Computational Creativity: Three Generations of Research and Beyond

Debasis Mitra

Department of Computer Science
Florida Institute of Technology
dmitra@cs.fit.edu

Abstract

In this article we have classified computational creativity research activities into three generations. Although the respective system developers were not necessarily targeting their research for computational creativity, we consider their works as contribution to this emerging field. Possibly, the first recognition of the implication of intelligent systems toward the creativity came with an AAAI Spring Symposium on *AI and Creativity* (Dartnall and Kim, 1993). We have here tried to chart the progress of the field by describing some sample projects. Our hope is that this article will provide some direction to the interested researchers and help creating a vision for the community.

1. Introduction

One of the meanings of the word "create" is "to produce by imaginative skill" and that of the word "creativity" is "the ability to create,' according to the Webster Dictionary. However, the intriguing act of human creativity has meant different things to different communities of scholars in different fields. Also, its relationship with "innovation" and otherwise general "intelligent activities" is not very crisp.

Over last half a century computer scientists seems to have used a shallow working understanding for creativity in order to develop computational tools that engage in creative activities, as perceived by the community at that point in history. We will divide these types of researches in three genres. The classification is not necessarily chronological but has some temporal insinuation and so, we call them as *generations*. The list of the works that we will discuss is not exhaustive. The purpose of this article is not to provide a survey of the area. Rather, we will choose some sample works as examples for each generation. Lastly, we will discuss the works outside the scope of computational creativity and some recent developments that may have strong bearings to the discipline.

2. Philosophical Angles

Philosophers try to understand creativity from the historical perspectives – how different acts of creativity (primarily in science) might have happened. Historical investigation of the process involved in scientific discovery relied heavily on philosophical viewpoints. Within philosophy there is an ongoing old debate regarding whether the process of scientific discovery has a normative basis. Within the computing community this question transpires in asking if analyzing and computationally emulating creativity is feasible or not. In order to answer this question artificial intelligence (AI) researchers have tried to develop computing systems to mimic scientific discovery processes (e.g., BACON, KEKADA, etc. that we will discuss), almost since the beginning of the inception of the former field (AI). Scientific historical data and philosophical perspectives are often input or embedded within such systems.

Two conflicting views about how science is (or should be) conducted comes from Francis Bacon(http://en.wikipedia.org/wiki/Francis_Bacon) and Rene Descartes (http://en.wikipedia.org/wiki/Ren%C3%A9_Descartes, both of them worked in the seventeenth century) within the context of modern Western Science. The former proposed a methodology based on observation, meticulous data collection and analysis, generalization from data with inductive reasoning. Descartes is a proponent of epistemological method based on pure deductive reasoning. The basis of these two contradictory approaches lies deep within the Greek philosophy. Locke (Locke, 1690) connected these two views by some putative analyses on how mind reasons with observations. However, modern scientists, since the days of Galileo, never took up one of these extreme positions or the other. Rather, the approach was always based on controlled experiments for data gathering whenever that is feasible. Theoretical science always revolved around laboratory science, and the importance of theory is never underestimated. A general philosophical underpinning of this dual approach was most noticeably provided by Karl Popper (1934), when he defined science as targeted toward falsification of hypotheses. Popper argued that a hypothesis cannot be verified inductively but falsified conclusively through experiments or observation. Thomas Kuhn (1962) subsequently contradicted this view of Popper in his seminal work on the sociological aspects of science. However, it seems targeting the statistical significance of a null hypothesis forms the basis of most contemporary empirical activities. Kuhn divided science into evolutionary or paradigmatic stage, and revolutionary stage when a paradigm gets replaced by another one, e.g. Newtonian mechanics by quantum mechanics. Kuhn's influence on attempts to understand and enhance human creativity cannot be understated. Current emphasis on providing appropriate social and group environment for creativity and invention is almost a dogma to the point that individual aspects of creativity is viewed with suspicion by many.

Psychologists almost avoided the issue of human creativity, except for some researchers who tried to understand the condition of mind that engages in creative activities (Csikszentmihalyi, 1996). Others tried to study the personalities of so called creative individuals. A recent trend is to study on how to improve individual creativity (Epstein, 1996) from a pragmatic view. Neurobiological study on the process of intellectual creation is often linked to the gross psychological behavior of creative individuals under observation (http://ideaflow.corante.com/archives/brain_chemistry_creativity/). Although these works do not seem to relate directly to automated creative activity, computational creativity researchers will ignore them to their peril.

3. First Generation

Samples of first generation tools that dazzled the community with their "creative" capabilities are the early artificial intelligence (AI) tools, like the game programs, the theorem-provers, and the expert systems. Trying to empirically prove that computers can be creative, these early AI researchers took up the contemporary challenges. One of the earliest such systems is the *Logic Theorist* (LT) written by Allen Newell, J.C. Shaw and Herbert Simon in 1956 (http://www.aaai.org/AITopics/bbhist.html). The same group developed the *General Problem Solver* around the same time primarily for theorem proving. In 1958 Gelernter and Rochester extended such theorem provers for geometry that did diagrammatic reasoning. Samuel at IBM developed the first game-playing program for checkers over the fifties that achieved sufficient skill to challenge a world champion. Interestingly, this is also the first program that can learn from experience and improve its skills. Feigenbaum, Lederberg, Buchanan, Sutherland demonstrated in 1967 with their *Dendral* program that computers can do sophisticated scientific reasoning based on precompiled knowledge. This is the first expert system and it interpreted mass spectra of organic chemical compounds. Although these programs impressed lay persons with the creative power of mechanistic computing, they cease to be considered as "creative enough" by the researchers as they generalized their experience and found some common algorithmic notions behind these programs that appeared not so novel after all.

These first generation creative systems were almost based on Descartesian principle of pure reasoning as opposed to Baconian learning from observation. They had elucidated three computing problems: (1) Blind search is not enough and faster search mechanisms are keys to the development of such smart systems. (2) There are general purpose (weak) heuristics and domain specific (strong) heuristics that improve the search. (3) Pure reasoning is not enough and availability of knowledge and the capability to learn are needed. These experiences ushered a new era in computational creativity as scientists took up different types of problems to prove the same hypothesis that computers can be creative.

4. Second Generation

Armed with the experience gained from the first generation of creative systems, two groups of researchers went into two directions. One group attempted to broaden the scope of the first generation systems. *Meta-Dendral* (Lidsay et al., 1980) adds learning capability to Dendral that improves with experience. Results from Meta-Dendral were publication worthy in scientific literature. Some groups used their program's weak heuristics and created shells that can address other domains than the original one (e.g., *EMYCIN* shell as an empty-MYCIN program developed out of *MYCIN* for medical diagnosis). Although this direction of works initiated the technology of *expert Systems* that resulted in development of many commercial intelligent systems, it did not contribute much toward computational creativity.

The other direction took up the challenge of developing systems for problems more related to creativity. Doug Lenat's (1984) *AM* program proposed theorems and conjectures in number theory starting with some axiomatic notions on set theory, and a bag of "common sense" rules (http://www.cyberconf.org/~cynbe/muq/muf3_21.html). Lenat's program also showed how younger mathematicians without much experience could contribute so much in the field, which is a known fact about mathematical research. Many of the second generation automated creative software led to much broader projects. For instance, Lenat's faith in common sense knowledge behind any higher level creative activity led him to develop the *CYC* project (Guha and Lenat, 1991), which is still continuing.

Cheeseman et al. (1996) developed the *AutoClass* program that classified astronomy data from sky-survey in unsupervised mode. It identified interesting astronomical objects, even discovered galaxies with distinct color distribution even though the input data did not include color information. This program ushered a new era of *Datamining* which found a niche application in the business world. In contrast to Descartesian pure reasoning-based paradigm this is also a venture into Baconian pure observation-based science. The discovery and the "aha" or "Eureka" comes by careful abstraction process over data (as Kepler did in finding laws of planetary motion from Tycho Brahe's data, as opposed to Archimedes' discovering density measurement technique by pure reasoning).

A conscious Baconian approach was taken by Langley's group (Bradshaw et al, 1987) in developing the program *Bacon.* Bacon derived algebraic formulas describing the relations between data. A small amount of supervision is needed in defining the variables over which the relation is to be discovered. Powerful weak heuristics are used in the program. It discovered Plank's law of black-body radiation, Boyel's law of gas, Kepler's laws of planetary motion, etc. Bacon, like AutoClass has no knowledge about science and works almost exclusively within the data space. Some additional knowledge could have made Bacon capable of suggesting new experiments (for missing or inconclusive data), or deriving additional theoretical results (e.g., postulating quantization of energy after deriving Plank's law). Domain specific knowledge can also reduce search space as observed from the first generation of tools.

The next system that we have picked up for our study within this generation is the *Kekada* developed by Kulkarni and Simon (1988). It was primarily targeted toward emulating Hans Kerb's discovery process of urea synthesis. Kekada suggested experiments, analyzed their results and gradually led to the discovery. It considered some primitive sociological factors like the expertise of the involved scientist, availability of resources, which constitute a great constraint on the discovery process.

Kekada was based on Simon and Lea's (1984) two space cognition model: rule space and instance space, with heuristics for both of these spaces. Heuristics are for experiment proposing, hypothesis proposing, problem generation, expectation setting, etc. Qualitative abstracted information from the experiments needed to be entered interactively. If the result violated the expectation bounds for an experiment, then a "surprise" occurred, and resolving the issue got priority. Thus, it emulated the important psychological aspect of human curiosity. Kekada combined the deductive reasoning with inductive data abstraction that is the common methodology of modern science. The prospect of Kekada was nicely described by Kulkarni and Simon, *"This was viewed, in turn, as a first step toward characterizing the heuristics used by scientists for planning and guiding their experimental work."* Although Kekada was a strongly domain specific software, Kekada-like tools could be wonderful aids for the experimental scientists.

In contrast to the models of individual discoveries in the previously described work, Taggard and Nowak tried to experiment with the sociological angle of the creative process in science in the Kuhnian sense. They emulated the "plate tectonics" revolution in geology as a paradigm shift. The acceptance of the fact that plates move on earth surface is considered as a revolutionary phenomenon in geology. Their program *ECHO* was built upon some underlying theses: (1) If scientific knowledge is represented with propositional and conceptual network, then a scientific revolution involves major transformation in the network. (2) Links in the network are mainly *kind-of* and *part-of* types. (3) New theoretical concepts normally arise by mechanisms of conceptual combination. (4) The network is primarily structured from the perspective of explanatory coherence. (5) Hypotheses are generated by abduction. (6) Transition to a new conceptual network (or scientific revolution in Kuhnian sense) happens because of a better explanatory coherence. Individual scientists are implicit overlapping pieces in this network, they do not appear in the model, and the creative process takes place over the whole community.

5. Third Generation

The second generation automated creativity tools were experimental in nature. Each was more geared toward proving a case. In contrast, the third generation tool developers are bolder in their approach. Armed with matured algorithmic techniques (Kakobas, 1993), they target their tools to produce practically useful results

Valdes-Perez' group at Carnegie Mellon University developed the *PAULI* system for elementary particle physics research. With significant amount of knowledge on particle physics embedded within the system, it primarily reasons on the conservation of different quantum parameters (e.g., charge, Baryon number, etc.) over the particle interactions (Valdes-Perez, 1994). The input to the system is a high level of information abstracted from the experimental data. Some of its results were published in Physics journals.

John Koza's group at Stanford University has applied Genetic Programming for invention in many disciplines (Koza et al., 1999). The list is too long to mention. Some of the notable ones are in circuit design, optical lens design, economic modeling, antenna design, discovery of protein motifs, and reverse engineering of metabolic pathways. A common technique behind all these different applications is to parameterize the topology of the underlying knowledge structure and then letting the genetic program to modify it (Koza et al., 2003). Some of the designs have had previous patents, or have patent-pending status (first time, in the name of a Software rather than that of a human inventor).

Gero and Kazakov (1996) used a very similar technique of genetic engineering for computer aided design. Only subtle difference between this approach and that of Koza's group seems to be in the way the underlying knowledge is represented. Gero (now at George Mason University) and Kazakov used a more formal logical KR technique with somewhat qualitative knowledge-base. The success of the evolutionary programming approach behind both of these groups' works suggest that creativity may need some boost from the random number generator in order to get out of local optima, or in other words, to get out of a conventional thinking pattern!

Doug Lenat's faith in the importance of common sense knowledge led to the *CYC* project (Lenat and Guha, 1990). CYC is an encyclopedic knowledge-base of gigantic proportion. The group developed its own logic-based knowledge representation technique and hand coded knowledge aided by interactive queries from the system. An expectation behind developing the system is that added with domain specific expert knowledge it will be able to be creative. The group has formed a commercial venture called *Cycorp, Inc.* (http://www.cyc.com/), and some free versions of the software are released both for commercial and research communities. To the best of our knowledge, no new creativity-related results have been reported in the literature yet.

6. And Beyond

On the periphery of automated creativity some computer scientists and epistemologists are always challenged by developing mechanisms for aiding human creativity. It is difficult to discern between *productivity tools* (e.g., spreadsheet or CAD tools) and *creativity tools*. However, some of such systems' contributions toward helping creative activities are undeniable. For instance, a tool to "visualize" music with the wave-forms and colors that actually help musicians to create new music is a creativity tool (see sidebar by Linda Candy in Shneiderman, 2007).

In a recent article Ben Shneiderman (2007) has described three schools of thoughts on what is needed for human creativity. (1) Structured approach, where meta-level knowledge clearly tells how to produce new concepts or design from existing information. (2) Inspirational approach, where right ambience needs to be provided for creativity. Archimedes' bath-tub or Newton's falling apple could be examples of such ambience where the creator is "inspired" to think out of the box. (3) Social approach, where communication with individuals working on the same problem provide complementary knowledge. Shneiderman discusses these three approaches in the context of developing creativity tools. Such a tool may provide appropriate structure, or right ambience, or help in dynamically communicating with a group knowledge-base.

One of the examples of structured approaches is TRIZ (1997). Although anyone is yet to develop any software based on TRIZ, it is a generic scheme (almost algorithmic) for designing new product or concept from existing ones. It provides a finite set of rules on what to look for the purpose of making appropriate modifications. In some sense it is like the evolutionary programming approach of Koza's group or Gero's group (described in the last section), where the objective functions would be based on TRIZ' structured suggestions. However, TRIZ does not provide a quantitative way of measuring the improvement and cannot get out of any local optima over the design space. One of the criticisms against TRIZ is that so far it produced inventions of only incremental nature.

Some researchers work on a much broader scale to improve human creativity in general. Janet Kolodner (2002) at Georgia Tech attempts to help creative activities amongst children. Her *Learning By Design*™ facilitates students in learning and applying concepts in science by going through iterative goal-oriented cycles of design/redesign and investigate/explore in a group setting. Case-based reasoning software helps in this facilitating process.

On a similar vein, Bruce Porter's group at University of Texas at Austin developed the *AURA* knowledge representation system that is being adapted by SRI and Vulcan, Inc., in their HALO project (Chaudhri et al., 2007). AURA is primarily an ontology system. SRI group has used it to develop a knowledge-base for a selected section of school physics. It can interactively help students to solve related physics problems. Their tool shows a strong promise in aiding creative activities in science. A broader knowledge-base can provide help in physicists solving unknown problems interactively. For the purpose of participating in creative activity it has to be more dynamic and be ready to modify the structure of knowledge than what an ontological system can provide (personal discussion with Mary-Lou Maher).

In this article we are primarily concerned with automated creativity that relates to science. Computational creativity is a thriving topic in arts and music (*Computer Music* journal is an example of such a community, see http://204.151.38.11/cmj/). However, some recent developments within the Computer Games cannot be ignored even from a study on creativity in science.

Game designers have to embed some amount of physics in their software. Primarily for the sake of faster processing, but secondarily for the sake of providing alternative reality to the users, they often tweak with the laws of nature in their programs. This has developed into a whole sub-discipline called *Game Physics* (Eberly, 2004), and vendors have emerged to provide independent game physics software modules and independent processors. Imagination is always behind the power of creativity and the game designers embed such imaginative alternative physics in their software. Although in most of such software the underlying physics is hard (procedurally) coded, sometimes they even provide their users to create alternative reality, as in *Second Life*. Experiences gained from developing and using such software someday may provide the background for generating a whole set of creativity tools targeted toward science.

7. Conclusion

In this article we tried to map research works on intelligent systems (or on its periphery) that are related to computational creativity. Our focus was primarily on the scientific or engineering domain, because such a domain provides rich repertoire of creativity activities. Also, the benefit to computer science in interacting with natural science is a well established fact. The samples picked up here are more to show the thematic progress of such research rather than to provide a broad review. We have loosely classified computational creativity research into three generations. We have also provided some philosophical underpinnings of the scientific discovery process in this article.

The first generation tools are the early AI systems, which were primarily to demonstrate that some intelligent activities are quite algorithmic. However, researchers soon realized that the key challenge lie in controlling the search space and in finding domain related heuristics. Importance of representing knowledge was also elucidated in this stage. The second generation tools were developed to prove that computers can be actually creative. Psychological and social angles of such creative activities were deliberately modeled. The potential that these projects created can hardly be understated. However, the direct focus on modeling a creative act in science was not followed in the subsequent generation. In fact the activities on computational creativity in third generation do not seem to be as extensive as that in the second generation. The momentum seems to have subsided. However, what we classify as third generation systems are far more matured and are targeted toward broad ranges of practical problems rather than to prove any small set of hypotheses, as the second generation tools were. The primary purpose of these projects is also to demonstrate the prowess of the underlying methodologies, not so much to contribute in understanding the creative process. In the US, a recent thrust from the National Science Foundation with their *CreativeIT* initiative raises some hope to revive the field. The initiative followed a series of workshops (Nakakoji, 2005) sponsored by NSF that provided some guidelines for computational creativity research.

Lastly, we have presented here some samples on the creativity support projects. Possibly the future of research on computational creativity lies in this direction that is of so much interest to the society currently.

Acknowledgement: Author gratefully acknowledges discussions with Subrata Dasgupta who initiated the former to many of the philosophical issues related to the scientific discovery process. Some ideas in this article came from a discussion with Mary Lou Maher. This work is supported by a grant (IIS-0732566) from the *CreativeIT* initiative of the US National Science Foundation.

References

Bradshaw, G. L., Langley, P., Simon, H. A., and Zytkow, J. M. (1987) *Scientific Discovery: Computational Explorations of the Creative Processes*. The MIT Press, Cambridge, MA, USA.

Cheeseman, P., and Stutz, J. (1996) Bayesian classification (AutoClass): theory and results. In *Advances in Knowledge Discovery and Data Mining*, Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (Eds.), AAAI Press/MIT Press.

Csikszentmihalyi, M. (1996) *Creativity — Flow and the Psychology of Discovery and Invention,* Harper Collins Publishers, New York, NY, USA.

Dartnall, T., and Kim, S. (1993) Editors. Artificial Intelligence and Creativity: Papers from the 1993 Spring Symposium, *Technical Report SS-93-01*, AAAI, Menlo Park, CA, USA.

Eberly, D. H. (2004) *Game Physics (Interactive 3d Technology Series)*. Elsevier/Morgan Kaufman, San Francisco, CA, USA.

Epstein, R. (1996, July/Aug) *Psychology Today*, Article ID: 1096 http://psychologytoday.com/articles/pto-19960701-000034.html

Gero, J. S. and Kazakov, V. (1996). Evolving building blocks for design using genetic engineering: A formal approach. In Gero, J. S. (ed.) *Advances in Formal Design Methods for CAD*, Chapman and Hall, London, pp. 31-50.

Guha, R. V., and Lenat, D. B. (1991) CYC: A mid-term report. *Applied Artificial Intelligence* 5 (1): 45-86 (1991).

Kakobas, S. (1993) Elements of Scientific Creativity. In *Artificial Intelligence and Creativity: Papers from the 1993 Spring Symposium*, (Eds.) Dartnall, T., and Kim, S. *Technical Report SS-93-01*, pp. 39-46, AAAI, Menlo Park, CA, USA.

Karl Popper (1934) *Logik der Forschung*.

Kolodner, J.L. (2002). Learning by Design™: Iterations of Design Challenges for Better Learning of Science Skills. *Japanese Bulletin of Cognitive Science.*

Koza, J.R., Bennett, F.H., Andre, D., & Keane, M.A. (1999). *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann.

Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., & Lanza, G. (2003). *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Springer.

Kuhn, T. S. (1962) *The Structure of Scientific Revolutions*, Chicago: Univ. of Chicago Press.

Kulkarni, D., and Simon, H. A. (1988) The process of scientific discovery: The strategy and experimentation. *Cognitive Science*, 12: 139-175.

Lenat, D. B., and Brown, J. S. (1984): Why AM and EURISKO Appear to Work. *Artificial Intelligence* 23(3): 269-294.

Lenat D. B., and R. V. Guha. (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley.

Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., and Lederberg, J. (1980) *Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project.* McGraw-Hill Book Company.

Locke, J. (1690) *An Essay Concerning Human Understanding.*

Nakakoji, K. (September 23, 2005) "Seven Issues for Creativity Support Tool Researchers," in the *Report of NSF Workshop on Creativity Support Tools*. ftp://ftp.cs.umd.edu/pub/hcil/CST-NSF/Papers/sevenissues.htm

Shneiderman, B. (2007) Creativity support tools: Accelerating discovery and innovation. *Communications of the ACM*, 50(12): 20-32.

Simon, H. A., and Lea (1984) Problem solving and rule induction: A unified view. In Greg, L. (Ed.), *Knowledge and Cognition*, Lawrence Erlbaum Associates, Inc. Hillside, NJ, USA.

Thaggard, P., and Nowak, G. (1990) The conceptual structure of the geological revolution. In *Conceptual Models of Scientific Discovery and Theory Formation*, (Eds.) Shrager, J., and Langley, P., pp. 27-65, Morgan Kaufman Publishers, Inc., Palo Alto, CA, USA.

TRIZ journal archive, July (1997) "40 Inventive Principles With Examples" and the "Contradiction matrix." http://www.triz-journal.com/archives/1997/07/

Valdes-Perez, R. E. (1994) Algebraic reasoning about reactions: Discovery of conserved properties in particle physics, *Machine Learning Journal*, 17 (1): 47-67.

Valdes-Perez, R. E. (1996) A New Theorem in Particle Physics Enabled by Machine Discovery. *Artificial Intelligence*, 82 (1-2): 331 – 339.

Chaudhri, V. K., John, B. E., Mishra, S., Pacheco, J., Porter, B., and Spaulding, A. (2007) Enabling Experts to Build Knowledge-Bases from Science Textbooks. Proceedings of the ACM conference *K-CAP*, Whistler, BC, Canada.