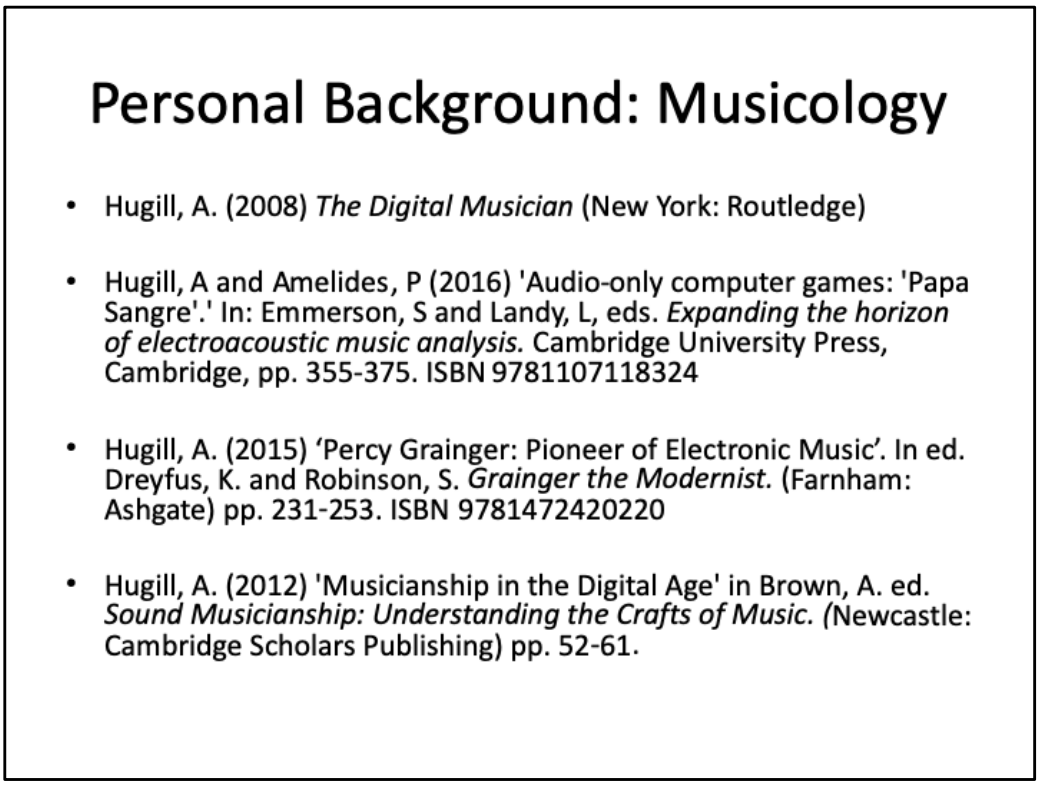
Creative Computing: concepts, theories, applications

Professor Andrew Hugill
Bath Spa University, 2017

Personal Background: Strategic Initiatives

- De Montfort University, Leicester:
  - 1995, Music, Technology and Innovation
  - 1999, Centre for Technology and the Arts
    - Included Centre for Textual Studies
  - 2005, Institute of Creative Technologies
- Bath Spa University
  - 2013, Centre for Creative Computing

De Montfort University, Leicester:
1995, Music, Technology and Innovation
1999, Centre for Technology and the Arts
Included Centre for Textual Studies
2005, Institute of Creative Technologies
Bath Spa University
2013, Centre for Creative Computing

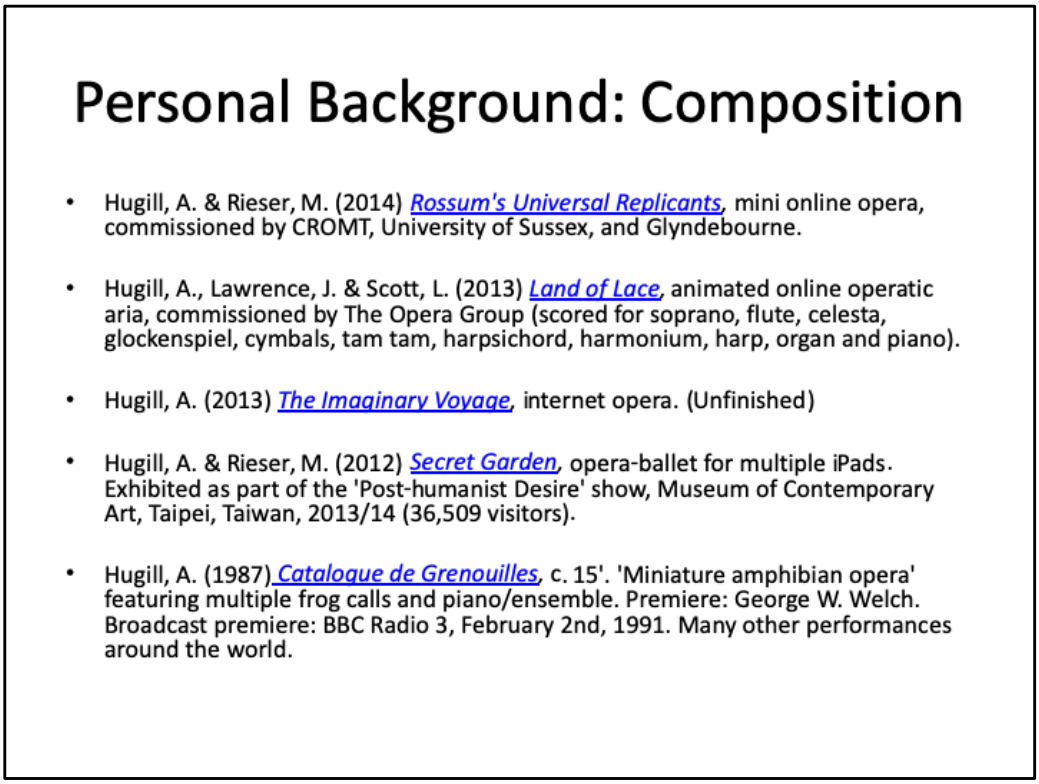


Hugill, A. (2008) *The Digital Musician* (New York: Routledge)

Hugill, A and Amelides, P (2016) 'Audio-only computer games: 'Papa Sangre'.' In: Emmerson, S and Landy, L, eds. *Expanding the horizon of electroacoustic music analysis.* Cambridge University Press, Cambridge, pp. 355-375. ISBN 9781107118324

Hugill, A. (2015) 'Percy Grainger: Pioneer of Electronic Music'. In ed. Dreyfus, K. and Robinson, S. *Grainger the Modernist.* (Farnham: Ashgate) pp. 231-253. ISBN 9781472420220

Hugill, A. (2012) 'Musicianship in the Digital Age' in Brown, A. ed. *Sound Musicianship: Understanding the Crafts of Music. (*Newcastle: Cambridge Scholars Publishing) pp. 52-61.



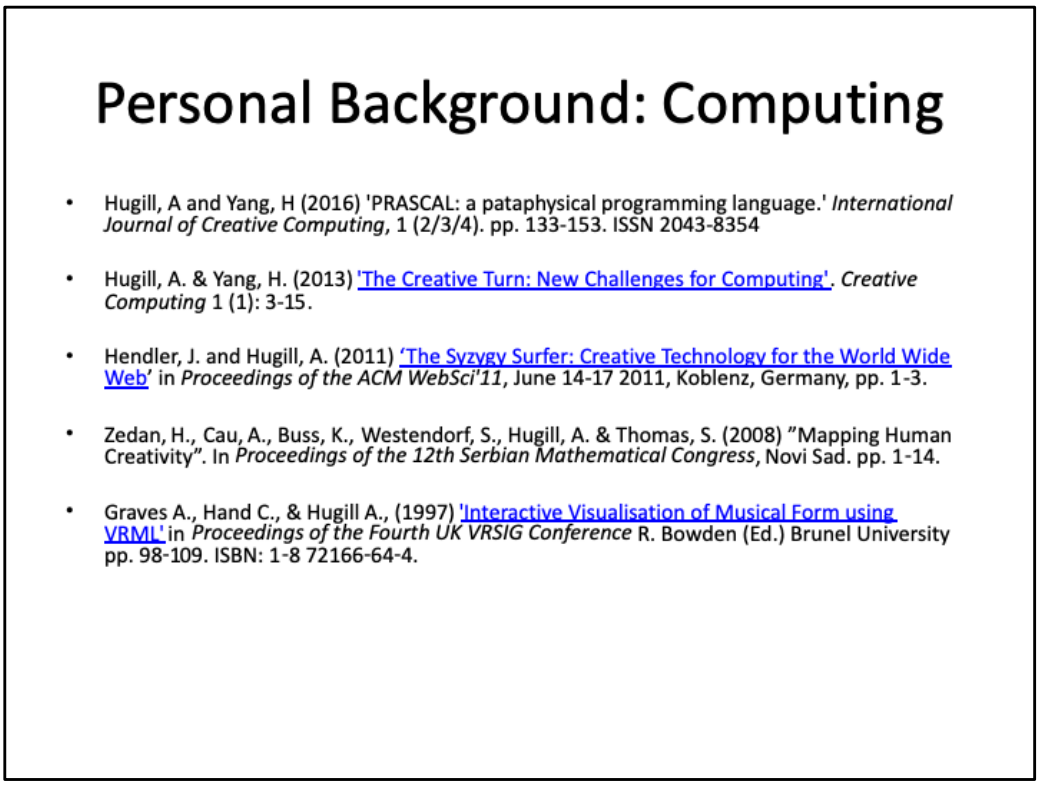
Hugill, A. & Rieser, M. (2014) *Rossum's Universal Replicants*, mini online opera, commissioned by CROMT, University of Sussex, and Glyndebourne.

Hugill, A., Lawrence, J. & Scott, L. (2013) *Land of Lace*, animated online operatic aria, commissioned by The Opera Group (scored for soprano, flute, celesta, glockenspiel, cymbals, tam tam, harpsichord, harmonium, harp, organ and piano).

Hugill, A. (2013) *The Imaginary Voyage*, internet opera. (Unfinished)

Hugill, A. & Rieser, M. (2012) *Secret Garden*, opera-ballet for multiple iPads. Exhibited as part of the 'Post-humanist Desire' show, Museum of Contemporary Art, Taipei, Taiwan, 2013/14 (36,509 visitors).

Hugill, A. (1987) *Catalogue de Grenouilles*, c. 15'. 'Miniature amphibian opera' featuring multiple frog calls and piano/ensemble. Premiere: George W. Welch. Broadcast premiere: BBC Radio 3, February 2nd, 1991. Many other performances around the world.



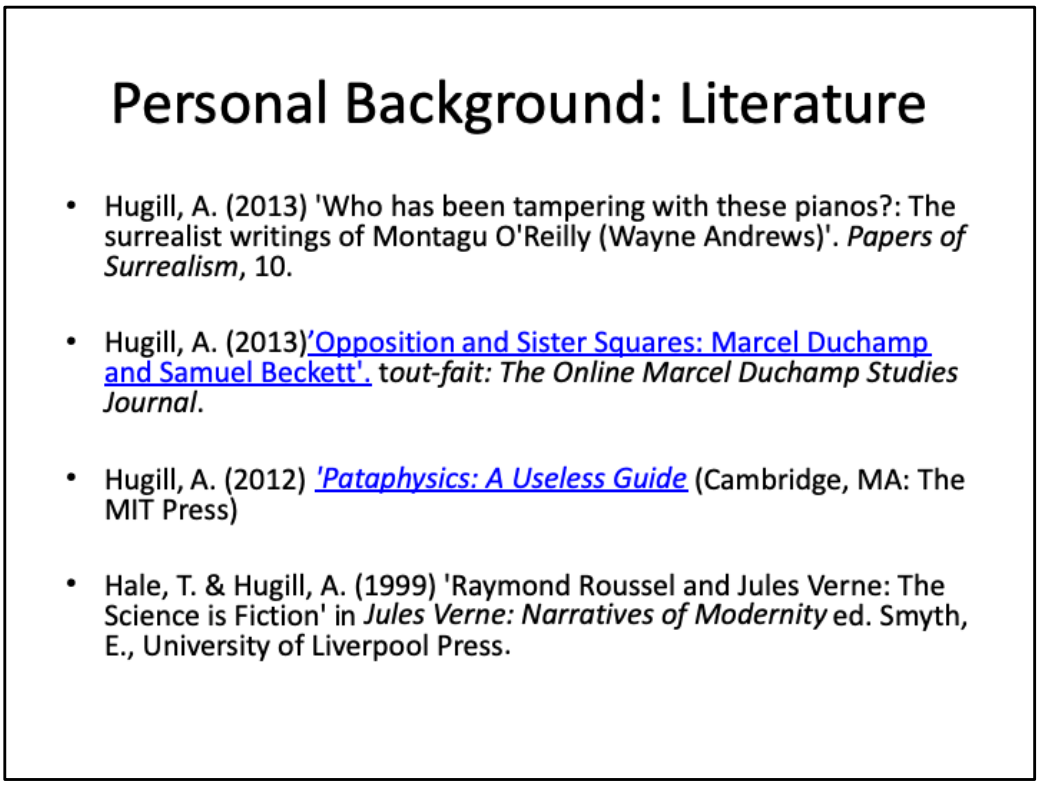
Hugill, A and Yang, H (2016) 'PRASCAL: a pataphysical programming language.' *International Journal of Creative Computing*, 1 (2/3/4). pp. 133-153. ISSN 2043-8354

Hugill, A. & Yang, H. (2013) 'The Creative Turn: New Challenges for Computing'. *Creative Computing* 1 (1): 3-15.

Hendler, J. and Hugill, A. (2011) 'The Syzygy Surfer: Creative Technology for the World Wide Web' in *Proceedings of the ACM WebSci'11*, June 14-17 2011, Koblenz, Germany, pp. 1-3.

Zedan, H., Cau, A., Buss, K., Westendorf, S., Hugill, A. & Thomas, S. (2008) "Mapping Human Creativity". In *Proceedings of the 12th Serbian Mathematical Congress*, Novi Sad. pp. 1-14.

Graves A., Hand C., & Hugill A., (1997) 'Interactive Visualisation of Musical Form using VRML' in *Proceedings of the Fourth UK VRSIG Conference* R. Bowden (Ed.) Brunel University pp. 98-109. ISBN: 1-8 72166-64-4.



Hugill, A. (2013) 'Who has been tampering with these pianos?: The surrealist writings of Montagu O'Reilly (Wayne Andrews)'. *Papers of Surrealism*, 10.

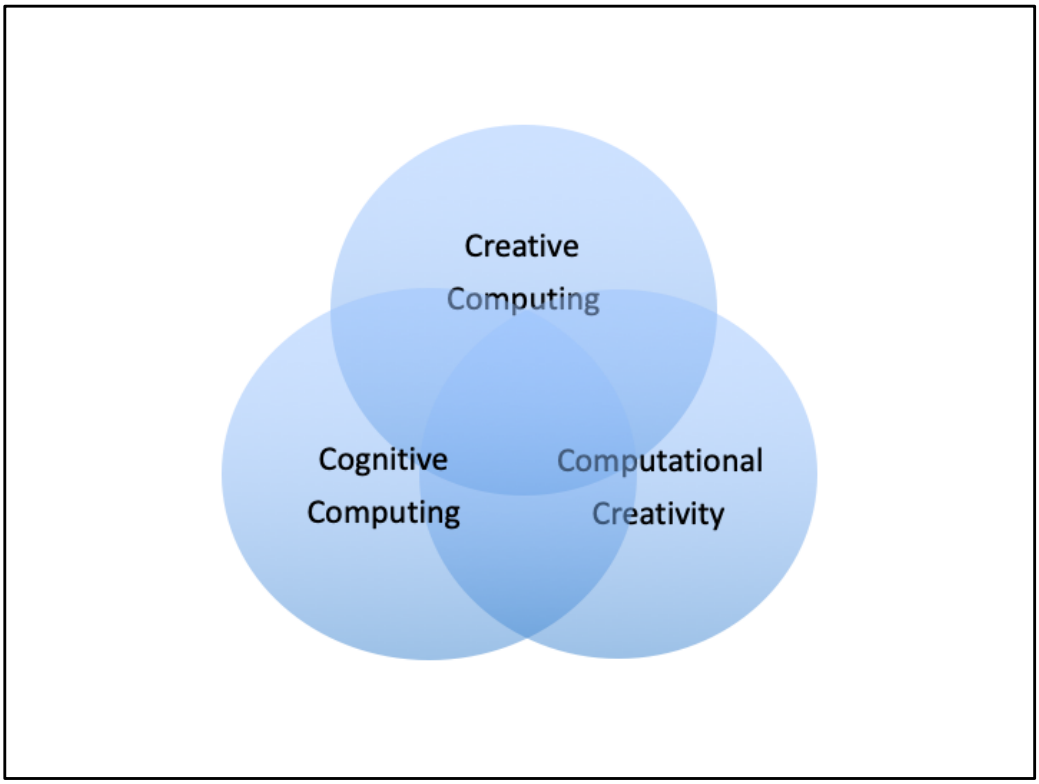
Hugill, A. (2013)'Opposition and Sister Squares: Marcel Duchamp and Samuel Beckett'. t*out-fait: The Online Marcel Duchamp Studies Journal*.

Hugill, A. (2012) *'Pataphysics: A Useless Guide* (Cambridge, MA: The MIT Press)

Hale, T. & Hugill, A. (1999) 'Raymond Roussel and Jules Verne: The Science is Fiction' in *Jules Verne: Narratives of Modernity* ed. Smyth, E., University of Liverpool Press.

Creative Computing @BSU

- Centre for Creative Computing
  - Core staff team (7), plus wider team, postdocs, VFs
  - International Journal of Creative Computing
  - International Symposium on Creative Computing
  - Funded Projects
  - PhD Cohort
- MSc in Creative Computing
- BSc Creative Computing (Animation, Gaming, Web Technologies pathways)


Creative
Computing
Cognitive
Computing
Computational
Creativity

Cognitive Computing

- Cognitive computing is the simulation of human thought processes in a computerized model.
- Cognitive computing involves self-learning systems that use data mining, pattern recognition and natural language processing to mimic the way the human brain works.
- Watson Paths

Watson


PIENSE
THINK
ΣΚΕΨΟΥ
DENKE
PENSER
$24,000
(Who is Stoker?
$ 1,000
$77,147
Who is Bram Stoker?
$ 17,973
$21,600
WHO IS BRAM STOKER?
$5600

Computational Creativity

- The goal of computational creativity is to model, simulate or replicate creativity using a computer.

- The Painting Fool
- Musical Metacreation

Creative Computing: Aims

- Human levels
  - Survival level - clothing, eating, housing and traveling
  - Social level - working and contributing to society
  - Spiritual level - being successful, being entertained, thinking philosophically, satisfying curiosity, etc.
- Creative Computing aims to help with the third level.
  - aims to serve human creativity better
  - aims to reconcile subjective ambiguity with objective precision
  - aims to do computing more creatively.

3 Types of Creative Computing development

1. Creative development of a computing product
2. Development of a Creative Computing product
3. Development of a computing environment to support creativity.

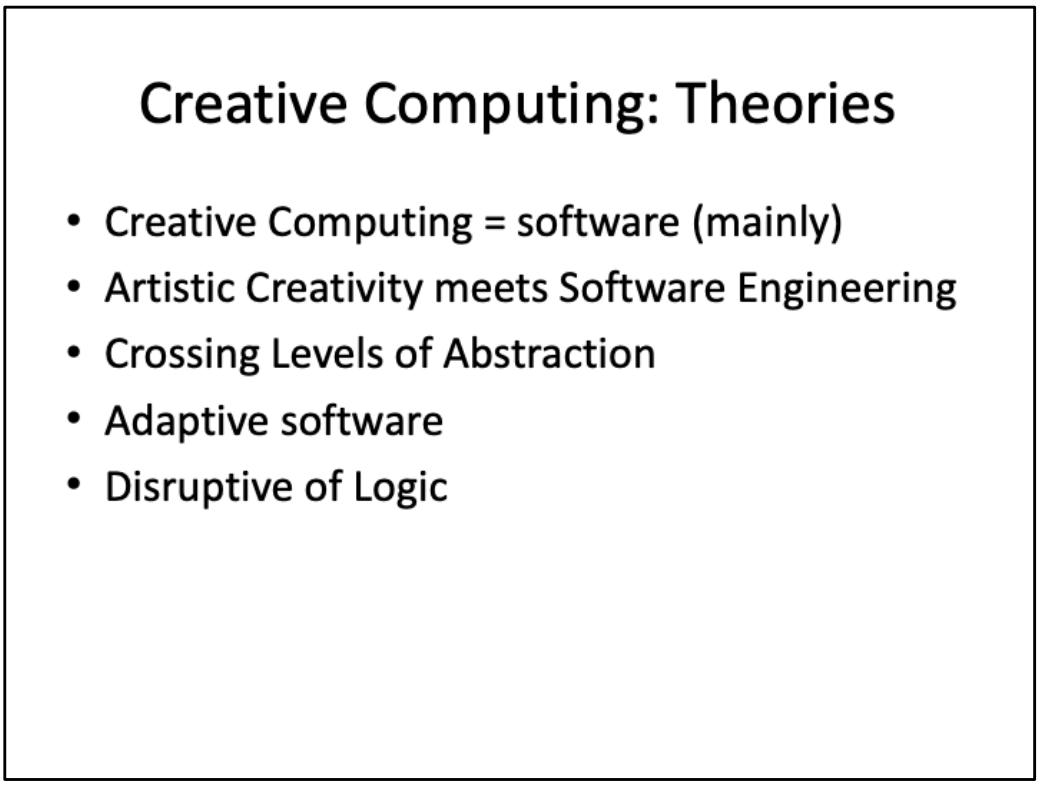
Creative development of a computing product refers to the way in which people work in the process of software development. This focus on people enables the deployment of creative methods. In order to develop software creatively, people focus on Requirements Engineering, Software Design, Software Construction, Software Testing and Software Maintenance (SWEBOK).

Development of a Creative Computing product, by contrast, focuses upon the product itself. The aim is to improve creativity and expressiveness in the software itself. Software Quality is therefore the key indicator of this type.

The development of a computing environment to support creativity requires a focus on Software Configuration Management, Software Engineering Management, Software Engineering Process and Software Engineering Tools and Methods. The overall goal is to create a general purpose platform for computing products

Creative Computing: Concepts

- Creative Computing is a new ecosystem for computing that comprises a dynamic cycle of artefact building and critical feedback.
- This ecosystem relies upon a complete interdependency of human and machine, each relating to the other in a progressive and symbiotic relationship.
- Its processes are divergent, nonlinear, convoluted, recursive and multidimensional. In short, it is not monolithic.
- Its main functions are novelty and surprise, delivering unexpected outcomes, whether in the form of software, hardware or communications interactions.
- Creative Computing is about *good data* rather than *big data*. It aims to improve human creativity by improving data quality.

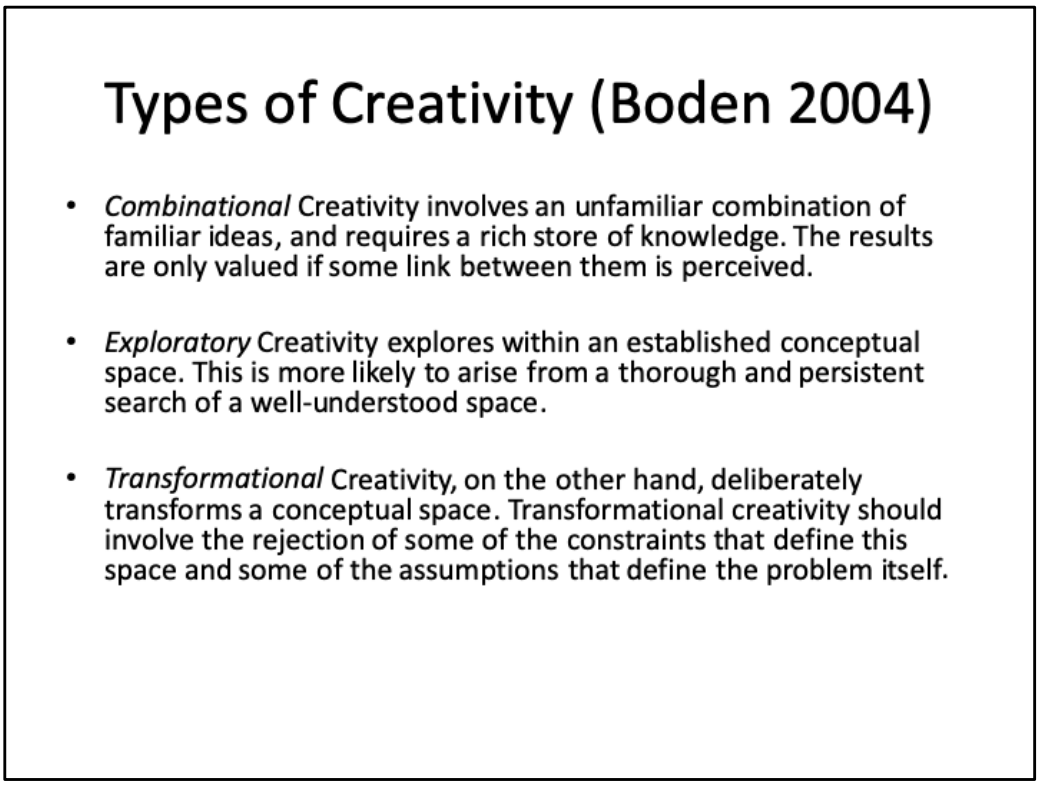


The **rationalist** paradigm, which was common among theoretical computer scientists, defines computer science as a branch of mathematics, treats programs on a par with mathematical objects, and seeks certain, a priori knowledge about their 'correctness' by means of deductive reasoning.
The **technocratic** paradigm, promulgated mainly by software engineers, defines computing as an engineering discipline, treats programs as mere data, and seeks probable, a posteriori knowledge about their reliability empirically using testing suites.
The **scientific** paradigm, prevalent in the branches of artificial intelligence, defines computer science as a natural (empirical) science, takes programs to be entities on a par with mental processes, and seeks a priori and a posteriori knowledge about them by combining formal deduction and scientific experimentation.



Creativity is a famously difficult term to define, despite the large amount of research from many different disciplinary perspectives. The commonly encountered definitions which stress novelty, surprise and usefulness often raise as many questions as they answer (Boden 2004); Kaufman & Sternberg 2010); Runco & Jaeger 2012; Weisberg 2015). What is novel to one person may not be so novel to another, as Margaret Boden seems to indicate with her division of 'P-creativity' (short for psychological creativity) as the personal kind of creativity that is novel in respect to the individual mind and 'H-creativity' (short for historical creativity) as fundamentally novel in respect to the whole of human history (Boden 2004).

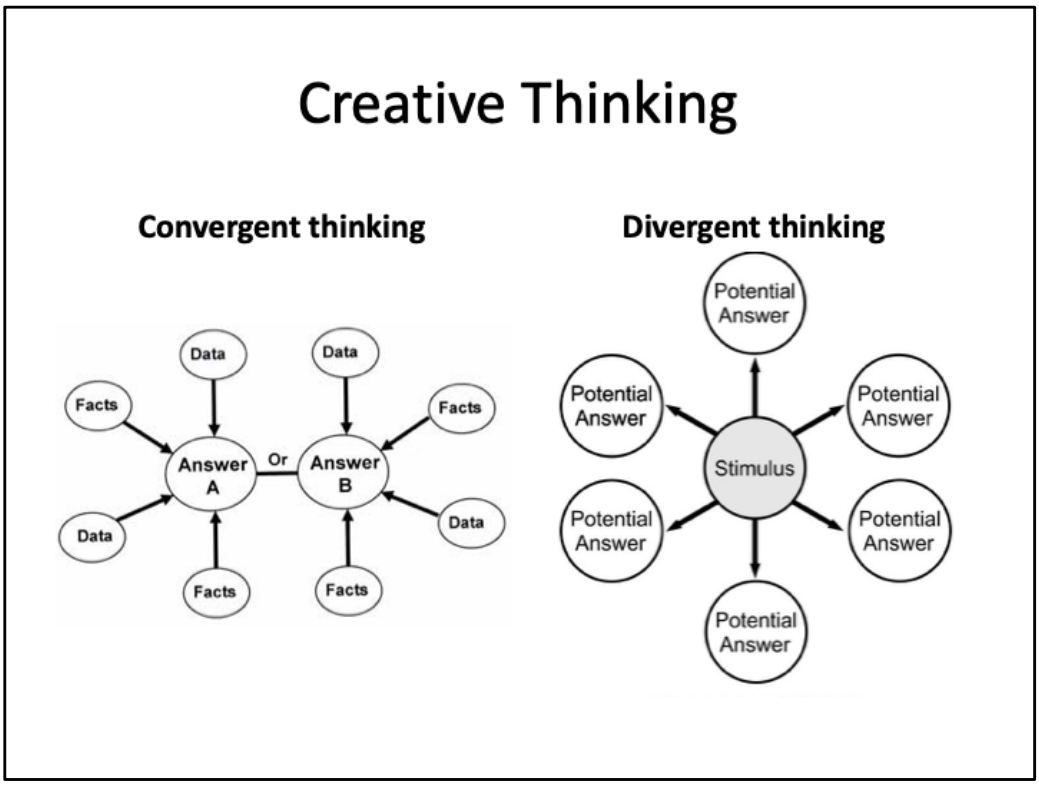
Usefulness seems to be allied to a political concept of creative *industry.* If a creative act produces something useful, so the argument goes, then it must be valuable. If it has value, then creativity itself shares that value and should be encouraged and rewarded. This argument is agreeably justifying to those of us who routinely work in creative fields, but it does gloss over the reality that much creativity is useless, at least in the sense that it is unproductive. The majority of creative work of both artists and scientists is discarded as poor quality. The amount of creativity in such work is roughly the same as work which is more successful, or sometimes a little more or a little less. In the end, it is not the *creativity* which makes it useful.



Combinational Creativity involves an unfamiliar combination of familiar ideas, and requires a rich store of knowledge. The results are only valued if some link between them is perceived.

Exploratory Creativity explores within an established conceptual space. This is more likely to arise from a thorough and persistent search of a well-understood space.

Transformational Creativity, on the other hand, deliberately transforms a conceptual space. Transformational creativity should involve the rejection of some of the constraints that define this space and some of the assumptions that define the problem itself.



The common factor in all the above is **creativity**, my fourth theme.

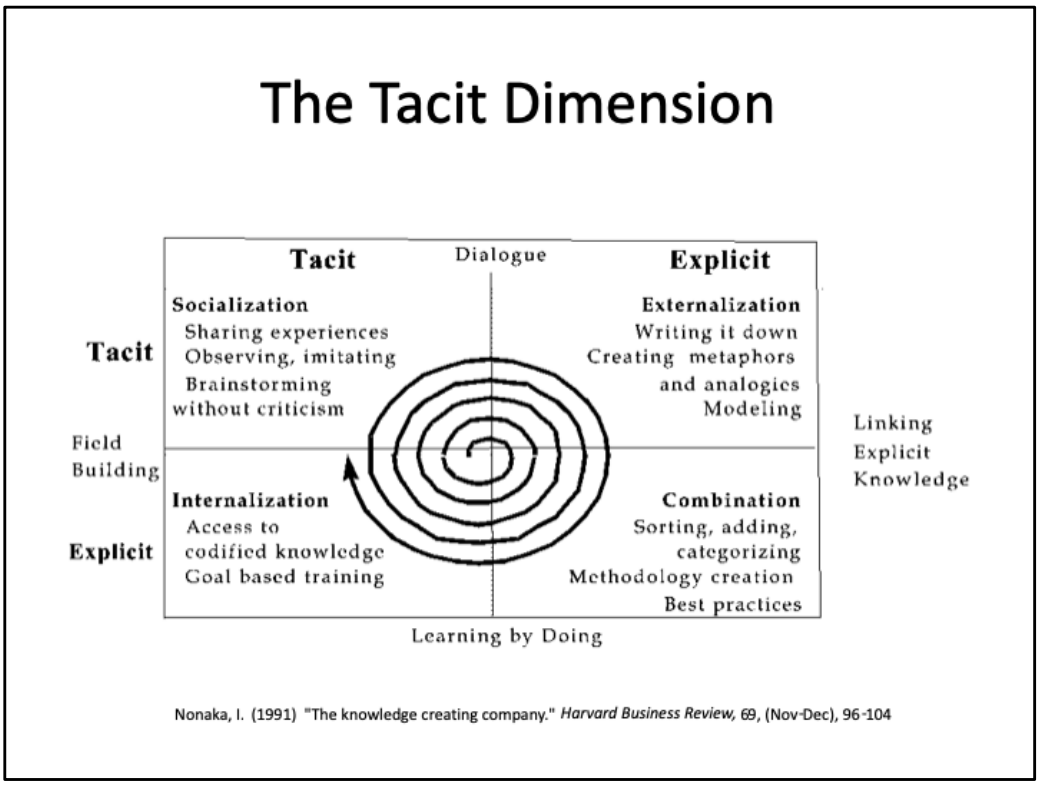
The purpose of creative computing is to get computers to help humans be more creative. My hypothesis is that to do so we need to embed creativity in the computer systems themselves. I'm wondering how the knowledge of creativity that is so abundant at Bath Spa University may be transferred. I do believe that creativity can be taught. Indeed, I have spent a great deal of my career doing exactly that. But teaching it is not at all straightforward. I generally observe that people are far more creative than they realise, but that convergent thinking (constrained by practicality) usually wins out over divergent thinking (unconcerned by practicality). To be creative requires a mix of these two, with a strong sense of timing. There has to be sufficient divergence to entertain unlikely possibilities, but sufficient convergence to know when and how these may become a reality. To my mind, this is a matter of bringing tacit knowledge into the procedural domain. And that is a statement that requires an explanation.

Knowledge Types



We can divide knowledge into three main categories. Propositional knowledge is ‘facts’ – commonly agreed certainties. Procedural knowledge concerns processes. The third category was identified by Mike Polanyi in the 1960s as “tacit knowledge”. This is the knowledge that we have without knowing how or why we have it. As Polanyi famously said: “we know more than we can tell”.

Consider the Great British Bake Off. 10 people are given the same ingredients, the same recipe, the same time, the same conditions. They produce ten different cakes. Each cake is a proposition. The process by which they created the cake is procedural. But the differences in quality and style – these are the evidence of tacit knowledge. Perhaps the individuals learned something from their family background. Perhaps they read something, or saw someone baking in a way that registered. Or perhaps something else is going on. At any rate, it is the translation of the tacit knowledge into the process that delivers the proposition.



Tacit knowledge has informed a whole new interdiscipline of knowledge management. One of its leading theorists, Ikujiro Nonaka, has produced, in an act of unconscious pataphysics, this “knowledge spiral”. In his view, explicit and tacit knowledge complement one another through four stages: socialization, externalization, combination and internalisation. This is the SECI model. Apart from the symbol of the spiral, this theory is pataphysical in the way that it encodes certain illogicalities. For example, if the defining characteristic of tacit knowledge is that it cannot be articulated, how can it be made explicit? It is exactly this kind of ambiguous reasoning that provides the conditions for creativity.

Tacit knowledge was first defined by Mike Polanyi, most notably in his book *The Tacit Dimension* (1958), in which he declared “we know more than we can tell”. Tacit knowledge is the knowledge we have without knowing or understanding how we have it, such as the ability to recognise a face or ride a bicycle. Tacit knowledge has a number of characteristics that distinguish it from the other two forms of knowledge. It is acquired through experience rather than conscious study. It is subjective, varying from person to person. It is not codified and is transferred in an intuitive way. Tacit knowledge is often seen as the origin of creativity, perhaps most enthusiastically in the field of Knowledge Management, where it is seen as a key to competitive advantage through innovation. In their book, *Enabling Knowledge Creation: How to Unlock the Mystery of Tacit Knowledge and Release the Power of Innovation,* von Krogh, Ichijo, and Nonaka argue that “recognizing the value of tacit knowledge and figuring out how to use it is the key challenge in a knowledge-creating company” (ref p. 7).

Universities are apparently less proactive in recognising the value of tacit knowledge. Traditional assessment methods tested mostly propositional knowledge through sit-down examinations. In the past fifty years or so, an increasing acceptance of projects and coursework have led to more inclusion of procedural knowledge. But tacit knowledge, while always evident in the students, has been viewed as inherently beyond the fair implementation of objective standards. This becomes an acute problem in creative disciplines such as the arts, where tacit knowledge is a key part of the origination and production of work. Academics working in these fields have perforce become skilled at harnessing tacit knowledge in the pursuit of more procedural and propositional goals. They are, in effect, teaching creativity. Yet the question of tacit knowledge itself is only indirectly addressed.

Creativity is an endlessly emergent force involving ideation and problem-solving. An individual or group may deploy both convergent and divergent thinking. Whereas convergent thinking assumes that a problem has a single correct solution (Kneller 1971), divergent thinking refuses to accept such a conservative outlook and seeks multiple, alternative solutions (Kim & Pierce 2013). Achieving an appropriate balance over time between these two modes of thinking is key to creativity. Multiple divergent solutions, while often highly stimulating, may also be ultimately unproductive. But without such divergence it is unlikely that any novel solution will be achieved. There must be a balance between the two states, with neither unduly inhibiting the other. The eventual product of this process provides evidence of the creativity that existed. Thus, the creative knowledge of the individual emerges from the tacit into the procedural domain. In other words, what you know more than you can say becomes what you can say more than you can know.

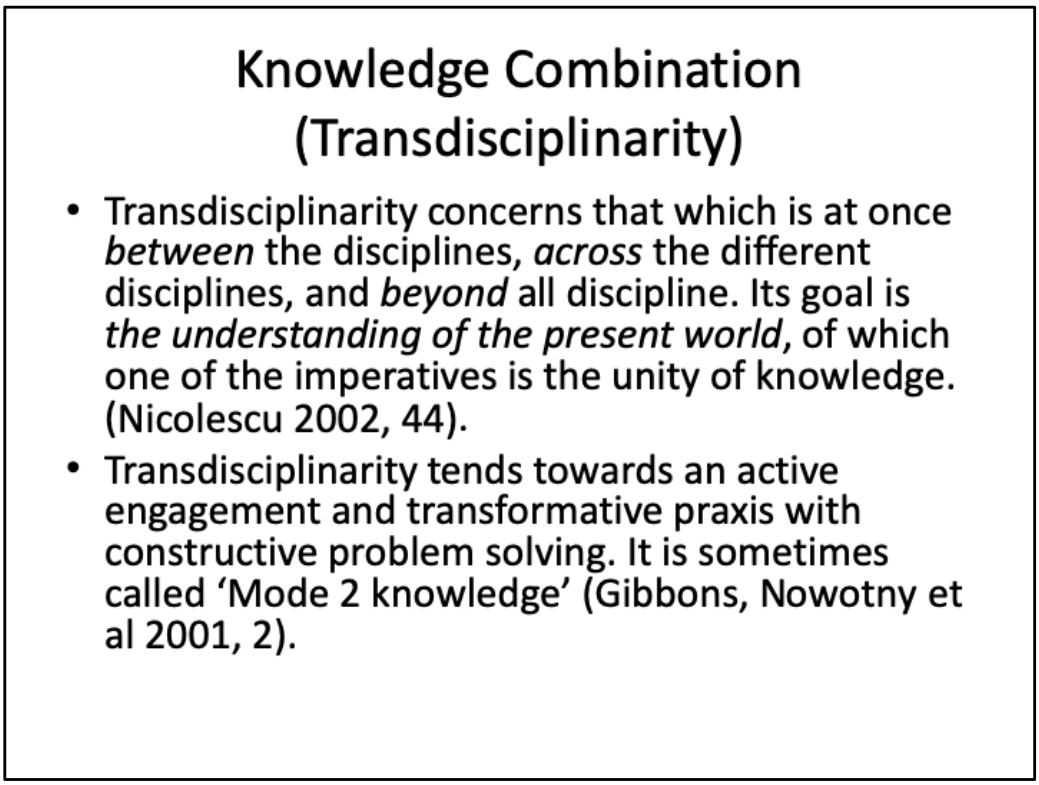
The distinction between creativity in Art and creativity in Science, then, lies in the extent to which the presence of tacit knowledge is acknowledged. In both cases, creativity is present, but in a different form. The resulting challenge is for the transdisciplinarian working with creativity to carry that understanding into the different fields. While this paper will focus mainly on artists who move into scientific disciplines, there are many examples of motion in the opposite direction. One thinks, for example, of the surrealist paintings of the zoologist Desmond Morris or the performance art work of the chemist George Brecht. Albert Einstein, who was a good amateur musician, was clear about the importance of what he called “intuition”:

"All great achievements of science must start from intuitive knowledge. I believe in intuition and inspiration.... At times I feel certain I am right while not knowing the reason." (Calaprice, 2000, 287).

The mathematician Henri Poincaré had similar thoughts. In *Science and Method* he

describes how his atomic thoughts are hooked to the wall of his mind during a still period but, when the unconscious gets to work:

some of them are detached from the wall and set in motion. They plough through space in all directions... like the gaseous molecules in the kinetic theory of gases. Their mutual collisions may then produce new combinations.



Transdisciplinarity is nothing new: the fabled Renaissance polymath saw few boundaries between 'art' and 'science'. Something of that spirit persists today, but the motivations of those who practice transdisciplinarity today are rather different from the all-encompassing vision of figures such as Leonardo da Vinci or René Descartes. *Complexity* is the reality that makes us all humble. Once upon a time, there was a single person called an opthalmologist who was expert in all aspects of the human eye, including disease and surgery. Nowadays there are many specialists who are expert in this or that particular aspect of ocular disorders. The astonishing complexity of the human eye requires this level of specialism. The same is true in oncology, or astrophysics, or literature, or indeed *any* academic discipline .

This complexity gives rise to the realisation that complex problems cannot be solved through the lens of a single discipline. Given that the fields of human knowledge are now too vast to be understood by any one individual, the result is often multidisciplinary projects on a grand scale. In general, multidisciplinary projects consist of teams of specialists addressing a common problem. The strength of these projects is that, by combining expertise in this way, a more complete solution may be found. However, the disciplinary boundaries remain unchanged in such scenarios and the specialists frequently end the projects by retiring back into their discrete fields.

The transdisciplinarian, by contrast, is one who actively moves across disciplines. Basarab Nicolescu asserts:

Transdisciplinarity concerns that which is at once *between* the disciplines, *across* the different disciplines, and *beyond* all discipline. Its goal is *the understanding of the present world*, of which one of the imperatives is the unity of knowledge. (Nicolescu 2002, 44).
Helga Nowotny and others observe that this entails the creation of a new form of knowledge:

Transdisciplinarity tends towards an active engagement and transformative praxis with constructive problem solving. It is sometimes called 'Mode 2 knowledge' (Gibbons, Nowotny et al)

This kind of knowledge includes that which is embedded in creative artefacts and processes. The European Research Council's scientific committee made this clear in a recent communication to its panel members:
[...] we firmly believe that art, like science, is a form of human creativity that found its institutionalization in the modern world. With the emergence of new technologies we expect that the borders between science and art will become even more blurred, a fruitful development that will expand the realm of human imagination. It will also impact the use of and the further development of new technologies in areas which are emerging now or are yet to be opened up. [...] Like in engineering, where the result of research might not be scientific papers, but an artifact or novel process, the humanities shall remain open to this aspect of knowledge-production (Nowotny 2012).

Human-centred Computing



Let's challenge the educators in the room…

Students rightly and predictably expect to learn how to use various key pieces of software or how to undertake programming in certain common languages. Often these are challenging tasks that take many months to master. So, course are often designed around the need to produce "power users". This is frequently justified with reference to industry and future employment.

Yet when I visit industry, what I find very often is CEOs bemoaning the lack of creativity in graduates. "Their technical skills are fine", they say to me, "but they don;t know how to be creative, how to work in a team, how (even) to hold a conversation with one another".

What tends to happen on courses is a triumph of technique over content. Clichés become pertfectly acceptable if they demonstrate appropriate skills.

What is happening here, in my opinion, is that we are developing our computer skills without developing the humans that use the computers. Creativity needs to be taught to *people* rather than computers. And creativity can be taught, make no mistake.

What are the elements of human-centred computing? Well, they are all human:

ambiguity; style; deviation; contradiction; surprise; etc. It is these elements that creative computing should exist to foster and develop. The challkenge is that they are the opposite of computers as currently being engineered. Computers deal in precision, uniformity, conformity, disambiguation, predictability, etc.

If I was to be artistic, I would suggest that we need more poetry in our engineering. Indeed, from an academic point of view, I would love to see a software engineering course that contained a compulsory module in poetry. Mathematics, after all, has come close to poetry recently.

As I suyrvey the current computational landscape, I have a sense that this spirit is slowly developing, somewhat in the teeth of engineering. When Robinson Crusoe, marooned on his desert island, encountered the cannibals that were about to devour his soon-to-be servant Friday, he was seized with doubt. Should he intervene in another's strange cultural practices? How did he know whether what he was seeing was right or wrong? In the end he intervened, and in doing so asserted a Western Christian set of values which included the effective slavery of his personal savage.

The Mars Rover is an extrme example of human-centred computing, since every aspect of the project carries the vague and slightly terrified ambitions of a species that nevertheless seeks to impose its will on another planet. Perhaps it will come across the equivalent of a footprint in the sand. Perhaps it already has, in the form of evidence of water. But will it be able to recognise the true significance of this when it does?

In the meantime, we need to insist that Creative Computing places the human at the centre of the picture. Whereas other research tries to go on improving the technology of robots, our research should improve the ways in which human beings come to an understanding of their relationships with these strange entities that we are creating.

Creative Applications



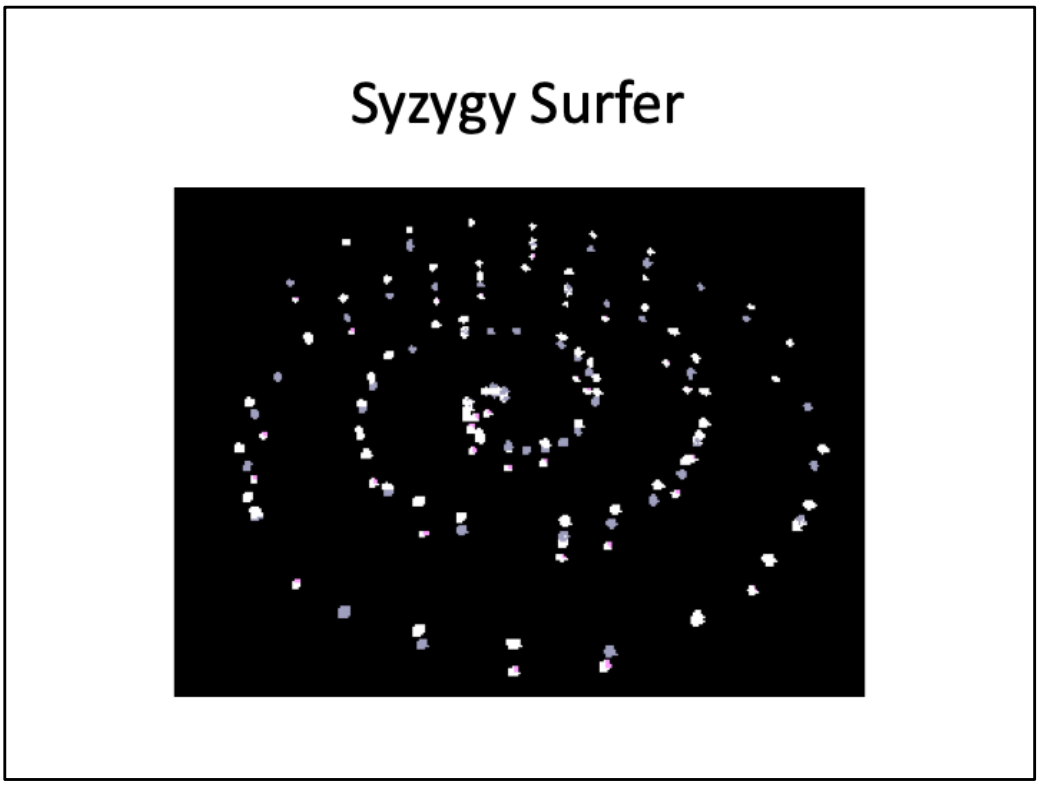
Let’s take an example of any widely used creative application, e.g. Adobe Photoshop.. When you launch this application, two things are immediately apparent. First: your creativity is assumed from the start. Even if you are going to use some of the in-built AI capabilities of the software, it is clear that you are the author of any creative decision. The second evident requirement is that you need to develop the necessary skillset to make the software work for you. In the case of Photoshop, this is quite a daunting prospect. Other software is less demanding (interestingly, it is called more ‘intuitive’). But all software requires of the user that they adapt to its requirements.

Now in one sense this is entirely equivalent to an artist’s encounter with the materials of their art, be it paint and canvas, or musical instrument, or the written word. Indeed, the resistance offered by the medium is generally a key part of artistic creation. The medium itself embodies the knowledge of the artist. The computer simulates this state of affairs, in some cases memorising user interactions so that they may be recalled later and replayed. But there is a difference too: in a computer this need not necessarily be so. When Michelangelo chipped at a block of marble, there was no ‘undo’ command, no easy way to rectify a mistake (some of the best art comes from finding creative solutions to errors). The computer environment offers not just facility but also forgiveness of error. But this apparent flexibility conceals a different kind of constraint: one on creativity.

Computer constraints are built into the software and are often driven by what is seen

as commercial necessity. My audio software, for example, defaults to a timeline that positively encourages the laying down of a drum track with a certain number of beats per minute. In order to use the software, I have to begin by changing these defaults and in effect working *against the grain.* Why not use different software? Well, I do, but sometimes this is still the best package for what I am trying to achieve. But the constraints go even deeper than this, in ways which I am sure you all recognise. The computer conditions the very concept with which one begins a session. In music, that normally relates to time-based processes. Sometimes such meddling with one's initial idea is annoying, and sometimes it is welcome, but it is always present. In most cases, artists become interested in some aspect of the software with which they can meddle in productive acts of creative abuse. That is: they use tools in ways other than that which were envisaged by the builders. This is common practice, and often produces some of the best artistic results.

But for Creative Computing as a field, the notion of a software environment which works to develop, rather than inhibit, a user's creativity is surely the most exciting possibility. Softeware that rewrites itself on the fly in response to user interaction. Software that harnesses big data to provide models and pathways for the user. But most importnatly, software that addresses the creative impulse of the user, however illogical and unformed that may be, and responds sympathetically to its complexity, rather than imposing a structure upon it. This is a difficult challenge, for it is hard to see how it might be implemented. But it's difficulty should not prevent us from trying to achieve it. Software that has the formlessness of a lump of clay, the incoherence of a half-remembered scene, the vagueness of a passing fancy, that encourages divergent thinking and ideation. There is plnty of time for the convergent, problem-solving part later, and existing software is great for that.



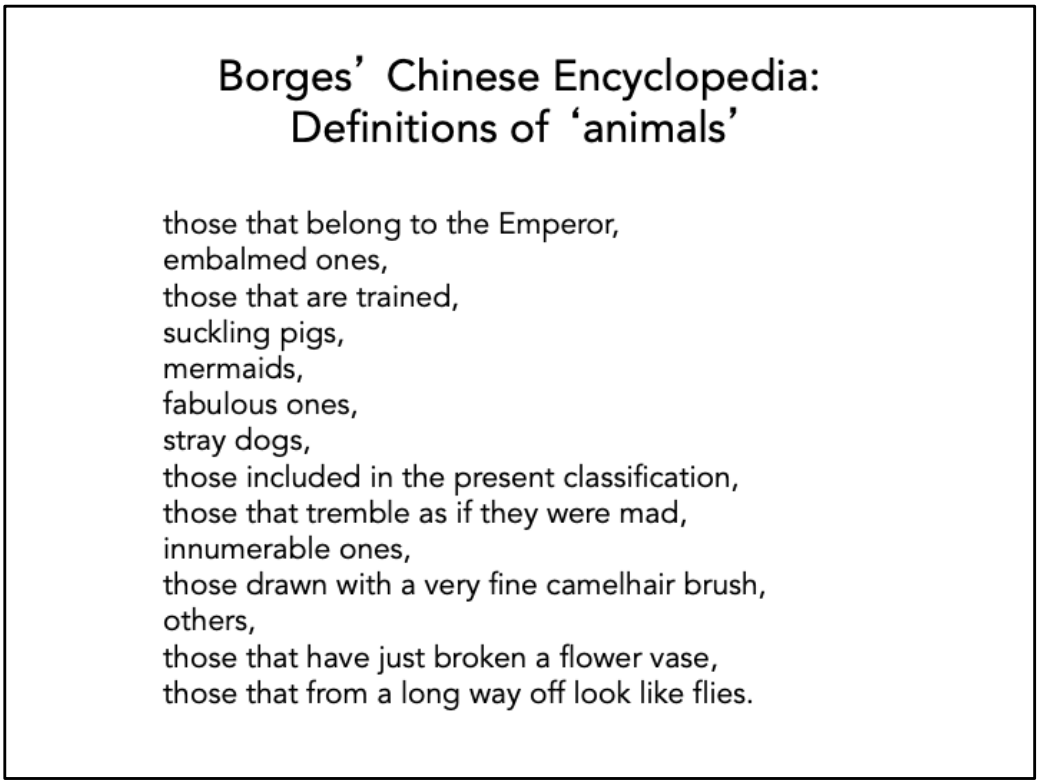
With Jim Hendler I proposed the Syzygy Surfer, a new kind of search engine which takes you not where you thought you wanted to go, but somewhere you didn't expect to go but are pleased you have reached when you get there! The results are not random but linked poetically. We will see some examples of applications of this technology later.

Jarry said: pataphysics is to metaphysics as metaphysics is to physics.

So I say: patadata is to metadata as metadata is to data.

The Syzygy Surfer benefits from the ambiguities thrown up by the process of metadata creation, precisely those human 'shortcomings' identified by Cory Doctorow (2001) in his 'Metacrap' thesis: lying, laziness, error, subjectivity, plurality, and so on.

We insert our pataphysical declensions into the metadata harvest in a way that enriches creative interaction, rather than just delivering a series of obvious mistakes.



Consider these definitions of animals in a Chinese encyclopedia described by Jorge Luis Borges in his story 'The Analytical Language of John Wilkins'. Making a categorisation like Borges' cannot be done via statistical regularity nor can it be achieved using traditional categorical knowledge. We need something that can permute knowledge about the regularities in the world, but in a not-totally-random way. We must exploit connections among concepts in a new way so as to generate relations that seem interesting, but unusual.

Borges extended

Borges' definitions of animals annotated:

Class	Relations
Those that belong to the Emperor	culturally specific, royal, possessive
embalmed ones	past, preserved
those that are trained	made uniform, commonalities
suckling pigs	young, edible, noisy
mermaids	seductive, semi-human
fabulous ones	mythical, imaginary
stray dogs	outside known networks, rough and ready
those included in the present classification	organized, self-similar
those that tremble as if they were mad	uncontrolled, incoherent
innumerable ones	general, multiple
those drawn with a very fine camelhair brush	culturally specific, aesthetically distinct
others	non-categorical, unspecifiable
those that have just broken a flower vase	present, errant, mischievous
those that from a long way off look like flies	swarming around a particular topic

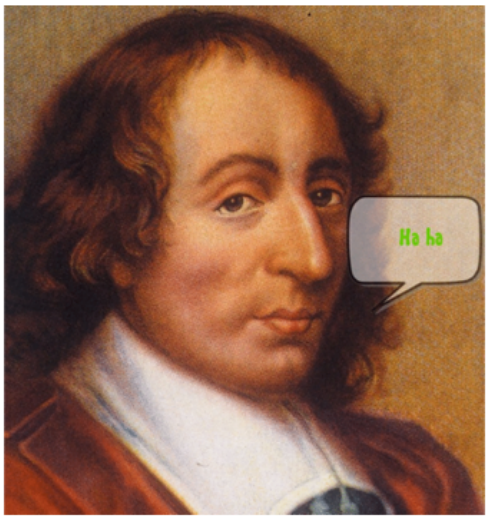
The Syzygy Surfer expands Borges' Chinese encyclopaedia by exploiting the evocative relations that the categories depend on.

Stating that these relationships exist is not a problem, but defining the features of them unambiguously would be arbitrarily hard. The obviously creative categorisation of Borges is difficult, if not impossible, to directly build in such a way that a web system, when reasoning using these, could do its job.

But I would argue that this very dichotomy can be used to enable the sort of creative encounter on the web that we are looking for. The trick is to find precise, unambiguous algorithms that the computer can apply to the growing metadata-based knowledge on the web, but which when realised can enable a human to envision the existence of relationships that the machine would not be able to represent or use directly. This behaviour, properly done, should enable the machine to produce a non-random encounter that the human may see as either logical or surprising or both, but which will let the human conceptualise the inherent ambiguity that the machine is incapable of directly creating.

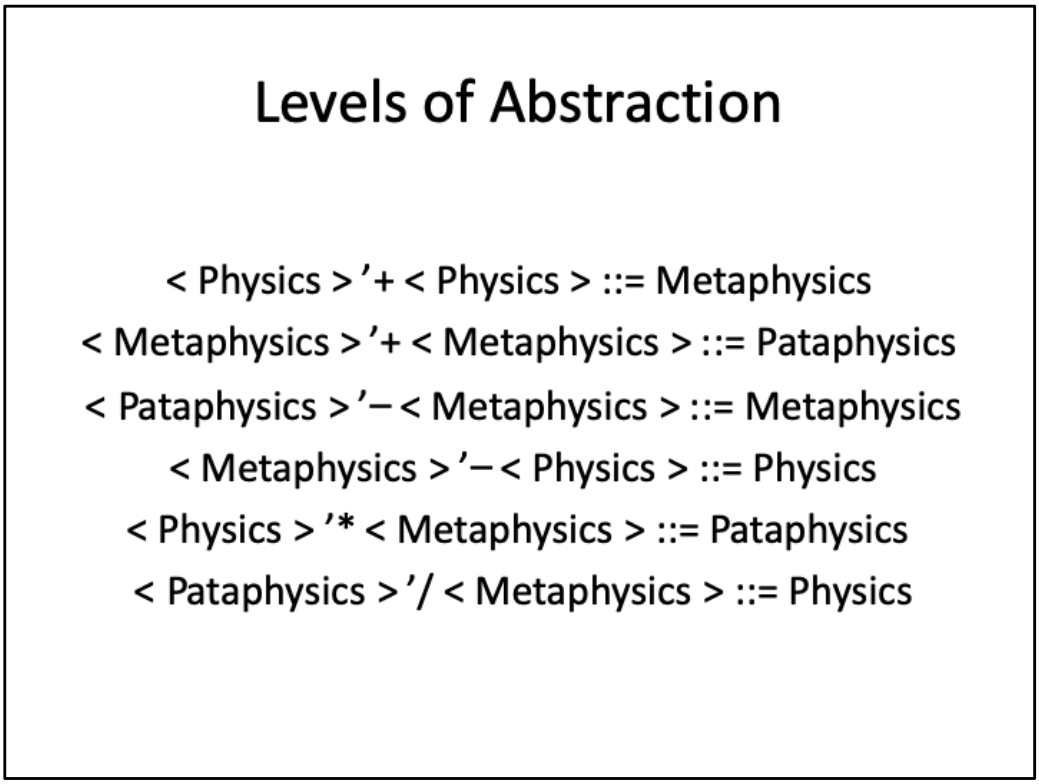
This has significant implications for the emerging technology of the Semantic Web, which creates machine-readable annotations that can be used in defining relationships between items on the web.

PRASCAL



All this introduces my theme of **logic,** or rather, **illogic**. How can a computer reason illogically? It seems to be something that is practically impossible.

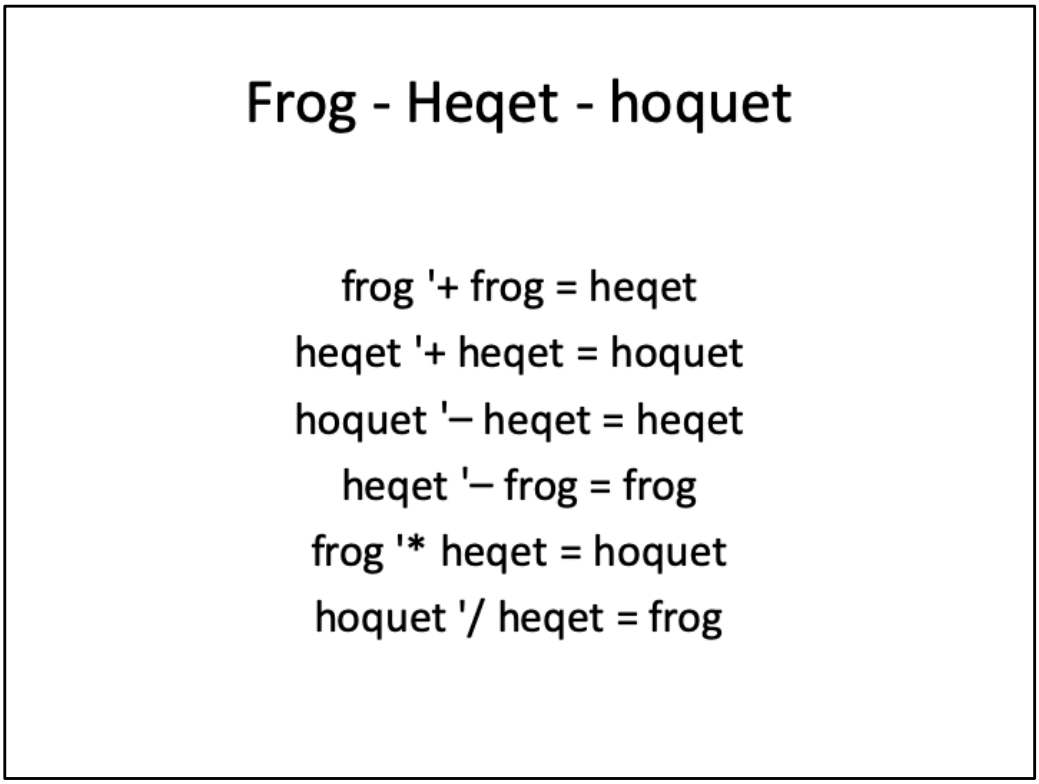
Hongji Yang and I have written a new programming language, which we call PRASCAL. Computer scientists amongst you will be aware of a rather ancient language called ‘Pascal’. By adding an ‘r’ we have injected a rascally element into Pascal. Rather like Jarry did with Shakespeare to create Ubu, we have manipulated, distorted and transformed the original language to produce creative results.



PRASCAL allows us to cross levels of abstraction. Whereas computing traditionally travels from user requirements (high level) to machine language (low level), the challenge facing Creative Computing is to do so creatively. So, here is a little example of a creative crossing using pataphysical operators. If we say that the traditional mathematical operators may be pataphysically modified with a preceding apostrophe, then we arrive at this formula:

::= means "is defined as" in Backus-Naur form.

< Physics >'+ < Physics >::= Metaphysics
< Metaphysics >'+ < Metaphysics >::= Pataphysics
< Pataphysics >'– < Metaphysics >::= Metaphysics
< Metaphysics >'– < Physics >::= Physics
< Physics >'* < Metaphysics >::= Pataphysics
< Pataphysics >'/ < Metaphysics >::= Physics



Now let's render that in more natural terms. In this example: a frog is a frog, an instance of the physically real; heqet, the Ancient Egyptian frog goddess, is an example of the metaphysical level; and hoquet, which is a musical technique of antiphony, is a pataphysical clinamen swerve away by shifting a vowel sound. So we get:

frog '+ frog = heqet
heqet '+ heqet = hoquet
hoquet '– heqet = heqet
heqet '– frog = frog
frog '* heqet = hoquet
hoquet '/ heqet = frog
frog **'*** heqet = hoquet
hoquet **'/** heqet = frog

frog hic plus frog is heqet
heqet hic plus heqet is hoquet
hoquet hic less heqet is heqet
heqet hic less frog is frog
frog hic times heqet is hoquet
hoquet hic by heqet is frog

Uboolean Logic

X	Y	Z = X (AND ⩘)Y
F	F	*F*
F	T	*F*
F	FT	*F*
T	F	*F*
T	T	*T*
T	FT	*FT*
FT	F	*F*
FT	T	*FT*
FT	FT	*FT*

PRASCAL uses a new kind of logic'.

Traditional computer logic is binary. Things are either True or False, which corresponds to the 1 and 0 states of transistors. This is called Boolean Logic, after the mathematician George Boole (1815-1864). Our logic, by contrast, envisages three states: True, False and FalsTrue, in which something is simultaneously True *and* False. We call this, naturally, Uboolean Logic.

Now, this is not the first time that a many-valued logic has been proposed. There are examples going back to the 1930s. However, Uboolean differs from all these in a crucial way: whereas they propose an intermediate condition where the Truth or Falsehood of something is unknown or undecided, ours proposes a condition in which it is both False *and* True simultaneously: the pataphysical antinomy.

Here is a typical truth table, showing the results of a conjunction (there are similar tables for other operations). It is an explosive state of affairs whose only possible outcome is an imaginary solution.

If x is false and y is false, then z is false
If x is false and y is true then z is false
If x is false and y is falstrue then z is false
If x is true and y is false then z is false

If x is true and y is true then z is true
If x is true and y is falstrue then z is falstrue
If x is flstrue and y is false then z is false
If x is falstrue and y is true then z is falstrue
If both x and y are falstrue then z is falstrue.

Quantum Computing



What are the applications of PRASCAL? Well, it's early days, but we can immediately envisage some possibilities in the emerging technology of Quantum Computing. The slide shows an object which is claimed to be a quantum computer.

A quantum computer is a machine that operates according to the principles of quantum mechanics, the physics of very small things like electrons and photons. In a classical computer, a transistor stores a single "bit" of information. If the transistor is "on," it holds a "1." If it's "off," it holds a "0." In a quantum computer, information is held in a system that can exist in two states *at the same time*, thanks to what's called the superposition principle of quantum mechanics. This dramatically increases the speed and processing power of such machines. Uboolean logic is ready made for such contradictory scenarios.

We can also apply Uboolean logic to the problem of the wave-particle duality of light. This was much discussed by Einstein, amongst others, and used by Richard Feynman to demonstrate quantum mechanics in the famous two slits experiment. The essence is that light is both a wave *and* a particle: something that is physically impossible under Newtonian particle theory and Maxwellian electromagnetic wave theory. This is evidently a FalsTrue.



Andrew Hugill
a.hugill@bathspa.ac.uk