Mere Generation: Essential Barometer or Dated Concept?

Dan Ventura
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
ventura@cs.byu.edu

Abstract

The computational creativity community (rightfully) takes a dim view of supposedly creative systems that operate by *mere generation*. However, what exactly this means has never been adequately defined, and therefore the idea of requiring systems to exceed this standard is problematic. Here, we revisit the question of mere generation and attempt to qualitatively identify what constitutes exceeding this threshold. This exercise leads to the conclusion that the question is likely no longer relevant for the field and that a failure to recognize this is likely detrimental to its future health.

Introduction

For many of us in the computational creativity community the idea of artifact generation, in and of itself, has come to be looked on as something less than an accomplishment, even though in many of the domains in which we operate, the generation of even just reasonable artifacts is still well beyond the capabilities of any current system. Indeed, the expression *mere generation* has become something of a favored pejorative whose history reaches back at least to the meeting of the *Third International Conference on Computational Creativity* held in Dublin in 2012, during which the tagline, "Scoffing at Mere Generation for more than a Decade"[1], became a conference theme. This theme was explicitly revisited during the 2015 meeting of the conference in Park City, at which small buttons showing the expression "mere generation" struck through were included in the registration packets (see Fig. 1). Many of the conference attendees delightedly wore the buttons, but others at the event, especially those that may not have attended earlier conferences, were less enamored with or bemused by them, and may possibly have found them offensive.

It became clear that though at least some of us have been endorsing this dogma of disdain for many years, it is perhaps not as self-evident as we might think that it is. The purpose of this paper is to suggest that the idea of "mere generation" needs to be revisited by the community, that at the very least, we should clarify what is meant by the expression, and that, in fact, its use with respect to modern systems claiming correspondence with the field of computational creativity should probably be deprecated.

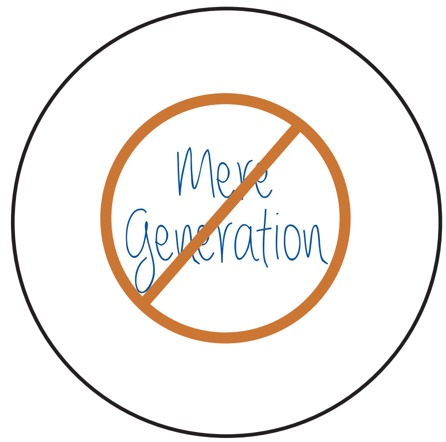

Figure 1: The button handed out in Park City at the *Sixth International Conference on Computational Creativity*, 2015. But, what does this mean, exactly?

This question of when a system has crossed the line from mere generation to something more is related to the question of system evaluation, which has begun to be addressed by work such as Ritchie's metrics (Ritchie 2007), Colton et al.'s FACE/IDEA framework (Colton, Charnley, and Pease 2011) and Jordanous' SPECS methodology (Jordanous 2012)[2]. The goal of these types of approaches—and it is a critical one—is to suggest viable ways of measuring, either absolutely or relatively the "creativity level" of (some aspect of) a system. In contrast, the purpose of this paper is to argue that, by any reasonable measure, we, as a field, have at some point crossed an important threshold on our quest for computationally creative systems.

The approach we will follow here is reminiscent of a *gedanken* experiment inspired by Ritchie's metrics (Ventura 2008)—we are going to examine a spectrum of candidate (computationally) creative processes, from *definitely-mere-generation* to *definitely-not-mere-generation*, characterized

[1] Coined by the host of that conference, the inimitable Tony Veale

[2] And even a recent blogpost at `http://www.bestofbotworlds.com/node/22`

by several prototypical algorithms that fall at different points along that spectrum. As we traverse the spectrum, we will at each point consider the existence (or lack thereof) of three characteristics: *novelty*, *value* and *intentionality*, as surrogate indicators for the existence of (some form of) creativity. Note that the first two are most commonly addressed with respect to product[3], while the third deals with process. For our purposes, these characteristics will be defined as follows[4]:

novelty: the quality of being new, original or unusual; this is relative to the population of artifacts in the domain in question and can apply in the personal or historical sense.

value: the importance, worth or usefulness of something; this would typically be ascribed by practitioners of the domain in question.

intentionality: the fact of being deliberative or purposive; that is, the output of the system is the result of the system having a goal or objective—the system's product is correlated with its process.

The prototype algorithms used to populate our spectrum are meant to be representative of the breadth of approaches under consideration by the field rather than definitive or exhaustive and are abstract enough that it is likely that the majority of historical and extant CC algorithms can be, to a fairly accurate approximation, typified by one of them or by some "convex combination" of a couple of them. However, we will not attempt to support that hypothesis here.

Instead, we will argue that the field of computational creativity as a whole has moved well past a critical threshold and that the derisory merely generative system of yore is nearly impossible to find amongst the systems that we see today, certainly those built by people conversant in the field and, in many instances, by those who are not (yet). So, we can still enjoy our scoffing, albeit without any real targets at which to direct our ridicule. This is not to suggest that our quest for computational creativity is complete—far from it—but we can say with conviction that we have thrown off our moorings and left port.

A Generation Odyssey

As a running rhetorical example, we will address the problem of generating artifacts from the Japanese poetic form *haiku*, perhaps the most famous example of which is the following by Bashō[5]

Furu ike ya
kawazu tobikomu
mizu no oto

[3]But also note that our use of "product" here is abstract, and that, in particular, the artifact produced might itself be a process.

[4]These definitions can be formalized, but for this discussion that will not be useful.

[5]A well-known translation, due to Donald Keene, that both faithfully captures the literal meaning and yet (necessarily) loses a great deal of the impact goes as follows:
The ancient pond
A frog leaps in
The sound of water

Algorithm 1 *Generation using a stochastic process.*

```
Create()
  a = {}
  while not done do
    a = a + random_atom()
  return a
```

Traditional haiku is a simple, elegant poetic form used for juxtaposing two ideas or images. It is characterized by a single-stanza structure with 17 total syllables, divided into three phrases with syllable counts 5-7-5. Themes that speak to the Japanese reverence for nature dominate the form; syntax is somewhat loose, deferring to structure; and implied semantics are favored over the explicit. We have chosen haiku because its realizations are small enough to allow analysis and demonstration of multiple examples while being complex enough to admit treatment of a range of important issues. From here on, when an example is useful, for understandability, we will sacrifice the purity of the original Japanese form and use English.

Randomization

It is difficult to conceive of a simpler form of generation than that of a stochastic process, so that is where we will begin. The first level of generation, then, consists only of producing a set of atomic elements, as shown in Algorithm 1. For the generation of *haiku*, this means simply generating some number of words and stopping, without regard for syllable count, line count or syntax (which is often not a huge concern in haiku anyway), let alone semantic cohesion or theme. An actual example, which was generated using a simple web app[6], is shown below[7].

sadistic ideal adopter
devil seducer diametric
accursed blabbermouth

Similarly meaningless output in other domains is easily conceived: a collage of random shapes (or even an image composed of random pixels), a musical composition of random notes (pitches and durations), a recipe composed of a random list of ingredients (and amounts), a neologism as a random sequence of letters, etc. There should be little question whether this system is merely generative; indeed, some may argue it doesn't even rise to that aspiration.

The output of such a stochastic system will almost certainly be novel, in the sense that as the size of the artifact increases, the likelihood that it has been generated before by any system (computational or otherwise) becomes vanishingly small. However, this novelty is not intentional—the system is randomly choosing an artifact without any notion of novelty. At the same time, system output will almost

[6]`http://www.textfixer.com/tools/random-words.php`

[7]Produced by randomly selecting a (random) number of words. Line breaks were chosen arbitrarily for formatting purposes and are not part of the artifact.

Algorithm 2 *Generation by plagiarizing an inspiring set* I.

Create(I)

  $a = \texttt{random\_select}(I)$

  return a

certainly not be valuable, by the same kind of probabilistic argument—as the size of the artifact increases, the likelihood that it has any meaning/utility/aesthetic quality becomes equally small. Another way to say the same thing: as a function of artifact size, the set of all possible candidate artifacts grows very quickly and, in particular, it grows much more quickly than does the set of valuable candidate artifacts.

Plagiarization

A significant improvement in artifact quality can be achieved by making use of an inspiring set that contains examples of quality artifacts, and the simplest use that can be made of such a set is blatant plagiarism. An abstract plagiarizing system is shown in Algorithm 2, and the output of this kind of system is significantly improved over that of Algorithm 1 in the sense that it will be real haiku, *per se*. It also has another advantage over Algorithm 1 in that it has acquired some knowledge of what haiku is. This is knowledge only in the most rudimentary sense: functionally, the system "knows" only that haiku is anything in its inspiring set and anything not in its inspiring set is not haiku. Of course, in a third sense, this system represents a step backwards from that of Algorithm 1 because it has no autonomy (the one thing that Algorithm 1 has going for it)—it cannot generate anything novel. Assuming that haiku on the web are all examples of quality artifacts (a patently ridiculous assumption that does not affect our current argument), such a plagiarizing system can be constructed by employing a simple Google search for haiku and choosing randomly from amongst the returned results. One such search, for example, turns up the following[8]:

A cricket disturbed
the sleeping child; on the porch
a man smoked and smiled

Similarly simple systems can easily be built for images, music, recipes, etc.

The output of a plagiarizing system reveals a character complementary to that of the stochastic: it will by definition not be novel, and, again by definition it will be valuable. Like the novelty of the stochastic system, though, this value lacks intention—the system is regurgitating an artifact without any notion of value (beyond that of implicitly ascribing value to the inspiring set). This time, the set of candidate artifacts is fixed and is a strict subset of the set of all valuable artifacts.

[8]Found at `http://examples.yourdictionary.com/examples-of-haiku-poems.html`

Algorithm 3 *Generation by memorizing an inspiring set* I.

Create(I)

  $model = \texttt{memorize}(I)$

  $a = \texttt{random\_from\_memory}(model)$

  return a

Memorization

A slightly more sophisticated version of Algorithm 2 builds a model of the inspiring set by memorization (see Algorithm 3). While the lookup-table approach of Algorithm 2 could be considered a form of memorization, what is meant here is that the inspiring set is re-represented in some way by the system, ideally without loss of fidelity. This is the first level at which building the system is not trivial (given commonly available resources)—a model that memorizes typically does so by overfitting data (that is, the parameters of the model are under-constrained by the data), so an inspiring set of interesting size will require a very powerful model for its memorization. As a result of this memorization, the system has "internalized" the inspiring set in a nontrivial way, though without learning any generalizing principles. If the memorization is perfect, the result is likely indistinguishable from regurgitation—any generated artifact will be a faithful copy of a member of the inspiring set, even though that set has been re-represented. However, this memorization process at least admits the possibility of some level of variance from the inspiring set, due to faulty memory, model capacity or fidelity issues, etc. For example, if the memorization process involves some form of compression, it is possible that the compression will not be lossless, resulting in errors during reproduction[9]:

A cricket disturbed
the creeping mold; on the porch
a man choked and died

These errors may, in fact, be thought of as features rather than bugs, and even packaged as a very simple form of creativity[10]; however, the system realistically only has the same level of "knowledge" of haiku as that of Algorithm 2. Indeed, because the system's goal is memorization, any errors in reproduction that it makes would likely go undetected (by the system itself) and it has no mechanism for evaluating the quality of a perturbation—the detrimental norm will not be distinguishable from the serendipitous exception (and in this sense, we have returned to the lack of knowledge exhibited by systems like Algorithm 1).

Because a memorization system is attempting to mimic the output of a plagiarizing system, the artifacts it produces will essentially be characterized the same way: value with lack of novelty, again without intention. When errors are introduced, the two characteristics are inversely affected: novelty likely increases while value likely decreases, for the

[9]A serendipitous, if morbid, perturbation of the cricket haiku done by the author

[10]Though doing so would likely be construed as hucksterism by our community

Algorithm 4 *Generation by generalizing an inspiring set* I.

Create(I)
  $model$ = `build_model`(I)
  a = `generalize_from_model`($model$)
  return a

same reasons given in discussing stochastic systems. Intention is unaffected by error as the system has no mechanism for evaluation. This time, the set of candidate artifacts intersects the set of all valuable artifacts, with the size of the intersection dependent on the fidelity of the memorization (assumed to be high).

Generalization

Another step along the spectrum regains some of the autonomy that was lost with the introduction of an inspiring set. Algorithm 4 shows a system that goes beyond memorization by modeling the inspiring set in such a way that generalization is possible. This is typically accomplished by some form of regularization of the model combined with a bias (either implicit or explicit). The resulting model output can demonstrate significant variance from the inspiring set, and the trick for producing reasonable output is to discover the right amount of regularization and the right bias, both non-trivial propositions. In the case of haiku, regularization might force the model to represent words as abstract entities, such as parts of speech, and a bias might favor poems with a syllable count of (or near) 17 and/or words related to nature. As an example, such a simple generalizing model could, given an inspiring set similar to the cricket haiku above, produce something like[11]

The snowflake reveals
a quiet rock near a tree
a fish blows or falls

Note that the model itself and either or both the regularization and the bias may be learned or explicitly designed, and any or all of these may be interpretable or they may not. In any case, a generalizing system must be acknowledged to have a significantly deeper knowledge about haiku than any of its three predecessors, even if that knowledge is still naïve or even somewhat incorrect. At this level we may begin to see the natural introduction of pastiche, if the model is particularly good.

Here for the first time, we begin to see artifacts that may non-vacuously exhibit both novelty and value. Novelty will be limited to the degree that the model makes explicit use of constructs found in the inspiring set. Value will be limited because any valid generalization of the inspiring set may be output. The set of candidate artifacts has increased significantly over that available to the plagiarizing and memorizing models, but is much smaller than that available to the stochastic. For the first time, the system can be said to have at least some limited (implicit) intentionality in both the novelty and value it produces: the model regularization enforces some level of generalization (and thus novelty) from the inspiring set by disallowing too much complexity; and the bias (can) enforce some notions of value.

Algorithm 5 *Generation by modeling an inspiring set* I *and filtering candidate artifacts via a fitness function* `fitness`().

Create(I, `fitness`())
  $model$ = `build_model`(I)
  while $score < \theta$ **do**
    a = `generate`($model$)
    $score$ = `fitness`(a)
  return a

Filtration

Moving farther along the spectrum, we see the first evidence of self-evaluation, in the form of an objective or fitness function. Algorithm 5 extends Algorithm 4 by filtering its generative results, using some notion of fitness. The model now may be designed for some other purpose than (just) generalization; the modeling step can now afford to "take more risk", because the generated artifacts are vetted after the generative step. In order to be useful, the fitness function should evaluate aspects of the model not already implicitly managed by the generalizing model. For example, if the model includes a bias for 17-syllable stanzas, it is likely redundant for the the fitness function to compute a score for syllable count. Instead, the fitness function will be most useful for measuring holistic characteristics of the artifact. In the case of haiku, these might include notions such as overall valence or affect of the stanza, semantic relationships among constituent words, novelty, etc., and several of them could be composed in some way to compute the fitness score. The use of such a filter would likely preclude the creation of the snowflake haiku (which was generalized from the cricket haiku) because it would score poorly for semantic cohesion, and as a result probably low in (at least) affect as well. However, another haiku generalized by the same model could score significantly better, passing the fitness threshold and therefore being output as a viable artifact[12]:

The sunlight reveals
a quiet path near a brook
a tree drinks or sleeps

Use of a filtering function consequent to a generative step can be thought of (somewhat simplistically) as analogous to a musician listening to her composition after writing it or a chef tasting a dish after he conceived the recipe for it (we will see a better approximation to this in systems further along the spectrum).

Just as with the generalizing model, the filtering model can produce both novelty and value, and for the same rea-

[11] Created by the author using a part-of-speech generalization of the cricket haiku, and strong bias for 17 syllables and selection of nature-related words of the requisite parts of speech

[12] Also created by the author using the same generalization model as the previous section, with serendipitous word choices that increase the semantic cohesion and affect

Algorithm 6 *Knowledge-based generation by modeling an inspiring set* I*, employing a fitness function* `fitness()` *and leveraging a knowledge base* K.

Create(I, `fitness()`, K)
  $model$ = `build_model`(I)
  while $score < \theta$ **do**
    a = `generate`($model$, K)
    $score$ = `fitness`(a)
  return a

sons. However, both the value and the novelty are likely to be increased because for the first time we see explicit intention in the form of the fitness function. Further, since the fitness function can, at least notionally, address both value (by filtering for semantics, affect, etc.) and novelty (by filtering using some form of distance from inspiring set), both characteristics can be said for the first time to be intentional. This is a significant milestone.

Inception

Yet another level of generation is attained with the addition of a knowledge-base, which is used to affect/augment the model, consequently injecting additional depth and nuance into the generalization process, thereby allowing the fitness threshold to be increased, and leading to better artifacts. In Algorithm 6, the knowledge-base is incorporated solely into the generative step, but variations can include it in the modeling step and/or the fitness evaluation as well. It can be very general or domain specific. In the case of haiku, useful domain knowledge would include such things as semantic relationships amongst words, alternative grammatical constructs and exceptions, common facts, metaphor, etc.

Such a system might produce a variation on the tree haiku like the following[13]:

In golden torpor
while insects hum over a rill
an old oak dozes

In the excellent movie *Inception*, Leonardo DiCaprio and his team are given the task of infiltrating a man's mind, while he is in an induced dream-state, in order to implant an idea. The tricky part is that for the idea to germinate, the man must not realize that it has been implanted but must instead believe that it originated with himself. To avoid detection, the team induces a dream-within-a-dream-within-a-dream scenario, obfuscating their presence by constructing multiple levels of indirection. At least as an end game, the CC community faces a similar challenge—how to inject knowledge into a computationally creative system without leaving the injector's fingerprints all over the resulting artifacts. We leave this as a challenge for the future.

Since the knowledge-based model builds on the filtering model, both intentional novelty and intentional value can be found here as well, and, making use of the additional knowledge now available, that intention can be more nuanced, resulting in a concomitant increase in value (and possibly in novelty as well).

[13]Again created by the author by making use of synonomy and other relational semantics, metaphor, and grammatical variation to modify the tree haiku

Algorithm 7 *Creative generation by modeling an inspiring set* I *and leveraging a knowledge base* K *followed by evaluating the perception of the generated artifact.*

Create(I, K)
  $model$ = `build_model`(I)
  while $score < \theta$ **do**
    a = `generate`($model$, K)
    $score$ = `evaluate`(`perceive`(a))
  return a

Creation

The final stop on our journey abstracts the system's evaluation mechanism and introduces perceptual ability, as shown in Algorithm 7. This new ability means the system can ground concepts perceptually, giving it at least a rudimentary ability to understand the world. Leveraging this understanding leads to additional improvement in results. The most obvious perceptual abilities that might be incorporated into such a system include vision, audition, chemical analysis (smell/taste) and touch. With these, a system can look at the haiku as well as listen to it being read aloud, allowing the evaluation of factors such as visual appearance, prosody, (both visual and aural) flow, etc.

Just these basic perceptual abilities have the potential to significantly improve results, but there is no reason that computational systems need be limited to just these. Additional derived and invented perceptual capabilities can be conceived, including other types of signal processing (radiation, atmospheric pressure, network flow), and abstract percepts such as the detection of affective and social cues, etc.

Here is a nice English haiku[14] that cleverly plays on Bashō's famous poem and that could notionally be created by such a system:

By an ancient pond
a bullfrog sits on a rock
waiting for Bashō

Intentional novelty and value are featured here as well, but with the advantage that the intentionality is now perceptually grounded. The benefit of this should be evident: grounding allows the possibility of natural cross-domain creativity (write a haiku that describes what silence looks like), and it improves the possibility of mutual comprehension (assuming shared percepts).

An intentional detour

Before ending our expedition and considering what we may have learned, we must first discuss a somewhat orthogonal but important concern about how intention might be

[14]Written by Scott Alexander

Algorithm 8 *Creating haiku through random generation filtered by a fitness function,* `Fitness()`*, which returns a score that is computed as a convex combination of feature values that measure the goodness of the artifact along the characteristic dimensions of syllable count, line count, theme and semantics.*

```
Haiku()
    while score < θ do
        a = generate()
        score = Fitness(a)
    return a
Fitness(a)
    y = syllable_count(a)
    l = line_count(a)
    t = theme(a)
    s = semantics(a)
    return α_y y + α_l l + α_t t + α_s s
```

"located" in a CC system. Consider the approaches of Algorithms 8 and 9 for creating haiku. The first is a pure generate and test procedure, albeit with a (postulated) sophisticated test mechanism. The second is an iterative, controlled generative procedure[15]. In what ways do they differ? One difference might be temporal, as Algorithm 8 may take significant time to produce an artifact whose fitness is above threshold. On the other hand, this approach may be capable of generating haiku that Algorithm 9 can not, because its generation process is not limited in any way. However, given enough time for Algorithm 8 and enough breadth of theme and vocabulary for Algorithm 9, one might argue that they are equivalent in their potential observable behavior (that is, in the set of artifacts that they can generate). Further, both approaches employ the same domain knowledge about haiku (structure, theme, semantics, etc.).

The real difference between the two is in *where* that knowledge is leveraged. In Algorithm 8, the knowledge is used as a *post hoc* filtering mechanism. In Algorithm 9, the knowledge is used to restrict the generation process *in situ*. The question is, *are these approaches fundamentally different in their creative ability*? Also, note that the question is no closer to resolution if one considers the meta-creative case in which the system may change its domain knowledge/summative criteria through learning, interaction, environmental effects, etc.—such changes could be effected in either the fitness function or in the generative process[16].

Another, related, difference between the two is in what

[15]Note that in both cases, the summative characteristics/domain knowledge shown (structure, thematic range, semantics) is meant to be representative only; the idea is that any and all such knowledge would be incorporated into both algorithms, either as a part of the fitness function or as a part of the generative process, respectively.

[16]*How* these changes might be effected is another question; at first blush, it seems that perhaps self-modification of the fitness function could be significantly easier than self-modification of the generative process, but that might be a consequence only of the way those two constructs have been rendered here.

Algorithm 9 *Creating haiku through an iterative process of first choosing a theme, then choosing theme-appropriate candidate words, then selecting some subset of those words (along with helper words) that contains 17 syllables and can be broken into three lines, then ordering the words to convey an acceptable level of semantics.*

```
Haiku()
    while not done do
        theme = choose_theme()
        wordset = find_words(theme)
        while not 17 syllables in three lines do
            words = select_words(wordset)
            while unacceptable semantics do
                a = reorder(words)
    return a
```

the system can explain: the filtering system of Algorithm 8 can explain *why* the artifact is novel and has value, but it can not give a satisfactory account of how the artifact was produced; the generative system of Algorithm 9 can to some approximation explain not only the novelty and value of its output but also the reason it was generated. Interestingly, there exist human creators of both ilks as well: those that are method-conscience and those that are not.

Where in the World are We?

The journey of generation that we have just taken is illustrated in Figure 2, with the stochastic system defining one extreme of the spectrum, while the other is left undefined. The ordering of the various approaches and the relative spac-

Figure 2: A spectrum of generative systems. The threshold beyond which our systems are no longer merely generative lies somewhere, but it is not clear where. Even more concerning, as our systems become more sophisticated and we progress farther afield, we as a community may continue to insist that the threshold is just beyond our currently charted territory. (Original artwork courtesy of Krey Ventura.)

ing between them may be debatable, and the exact placement of a particular system on the spectrum may be unclear, but the general picture is accurate to some approximation sufficient for the current discussion. Given this, we can now address two critical questions:

1. *Where is the threshold*—how far along the spectrum must one go in order to be safe from the label mere generation?
2. *Where are we*—where on the spectrum is a typical modern computationally creative system?

The edge of the world

What exactly must a system do (or what characteristics must it possess) to avoid being damned as mere generation? One might argue that the question doesn't really matter, because the expression is just meant as a catchy maxim that articulates a philosophy we as a field profess; however, the question actually matters a great deal, because we are often guilty of employing this philosophical tenet as a concrete measuring stick, with the common result that a system fails to measure up. This is problematic because we can't then concretely say why it fails to meet the standard—it fails simply because it is merely generative.

As a way to instigate a discussion on the matter, we offer several "lines in the sand" which, if achieved, could be considered sufficient to show that a system can no longer be considered merely generative:

1. it can be demonstrated to possess any knowledge whatsoever
2. it can be demonstrated to possess knowledge that it has had some hand in structuring/acquiring
3. it can be demonstrated that it has some reasonable chance of producing both novelty and value
4. it can be demonstrated that it has some reasonable chance of producing both novelty and value and at least one of these is intentional
5. it can be demonstrated that it has some reasonable chance of producing both novelty and value and both of these are intentional

These candidate thresholds are ordered by increasing level of demand, and they correspond roughly to demarcating mere generation as solely randomization up to and including generalization. A reasonable argument can likely be made for any of these, and we argue that anything more demanding will exclude real computational creativity. We further argue that the line should be drawn to be as inclusive as possible, while respecting the spirit of rejecting mere generation.

A related concern is the potential for an insidious shifting of this threshold over time—as increasingly sophisticated and accomplished systems are developed, the threshold is continuously shifted beyond their reach—not by the lay person, but *by the community itself*. This can be argued to be a natural consequence of and even stimulus for progress; however, it is at least as likely that the effect is instead a depression of growth—without some magical talisman unattainable by mere mortals, no matter how far afield we sail, we will never see the shores of Valinor[17].

In the past, when the bounds of the world were not yet understood, it was not uncommon for seafarers to fear traveling into the unknown. While we as a community do not fear going where no one has gone before, it is possible that we are overly tentative about admitting that perhaps we already have. And, in fact, we have, not in the exceptional case at this point, but in the common one. Certainly, we aren't where we want to be yet, but we need to own the fact that we, at least, are not in port any longer. We have sailed beyond the edge of the map.

Triangulating our position

In other words, by any reasonable measure, we as a community (taking that term to include many researchers and systems that have not yet participated in a titular CC event and may not even be aware that the field exists) have moved *en masse* beyond the threshold of mere generation.

As an example, consider the soon-to-be-released *No Man's Sky*[18] being developed by Hello Games. It is being touted as a science fiction exploration game set in a vast, open universe created entirely by procedural generation (see Figure 3). Early press has seemed positive and previews look pretty spectacular. If it is, in fact, as large as it claims to be[19] and is purely procedurally generated as claimed (and how could it be anything else at that scale), and we are tempted to dismiss it as mere generation, we likely miss out on something pretty extraordinary, miss out on potentially growing our field and, what's worse, potentially risk losing our credibility.

Taking the most conservative threshold mentioned above (intentional novelty and intentional value) and being conservative in our analysis of the output potential of the various types of systems, we would require a system to have, at the least, the ability to filter artifacts. If we are more liberal in our analysis of the prototype models, or our placement of threshold, it is even easier to argue that we have made the leap, and that scoffing at mere generation has made the transition from inviolate charge to historical amusement.

The problems of the day are more complicated, as they should be, including questions like:

- how can we build computationally creative systems that are more autonomous (i.e. that have fewer of the designer's fingerprints all over them)?[20]

[17]If you haven't read *The Lord of the Rings* trilogy and *The Silmarillion*, you should

[18]Release dates: June 21 in North America, June 22nd in Europe, and June 24th in the UK, `http://www.no-mans-sky.com/`

[19]1.8×10^{19} worlds is the latest estimate, according to `http://www.gamespot.com/articles/how-to-play-no-mans-sky-a-detailed-breakdown/1100-6435316/`

[20]Note that the position taken here suggests that imbuing a system with greater autonomy might have the unfortunate effect of pushing it back onto the wrong side of the mere generation threshold due to a (hopefully temporary) precipitous drop in output qual-

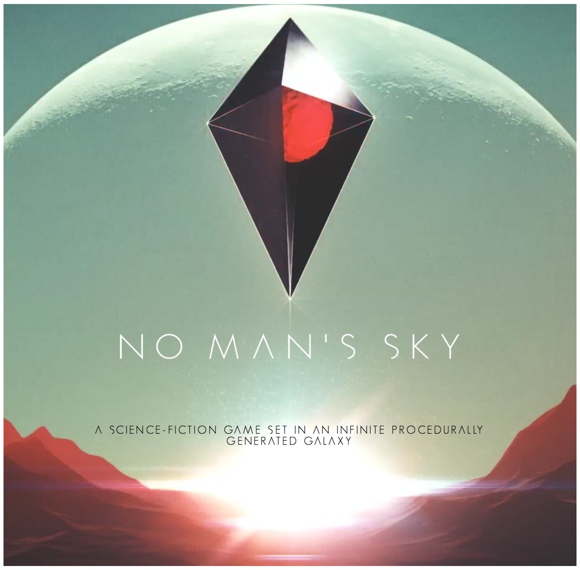

Figure 3: Can an "infinite procedurally generated galaxy" be produced by mere generation? (Image from `www.no-mans-sky.com`)

- how can we scale our systems to the real world (i.e. produce real artifacts of value and novelty)?
- how can we apply CC ideas to new domains (i.e. those perhaps less obviously amenable to computational approaches)?
- how can we do cross-domain (computational) creativity?

As an example apropos the last question, consider the problem of crossing humor with haiku. One way to do it, of course, is simply to write haiku with humorous themes[21], but a more sophisticated approach might consider parodying the form itself, as demonstrated by this cleverly horrible haiku-like poem[22]:

Anything can be
a haiku if you try hard
eno - u - u - gh

Last Words

We have argued for a need to revisit the idea of mere generation, for a need to better define what it means, that its use as a measuring stick for modern systems is outdated, and that continuing its use will be detrimental to our field. We have considered a spectrum of generation, populated with prototypical computational creativity algorithms and have argued that these are both abstract enough and varied enough in complexity to adequately represent the breadth of approaches in our field. Using this spectrum, we've argued that, as a field, we have surpassed the mere generation threshold.

Yet, our field seems to be growing very slowly, for all its appeal. In particular, our flagship conference seems to be characterized by a high rate of churn, the participants a combination of a small core of regular contributors and a larger contingent of hopeful initiates that fail to persist. Some churn is to be expected, and is likely even healthy, but too high a rate is detrimental, and it is very possible that such a high rate is correlated with our continued scoffing at mere generation.

We have a very long ways to go before we find our first computational Da Vinci, O'Keeffe, Einstein, Freud, Mozart, Turing or Dickens. But, we have come a bit farther as a field than we give ourselves credit for, and in particular, we are well past the doldrums of mere generation and exploring the uncharted territory beyond. We should take care not to overstate our achievements, but at the same time, we should take equal care not to understate them either. Further, there are many systems and researchers that have, even without the benefit of our collective wisdom/disdain, managed to navigate quite a ways into the wilds themselves, and it would be judicial of us to acknowledge this and make connections with them, expanding our understanding and our field.

ity. This seems consistent with a community view that esteems (intentional) novelty and value in its systems—if the change was a real advance, system behavior will eventually improve such that it surpasses the less-autonomous, and, in doing so, will easily find itself again on the right side of the mere generation threshold.

[21] Actually this has been done for centuries, at least to some extent, in the *senryū* poetic form

[22] This haiku was discovered by accident at `http://shirt.woot.com/offers/haiku-3`. If there is a better attribution, it is not known.

Acknowlegements

A tip of the hat to the anonymous reviewer who suggested the obvious-in-hindsight yet brilliant idea of making the title a haiku.

References

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*, 90–95.

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.

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

Ventura, D. 2008. A reductio ad absurdum experiment in sufficiency for evaluating (computational) creative systems. In *Proceedings of the 5th International Joint Workshop on Computational Creativity*, 11–19.