeland, 'Accelerating Turing Machine', *Minds and Machines*, **12 pp. 281-301**, (2002).
[3] J. Copeland, 'Hypercomputation', *Minds and Machines*, **12 pp. 461-502**, (2002).
[4] J. Copeland, 'The Church-Turing thesis', *Stanford Encyclopedia of Philosophy*, (2002).
[5] M. Davis, 'The Myth of Hypercomputation', *in Christof Teuscher (ed.), Alan Turing: the life and legacy of a great thinker, Springer*, (2006).
[6] R Fraser and S. Akl, 'Accelerating Machine', *Technical Report*, **2006-510**, (2006).
[7] S. Kleene, *Logique Mathématique*, Gabay, 1971.
[8] B. Loff and J. Felix Costa, 'Five Views of Hypercomputation', *International Journal of Unconventional Computing*, **5, pp. 191-207**, (2009).
[9] J.R. Lucas, 'Minds, Machines and Gödel', *Philosophy*, **36 pp. 112-127**, (1961).
[10] W. Mass, 'Networks of Spiking Neurons: the Third Generation of Neural Networks Models', *Neural Networks*.
[11] Y. Matiiassevitch, *Le Dixième Problème de* H*ilbert: son Indécidabilité*, Masson, 1995.
[12] J. Norton, 'A quantum mechanical supertask', *Foundations of Physics*, **8, Vol. 29**, (1999).
[13] J.R. Searle, 'Minds, Brains and Programs', *Behavioral and Brain Sciences*.
[14] O. Shagrir and I. Pitowski, 'Physical Hypercomputation and the Church-Turing Thesis', *Minds and Machines*, **13 pp. 87-101**, (2003).
[15] H.T. Siegelmann, 'Computation beyond the Turing Limit', *Science*, **268 pp. 545-548**, (1995).
[16] A.M. Turing, 'On Computable Numbers, with an Application to the Entscheidungsproblem', *Proceedings of the Mathematical Society*, **42**, (1936).
[17] A.M. Turing, 'Systems of Logic Based on the Ordinals', *Proceedings of the London Mathematical Society*, **45 pp. 161-228**, (1939).
[18] Hector Zenil and Francisco Hernandez-Quiroz, 'On the Possible Computational Power of the Human Mind', *Worldviews, Science and US, Gershenson, Aerts and Edmonds (eds.), World Scientific*, (2008).

Proceedings of AISB '11: Computing & Philosophy
Dimitar Kazakov and George Tsoulas (eds.)
ISBN 978-1-908187-03-1

Published by the Society for the Study of Artificial Intelligence and the Simulation of Behaviour
Printed by the University of York, York, UK