Implementing feedback in creative systems: A workshop approach

Joseph Corneli[1] **and Anna Jordanous**[2]
[1] Department of Computing, Goldsmiths College, University of London
[2] School of Computing, University of Kent

Abstract

One particular challenge in AI is the computational modelling and simulation of creativity. Feedback and learning from experience are key aspects of the creative process. Here we investigate how we could implement feedback in creative systems using a social model. From the field of creative writing we borrow the concept of a Writers Workshop as a model for learning through feedback. The Writers Workshop encourages examination, discussion and debates of a piece of creative work using a prescribed format of activities. We propose a computational model of the Writers Workshop as a roadmap for incorporation of feedback in artificial creativity systems. We argue that the Writers Workshop setting describes the anatomy of the creative process. We support our claim with a case study that describes how to implement the Writers Workshop model in a computational creativity system. We present this work using patterns other people can follow to implement similar designs in their own systems. We conclude by discussing the broader relevance of this model to other aspects of AI.

1 Introduction

In educational applications it would be useful to have an automated tutor that can read student work and make suggestions based on diagnostics, like, is the paper wrong, and if so how? What background material should be recommended to the student for review?

In the current paper, we "flip the script" and look at what we believe to be a more fundamental problem for AI: computer programs that can themselves learn from feedback. After all, if it was easy to build great automatic tutors, they would be a part of everyday life. As potential users (thinking from both sides of the desk) we look forward to a future when that is the case.

Along with automatic tutoring, computational creativity is a challenge within artificial intelligence where feedback plays a vital part (for example Pérez y Pérez, Aguilar, & Negrete, 2010; Pease, Guhe, & Smaill, 2010). Creativity cannot happen in a 'silo' but instead is influenced and affected by feedback and interaction with others (Csikszentmihalyi, 1988; Saunders, 2012). Computational creativity researchers are starting to place more emphasis on social interaction and feedback in their systems and models (Saunders, 2012; Gervás & León, 2014; Corneli et al., 2015). Still, nearly 3 in 4 papers at the 2014 International Conference for Computational Creativity[1] failed to acknowledge the role of feedback or social communication in their computational work on creativity.

To highlight and contribute towards modelling feedback as a crucial part of creativity, we propose in this paper a model of computational feedback for creative systems based on Writers Workshops (Gabriel, 2002), a literary collaborative practice that encourages interactive feedback within the creative process. We introduce the Writers Workshop concept (Section 2) and critically reflect on how it could encourage serendipity and emergence in computational models of intelligence and creativity. These considerations lead us to propose a Writers Workshop computational model of feedback in computational creativity and AI systems (Section 2.1), the central contribution of this paper. In Section 3 we consider how the Writers Workshop model fits into previous work in various related areas. While we acknowledge that this paper is offering a roadmap for this model rather than a full implementation, we consider how the model could be practically implemented in a computational system and report our initial implementation work (Section 4). In concluding discussions, we reflect on divergent directions in which this work could potentially be useful in the future.

2 The Writers Workshop

Richard Gabriel (2002) describes the practise of Writers Workshops that has been put to use for over a decade within the Pattern Languages of Programming (PLoP) community. The basic style of collaboration originated much earlier with groups of literary authors who engage in peer-group critique. Some literary workshops are open as to genre, and happy to accommodate beginners, like the Minneapolis Writers Workshop[2]; others are focused on professionals working within a specific genre, like the Milford Writers Workshop.[3]

[1] ICCC is the key international conference for research in computational creativity.

[2] http://mnwriters.org/how-the-game-works/

[3] http://www.milfordsf.co.uk/about.htm

The practices that Gabriel describes are fairly typical:

- Authors come with work ready to present, and read a short sample.
- This work is generally work in progress (and workshopping is meant to help improve it). Importantly, it can be early stage work. Rather than presenting a created artefact only, activities in the workshop can be aspects of the creative process itself. Indeed, the model we present here is less concerned with after-the-fact assessment than it is with dealing with the formative feedback that is a necessary support for creative work.
- The sample work is then discussed and constructively critiqued by attendees. Presenting authors are not permitted to rebut these comments. The commentators generally summarise the work and say what they have gotten out of it, discuss what worked well in the piece, and talk about how it could be improved.
- The author listens and may take notes; at the end, he or she can then ask questions for clarification.
- Generally, non-authors are either not permitted to attend, or are asked to stay silent through the workshop, and perhaps sit separately from the participating authors/reviewers.[4]

Essentially, the Writers Workshop is somewhat like an interactive peer review. The underlying concept is reminiscent of Bourdieu's *fields of cultural production* (Bourdieu, 1993) where cultural value is attributed through interactions in a community of cultural producers active within that field.

2.1 Writers Workshop as a computational model

The use of Writers Workshop in computational contexts is not an entirely new concept. In PLoP workshops, authors present design patterns and pattern languages, or papers about patterns, rather than more traditional literary forms like poems, stories, or chapters from novels. Papers must be workshopped at a PLoP or EuroPLoP conference in order to be considered for the *Transactions on Pattern Languages of Programming* journal. A discussion of writers workshops in the language of design patterns is presented by Coplien and Woolf (1997).

The steps in the workshop can be distilled into the following phases, each of which could be realised as a separate computational step in an agent-based model:

1. Author: `presentation`
2. Critic: `listening`
3. Critic: `feedback`
4. Author: `questions`
5. Critic: `replies`
6. Author: `reflections`

The `feedback` step may be further decomposed into `observations` and `suggestions`. This protocol is what we have in mind in the following discussion of the Writers Workshop.[5]

Dialogue example

Note that for the following dialogue to be possible computationally, it would presumably have to be conducted within a lightweight process language. Nevertheless, for convenience, the discussion will be presented here as if it was conducted in natural language. Whether contemporary systems have adequate natural language understanding to have interesting interactions is one of the key unanswered questions of this approach, but protocols such as the one described above are sufficient to make the experiment.

For example, here's what might happen in a discussion of the first few lines of a poem, "On Being Malevolent". As befitting the AI-theme of this workshop, "On Being Malevolent" is a poem written by an early user-defined flow chart in the FloWr system (known at the time as Flow) (Colton & Charnley, 2014).

> FLOW: "*I hear the souls of the damned waiting in hell. / I feel a malevolent spectre hovering just behind me / It must be his birthday*."
>
> SYSTEM A: I think the third line detracts from the spooky effect, I don't see why it's included.
>
> SYSTEM B: It's meant to be humourous – in fact it reminds me of the poem you presented yesterday.
>
> MODERATOR: Let's discuss one poem at a time.

Even if, perhaps and especially because, "cross-talk" about different poems bends the rules, the dialogue could prompt a range of reflections and reactions. System A may object that it had a fair point that has not been given sufficient attention, while System B may wonder how to communicate the idea it came up with without making reference to another poem. Here's how the discussion given as example in Section 2 might continue, if the systems go on to examine the next few lines of the poem.

> FLOW: "*Is God willing to prevent evil, but not able? / Then he is not omnipotent / Is he able, but not willing? / Then he is malevolent.*"
>
> SYSTEM A: These lines are interesting, but they sound a bit like you're working from a template, or like you're quoting from something else.
>
> SYSTEM B: Maybe try an analogy? For example, you mentioned birthdays: you could consider an analogy to the conflicted feelings of someone who knows in advance about her surprise birthday party.

[4] Here we present Writers Workshops as they currently exist; however this last point is debatable. Whether non-authors should be able to participate or not is an interesting avenue for experimentation both in human and computational contexts. The workshop dialogue itself may be considered an "art form" whose "public" may potentially wish to consume it in non-participatory ways. Compare the classical Japanese *renga* form (Jin'Ichi, Brazell, & Cook, 1975).

[5] The connections between Writers Workshops and design patterns, noted above, appear to be quite natural, in that the steps in the workshop protocol roughly parallel the typical components of design pattern templates: *context*, *problem*, *solution*, *rationale*, *resolution of forces*.

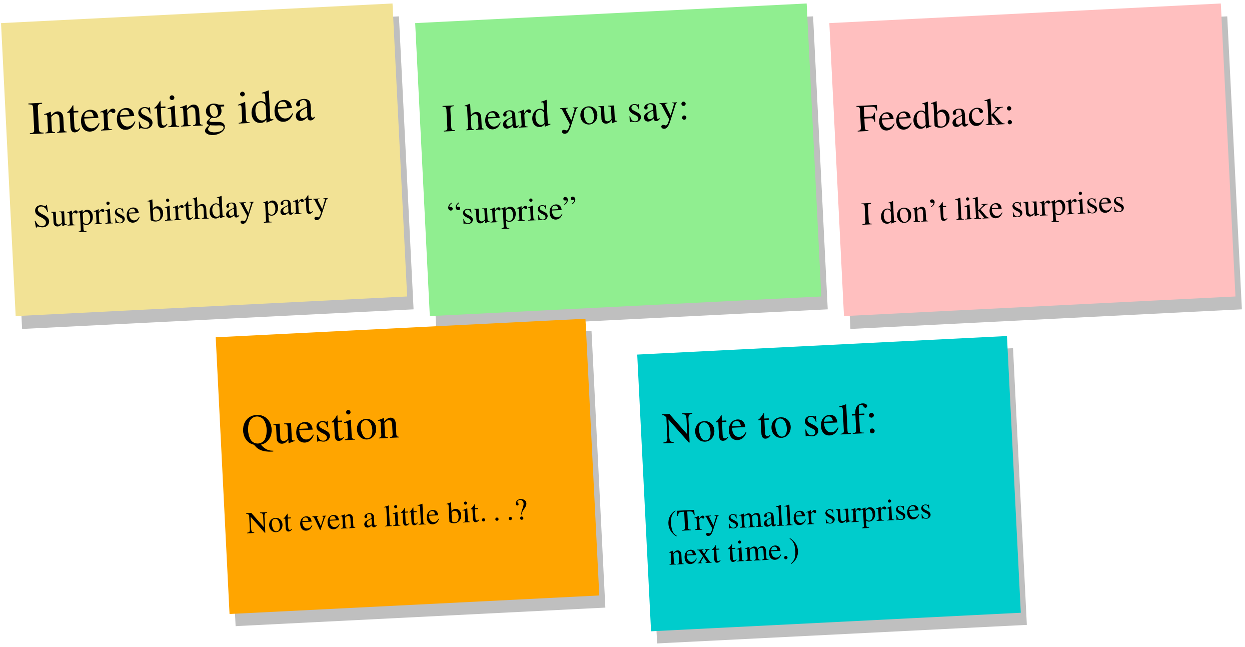

Figure 1: A paper prototype for applying the *Successful Error* pattern following a workshop-like sequence of steps

This portion of the discussion shifts the focus of the discussion onto a line that was previously considered to be spurious, and looks at what would happen if that line was used as a central metaphor in the poem.

> FLOW: Thank you for your feedback. My only question is, System B, how did you come up with that analogy? It’s quite clever.
> SYSTEM B: I’ve just emailed you the code.

Whereas the systems were initially reviewing poetry, they have now made a partial genre shift, and are sharing and remixing code. Such a shift helps to get at the real interests of the systems (and their developers). Indeed, the workshop session might have gone better if the systems had focused on exchanging and discussing more formal objects throughout.

2.2 How the Writers Workshop can lead to computational serendipity

Learning involves engaging with the unknown, unfamiliar, or unexpected and synthesising new understanding (Deleuze, 2004 [1968]). In the workshop setting, learning can develop in a number of unexpected ways, and participating systems need to be prepared for this. One way to evaluate the idea of a Writers Workshop is to ask whether it can support learning that is in some sense *serendiptious*, in other words, whether it can support discovery and creative invention that we simply couldn’t plan for or orchestrate in another way.

Figure 1 shows a paper prototype showing how one of the “patterns of serendipity” that were collected by Van Andel (1994) might be modelled in a workshop-like dialogue sequence. The patterns also help identify opportunities for serendipity at several key steps in the workshop sequence.

Serendipity Pattern: *Successful error*. Van Andel describes the creation of Post-it$^{\text{TM}}$ Notes at 3M. One of the instrumental steps was a series of internal seminars in which 3M employee Spencer Silver described an invention he was sure was interesting, but was unsure how to turn into a useful product: weak glue. The key prototype that came years later was a sticky bookmark, created by Arthur Fry. In the Writers Workshop, authors similarly have the opportunity to share things that they find interesting, but that they are not certain about. The author may want to ask a specific question about their creation: Does x work better than y? They may flag certain parts of the work as especially problematic. They may think that a certain portion of the text is interesting or important, without being sure why. Although there is no guarantee that a participating critic will be able to take these matters forward, sometimes they do – and the workshop environment will produce something that the author wouldn’t have thought of alone.

Serendipity Pattern: *Outsider*. Another example from van Andel considers the case of a mother whose son was aflicted by a congenital cateract, who suggested to her doctor that rubella during pregnancy may have been the cause. In the workshop setting, someone who is not an “expert” may come up with a sensible idea or suggestion based on their own prior experience. Indeed, these suggestions may be more sensible than the ideas of the author, who may be to close to the work to notice radical improvements.

Serendipity Pattern: *Wrong hypothesis*. A third example describes the discovery that lithium can have a therapeutic effect in cases of mania. Originally, lithium carbonate had merely been used a control by John Cade, who was interested in the effect of effect of uric acid, present in soluble lithium urate. Cade was searching for causal factors in mania, not therapies for the condition: but he found that lithium carbonate had an unexpected calming effect. Similarly, in the workshop, the author may think that a given aspect of their creation is the interesting "active ingredient," and it may turn out that another aspect of the work is more interesting to critics. Relatedly, the author may not fully comprehend a critic's feedback and may have to ask follow-up questions to understand it.

Serendipity Pattern: *Side effect*. A fourth example described by van Andel concerns Ernest Huant's discovery that nicotinamide, which he used to treat side-effects of radiation therapy, also proved efficacious against tuberculosis. In the workshop setting, one of the most important places where a side-effect may occur concerns feedback from the critic to the author. In the simple case, feedback may trigger revisions to the work under discussion. In a more general, and more unpredictable case, feedback may trigger broader revisions to the generative codebase.

This collection of patterns shows the likelihood of unexpected results coming out of the communication between author and critics. This suggests several guidelines for system development, which we will discussed in a later section.

Further guidelines for structuring and participating in traditional writers workshops are presented by Linda Elkin in (Gabriel, 2002, pp. 201–203). It is not at all clear that the same ground rules should apply to computer systems. For example, one of Elkin's rules is that "Quips, jokes, or sarcastic comments, even if kindly meant, are inappropriate." Rather than forbidding humour, it may be better for individual comments to be rated as helpful or non-helpful. Again, in the first instance, usefulness and interest might be judged in terms of explicit criteria for serendipity; see (Corneli, Pease, Colton, Jordanous, & Guckelsberger, 2014; Pease, Colton, Ramezani, Charnley, & Reed, 2013). The key criterion in this regard is the *focus shift*. This is the creation of a novel problem, comprising the move from discovery of interesting data to the invention of an application. This process is distinct from identifying routine errors in a written work. Nevertheless, from a computational standpoint, noticing and being robust to certain kinds of errors is often a preliminary step. For example, the work might contain a typo, grammatical or semantic error, while being logically sound. In a programming setting, this sort of problem can lead to crashing code, or silent failure. In general communicative context, argumentation may be logically sound, but not practically useful or poorly exposited. Finally, even a masterful, correct, and fully spellchecked piece of argumentation may not invite further dialogue, and so may fail to open itself to further learning. Identifying and engaging with this sort of deeper issue is something that skillful workshop participants may be able to do. Dialogue in the workshop can build on strong or less strong work – but provoking interpretative thoughts and comments always require a thoughtful critical presence and the ability to engage. This can be difficult for humans and poses a range of challenges for computers – but also promises some interesting results.

3 Related work

In considering the potential and contribution of the Writers Workshop model outlined in Section 2, we posit that the Writers Workshop model is useful for encouraging feedback in computational systems, and in particular systems that are designed to be creative or serendipitous.

Feedback has long been a central concept in AI-related fields such as cybernetics (Ashby, 1956; Seth, 2015). Feedback about feedback (and &c for higher orders) is understood to be relevant to thinking about *learning* and *communication* (Bateson, 1972). We now consider the importance of the roles that communicative feedback play in computational creativity and computational serendipity and discuss previous related work in incorporating feedback into such computational systems.

3.1 Feedback in computational creativity

Creativity is often envisaged as involving cyclical processes (e.g. Dickie's (1984) art circle, Pease and Colton's (2011) Iterative Development-Expression-Appreciation model). There are opportunities for embedded feedback at each step, and the creative process itself is "akin to" a feedback loop. However, despite these strong intimations of the central importance of feedback in the creative process, our sense is that feedback has not been given a central place in research on computational creativity. In particular, current systems in computational creativity, almost as a rule, do *not* consume or evaluate the work of other systems.[6]

Gervás and León (2014) theorise a creative cycle of narrative development as involving a Composer and an Interpreter, in such a way that the Composer has internalised the interpretation functionality. Individual creativity is not the poor relation of social creativity, but its mirror image. Nevertheless, even when computer models explicitly involve multiple agents and simulate social creativity (like Saunders & Gero, 2001), they rarely make the jump to involve multiple systems. The "air gap" between computationally creative systems is very different from the historical situation in human creativity, in which different creators and indeed different cultural domains interact vigorously (Geertz, 1973).

3.2 Feedback in computational serendipity

The term computational serendipity is rather new, but its foundations are well established in prior research.

Grace and Maher (2014) examine *surprise* in computing, seeking to "adopt methods from the field of computational creativity [...] to the generation of scientific hypotheses." This is an example of an effort focused on computational *invention*.

An area of AI where serendipity can be argued to play an important part is in pattern matching. Current computer programs are able to identify known patterns and "close matches" in data sets from certain domains, like music (Meredith, Lemström, & Wiggins, 2002). Identifying known

[6] An exception to the rule is Mike Cook's AppreciationBot (`https://twitter.com/AppreciationBot`), which is a reactive automaton that "appreciates" tweets from MuseumBot.

patterns is a special case of the more general concept of *pattern mining* (Bergeron & Conklin, 2007). In particular, the ability to extract *new* higher order patterns that describe exceptions is an example of "learning from feedback." Deep learning and evolutionary models increasingly use this sort of idea to facilitate strategic discovery (Samothrakis & Lucas, 2011). Similar ideas are considered in business applications under the heading "process mining" (Van Der Aalst, 2011).

In earlier work (Corneli et al., 2014, 2015), we used the idea of dialogue in a Writers Workshop framework to sketch a "theory of poetics rooted in the making of boundary-crossing objects and processes" and described (at a schematic level) "a system that can (sometimes) make 'highly serendipitous' creative advances in computer poetry" while "drawing attention to theoretical questions related to program design in an autonomous programming context."

3.3 Communications and feedback

The Writers Workshop heavily relies on communication of feedback within the workshop. Gordon Pask's conversation theory, reviewed in (Pask, 1984; Boyd, 2004), goes considerably beyond the simple process language of the workshop, although there are structural parallels. We see that a basic Pask-style learning conversation bears many similarities to the Writers Workshop model of communicative feedback (Boyd, 2004, p. 190):

1. Conversational participants are carrying out some actions and observations;
2. Naming and recording what action is being done;
3. Asking and explaining why it works the way it does;
4. Carrying out higher-order methodological discussion; and,
5. Trying to figure out why unexpected results occured.

Variations to the underlying system, protocol, and the schedule of events should be considered depending on the needs and interests of participants, and several variants can be tried. On a pragmatic basis, if the workshop proved quite useful to participants, it could be revised to run monthly, weekly, or continuously.[7]

4 Case study: Flowcharts and Feedback

This section describes work that is currently underway to implement the Writers Workshop model, not only within one system but as a new paradigm for collaboration among disparate projects. In order to bring in other participants, we need a neutral environment that is not hard to develop for: the FloWr system mentioned in Section 2.1 offers one such possibility. The basic primary objects in the FloWr system are *flowcharts*, which are comprised of interconnected *process nodes* (Charnley, Colton, & Llano, 2014; Colton & Charnley, 2014). Process nodes specify input and output types, and internal processing can be implemented in Java, or other languages that interoperate with the JVM, or by invoking external web services. One of the common applications to date is to generate computer poetry, and we will focus on that domain here.

A basic set of questions, relative to this system's components, are as follow:

1. *Population of nodes*: What can they do? What do we learn when a new node is added?
2. *Population of flowcharts*: Pease et al. (2013) have described the potentially-serendipitous repair of "broken" flowcharts when new nodes become available; this suggests the need for test-driven development framework.
3. *Population of output texts*: How to assess and comment on a generated poetic artefact?

In a further evolution of the system, the sequence of steps in a Writers Workshop could itself be spelled out as a flowchart. The process of reading a poem could be conceptualised as generating a semantic graph (Harrington & Clark, 2007; Francisco & Gervás, 2006). Feedback could be modelled as annotations to a text, including suggested edits. These markup directives could themselves be expressed as flowcharts. A standardised set of markup structures may partially obviate the need for strong natural language understanding, at least in interagent communication. Thus, we could agree that `observations` will consist of stand-off annotations that connect textual passages to public URIs using a limited comparison vocabulary, and `suggestions` will consist of simple stand-off line-edits, which may themselves be marked up with rationale. These restrictions, and similar restrictions around constrained turn-taking, could be progressively widened in future versions of the system. The way the poems that are generated, the models of poems that are created, and the way the feedback is generated, all depend on the contributing system's body of code and prior experience, which may vary widely between participating systems. In the list of functional steps below, all of the functions could have a subscripted "$\mathcal{E}$", which is omitted throughout. Exchanging path dependent points of view will tend to produce results that are different from what the individual participating systems would have come up with on their own.

I. Both the author and critic should be able to work with a model of the text. Some of the text's features may be explicitly tagged as "interesting." Outstanding questions may possibly be brought to the attention of critical listeners, e.g. with the request to compare two different versions of the poem (`presentation`, `listening`).

   1. *A model of the text.* $m : T \rightarrow M$.
   2. *Tagging elements of interest.* $\mu : M \rightarrow I$.

II. Drawing on its experience, the critic will use its model of the poem to formulate feedback (`feedback`).

   1. *Generating feedback.* $f : (T, M, I) \rightarrow F$.

[7] For a comparison case in computer Go, see http://cgos.computergo.org/.

III. Given the constrained framework for feedback, statements about the text will be straightforward to understand, but rationale for making these statements may be more involved (`questions`, `replies`).

   1. *Asking for more information.* $q : (M, F, I) \to Q$.
   2. *Generating rationale.* $a : (M, F, Q) \to \Delta F$.

IV. Finally, feedback may affect the author's model of the world, and the way future poems are generated (`reflection`).

   1. *Updating point of view.* $\rho : (M, F) \to \Delta \mathcal{E}$.

The final step is perhaps the most interesting one, since it invites us to consider how individual elements of feedback can "snowball" and go beyond line-edits to a specific poem to much more fundamental changes in the way the presenting agent writes poetry. Here methods for pattern mining, discussed in Section 3.2, are particularly relevant. If systems can share code (as in our sample dialogue in Section 2.1) this will help with the rationale-generating step, and may also facilitate direct updates to the codebase. However, shared code may be more suitably placed into the common pool of resources available to FloWr than copied over as new "intrinsic" features of an agent.

Although different systems with different approaches and histories are important for producing unexpected effects, "offline" programmatic access to a shared pool of nodes and existing flowcharts may be useful. Outside of the workshop itself, agents may work to recombine nodes based on their input and output properties to assemble new flowcharts. This can potentially help evaluate and evolve the population of nodes programmatically, if we can use this sort of feedback to define fitness functions. The role of temporality is interesting: if the workshop takes place in real time, this will require different approaches to composition that takes place offline (Perez, Samothrakis, Lucas, & Rohlfshagen, 2013). Complementing these "macro-level" considerations, it is also worth commenting on the potential role of "micro-level" feedback within flowcharts. Local evaluation of output from a predecessor node could feed backwards through the flowchart, similar to backpropagation in neural networks. This would rely on a reduced version of the functional schema described above.

5 Concluding discussion and future directions

We have described a *general* and *computationally feasible* model for using feedback in AI systems, particularly creative systems. The Writers Workshop concept, borrowed from creative writing, is transformed into a model of a structured approach to eliciting, processing and learning from feedback. To better evaluate how the Writers Workshop model helps us advance in our goal of incorporating feedback into artificial creativity, we critically considered how the model fits into related work. In particular, we found that serendipity, a key concept within creativity and AI more generally, is a concept with which the Writers Workshop model could assist computational progress. In this respect, we should highlight the difference between "global" analytics describing the collection of nodes and flowcharts in the FloWr ecosystem, and the path-dependent process of analysis and synthesis that takes place in a workshop setting.

Our preliminary implementation work (Section 4) shows that the model can be transfered to a functional implementation. This work highlights several considerations relevant to further work with the Writers Workshop model:

- Each contributing system should come to the workshop with at least a basic awareness of the workshop protocol, with work to share, and prepared to give constructive feedback to other systems.
- The workshop itself needs to be prepared, with a suitable communication platform and a moderator or global flowchart for moving the discussion from step to step.
- A controlled vocabulary for communications and interaction would be a worthwhile pursuit of future research, perhaps based on an ontology inspired by the Interaction Network Ontology.[8]
- In order to get the most value out of the workshop experience, systems (and their wranglers) should ideally have questions they are investigating. As discussed above, prior experience plays an important role in every step. This opens up a range of issues for further research on modeling motivations and learning from experience.
- Systems should be prepared to give feedback, and to carry out evaluations of the helpfulness (or not) of feedback from other systems and of the experience overall.

Developing systems that could successfully navigate this collaborative exercise would be a significant advance in the field of computational creativity. Since the experience is about learning rather than winning, there is little motivation to "game the system" (cf. Lenat, 1983). Instead the emphasis is squarely upon mutual benefit: computational systems helping to develop each other through communication and feedback.

The benefits of the Writers Workshop approach could innovate well beyond models for feedback and communication within a particular environment or restricted domain. Following the example of the Pattern Languages of Programming (PLoP) community, we propose that the Writers Workshop model could be deployed within the Computational Creativity community to design a workshop in which the participants are computer systems instead of human authors. The annual International Conference on Computational Creativity (ICCC), now entering its sixth year, could be a suitable venue.

Rather than the system's creator presenting the system in a traditional slideshow and discussion, or a system "Show and Tell," the systems would be brought to the workshop and would present their own work to an audience of other systems, in a Writers Workshop format. This could be accompanied by a short paper for the conference proceedings written

[8] The Interaction Network Ontology primarily describes interactions within humans as opposed to within human societies; a distinct *Social* Interaction Ontology does not seem to exist at present. However, the classes of the Interaction Network Ontology appear to be quite broadly relevant. This ontology is documented at `http://www.ontobee.org/browser/index.php?o=INO`. Its URI is `http://svn.code.sf.net/p/ino/code/trunk/src/ontology/INO.owl`.

by the system's designer describing the system's current capabilities and goals. If the Workshop really works well, future publications might adapt to include traces of Workshop interactions, commentary from a system on other systems, and offline reflections on what the system might change about its own work based on the feedback it receives. Paralleling the PLoP community, it could become standard to incorporate the workshop into the process of peer review for the new *Journal of Computational Creativity*.[9] AI systems that review each other would surely be a major demonstration and acknowledgement of the usefulness of feedback within AI.

In closing, we wish to return briefly to the scenario of computer generated feedback in educational contexts that we raised at the beginning of this paper and then set aside. The elements of our functional design for sharing feedback among computational agents has a range of features that continue to be relevant for generating useful feedback with human learners. Students are experience-bound, and a robust approach to formative assessment and feedback should take into account the student's historical experience, so far as this can be known or inferred. In order for feedback, recommendations, and so on to adequately take individual history into account, sophisticated modelling and reasoning would be required. Nevertheless, from the point of view of participating computational agents, a student may simply look like another agent. It is in this regard that computational models of learning from feedback are seen as fundamental.

Acknowledgement

Joseph Corneli's work on this paper was supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant number 611553 (COINVENT).

References

Ashby, W. R. (1956). *An introduction to cybernetics.* London, UK: Chapman & Hail Ltd.

Bateson, G. (1972). *Steps to an ecology of mind.* Chicago: University of Chicago Press.

Bergeron, M., & Conklin, D. (2007). Representation and discovery of feature set patterns in music. In *International Workshop on Artificial Intelligence and Music, at IJCAI-07, The Twentieth International Joint Conference on Artificial Intelligence* (pp. 1–12).

Bourdieu, P. (1993). *The field of cultural production: Essays on art and literature.* Cambridge, UK: Polity Press.

Boyd, G. M. (2004). Conversation theory. In D. H. Jonassen (Ed.), *Handbook of research for educational communications and technology* (2nd ed., pp. 179–197). Lawrence Erlbaum.

Charnley, J., Colton, S., & Llano, M. T. (2014). The FloWr framework: Automated flowchart construction, optimisation and alteration for creative systems. In *Proceedings of the 5th International Conference on Computational Creativity.*

Colton, S., & Charnley, J. (2014). Towards a Flowcharting System for Automated Process Invention. In D. Ventura, S. Colton, N. Lavrac, & M. Cook (Eds.), *Proceedings of the Fifth International Conference on Computational Creativity.*

Coplien, J. O., & Woolf, B. (1997). A pattern language for writers' workshops. *C++ report*, *9*, 51–60.

Corneli, J., Jordanous, A., Shepperd, R., Llano, M. T., Misztal, J., Colton, S., & Guckelsberger, C. (2015). Computational Poetry Workshop: Making Sense of Work in Progress. In *Proceedings of the Sixth International Conference on Computational Creativity.* Retrieved from http://metameso.org/~joe/docs/poetryICCC-wip.pdf

Corneli, J., Pease, A., Colton, S., Jordanous, A., & Guckelsberger, C. (2014). *Modelling serendipity in a computational context.* Retrieved from http://arxiv.org/abs/1411.0440 (Under review.)

Csikszentmihalyi, M. (1988). Society, culture, and person: a systems view of creativity. In R. J. Sternberg (Ed.), *The nature of creativity* (chap. 13). Cambridge, UK: Cambridge University Press.

Deleuze, G. (2004 [1968]). *Difference and repetition.* Bloomsbury Academic.

Dickie, G. (1984). *The art circle: A theory of art.* Haven.

Francisco, V., & Gervás, P. (2006). Automated mark up of affective information in English texts. In *Text, speech and dialogue* (pp. 375–382).

Gabriel, R. P. (2002). *Writer's Workshops and the Work of Making Things: Patterns, Poetry...* Addison-Wesley Longman Publishing Co., Inc.

Geertz, C. (1973). *The interpretation of cultures: Selected essays.* Basic Books (AZ).

Gervás, P., & León, C. (2014). Reading and Writing as a Creative Cycle: The Need for a Computational Model. In *5th International Conference on Computational Creativity, ICCC 2014.* Ljubljana, Slovenia.

Grace, K., & Maher, M. L. (2014). Using Computational Creativity to Guide Data-Intensive Scientific Discovery. In Y. Gil & H. Hirsh (Eds.), *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence.* (Discovery Informatics Workshop: Science Challenges for Intelligent Systems.)

Harrington, B., & Clark, S. (2007). ASKNet: Automated Semantic Knowledge Network. In A. Howe & R. Holte (Eds.), *Procs. of the Twenty-Second AAAI Conference on Artificial Intelligence* (Vol. 2, pp. 889–895). AAAI Press.

Jin'Ichi, K., Brazell, K., & Cook, L. (1975). The Art of Renga. *Journal of Japanese Studies*, 29–61.

Lenat, D. B. (1983). EURISKO: a program that learns new heuristics and domain concepts: the nature of heuristics III: program design and results. *Artificial Intelligence*, *21*(1), 61–98.

Meredith, D., Lemström, K., & Wiggins, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, *31*(4), 321–345.

[9] http://www.journalofcomputationalcreativity.cc

Pask, G. (1984). Review of conversation theory and a protologic (or protolanguage), Lp. *ECTJ*, *32*(1), 3-40. Retrieved from http://dx.doi.org/10.1007/BF02768767 doi: 10.1007/BF02768767

Pease, A., & Colton, S. (2011). Computational creativity theory: Inspirations behind the FACE and the IDEA models. In *Proceedings of the Second International Conference on Computational Creativity.*

Pease, A., Colton, S., Ramezani, R., Charnley, J., & Reed, K. (2013). A Discussion on Serendipity in Creative Systems. In *Proceedings of the Fourth International Conference on Computational Creativity.*

Pease, A., Guhe, M., & Smaill, A. (2010). Some aspects of analogical reasoning in mathematical creativity. In *Proceedings of the International Conference on Computational Creativity* (p. 60-64). Lisbon, Portugal.

Perez, D., Samothrakis, S., Lucas, S., & Rohlfshagen, P. (2013). Rolling horizon evolution versus tree search for navigation in single-player real-time games. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation* (pp. 351–358).

Pérez y Pérez, R., Aguilar, A., & Negrete, S. (2010). The ERI-Designer: A computer model for the arrangement of furniture. *Minds and Machines*, *20*(4), 533-564.

Samothrakis, S., & Lucas, S. (2011). Approximating n-player Behavioural Strategy Nash Equilibria Using Coevolution. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 1107–1114).

Saunders, R. (2012). Towards autonomous creative systems: A computational approach. *Cognitive Computation*, *4*(3), 216–225.

Saunders, R., & Gero, J. S. (2001). The digital clockwork muse: A computational model of aesthetic evolution. In *Proc. Annual Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour (SSAISB)* (pp. 12–21).

Seth, A. (2015). The cybernetic bayesian brain: From interoceptive inference to sensorimotor contingencies. In T. Metzinger & J. Windt (Eds.), *Open mind project* (p. 1-24). Frankfurt: MIND Group.

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

Van Der Aalst, W. (2011). *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media.