Towards a Creativity Commons

Joseph Corneli
Department of Computing
Goldsmiths College, University of London
j.corneli@gold.ac.uk

May 23, 2014

Prepared for delivery at the Workshop on the Ostrom Workshop (WOW5) conference,
Indiana University Bloomington, June 18–21, 2014.

Abstract

A preliminary treatment of a social creativity is presented that aims to be "mechanism independent" in the sense that it could equally well describe the behavior of simulated societies of computational agents or the embodied behavior of humans engaging in commons-based peer production. The idea of project-embedded problem solving is used as a connection between these two domains. Mathematical thinking provides an inspiring example, and the methodology that is proposed is an interpretation of the highly heterodox interdisciplinary theory of design patterns.

1 Introduction

From the perspective of creativity, licenses are a blunt instrument. Copyleft licenses (CC-By-SA, GNU GPL, etc.), non-copyleft licenses (CC-By, MIT, etc.), copyright waivers (CC-Zero), and the public domain support a vibrant culture of creative reappropriation. However, if the taxonomy of licenses is understood as our native theory of "free culture," then it seems we are collectively turning a blind eye to the all-important *creativity* that makes it all work. One thing that's not made particularly clear in the legal approach is that creative work has a lifecycle: at minimum, a beginning, middle, and an end. We need to pay more attention to understanding the intricacies of the "middle bit." In short, despite their important role as protectionary measures, licenses are not of great help in explaining or directly facilitating the genesis of cultural works.

Similarly, (general) economic behaviors, like gift exchange, potlach, or division of labor, can contribute incentives, norms, or boosts to efficiency, but they are typically second-order phenomena, since they assume that there is something there to exchange or destroy, or that there is a process to subdivide and enact. Of course, creativity very often develops at the social, symbolic, or economic level, but this is all the more reason not to take these phenomena as fundamental *explicata* for creative culture. In addition, as Shaw and Hill [20] make clear, the particular economic form known as commons-based peer production is in no way a political or social panacea. On the one hand, "*many successful peer production projects exhibit strong inequalities of participation and deeply entrenched leadership*" – on the other, many free/open software projects never attract collaborators at all. As researcher and free software activist B. Mako Hill argued earlier, "*The benefits of collaboration become something to understand, support, and work towards, rather than something to take for granted*" [11].

This paper approaches that challenge by trying to understand creativity in a mechanism independent way. The hypothesis that computers can simulate humans is not directly useful for building a theoretical model of creativity. What is more interesting is a social-computational perspective that says that certain combinations of humans and computers can simulate relatively advanced computer systems. For example, from the point of view of any given user, a peer produced website might be said to roughly approximate an artificial intelligence with an agent-based architecture. Similarly, the history of mathematics might be thought of "as if" it were an analogous long-running computational process, carried out mostly on paper. The fact that mathematics actually developed in the minds of humans is, clearly, not to be neglected. Nevertheless, the complementary fact that mathematics can be communicated between humans means that it can exist, to some extent, *in charta* or *in silica*. Mathematical creativity can accordingly be thought about in a way which sets aside the question of whether it is carried out by humans, societies, or computers, and which focuses, instead, on the fundamental features of mathematical form. These typically include definitions, theorems, and proofs along with explanations, remarks, questions, and so forth. Mathematics is by no means the only domain that can be studied in this way. Mathematics is, rather, an inspiring and well-studied example, and one can point to practical advances in fields like automated theorem proving that are associated with a computational approach. This example helps to motivate the mechanism independent study of the social-computational dynamics that are involved in the creation of software and other cultural works.

2 Building a mechanism independent model of creativity

Embodiment and sociality are the antidote to "as if" thinking for humans and computers. Tim Ingold [12] makes a distinction between "as if" thinking conducted, for example, by expert systems, and in-the-world thinking carried out by humans. In addition to their crafty minds, one thing that individual humans definitely have, which computers generally do not, is a robust embodiment. Ingold classifies writing in the same "as if" category as expert systems, since written expression tends to be equally disembodied. Manual DeLanda [7] (from whom I've borrowed the idea of "mechanism independence") talks about a potential antidote to "as if" thinking in computer systems: if we can give the computer even a limited embodiment, putting it in touch with a virtual world – then it doesn't have to have everything worked out in advance. In the case of mathematical thinking, individual humans are in contact with a rich phenomenal world, in which they find themselves capable of drawing lines in sand or on slate, discourse with other humans, and visualization or embodied feeling-ones-way through complex mental projections. A minimum mechanism independent model of creativity should include *agents* and *phenomena*. In particular, agents exist simultaneously as part of one another's phenomenal world, and this is referred to as *sociality*.

Agents relate around works and acts. Works, skills, acts, and agents themselves can all be created. Property and ownership are some of the rules that govern the relationship of agents to works, and these rules can become arbitrarily complex. Thus, for example, CC-By-SA-NC says "You can do (almost) whatever you would like with this work – as long as you give me credit, share any derivative works under the same license – and don't sell them." However, one ought not assume that creativity itself takes place exclusively in the context of one big commons, for instance, a given license "pool" like that defined by CC-By-SA-NC. Rather, creativity takes place within working groups, families, homesteads, etc., and it may be understood to evolve within nested contexts at several scales. For instance, creative work done within a given kin group might be comprehensible by other members of their linguistic community, and may be shared at the level of expression within a cultural commons, even if the work itself remains the property of its creators [13]. Even works that are shared more broadly tend to be created, enjoyed, and reproduced in smaller local contexts. In his public talks, Creative Commons cofounder Lawrence Lessig often quotes the composer John Phillip Sousa: "*When I was a boy... in front of every house in the summer evenings you would find young people together singing the songs of the day or the old songs.*"

Mathematical thinking: decompose processes into theories and problem solving projects. Quite a bit of background knowledge typically goes into stating, formulating, understanding or proving any given theorem. To speak metaphorically, the relevant definitions and lemmas are something like a neural substrate, in which the new theorem is expressed as a thought. Individual proofs can be broken down into subtrees and formal proof steps, but mathematical insight is better studied from the informal perspective of problem solving heuristics. One classic heuristic is to look for simpler versions of a hard problem and to try to solve those. Relevant social heuristics include asking for help. At any given point in a mathematical thought process, there may be many problem solving projects taking place on different levels: for example, one may be simultaneously carrying a digit, factoring a polynomial, finding a root, and designing a mechanical structure. A given mathematical theory or culture will suggest relevant approaches. In general there can be multiple agents, goals, and approaches at work.

The more general case: emergent design patterns. If mathematics is the formal theory of patterns, it can be quite useful to compare it to a relatively *informal* heterodox theory of patterns coming from the field of design. First introduced by the architect Christopher Alexander [1, 3], the design pattern methodology spread from architecture to software [9], and later, to other fields, including public affairs [19] and education [4]. Alexander's patterns are presented in a tree-like structure called a *pattern language*, ordered in a top-down manner from large-scale to small-scale levels of application, with each pattern presented in terms of a *picture*, a *context* (including links to relevant larger patterns), the *problem* that the pattern addresses, the *solution*, a *diagram*, and *links to smaller patterns* [3, pp. x-xi]. A relatively convincing implementation of Alexander's idea of patterns as a "*living language*" [3, p. xvii] was realized with one of the earliest applications of wiki software developed by Ward Cunningham: the Portland Pattern Repository.[1] Taking the idea of a pattern language as a point of departure, the *Peeragogy Handbook* [18] includes a catalog of patterns for peer production and peer learning that the authors discovered while working on the book. The patterns are presented using a template that is closely aligned to the

[1] `http://c2.com/ppr/`

structure coming from Alexander (Table 1). Each pattern is essentially a short textual abstract, describing an active dimension of the Peeragogy project. This emphasis on the active aspect is made particularly clear through the "What's Next" facet, which concretely links the patterns in the *Handbook* to the current activities being conducted within the aegis of the Peeragogy project. Connecting the patterns to the project in this way helps to convey their "living" nature, and the pattern catalog evolves together with the project. The notion of pattern-finding as a process related to, but distinct from abstraction, is described by Richard Gabriel, who emphasizes that the "*patterns and the social process for applying them are designed to produce organic order through piecemeal growth*" [9, p. 31]. In its original form, this statement describes the generative use of patterns to create artifacts (buildings, object oriented programs, etc.). However, this criterion is applied here to the growth and development the pattern language itself.

> **Title**: Encapsulate the idea – possibly include a subtitle.
>
> **Definition**: Explain the idea and the context in which it is meaningful.
>
> **Problem**: Explain why there's some issue to address here.
>
> **Solution**: Talk about an idea about how to address the issue.
>
> **Challenges**: Talk about what can go wrong.
>
> **What's Next**: Talk about specific next steps.
>
> *Illustrations, and the following, are optional:*
>
> [**Objectives**: Explain the purpose(s) of the proposed solution's functioning, if not fully specified by the description of the solution itself.]
>
> [**Examples**: Present example(s) that have been encountered, if this aids comprehension.]
>
> [**References**: Citations, if relevant.]

Table 1: Pattern template from the *Peeragogy Handbook*

Example: emergent design for peer learning. Patterns offer a heuristic but nonetheless relatively rigorous way to do ongoing "emergent" design. Some of the central patterns described in the *Peeragogy Handbook* indicate that peer learning projects often organize around a temporal *Heartbeat* of meetings, divide the work among several *Roles*, and have to decide what, and whether, to *Use or make* (often remixing earlier works). We've observed that even with a high-level *Roadmap* in mind, participants tend to drift until they pick *A Specific Project* and get down to work. With these and other patterns in mind, the analyst has a simple language for talking about what works and what doesn't work in the online peer learning context. The process of analysis is rigorous in the sense that either an existing pattern matches the topics under discussion, or new patterns should emerge. For example, in [6], the Peeragogy pattern language was applied to study user feedback about a new system that was developed for the mathematics website PlanetMath.org, resulting in three new patterns (*Minimum Viable Project*, *Frontend and Backend*, and *Spanning Set*) that expanded on or clarified earlier patterns. Each of these has its own "What's Next" within the context of a specific project.[2] The *Peeragogy Handbook* collects a dozen patterns and a half-dozen antipatterns (using the same template but emphasizing "Challenges") composed after the fashion described here. As an initial set of patterns describing peer production and peer learning, this catalog forms a candidate description of a creativity commons (Table 2, with details at `http://peeragogy.org/practice`). There are hundreds of other patterns that have been written over the years, but generally without the project-embedded focus adopted here – which puts this work in comparatively direct contact with phenomenal experience.

[2] For illustrative purposes, the *Spanning Set* pattern is reproduced in Table 3 in the Annex to this article.

Patterns

Wrapper – Front end appearance to participants. Consolidate and summarize.

Discerning a pattern – Found a pattern? Give it a title and record an example. (Woah, meta!)

Polling for ideas – Invite brainstorming, collecting ideas, questions, and solutions.

Creating a guide – Overviews expose the lay of the land. Collecting content and stories.

Newcomer – Welcoming "beginner's mind."

Roadmap – Plans for future work, direction towards a goal, dynamic.

Roles – Specialize and mix it up. Play to participants strengths and skills.

A Specific Project – Lightbulb moment: Most specific projects involve learning!

Carrying capacity – Know your limits, find ways to get other people involved.

Heartbeat – The "heartbeat" of the group keeps energy flowing.

Moderation – When leaders step back, dynamics can improve; moderator serves as champion and editor.

Use or make? – Repurposing, tinkering, or creating from scratch?

Anti-patterns

Isolation - A tale of silos, holes, and not-invented-here.

Magical thinking – "One meeting will (not) change everything!"

Messy with Lurkers – What happens when joining is low-cost and completion is low-benefit.

Misunderstanding Power – The workload is almost never evenly distributed.

Navel Gazing – "I have this really great idea. . . "

Stasis – What drives collaborative projects? (Whatever it is, it doesn't always work.)

Stuck at the level of weak ties – Can we deepen the connection?

Table 2: Catalog of patterns from the *Peeragogy Handbook*

3 Outlook and next steps

Heuristic methods. As Alexander [2] indicates, it will be a challenge for software developers, architects, and others – educators and researchers for example – who are inspired by the design pattern methodology to work together to transform the lived environment. An especial challenge is to find a way to work across different fields and different levels of analysis, "*integrating different types of theories and concepts in such a way that one can formulate meaningful propositions embodying findings now sequestered in separate fields of study*" [10, p. 44]. Although it makes certain epistemological and ontological assumptions that may not be appropriate in every conceivable case (regarding, in particular, the possibility of a "problematic" understanding of reality) – the adapted design pattern methodology described here should nevertheless be widely applicable as an informal dialectical method; cf. [8, p. 203]. The Peeragogy project will continue to develop its catalog of design patterns in practice based on active participation, with a new quarterly peer produced development course beginning in June, 2014, featuring rolling enrollment.[3] Participants will be invited to develop new candidate patterns starting from a simple reporting protocol (adapted from [23]):

> *(1) Review what was supposed to happen. (2) Establish what is happening. (3) Determine what's right and wrong with what we are doing. (4) What did we learn or change? (5) What else should we change going forward?*

Computational methods. Keeping in mind the core aim of mechanism independence, a parallel research agenda involves implementing a computational system that can work with design patterns using a society-of-mind style architecture [14]. Although design patterns have been used extensively to guide programming, there are still many challenges associated with moving from an informal heuristic modeling approach to a formal computational model. Related work using artificial agents in a narrative reasoning context was discussed in [22], with a detailed development in [21]. This work emphasized reflexive aspects of mentation and multiple ways to think about problems [15]. The orientation described here suggests a more cultural and less psychological tack. For instance, microtheories may be useful for describing local cultural contexts, and this resonates with the Ostrom-inspired view on the commons [17]. Another key question to examine is the way different local contexts link together. Nowak takes an evolutionary game theory approach to this question [16]. Here, one would rather consider hybrid cultures and blended theories: and the prospect for referring this work to ongoing research on *computational blending* looks good. Mathematics continues to be a motivating example, and the first order of business will be to develop an extension of the pattern catalog oriented to *mathematical* creativity.

4 Conclusion

As a collection of design patterns focused on peer learning and peer production projects, the Peeragogy pattern catalog can begin to be used right now as a heuristic theory of social creativity. It is simultaneously rigorous and anti-fragile: individual patterns need to be negotiated and reasoned through relative to the existing catalog, while the catalog as a whole will improve with discussion and debate. As a system connected to practice within the Peeragogy project, the pattern catalog is very much subject to change and further development. The individual "What's Next" steps serve as a distributed *Roadmap* for the Peeragogy project – and as the work done in the context of PlanetMath shows, they may be useful as a "living language" spread across several peer production sites. Although the computational system described above remains projected and conjectural, the design sketch usefully expands on an earlier discussion of multi-agent architectures for computational creativity, hinted at in [5]. This earlier article focused on critical assessments of creative progress. Here, creative progress is viewed as connected to formulating and working on problems, or, rather more profoundly, asking how a society "*is able to pose the problems set within it and to it by the differential relations it incarnates*" [8, p. 235]. The problem posed here, in a necessarily preliminary manner, asks how we can understand peer production in a computational way.

5 Acknowledgements

This work has been supported through EC funding for the project COINVENT 611553 by FP7. I would like to thank Helene Finidori for helpful conversations.

[3] `http://commonsabundance.net/groups/peeragogy/`

References

[1] Christopher Alexander. *The timeless way of building*. Oxford University Press, 1979.

[2] Christopher Alexander. "The origins of pattern theory: The future of the theory, and the generation of a living world". In: *Software, IEEE* 16.5 (1999), pp. 71–82.

[3] Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.

[4] Joseph Bergin et al. *Pedagogical Patterns: Advice for Educators*. Joseph Bergin Software Tools, 2012.

[5] Simon Colton et al. "Assessing Progress in Building Autonomously Creative Systems". In: *Proceedings of the Fifth International Conference on Computational Creativity*. Ed. by Dan Ventura et al. 2014.

[6] Joseph Corneli. "Peer produced peer learning: A mathematics case study". PhD thesis. The Open University (examined), 2014.

[7] Manuel DeLanda. *Philosophy and simulation: the emergence of synthetic reason*. Continuum, 2011.

[8] Gilles Deleuze. *Difference and repetition*. (trans. 2004, Paul Patton, London: Bloomsbury Academic). 1968.

[9] Richard P. Gabriel. *Patterns of Software*. Oxford University Press New York, 1996.

[10] Clifford Geertz. *The interpretation of cultures: Selected essays*. Basic Books (AZ), 1973.

[11] Benjamin Mako Hill. *When Free Software Isn't (Practically) Better*. Licensed via CC-By-SA. 2011. URL: `http://www.gnu.org/philosophy/when_free_software_isnt_practically_better.html`.

[12] Tim Ingold. "Technology, language, intelligence: a reconsideration of basic concepts". In: *Tools, language and cognition in human evolution* (1993), pp. 449–472.

[13] James Leach. "Modes of Creativity and the Register of Ownership". In: *CODE: Collaborative ownership and the digital economy* 29 (2005), p. 38.

[14] Marvin Minsky. *The Society of Mind*. New York: Simon and Schuster, 1988.

[15] Marvin Minsky. *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon & Schuster, 2007.

[16] Martin A. Nowak. "Five rules for the evolution of cooperation". In: *Science* 314.5805 (2006), pp. 1560–1563.

[17] Elinor Ostrom et al. "Revisiting the commons: local lessons, global challenges". In: *Science* 284.5412 (1999), pp. 278–282.

[18] Howard Rheingold et al. *The Peeragogy Handbook*. Arlington, MA: Pierce Press and Chicago: PubDomEd Press. Published with a CC-Zero copyright waiver. 2014. URL: `http://peeragogy.org`.

[19] Douglas Schuler. *Liberating voices: A pattern language for communication revolution*. MIT Press, 2008.

[20] Aaron Shaw and Benjamin Mako Hill. "Laboratories of Oligarchy?: How the iron law extends to peer production". In: *Journal of Communication* (forthcoming).

[21] Push Singh. "EM-ONE: an architecture for reflective commonsense thinking". PhD thesis. Massachusetts Institute of Technology, 2005.

[22] Push Singh and Marvin Minsky. "An architecture for cognitive diversity". In: *Visions of mind: architectures for cognition and affect* (2005), pp. 312–331.

[23] US Army. *Training the Force (Field Manual No. 7-0)*. Oct. 2002.

Annex

Spanning Set: Follow the paths in the grass

Definition: With a well-constructed information access system, you may be able to get what you need without digging. If you do need to dig, it is very good to get some indication about which direction to dig in. At the level of content, this may be achieved by using high-level "topic articles" as a narrative map to the content. In general, the Spanning Set may include people as well as less dynamic media objects. In general, a spanning set is comprised of a set of fundamental actions (e.g. asking a question) and fundamental relationships between resources. In the case of media objects, it may reduce to the pattern *Creating a Guide*, but the guide may need to be updated regularly to remain relevant.

Problem: People need to know what can be done with a given resource, and this isn't always obvious. Relying on a single knowledgeable guru figure isn't always possible.

Solution: A spanning set of a system's features, categories, and relations can be comprised of many different kinds of components: for example, a "start menu" or pop-up window showing keyboard shortcuts that shows what can be done with a given tool; a schedule of office hours so that people know how to find help; and topic-level narrative guides to content.

Examples: One social version of a Spanning Set is the classical master/apprentice system, in which every apprentice is supervised by a certified master. In a standard course model, there is one central node, the teacher, who is responsible for all teaching and course communication and answering student questions. In large courses, this model is sometimes scaled up through tutors and TAs. In a large MOOC, a "spanning set" of peer tutors could help give everyone personal attention. In Q&A sites, these roles are fully disaggregated and distributed.

Challenges: From the perspective of *Frontend and Backend*, principles and features are visible as part of a system's "frontend" – but the actual spanning set of relevant behaviors tends to be emergent. If any individual tries to span too much, they will get spread too thin; see *Carrying Capacity*.

What's Next: As a project with an encyclopedic component, PlanetMath can be used to span and organize a significantly larger body of existing material. We have come up with a high-level design for a "cross-index" to the mathematics literature. We're working on a prototype for Calculus.

Table 3: The *Spanning Set* pattern