A Discussion on Serendipity in Creative Systems

Alison Pease, Simon Colton, Ramin Ramezani, John Charnley and Kate Reed
Computational Creativity Group
Department of Computing
Imperial College, London
ccg.doc.ic.ac.uk

Abstract

We investigate serendipity, or happy, accidental discoveries, in CC, and propose computational concepts related to serendipity. These include a *focus-shift*, a breakdown of serendipitous discovery into *prepared mind*, *serendipity trigger*, *bridge* and *result* and three dimensions of serendipity: *chance*, *sagacity* and *value*. We propose a definition and standards for computational serendipity and evaluate three creative systems with respect to our standards. We argue that this is an important notion in creativity and, if carefully developed and used with caution, could result in a valuable new discovery technique in CC.

Introduction and motivation

A serendipitous discovery is one in which chance plays a crucial role and which results in a surprising, and often unsought, useful finding. This may result in a new product, such as Viagra, which was found when researching a drug for angina; an idea, such as acid rain, which was found when investigating consequences of tree clearance; or an artefact, such as the Rosetta Stone, discovered when demolishing a wall in Egypt.

In this paper we describe serendipitous discovery firstly in a human, and secondly in a computational context, and propose a series of associated computational concepts. We follow a modified version of Jordanous's evaluation guidelines for CC (Jordanous 2012), and consider three computational case studies in terms of our concepts and standards for serendipity. We finish by discussing whether serendipity in computers is either possible or desirable, and placing our ideas in the context of related work.

Eminent scientists have emphasised the role of chance in scientific discoveries: for instance, in 1679 Robert Hooke claimed: "The greatest part of invention being but a lucky bitt (*sic*) of chance" (cited in (Van Andel 1994, p. 634)), and, in 1775, Joseph Priestly said: "That more is owing to what we call *chance* ... than to any proper design, or preconceived theory in the business" (cited in (Merton and Barber 2004, p. 162)). In 1854, Louis Pasteur made what Merton and Barber refer to as "one of the most famous remarks of all time on the role of chance" (Merton and Barber 2004, p. 162) in his opening speech as Dean of the new Faculté des Sciences at Lille: "Dans les champs de l'observation le hasard ne favorise que les esprits préparés" (cited in (Van Andel 1994, p. 634 – 635)) ("In the fields of observation, fortune favours prepared minds"). Contemporary writers on serendipity include the psychologists Nickerson: "serendipity is widely acknowledged to have played a significant role in many scientific discoveries" (Nickerson 1999, 409) and Simonton: "Serendipity is a truly general process for the origination of new ideas" (Simonton 1995, p. 469); scientific journalist Singh: "The history of science and technology is littered with serendipity" (Rond and Morley 2010, p. 66); cognitive scientist and popular CC writer Boden remarks that "Chance is held to be a prime factor in many creative acts" (Boden 1990, p. 233). Equating serendipity with unexpected findings, Dunbar and Fugelsang used observational studies of scientists "in the wild" and brain imaging studies of scientific thinking to show that over half of scientists' findings are unexpected (Dunbar and Fugelsang 2005).

The word *serendipity* was coined in 1754 by Horace Walpole, as describing a particular kind of discovery. He illustrated the concept by reference to a Persian folk tale, *The Travels and Adventures of Three Princes of Serendip*, in which the princes go travelling and together make various observations and Holmesian inferences: "They were always making discoveries, by accidents and sagacity, of things which they were not in quest of" (cited in (Merton and Barber 2004, p. 2)) One such example occurs when a camel driver asks if they have seen his lost camel, and they display such detailed knowledge of the camel that the driver accuses them of stealing it. They justify their knowledge based on their observations and abductive inferences. In the last 260 years (and the last 60 in particular), this notion of a *happy, accidental discovery* has gone from being an arcane word and concept, to being part of commonplace language.

Serendipity is a value-laden concept, and has been considered both to depreciate and enhance a scientist's achievement, leading to accounts in which the role of serendipity in a discovery is either under or overrated. Despite this difficulty, there are numerous examples of serendipity in scientific discovery, some of which have been gathered into collections ((Roberts 1989) contains over 70 examples, (Rond and Morley 2010) contains examples in cosmology, astronomy, physics and other domains, and (Van Andel 1994) claims to have over 1000 (unpublished) examples). Examples from these sources include numerous medical discoveries, when a side effect was found to be more useful than the original goal; Kekulé's 1865 dream-inspired discovery of the structure of the benzine ring; the discovery of a Quechua man with maleria, drinking water which happened to be tainted from the bark of cinchona trees, that quinine (found in the bark) can cure maleria; Goodyear's discovery of vulcanised rubber, when trying to make a rubber resistent to temperature changes, after accidentally leaving a mixture of rubber and sulfer on a hot stove and finding that it charred, rather than melted; Penzias and Wilson's discovery of the *echoes of the Big Bang*, in which they were testing for the source of noise that a radio telescope was picking up, discovering eventually from a physicist that these were echoes from the Big Bang; and the Rosetta Stone, which was found by a soldier who was demolishing a wall in order to clear ground. We consider three examples below:

1. In 1928, while researching influenza, Fleming noticed an unusual clear patch in a petri dish of bacteria cultures. Subsequent examination revealed that the lid of the petri dish had fallen off (thus invalidating the experiment) and mould had fallen into the dish, killing the bacteria – resulting in the discovery of penicillin.

2. In 1948, on returning home from a walk, de Mestral found cocklebурs attached to his jacket. While trying to pick them off, he became interested in what made them stick so tightly, and started to think about uses for a system designed on similar principles – resulting in the discovery of *Velcro*.

3. In 1974, Fry was struggling to use pieces of paper to mark pages in his choir book, when he recalled of a colleague's failed attempts to develop superglue. The colleague had accidentally made a glue so weak that two glued pieces of paper could be pulled apart – this resulted in the discovery of *Post-it* notes.

We are fortunate in that the sociologist Merton and historian Barber have written a detailed account of the word "serendipity", tracing its meaning from its coinage in 1754 to 1954 (and an extended afterword on its usage from 1954 - 2004) (Merton and Barber 2004). This is a tremendous resource for those who require an algorithmic level of detail of a hard-to-grasp concept. By basing our computational interpretation on this book we can claim that we are using the word in the same way as is used in common parlance. They highlight three things of particular interest: firstly, while Walpole was unambiguous that serendipity referred to an *unsought* finding, this criterion has dropped from dictionary definitions (only 5 out of 30 English language dictionaries from 1909 - 2000 explicitly say "not sought for" (Roberts 1989, pp. 246–249); secondly, while serendipity originally described an event (a type of discovery), it has since been reconceptualised as a psychological attribute (of the discoverer); thirdly, they argue that the psychological perspective needs to be integrated with a sociological one.[1]

[1] Serendipity is usually discussed within the context of discovery, rather than creativity: in this paper we assume an association between the two.

Serendipitous discovery in a computational context

We identify characteristics of serendipitous discovery and propose corresponding computational concepts.

The focus-shift.

Serendipitous discovery often (perhaps always) involves a shift of focus. In our examples we see focus-shifts in the context of an unsuccessful (but valid) experiment (Viagra); a mistake (leaving the lid off a petri dish, thus invalidating an experiment); previously discarded refuse (weak glue); an accident (letting rubber touch hot stove); an object which is being removed (the Rosetta Stone); and something which was considered to be a nuisance (the noise in the Big Bang example, the burs on jacket), unimportant (side effects in medical drugs), or irrelevant (a dream). In all of these cases there is a radical change in the discoverer's evaluation of what is interesting: we can think of this as a reclassification of signal-to-noise (literally, in Penzias and Wilson's case).

There is not always a main focus: for instance, de Mestral was out walking when he came across the seeds of his discovery. In cases where there is a focus, this might be abandoned in favour of a more interesting or promising direction, or may be achieved alongside the shift in focus. In computational terms we could model a focus-shift by enabling a system to "change its mind" that is, to re-evaluate an object as interesting, which it had previously judged to be uninteresting.

Components.

We break down the components in serendipitous discovery as follows:

Prepared Mind: This is the discoverer's previous experiences, background knowledge, store of unsolved problems, skills and current focus. It corresponds to the set of background knowledge, unsolved problems, current goal, and so on in a system.

Serendipity Trigger: This is the part of the examples discussed which arises immediately prior to the discovery. Examples include a dream, a petri dish with a clear area, cocklebурs attached to a jacket and discarded glue. It corresponds to the example or concept in a system, which precedes the discovery.

Bridge: The techniques which enables one to go from the trigger to the result. These include reasoning techniques such as abduction (Fleming uses abductive inference to explain the surprising observation of the clear patch in petri dish); analogical reasoning (de Mestral constructed a target domain from the source domain of burs hooked onto fabric); and conceptual-blending (Kekulé blended molecule structure with a vision of a snake biting its tail and invented the concept of benzine ring). In AI, some reasoning techniques are more associated with creativity than others. For instance, analogical reasoning, conceptual-blending, genetic algorithms and automated theory formation techniques have featured heavily in CC publications. This is a good start for the reasoning techniques we identify here. Another key attribute is the ability to perform a focus-shift at an opportune time.

Result: This is the discovery itself. This may be a new product (such as Velcro), artefact (such as the Rosetta Stone), process (vulcanisation of rubber), hypothesis (such as "penicillium kills staphylococcus bacteria"), use for an object (such as quinine), and so on. The discovery may be an example of a *sought* finding (classified by Roberts as *pseudoserendipity* (Roberts 1989, p. x)), in which case the solution arises from an unknown, unlikely, coincidental or unexpected source.

Three dimensions of serendipity.

1. **Chance:** The *serendipity trigger* is unlikely, unexpected, unsought, accidental, random, surprising, coincidental, arises independently of, and before, the result. The value of carefully controlled randomness in CC and AI systems is well-established. For instance, GA systems, which are popular in CC, employ a user-defined mutation probability, usually set to around 5-10%. Introducing randomness into search has also proved profitable in other systems. Likewise, the role that surprise plays in CC is well explored.
2. **Sagacity:** This dimension describes the attributes, or skill, on the part of the discoverer (the *bridge* between the *trigger* and the *result*). In many of these examples others had been in the same position and not made the discovery. This skill involves an *open mind* (an ability to take advantage of the unpredictable); ability to focus-shift; appropriate reasoning techniques; and ability to recognise value in the discovery.
3. **Value:** The *result* must be happy, useful (evaluated externally). Measuring the value of a system's results is a well-known problem in CC, and can be evaluated independently of the programmer and system or (as is more common) by the programmer alone.

A discovery does not have to score highly on each axis to be considered serendipitous. The chances of an unanticipated use being found for a drug under development may be quite high (i.e., the role that chance plays in such a discovery is low), and the sagacity needed to discern that quinine-infused water has cured malaria may be low. While the discoveries that Walpole describes were not always important, the examples given today (in (Roberts 1989; Rond and Morley 2010; Van Andel 1994)) describe valuable, often domain-changing, discoveries. Arguably, the discovery of penicillin is the most serendipitous of our examples, since two improbable events were involved: the combination of penicillium mould and staphylococcus bacteria, and the accident of the petri dish lid falling off; it took great skill to recognise the importance of the observation, and – having saved millions of lives – it is clearly of great value.

Environmental factors.

As Merton and Barber point out, serendipitous discovery is not achieved in isolation. The discoverer is operating in a messy world and engaged in a range of activities and experiences. We propose the following characteristics of the discoverers' environments, and computational analogs:

1. **Dynamic world:** Data was presented in stages, not as a complete, consistent whole. This corresponds to streaming from live media such as the web.
2. **Multiple contexts:** Information from one context, or domain was used in another. This is a common notion in analogical reasoning.
3. **Multiple tasks:** Discoverers were often involved in multiple tasks. This corresponds to threading, or distributed computing.
4. **Multiple influences:** All discoveries took place in a social context, and in some examples the "unexpected source" was another person. This corresponds to systems such as agent architectures, in which different software agents with different knowledge and goals interact.

The three-step model of SPECS.

Jordanous summarises her evaluation guidelines in three steps; to identify a definition of creativity, state evaluation standards, and apply the standard to your creative system (Jordanous 2012). Here we apply these steps to the notion of *serendipity*.

Step 1: Identify a definition of serendipity that your system should satisfy to be considered serendipitous. We propose the following definition of computational serendipitous discovery:

Computational serendipitous discovery occurs when a) within a system with a prepared mind, a previously uninteresting serendipity trigger arises partially due to chance, and is reclassified as interesting by the system; and b) when the system, by processing this re-evaluated trigger and background information together with abductive, analogical or conceptual-blending reasoning techniques, obtains a new result that is considered useful both by the system and by external sources.

Step 2: Using Step 1, clearly state what standards you use to evaluate the serendipity of your system. With our definition in mind, we propose the following standards for computational serendipity:

Evaluation standard 1: (i) The system has a prepared mind, consisting of previous experiences, background knowledge, a store of unsolved problems, skills and (optionally) a current focus or goal. (ii) The serendipity trigger arises partially as a result of chance factors such as randomness, independence of the end result, unexpectedness, or surprisingness.

Evaluation standard 2: The system: (i) uses reasoning techniques associated with serendipitous discovery: abduction, analogy, conceptual-blending; (ii) performs a focus-shift; (iii) evaluates its discovery as useful.

Evaluation standard 3: As a consequence of the focus-shift, a result which is evaluated as useful by an external source is found.

Step 3: Test your serendipitous system against the standards stated in Step 2 and report the results. In the following section we evaluate three systems against our standards.

Computational Case Studies

Armed with an analysis of serendipity in computational settings, we investigate here the value of these insights with respect to past, present and future creative systems. In particular, we describe and evaluate from a serendipity perspective: (a) an abductive reasoning system which has already been employed in a different context (b) a series of experiments with the HR automated theory formation system aimed at promoting serendipitous discovery, and (c) a proposed an extension to a framework for creative currently under development.

The GH system

Our first system models the sort of reasoning initially described by Walpole in the Princes of Serendip story. As described in (Ramezani and Colton 2010), *Dynamic Investigation Problems* (DIPs) are a type of hybrid AI problem specifically designed to model real life situations where a guilty party has to be chosen from a number of suspects, with the decision depending on a changing (dynamic) set of facts and constraints about the current case and a changing set of case studies of a similar nature to the current case. Such situations occur in criminal or medical investigations, for instance, and the GH solver has been named after the fictional medical investigator Gregory House, although his namesake of Sherlock Holmes would equally suffice. DIPs have been designed to be unsolvable either by machine learning rules from the case studies or solving the constraints as a Constraint Satisfaction Problem, hence requiring a hybrid learning and constraint solving approach.

The GH system is given facts about a current investigation, in the form of predicates known to be true which relate various attributes of the guilty suspect but do not identify it. The problems are noisy in that only some of these facts are pertinent to finding the guilty suspect and (optionally) some facts which are required are missing. GH is also given similar facts about a number of previous cases which are related in nature to the current case, with the facts given again in predicate form. The facts of the current case and those of the case studies are given in blocks at discrete time steps, and the software solves the partial problems at each time step. To find the solutions, the facts of the current case are interpreted as a CSP to be solved by the CLPFD solver in Sicstus Prolog. Before it attempts to find a solution, GH maps the attributes of the previous cases onto those of the current case, and then uses association rule mining via the Weka machine learning package to find empirically true relationships between the attributes described in the facts. These relationships are selectively added to the CSP in order to find a more precise solution. The DIPs are set up so that the CSP without the extra constraints can be solved by multiple suspects, while – if the correct extra constraints are mined from the case studies – there is only one correct solution. Presenting further details of DIPs or the GH system is beyond the scope of this paper, but suffice to say, we performed a series of experiments to explore the nature DIPs and the solutions that GH can find. For instance, when the DIPs have 4 pertinent constraints of arity five or less, and 100% of the constraints are available either in the current case or hidden in the case studies, GH has an error rate (i.e., choosing the wrong subject) of 10%. When only 50% of the pertinent facts can be found, the error rate rises to 31%.

Standard 1: (i) The system has a *prepared mind* consisting of past cases, background knowledge and an unsolved problem. (ii) the *serendipity trigger* corresponds to a new piece of data which means that a previous case is now relevant. *Chance* factors arise in the order and which data the system receives.

Standard 2: (i) The system uses induction, abduction and constraint solving as reasoning techniques; its abductive procedures are of particular interest. (ii) *Focus-shifts* can occur if a previous case is re-evaluated by the system as relevant to the current case. (iii) The *result* is the diagnosis or identification of the guilty party, and is judged by the system to be correct.

Standard 3: As a consequence of the previously irrelevant case being re-evaluated as relevant, the diagnosis is achieved. *Value* consists in external evaluations of whether the system has reached the correct solution.

Additionally, the environmental factors are partially well represented: the system operates in a *dynamic world*; and we can see reasoning about different cases as operating in *multiple contexts*. However, it only solves one *task* at a time, and there are not currently *multiple influences*.

Experiments in model generation

The HR program (Colton 2002) is an automated theory formation system which, starting with background knowledge describing concepts and examples of those concepts, uses production rules iteratively to construct new concepts from old ones. It forms conjectures empirically which relate one or more concepts, and evaluates concepts and conjectures using a number of measures of interestingness, which in turn drives a best-first heuristic search whereby the most interesting old concepts are used to produce new concepts. For instance, the *complexity* of a concept is the number of production rule steps that were used in its production, and the complexity of a conjecture is the average of the complexity of the concepts it relates. When working in domains of pure mathematics for which axioms are given, HR can interface with the Davis-Putnam style model generator MACE and the resolution theorem prover Otter to attempt to disprove/prove empirical conjectures respectively. Working in domains of finite algebra, we started HR with only the axioms of the domain, and the background concepts required to express those axioms. In particular, HR was given no example algebras, and hence each algebra introduced to the theory came as a counterexample to a false conjecture the software made due to lack of data. In all sessions, we used modest time resources for using MACE (5 secs) and Otter (3 secs).

HR was enhanced so that whenever it found a counterexample to a new false conjecture, it tested to see whether that counterexample broke any previously unsolved *open* conjecture (i.e., for which MACE could previously find no counterexample and Otter could find no proof). We found that such occurrences were very rare. In the three test domains of group theory (associativity, identity and inverse axioms), monoid theory (associativity, identity) and semigroup theory (associativity), when run in breadth first mode, i.e., with no heuristic search, we never observed this behaviour during sessions with tens of thousands of production rule steps. This is because the search strategy means that usually the simplest concepts and hence the simplest conjectures were made early on during the session, and as became increasingly harder to find counterexamples to the progressively more difficult false conjectures, it was never the case that a later conjecture was disproved with a counterexample that also disproved an earlier one.

To attempt to encourage the re-use of counterexamples, we ran random search strategies, whereby the next concept to use in production rule steps was chosen randomly, subject to a complexity limit of 10. This strategy worked for monoids and semigroups, but not for group theory. As an example, in monoid theory, after 1532 steps, this conjecture:

$\forall b, c, d(((b * c = d \wedge b * d = c \wedge d * b = c \wedge c * d = b \wedge (d \neq id)) \leftrightarrow$
$(b * c = d \wedge d * b = c \wedge c * d = b \wedge (d \neq id))))$

was disproved by MACE finding a counterexample. The counterexample also broke this previous open conjecture:

$\forall b, c, d(((b * c = d \wedge c * b = d \wedge c * d = b \wedge (\exists e(e * c = d \wedge e * d = c))) \leftrightarrow (b * c = d \wedge (\exists f(b * c = f)) \wedge (\exists g(g * c = b)) \wedge d * b = c \wedge c * d = b)))$

This was the sole example we saw in 2000 theory formation steps in monoid theory. In semigroup theory, such events were more common: there were three times when a new counterexample was used to solve a single open conjecture, and on one occasion ten open conjectures were disproved by one counterexample.

Standard 1: (i) In these experiments HR develops a *prepared mind* during the run. The background knowledge is user-given concepts, the examples which have arisen during the run and all of the developed concepts and conjectures. The open conjectures constitute the store of unsolved problems, the skills are the production rules and other procedural mechanisms. At the point just before the *serendipity trigger*, the counterexample which arose in the context of the low complexity conjecture, the current focus is to prove or disprove the current conjecture. (ii) While there is no randomness in the way that MACE generates the *serendipity trigger*, in the random runs there is randomness in the way that the conjecture which prompted the new example was generated. In addition, the example was generated independently of the end result.

Standard 2: (i) The system did not use any of the three reasoning techniques. (ii) It did re-evaluate the previously unsolved conjecture, once it was solved, but this was not the reason that focus shifted.

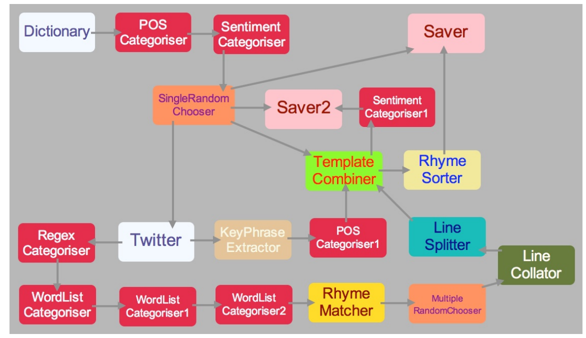

Figure 1: A poetry generating flowchart.

Standard 3: The *result* was the now-solved (previously open) conjecture. Apart from the fact that a theorem generally has higher status in mathematics than an open conjecture, we cannot claim that the solved conjectures were interesting. (None of them would appear in a textbook on group theory.) However, we can claim that, in this mode, if it was not for the example arising in a different context, the system would not have been able to solve the 18 open conjectures. We know this since it had already attempted to and failed within the time limits.

A flowcharting framework

In a project separate from our work on serendipity, we are building a flowcharting system to be used for Computational Creativity projects. Each node in the flowcharts undertakes a particular task on data types such as text and images, and the task can be generative or evaluative, or it could bring back data from websites or local databases. Without going into detail, the example flowchart in figure generates poems by compiling tweets mined from Twitter using a single adjective W as a search term, employing sentiment analysis and a rhyming dictionary along the way. The following is a stanza from a poem generated by the flowcharting system using this flowchart, where W was *malevolent*:

I hear the souls of the damned wailing in hell.
I feel a malevolent spectre hovering just behind me.
It must be his birthday.
Is God willing to prevent evil, but not able?
Then he is not omnipotent.
Is he able, but not willing?
Then he is malevolent.
It's only when his intelligence grows and he understands the laws of man that
He becomes malevolent and violent.
I don't find it malevolent, I find it affectionate.
Geeks do weird things and that can be hilarious for different reasons.

One of the purposes of the flowchart project is to have a platform for the development of creative systems that the whole Computational Creativity community to contribute to and benefit from. Our aim is to have a number of people developing nodes locally at various sites worldwide, then uploading them for everyone to share in building their own flowcharts via a GUI. We are specifically aiming for a domain independent framework, and to this end, our *inspiring examples* in building the system are the theory formation abilities of the HR system (Colton 2002), the painting abilities of The Painting Fool system (Colton 2013) and the poetry generation abilities described in (Colton, Goodwin, and Veale 2012). We currently have flowcharts which approximate the functioning of the original systems in all three cases.

Another main purpose of the project is to explore ways in which the software can automatically construct flowcharts itself - so that it can innovate at the process level. It is beyond the scope of this paper to describe how this will be done in detail, but one fact is pertinent: if/when such automated construction is possible, we will situate a version of the software on a server, constantly generating, testing and evaluating the flowcharts it produces, and making the artefacts it produces available, along with framing information (Charnley, Pease, and Colton 2012) about the process and the product. As new nodes are developed, they will be automatically made available to the system, and flowcharts will immediately be formed which utilise the new node.

The dynamic nature of this framework is clear: nodes will be accessing web services, so the data being used will be constantly changing; existing nodes will be updated and new nodes will be uploaded regularly; and new flowcharts will be created rapidly. In fact, we aim to increase this dynamic nature by having multiple such systems residing on various servers around the world, swapping nodes, flowcharts, outputs and meta-level information at regular intervals. We believe that this will increase the likelihood of chance encounters occurring to expect serendipity to follow. Moreover, the framework is not domain specific, and we will encourage the building of nodes which transfer information, say, from visual arts outputs to textual inputs, and vice versa. Thus, the environmental factors are extremely well-represented: the system operates in a *dynamic world* as it brings back data from websites or local databases, such as streaming from Twitter; the domain-independent aspect ensures that is can operate in *multiple contexts* (these will be concurrent, as in the example given in which the contexts are theory formation, painting and poetry). At any time-point there will be multiple *tasks* being undertaken by the various nodes, and, by feeding into each other these will provide *multiple influences*. We believe this will increase the likelihood of results/ideas/processes in one domain being serendipitously applied in another domain, hopefully with happy consequences.

Standard 1: (i) If we view the flowcharting system as a whole, then the *prepared mind* will be constructed via the nodes, consisting of the knowledge in the system at any time and the generative and evaluative procedures which the nodes are able to perform. Current goals will be the particular tasks that each node is involved in. (ii) The *serendipity trigger* to a particular node will arise via new information (for instance, from streaming such as Twitter) or sharing from other nodes. The sharing and updating could have a random element to it, but the main factor relating to *chance* will be that new information will arise in independent contexts, and thus will be independent of final *results*.

Standard 2: (i) As a platform for the development of creative systems that the whole CC community will contribute to and benefit from, the system as a whole will perform a variety of techniques, in particular those associated with creativity. Therefore, we expect that it will be able to perform abduction, analogy and conceptual-blending. (ii) The task that each node undertakes can be evaluative, and, if the system can perform automated construction of the flowcharts itself, it will constantly be evaluating the flowcharts it produces. Thus, *focus-shifts* should be possible; (iii) likewise, nodes will evaluate their own *results* (the artefacts that they produce).

Standard 3: The artefacts produced, such as the poem above, will be evaluated by external sources to determine the success of the whole project.

Discussion

With respect to the dynamic investigation problem and the model generation experiments described above, we can say that the former is realistic but not particularly serendipitous, while the latter is more serendipitous, but more artificial - in fact, we had to willingly make the system less effective to encourage incidents which onto which we might project the word serendipity. This raises the question of whether it is indeed possible to set up a computational situation within which such incidents genuinely occur. The flowchart system is the most promising in terms of making serendipitous discoveries. Of course, the evaluation standards themselves should be subject to evaluation, to make sure that they both reflect our intuitive notion of serendipity and are practical to apply to our CC systems.

We assume that in CC we are aiming to develop software which can surprise us, generate culturally valuable artefacts, and produce a good story about how it constructed the artefacts. There is tension between systematicity and serendipity, and it may be the case that incorporating serendipity into a creative system inhibits its ability to produce the desired artefacts. We take seriously the concern that modelling serendipity in CC may be either impossible or undesirable.

One can argue that, given the role of chance in serendipity, it is impossible to program such discoveries. Like have-a-go heroes, serendipity in our systems should be cherished but not encouraged. In response to such arguments, we have tried to characterise the sorts of environments which enhance the likelihood of making a chance discovery, and we have suggested computational analogs. Serendipity is not "mere chance" – the axes of sagacity and useful results are equally important. That serendipity-facilitating skills can be taught to people is not a new argument – much work written by scientists on serendipity is designed to teach others what skills are involved (see also (Lenox 1985)). Many (perhaps all) of the skills are standard skills of a scientist, and it may be argued that relevant machine learning techniques, such as anomaly detection and outlier analysis, already exist. We suggest that such techniques will be extremely useful, but probably not sufficient, for computational serendipitous discovery.

One might also argue that the same characteristics which aid serendipity would also aid negative serendipity. A system which allowed itself to be derailed from a task at hand might not achieve as much as one which maintains focus. Negative serendipity can be defined in various ways: Pek defines it as when: "A surprising fact or relation is seen but not (optimally) investigated by the discoverer", giving Columbus' lifelong belief that he had found a new route to Asia, rather than a new continent, as an example (Van Andel 1994, p. 369). We can also define it as a discovery which is *prevented* due to chance factors: this would be very hard to demonstrate, but relates to the "Person from Porlock" syndrome, where creative flow is interrupted due to an unwanted interruption. As well as negative serendipity, one might argue that a reliance on serendipity contrasts intelligence, and a system which uses a random search may exhibit less intelligent behaviour than one which follows a well developed heuristic search. Thus, in our HR experiment, enhancing its serendipity was a retrograde step for the system. We certainly would not advocate that all CC developers add serendipitous functionality to their existing software, which might detract from other functionality. Despite this, we suggest that serendipity is a feature which can be both possible and useful to model in future creative systems.

The examples of human serendipity all describe groundbreaking discoveries. In CC, we have learned that we must not aim to build systems which perform domain-changing acts of creativity, before systems which can perform everyday, mundane creativity (distinguished as "Big C" and "little c" creativity.) Similarly, we must expect to model "little s" serendipity before we are able to model "Big S" serendipity. The dimension which this affects the most is the third one – we must not expect the discoveries to be rated too highly with our embryonic models of computational serendipity. A useful intermediate way of evaluating the results might be with respect to other, non-serendipitous, results.

Related work

Many of the aspects we have identified as inherent in serendipitous discovery are already widespread computational techniques, and there are large bodies of work which will be particularly relevant. For instance, research into the role of problem reformulation in problem-solving, such as (Griffith, Nersessian, and Goel 2000), is relevant to the *focus-shift* aspect in that reformulation can trigger new solutions and re-evaluations. Our notion of focus-shift differs from problem reformulation, in that the focus may be on examples, artefacts, etc rather than problems, and the result of a focus-shift is a re-evaluation rather than re-representation. Problem-shift, where a problem evolves alongside possible solutions (see, for instance, (Helms and Goel 2012)), is also relevant.

Wills and Kolodner (Wills and Kolodner 1994) have analysed the processes involved in serendipitous recognition of solutions to suspended design problems, where the solutions overcome both functional fixedness and fixation on standard solutions. They propose a computational model which is based on the hypothesis that recognition arises from interaction between the processes of problem evolution and assimilation of proposed ideas into memory. Their analysis fits into our *sagacity* dimension as they elaborate skills needed to recognise value in unexpected places, and in particular ways in which the *focus-shift* can work.

There is related work on chance. For instance, Campbell's model of creativity, "blind variation and selective retention" (described in (Simonton 1999)), in which he draws an analogy between biological evolution and creativity, seems to be particularly pertinent for serendipity, with its emphasis on "blind" (Campbell elaborates his use of the term and discusses other candidates, including: chance, random, aleatory, fortuitous, haphazard, unrestricted, and spontaneous). This corresponds to our notion of *chance*.

Serendipity was formalised by Figueiredo and Campos in their paper 'The Serendipity Equations' (Figueiredo and Campos 2001). This paper used logical equations to describe the subtle differences between some of the many forms of serendipity. In practice none of the implemented examples rely on the computer to be the prepared mind. It is the user that is expected to have the 'aha' moment and thus the creative step. The computer is used to facilitate this by searching outside of the normal search parameters to engineer potentially serendipitous (or at least pseudo–serendipitous) encounters. One example of this is 'Max' created by Figueiredo and Campos (Campos and Figueiredo 2002). Here the user emails Max with a list of interests and Max finds a webpage that may be of interest to the user. Max expands the search parameters by using WordNet[2] to generate synsets for words of interest. Max also has the ability to wander; taking information from the first set of results and using these to find further pages. Other search examples include searching for analogies (Donoghue and Crean July 2002) and content (Iaquinta et al. 2008). These all use different strategies to provide new and potentially serendipitous information to the user (who must be the "prepared mind").

Further work and conclusions

The notion of serendipitous discovery is a popular and rather romantic one. Thus, when scientists or artists are framing their work for public consumption, they might tell a backstory about the role that serendipity played, which might enhance our perception of the value of the discovery or discoverer. In (Charnley, Pease, and Colton 2012), we outline the importance of producing framing information in CC. While the account of a discovery can be fictional (and thus could refer to a serendipity which did not happen), incorporating it into discovery mechanisms could result in richer framing information.

Challenging the idea that only humans can be serendipitous is a problem which is familiar to CC researchers. In the case of serendipity this may be even greater, since the notion of designing for serendipity can appear to be oxymoronic. Our message in this paper is that we should *proceed with caution* in this intriguing area.

[2]http://wordnet.princeton.edu/

Acknowledgements

We would like to thank our three reviewers who gave particularly thorough reviews and useful references. This research was funded by EPSRC grant EP/J004049.

References

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

Campos, J., and Figueiredo, A. D. 2002. Programming for serendipity. In *Proc. of the AAAI Fall Symposium on Chance Discovery – The Discovery and Management of Chance Events*.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In *Proc. of the 3rd ICCC*, 77–81.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-FACE poetry generation. In *Proc of the 3rd ICCC*.

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

Colton, S. 2013. The Painting Fool: Stories from building an automated painter. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity (forthcoming)*. Springer-Verlag.

Donoghue, D., and Crean, B. July, 2002. Searching for Serendipitous Analogies. In *European Conference on Artifical Intelligence (ECAI), Workshop on Creative Systems*.

Dunbar, K., and Fugelsang, J. 2005. Causal thinking in science: How scientists and students interpret the unexpected. In Gorman, M. E.; Tweney, R. D.; Gooding, D.; and Kincannon, A., eds., *Scientific and technical thinking*. Mahwah, NJ: Erlbaum. 57–79.

Figueiredo, A. D., and Campos, J. 2001. The Serendipity Equations. In Weber, R., and von Wangenheim, C. G., eds., *Proc. of ICCBR-4*.

Griffith, T. W.; Nersessian, N. J.; and Goel, A. 2000. Function-follows-form transformations in scientific problem solving. In *Prc. of the 22nd Annual Conference of the Cognitive Science Society*, 196–201. Cognitive Science Society.

Helms, M. E., and Goel, A. K. 2012. Analogical problem evolution in biologically inspired design. In *Design Computing and Cognition DCC'12 (J.S. Gero (ed))*. Springer.

Iaquinta, L.; Gemmis, M.; Lops, P.; Semeraro, G.; Filannino, M.; and Molino, P. 2008. Introducing Serendipity in a Content-Based Recommender System. *8th Int. Conf. on Hybrid Intelligent Systems* 168–173.

Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.

Lenox, R. S. 1985. Educating for the serendipitous discovery. *Journal of Chemical Education* 62(4):282–285.

Merton, R. K., and Barber, E. 2004. *The Travels and Adventures of Serendipity: A study in Sociological Semantics and the Sociology of Science*. New Jersey, USA: Princeton University Press.

Nickerson, R. S. 1999. Enhancing creativity. In Sternberg, R. J., ed., *Handbook of Creativity*. Cambridge, UK: Cambridge University Press. 392–430.

Ramezani, R., and Colton, S. 2010. Automatic generation of dynamic investigation problems. In *Proc of ARW*.

Roberts, R. M. 1989. *Serendipity: Accidental Discoveries in Science*. USA: John Wiley and Sons, Inc.

Rond, M. d., and Morley, I. 2010. *Serendipity: Fortune and the Prepared Mind (Darwin College Lectures)*. Cambridge University Press.

Simonton, D. K. 1995. Foresight in insight? In Sternberg, R. J., and Davidson (Eds.), J. E., eds., *The nature of insight*. MIT. 465–494.

Simonton, D. K. 1999. Creativity as blind variation and selective retention: Is the creative process darwinian? *Psychological Inquiry* 10(4):309–328.

Van Andel, P. 1994. Anatomy of the unsought finding. *The British Journal for the Philosophy of Science* 45(2):pp. 631–648.

Wills, L. M., and Kolodner, J. L. 1994. Explaining serendipitous recognition in design. In *Proc. of the 16th Annual Conference of the Cognitive Science Society*. Atlanta, GA.