Casually Evolving Creative Technology Systems

Matthew Lewis

The Advanced Computing Center for the Arts and Design
The Ohio State University, Columbus Ohio, USA
mlewis@accad.osu.edu

Abstract

This position paper will describe the early stages of a development effort in which an interactive evolutionary design approach is being applied to the domain of technology-based system development, in a new media art and design context. These systems connect different technologies and techniques in order to process and transform data of one type into another. For example, location data might be used to select a relevant network information feed, the text of which drives geometry generation, which in turn could be sent to a 3D printer that would produce a sculpture. While the flexibility of this problem domain is idiosyncratic to our interdisciplinary new media environment, the target physical context for using a creativity support tool, and the context's effect on creative output is rather the primary research focus. The goal is to explore the possibilities of what might be termed "casual design", analogous to the "casual gaming" genre in which games are played by anyone, where ever they might find a few spare minutes, rather than requiring significant time and hardware commitments.

Problem Domain and Solution Space

In my position at our research center, I'm fortunate to enjoy constant interactions between many disciplines, collaborating with and teaching art and design graduate students and faculty who are passionate about seeking creative applications of emerging technology. Their work usually involves broadly interdisciplinary knowledge. Recent and current projects include glass artists building software, architects making responsive sound environments, product designers integrating social networking software, and choreographers visualizing databases. It is not uncommon for a researcher to have a concept or process they want to work with, but to have great flexibility in how they are willing to collaborate. When innovative technology solutions are sought, like any researcher we rely on our knowledge of prior work, searching memories and archives for recent discoveries that might apply and connect in new ways. Since our center is an emerging technology resource for all of the art and design affiliated departments at our university, we are usually approached with a specific design problem, but the investigator may or may not have a particular technology or approach in mind. Usually a significant number of research questions remain open. The situation is reminiscent of the way writers often talk of building a story from a seed of a single character, location, or situation, and then finding where it leads.

Dagstuhl Seminar Proceedings 09291
Computational Creativity : An Interdisciplinary Approach
http://drops.dagstuhl.de/opus/volltexte/2009/2195

A typical iterative design search for creative solutions proceeds: “What if we tried this approach? What are our options? If we send the information to this software, then we could process it so that it could be displayed like this...” Active participation in this discussion requires not just an ongoing broad awareness of the relevant technical building blocks that constantly become available within our respective fields, but also the ability to recall the possible technical options that have been reviewed in recent months and years. This last part is becoming increasingly daunting: it is commonplace to discover a half-dozen fascinating and potentially relevant new technology capabilities a day via an array of easily customized information feeds such as news blogs, Twitter, Delicious, etc. Examples of the breadth of emerging technology resources students and faculty have at their disposal at our university should be instructive: physical computing sensors and actuators; 3D printing and laser cutting fabrication systems; multi-touch, stereo projection, and VR displays; motion tracking, analysis, and recognition; distributed computing; speech recognition and synthesis; computer controlled physical lighting and multi-channel sound; real-time 3d virtual characters, environments, and dynamics simulation software.

In practice, our need to develop creative systems by assembling hierarchies of different technologies tends toward a small set of purposes. Initially experience is gained combining interesting new tools or resources, so that we’ll be able to demonstrate their integration to others who would find this useful. We try to create simple, clear examples to serve as resources that others can use as flexible building blocks for their own systems (and to help them learn on their own.) Teams of individuals from different disciplines often develop such systems as part of collaborative production and research efforts.

Might this space of creative possibilities more formally be represented so that it could be explored via a generative system, for use as a creativity amplification tool? If a sufficiently rich solution space can be crafted such that it can be gradually navigated, with individual solutions presented rapidly and in parallel, then the use of an interactive evolutionary design interface would be feasible. In addition, this sort of interface would be ideal for the desired target context: expanding the location of creative work into our larger environment (as is occurring with most other computer-mediated activities) instead of remaining tied to desk, chair, and keyboard.

Personal Descriptive Ontology

My early efforts at representing a creative space of technology possibilities in as simple a manner as possible are proceeding using a minimal descriptive ontology. I'm purposely pursuing a very simple fixed set of categories, properties, and relationships sticking to a “tagging” strategy as much as possible. This will be explained more below. Note that “ontology” is being used in the computer science sense of the term, rather than the philosophical. This clarification seems worth mentioning in an interdisciplinary context in which there is often confusion over such terms [Zun06]. (E.g., “Autonomous”, another important term in generative design, seems to hold a similarly contentious position between CS and philosophy usages.)

My starting place is a set of available, interesting, and useful technical resources, a subset of which is listed above (e.g., “3D printing”, “VR”, “voice recognition”, and “motion tracking”). This collection of “techniques and technologies” has been abbreviated to the shorthand “teks” in this project. These represent brief descriptions of building blocks or tools in my subjective creative space. For someone else, this list of terms might represent colors, shapes, 2D design principles, or other abstract concepts, and their useful capabilities.

A “tek” then has what can be thought of roughly as inputs and outputs: things they “take” (or make use of) and things that they can “make” (or influence). Straightforward examples include voice recognition software taking in speech and producing text, or 3D printers using geometry files to make physical objects. I will refer to these inputs, sources, and outcomes as data “types”, (though this is a fairly unsatisfying term.) These are represented by nothing more than a simple lists of associated keywords: one for the types a tek can “take”, and a second list for the types of things a tek can “make”.

Here is how a few new media works might be described in this manner:

```
(speech)->voice_recognition->(text)->3Dsoftware->(image)->projector->(projection)

(sound)->microphone->(signal)->electronics->(magnetic_field)->ferrofluid->(form)

(person)->camera->(image)->face_id->(signal)->electronics->(voltage)->robot->(motion)
```

Some teks are narrowly defined by the few types they can handle (e.g., “voice recognition”), while others are extremely general, potentially interacting with dozens of teks or types (e.g., “script”, “signal”, “voltage”, “electronics”, “motion”, “object”). Some terms might serve as both types and teks, such as a person or a surface. The lack of formalization and constraint in their description should definitely give pause to categorization experts, as their boundaries are very ill defined. But I would argue this is actually a strength. It seems important that the system be defined so that one does not increasingly worry about categorization as the collection grows in complexity. I would like to throw in new terms and be confident that it will continue to work well enough. Those who have trusted sites like Delicious with their tag collections have first hand experience, while I suspect many who started down the road of attempting to maintain precise tag categories likely stopped at some point because of the interface “friction” involved with adding new tags. There is perhaps an interesting analogy to the expression function lists of evolutionary image making [Sim91]. Terms selected less will likely be removed from the list of available functions, and those that seem to yield greater fitness results will remain over time. Ultimately the goal is for new additions to the lists not to matter: throw them into the list and simply judge what comes out. This is clearly more in the spirit of a genetic algorithm approach than with carefully crafted classical AI rule-based systems.

While I've found a natural temptation to inch closer to creating a “proper” ontology with my urge to further detail my categories, the opposite approach, using a flat list of “tags” (again, simply unconstrained descriptive keywords) becomes quickly more appealing from a user interface and data entry standpoint. While the Semantic Web continues to inch forward, nearly all successful content management sites (for links, blogs, images, video...) now support simple tagging. For an accessible, influential article comparing ontologies and tags, see [Shi09].

This personal, narrow-domain ontology could certainly be made ever “flatter” like an entirely tag-based systems, or easily more clean and precise to be machine readable (and reasonable). My limited experience with formal ontology building software interfaces revealed that ontologies were challenging to "hold in mind", add to, and maintain. Lacking a good visualization, it was hard to consider all the necessary ramifications of changes beyond a certain (low) level of complexity. While I've found collections of tags much easier to make use of, one cannot be sure one is finding all of the options: there may be redundancies, misspellings, missed synonyms, plural versions, etc. This has proven surprisingly acceptable however for the non-critical domains in which tags are commonly used.

Furthermore AI's "common sense" problem applies in any comprehensive attempts to represent all relevant relationships between attributes of systems. It is my working hypothesis in this experiment that a manageable handful of tags for each tek will suffice to obtain sufficient richness in the solution space. This will be the case as long as the recognition of value and interpretation of terms resides in a human evaluator, rather than needing to be embedded in the system (as would be necessary for semantic web style machine evaluation and traversal of the solution space.)

Some systems in this domain are naturally easier to describe in these terms than others, just as some paintings can be described in terms of color and shape, while others cannot. What knowledge needs to be represented to describe the great new media works of artists Matt Kenyon and Doug Easterly [Ken09]? In one, a robot seeks and consumes Coke, which eats through its skin, eventually killing it. In another, reports from a web site on the death counts of soldiers trigger a wearable machine to cause pain. Must terms such as "pain, consumption, eating, death, and skin" explicitly be integrated? The following ambiguously captures the technology flow, but completely misses semantics:

```
(webfeed)->electronics->(motion)->object->(contact)->person->(sensation)
```

The system design will allow a given term to have a list of synonyms, words that the software will treat equivalently but which obviously will be interpreted differently by the reader. For example the more general "webfeed" might be instantiated as "stock_market", "deaths_reported", "weather_data", while "sensation" could be similarly substituted with "pain", "irritation", "pleasure", etc.

When one reads a brief description of a project like, "text from poems is projected, falling and accumulating on the viewer's shadow" or "packs of toy robot dogs search for toxins with sensors" the appeal of the idea can be quickly appreciated. Does the description need to be "rendered" in a natural language form? It does seem that a great deal of value is added when:

```
(touch)->wearable->(signal)->remote_server->(streaming_audio)->speaker->(sound)
```

…is presented as "touching the dress lets you hear the sounds of animals in a distant forest"?

With Jim Campbell's "Formula for Computer Art" in mind [Cam09] it seems worth remembering that the evaluation of something like a twitter feed driving color is likely to reside in the eye and the intent of the beholder, rather than to be embedded in the system:

```
(text)->twitter->(words)->electronics->(voltage)->LEDs->(color)
```

So the question remains, can this space be rich enough, that it can serve as a useful creativity tool for assisting an artist or designer? Current random examples generated from the solution space include system descriptions which require domain expert knowledge to evaluate and leave much room for interpretation:

```
(text)->print_on_demand->(surface)->conductive_thread

(paper)->thermo_ink->(visual)->robot->(gesture)->wearable

(color)->script->(number)->drawing->(drawing)->person->(gaze)->mocap

(heat)->spime->(sns)->webpage->(feed)->infovis->(shape)->jitter

(audio)->msp->(number)->infovis->(shape)->avatars->(gaze)->person
```

Once such a list of system descriptions is presented, the *interface* for interactively evolving descriptions simply requires the ability to select the most promising candidate individuals. The more difficult work of producing offspring generations via mating and mutating the selected "fittest" system descriptions will be discussed below.

The unconstrained art+design domain seems an ideal starting place for an investigation such as this, given its rich/smooth solution spaces that will hopefully make for easy to traverse high fitness regions of design space. I would expect highly constrained domains such as engineering to be more challenging, whereas fields such as writing might be even more appropriate.

Context: "The Street"

Returning from the representation to the problem domain, how are social and cultural processes affecting human creativity? My awareness of current technological developments in this space continues to shift away from conference proceedings, books, and journals to contemporary social networking sources such as Twitter, link sites, and RSS feed readers. A great deal of the creativity I encounter in the world comes to my attention first via "Look at this!" messages from these various feeds. Its not that I've stopped using previous sources of information, it's rather that more often than not I've already heard about the bits that are relevant to me shortly after the information became available. The cultural shift that I'm most interested in pursuing for this work however is the shift in *where* and *when* this knowledge is acquired. Using smartphones with high resolution screens, fast network connections, and data "in the cloud", emerging developments tend to be absorbed, evaluated, and integrated continuously: walking down the street or hallway, in stopped freeway traffic, standing in a line, or during lunch.

While handheld computers are moving the context of our online interaction for shopping, coordinating travel plans, and social interaction, they do not seem to be having much impact on creative design processes. It's been well observed that that most of our interactions with computers (and more generally, media) are shifting from extended periods of time to small constant bursts of activity: we watch video clips on YouTube instead of hours of TV, skim blog headlines and summaries in RSS readers instead of reading newspapers and magazines, and communication via text message, Facebook, and Twitter is replacing lengthy phone conversations. But creative work remains an exception among computer-mediated activities.

Creativity is often seen as requiring a good span of focused solitary time. Computer design software interfaces require a table for our mouse and monitor. Our development tools are not designed for 30 seconds in the checkout line or for use while out for a stroll.

Many of us likely find it difficult to find blocks of four hours to sit in a quiet room and focus on work the way it seems we used to be able to. Activities we once spent hours on such as games, movies, reading, and writing - we still do, but for much shorter periods of often-interrupted time. For each activity we have switched to micro equivalents: interfaces that allow each of these to be done anywhere, in chunks of a few minutes. This is even more feasible as the devices we always carry increase in capabilities. This points to an opportunity for evolving emerging technology systems as data-mapping processes, exploring systems describable in handfuls of words, a length many of us commonly encounter and evaluate in micro-blogging Twitter or Facebook streams.

Our RSS feed and Twitter grazing can be viewed as creative tasks, with interfaces increasingly crafted for search, selection, and exclusion. As I follow new information sources and delete others, I am customizing a system that presents me with novel creations, applications, and combinations, tailored to my interests. My selection of which data-feeds to follow carves out the space of creative results I will see. If I choose to read Kirn's digital motion blog, I will see a different class of descriptions of emerging technological marvels than if I subscribe to the feed of someone interested in robots, or architecture, or wearables. I'm exploring the space of possibilities, but again whenever and wherever I happen to have a few moments. A significant part of system development involves this sort of design-by-selection paradigm that interactive evolutionary design provides. The interface needs are simple: displaying individual designs in a given solution space. The design-by-selection nature of interactive evolutionary design also provides a great deal of flexibility in terms of both centralization vs. distributed storage and computation. The embarrassingly parallel nature of the problem also yields benefits across storage, memory, and time, which is ideal for the limitations of a handheld device.

The social and cultural processes surrounding human creativity are shifting as the ability to rapidly and constantly *share* one's creative work, as well as review that of others, continues to increase. Usually new technologies are presented and evaluated for interest in a “display hierarchy”: ideas are first reviewed as short text descriptions with keywords, e.g. in Twitter or RSS headlines. If they pique interest, a link can be followed to preview text and perhaps images. Additional curiosity might result in further investment of attention: watching video, running software, interaction, or perhaps commenting, or evaluating questions, or subscribing to an info feed. This can lead all the way to downloading or otherwise acquiring a product. Here are a few representative examples:

“@jbacker I'm thinking MAME + BeagleBoard + Pico Projector = retro gaming on-the-go/on-the-wall” –bjepson

“new work rendered on Amazon EC2 & candidate for large scale prints: http://is.gd/rAqL” -toxi

“playing with HTML5 + processingjs w/ COTA Bus live views: www.bit.ly/dossier & www.bit.ly/bigmap just select a route (no IE)” -th0ma5 “make real paper objects from models in secondlife” -ayou

“Biological Display with fluorescent Bacteria” -Processing Blogs

“access your 3D models on your iPhone, take a photograph of your surroundings, then place the 3D model in the photo” -Ponoko

These “elevator pitch” descriptions are enough to initially appreciate and evaluate the core of the idea. There has been a great deal of focus lately in particular on Twitter's 140 character limit, which seems to have dominated discussions about the power of conveying concepts concisely.

The Interactive Evolutionary Design System

My implementation has started from a web-server script, accessible from any web browser. It relies on a minimal grammar for the systems generated. My primary emphasis is on making it easy to visually skim the system descriptions, using simple strings of words, as opposed to paragraphs of natural language. In the best case, I am striving for fast feedback: making it easy to browse many generations of system descriptions quickly, and making intermediate results non-precious and disposable. One example real system [Win09] could be described by:

```
(touch)->keyboard->(words)->mechanical_turk->(image)->projector->(projection)
```

This might be automatically mutated by the system, most simply by changing one of its technologies: e.g., “voice recognition”, or a “script” monitoring a web feed might be substituted for “keyboard” as an alternate means for generating “words”:

```
(speech)->voice_recognition->(words)->mechanical_turk...
(web_feed)->script->(words)->mechanical_turk...
```

Given a set of technologies, the means for fitting them together will influence the perceived fitness of the space. If teks/types are randomly combined, the systems' fitness will be extremely low in general (with occasionally higher spikes). As the space is groomed to bring up the typical fitness, there will likely also be fewer surprises. Mutating interior teks and types is significantly more challenging. Of course the more we guard against relationships that “don't fit together”, the less chance there is for surprise. As an example, web page navigation and a wireless Wii balance board are two teks that don’t naturally “fit” together, but their association (via “surfing”) in one project produced a high fitness interesting system [Sch09].

I am currently working on implementing mating and mutating of system descriptions. The difficulty of doing this is proportional to the desire to make the components fit together logically. While it's easy to generate random poor combinations: it takes more work to bias the system towards logical connections. A motion-tracking camera can't be used with a voice recognition module without intermediate components. On the other hand, just seeing “motion_tracking + voice_recognition” likely suggests thoughts of one activating the other. For my domain, I'm working to make the “teks” and “types” fit together instead of reading as so much Dada poetry, by trying to match their inputs and outputs. As with most genetic algorithms however, it'll probably be useful to allow a few “unfit” individuals into the population.

When mating two systems with technologies linked by matching data types, if a standard crossover is attempted naively, there will be issues grafting the head of one system arbitrarily to the tail of the other. Ideally they would match, or another input/output pair could be selected, but more likely, one or more additional teks will need to be inserted to bridge the gap between different data-types. This initially appears analogous to classical AI rule-based planning, or theorem proving, in which a chain of "rules" must be found and applied to get from the current state to some desired later state. Given the necessarily short length of these system descriptions as dictated by the target interface constraints, I'm optimistic that such searches will be limited to sets of only one or two teks.

Beyond Text

The use of interactive evolutionary design for system development is relying upon text based description display initially, but my intent is to provide automated linking to images and resources corresponding to the tek and type components, to provide resources for *implementing* proposed systems. Ultimately I aspire to generate diagrams, as well as interactive prototypes.

The easier part of this task will be the generation of links to automate searching for the relevant combination of keywords describing the proposed system. Individual tek and type components can link to tutorials and reference materials for how to make use of them, or in some cases link to the sites where the software, hardware, services, or data can be acquired. Complimenting these information links will be publication links: a selected system proposal could be posted to one's network of contacts, who might provide feedback and pointers to further resources (e.g., via Twitter.)

More complex system representations could be pursued beyond text. Much as with more traditional design processes, ideas are first sketched out with minimal commitment of resources and at a high level with little details. Those possibilities that seem most promising can be considered in further depth, requiring a correspondingly greater amount of resources including attention, computation, and time. In visual evolutionary design systems this is sometimes implemented in terms of resolution: one starts with thumbnails, then might request a subset of individuals to be rendered at a higher resolution for more detailed inspection. System descriptions can be converted into system diagrams using procedural drawing and layout software. They might also generate interactive prototypes by converting these diagrams into data-handling components in the way that network data flow interfaces (like Max/MSP/Jitter [Cyc09]) produce "system diagrams" that actually accept, process, and output data according to shared interfaces. Environments like Processing and Houdini also make it relatively easy to "glue" together different technical components like sensor input, 3D graphics, sound generation and manipulation, etc. into functional systems.

While there is a long history of formal research into the generation and control of visual and interactive systems from natural language descriptions (e.g., [Coy01] and in classical AI planning), the spirit of this work is more inline with the goals of the handheld game "Scribblenauts" in which a seemingly impossible number of words can be written and combined to create functional game elements [5th09]. Their description reads, “This game is all about experimentation, imagination and endless replay value. Think of any person, place or thing, write it down, and watch it come alive! Write Anything, Solve Everything!”

Conclusion

Several important challenges in the field of evolutionary design apply here: How might we visualize the solution landscape and its fitness? How can we improve the interface for sculpting an evaluated solution landscape? Is this framework applicable to other domains? I primarily envision possibilities for the humanities, comparing and contrasting sets of ideas in related, complex bodies of work.

I view this approach as *making* both for and on “the street”. Kevin Kelly writes, “...stuff as it is actually used, and not how its creators planned on it being used. As William Gibson said, *The Street finds its own uses for things*.” [Kel09] A quarter century after Gibson wrote of the street, he now writes online throughout the day in small bits on Twitter. I now can make use of his ideas about technology minutes after he writes them, wherever I might be when I read them.

When Gibson first mentioned the street, I was a teen, and was coding my first generative software. As I was thirteen, my software naturally created dungeon worlds with magic monsters, explored by procedurally generated warriors seeking treasure. While I remember these worlds vividly, they were displayed to me only as short text descriptions. In the same way that Hitchcock was famously more interested in constructing the script than shooting it, similarly, I find the creativity involved in evolving the ideas more fascinating than debugging the wiring. In evaluating the relevance of such applications, the relationship between creativity and play should perhaps be reconsidered.

References

[5th09] 5th Cell, “Scribblenauts”, http://www.scribblenauts.com, accessed June 25, 2009.

[Cam09] Campbell, J. Formula for Computer Art. http://www.jimcampbell.tv/formula/, accessed June 25, 2009.

[Coy01] Coyne, B. and R. Sproat. “WordsEye: An Automatic Text-to-Scene Conversion System.” In SIGGRAPH 2001, pages 487 – 496. ACM SIGGRAPH, Addison Wesley, 2001.

[Cyc09] Cycling ’74. “Max/MSP/Jitter”, http://www.cycling74.com, accessed June 25, 2009.

[Kel09] Kelley, K. “Street Use”, http://www.kk.org/streetuse/, accessed June 25, 2009.

[Ken09] Kenyon, M. and D. Easterly. “SWAMP website”, http://www.swamp.nu/, accessed June 25, 2009.

[Sch09] Schneider, D. “Nintendo Wii balance board: Surfing the Internet”, http://edutechwiki.unige.ch/en/Nintendo_Wii_balance_board#Surfing_the_Internet, accessed June 25, 2009.

[Shi09] Shirky, C. “Ontology is Overrated: Categories, Links, and Tags”, http://www.shirky.com/writings/ontology_overrated.html, accessed June 25, 2009.

[Sim91] Sims, K. “Artificial Evolution for Computer Graphics”, Computer Graphics (Siggraph '91 proceedings), July 1991.

[Win09] Winningham, T. “They Gave You Pop Art”, http://doodle4noodle.appspot.com/static/popart/, accessed June 25, 2009.

[Zun06] Zuniga Shaw, N. and M. Lewis, “Inflecting Particles: locating generative indexes for performance in the interstices of dance and computer science”, Performance Research 11(2), pp 75-86, Taylor & Francis Ltd, 2006.