

On Designing “Stochastic Idea Generator”

Yoshiaki Mima

Future University Hakodate, Japan

Abstract. There are some software tools for supporting idea generation, with which user can generate various ideas with stochastic process. One of the most popular examples is the method so called “algorithmic architecture,” which is mainly used by architects. In such a system, use of random number is an essential factor for generating ideas beyond users’ expectation. In this paper, the meaning of the use of randomness in design process is discussed to figure out essential aspects of the stochastic approach. Then, two different types of software idea generators are introduced. As the necessary consequence, the effective control of randomness with language for concept construction is discussed.

Keywords: design process, random number, creativity, discovery, design control

1 Introduction

Finding, proposing, and implementing a new idea is important issue for creators. A lot of efforts has been made to get new ideas in many areas in our life, such as science, art, product design, mass production, education, business, etc. Many people has been discussed how to be innovative.

An innovative concept is considered to be “innovative,” because it is difficult to get reached with a simple combination of “existing methods.” To overcome the difficulty in finding good solutions, methods, which use random numbers to generate the candidates, are effective. For example, modern architects use the method called “algorithmic architecture. (Terzidis 2006)”

After finding “magic seed” for idea generation, people can generate good solutions for the time being; unfortunately, it will be difficult to get different results after creating some amount of the results. They are no more innovative.

In this paper, we will discuss topics “how to generate a new idea” in the following manner,

1. Review the existing methods for idea creation with or without random numbers in section 2;

2. Analyze the models for generating new ideas using random numbers and consider the merits and their limitation in section 3;

3. Propose a method to generate random ideas and a tool for controlling the idea generation tool to maximize the results in section 4;

4. Review the methods for idea generation.

2 Methods for Idea Generation

A lot of efforts had been made to be innovative. Yet another efforts had been made to construct general theory for creating the innovative idea (Altshuller 1999).

Before discussing “idea generation” in this paper, we will define the meaning of word “idea.” We are not going to discuss philosophical aspect of the word. Let us define the word “idea” as concrete and shareable information that express some status or provide some solutions, and “idea” must be described using recordable media. In other words, idea can be any sharable information such as an artwork that gives strong impressions to observers, a solution for some problems, and so on.

The meaning of the word “innovation” is difficult to define, in most cases, “innovative things” or “innovative matters” are expected to be something new. Generation of the innovative idea is difficult because it is an idea that cannot create from any established methods. In other words, if others know how to generate the target idea, it cannot be innovative.

Above discussion suggests, “Trying to find a systematic procedure for making innovation contains self contradiction.”

However, there are several conventional strategic methods for finding new ideas. In this section, we will review the approaches for “innovative idea” generation. The domains of the target ideas are widely spread, those idea creation processes contain common operational concept to enlarge the solution space for

evaluation and then make evaluation to choose the good solution from a lot of candidates.

2.1 Brain Storming

Brain Storming (Osborn 1953) is one of the simple, general, effective, and important methods to find good solutions.

When people use this method, the participants get together to create and share the ideas. At the stage of idea creation, participants are prohibited to give negative critiques to others. By allowing participants to make suggestions freely, unexpected good candidate ideas are expected to be appearing. This method helps to find a set of words or concepts of diverse topics.

2.2 KJ Method

KJ Method (Kawakita 1967) is a popular method for document idea preparation in Japan. This method was originally used as a method for preparing a report or academic paper from data gathered through the field works staying in different culture.

At the beginning, many small cards are prepared. On each paper, write the findings through the field works. Then gather the card, whose meaning is close.

As the result we have a cluster of cards. Then we will try to find the relations between the clusters. The orders between the clusters are also defined with the relation. With this process, finally, we will have a set of structured information, and some cards that do not have tight relations with other cards.

Using this method, a structured concept will be emerged.

2.3 Method for Innovation used in a Design Farm

Tom Kerry introduced a framework (Kerry 1999) for getting an innovative idea or solutions. Their steps of process, used in IDEO, are as follows:

1. Understand: define the problems to solve
2. Observe: make observation enough
3. Visualize and predict: build the design concept
4. Evaluate and refine: repeat refinement
5. Implement: the final stage for mass production

This method contains the brain storming in the step of "Visualize and predict" as sub-process. The most unique part of this method is that this method has a process for acquiring the wider perspectives for design, by making observation of target fields. By observing the target field, designers enlarges the chance to find better solutions.

2.4 Innovation with Repetitive Constructive Process

There is another model (Nakashima 2008) of innovative design. On this model the goal of the design is not specified as a complete static concept. On the process of finding good solutions, the target concept is dynamically changing and coming closer to the final solution with cyclic steps. Designer has a partial and incomplete vision of goal at the initial stage and makes a real physical model based on their mages. By evaluating the prototype the target concept will be made refined.

On this model, the change of target solution will guide the system to unexpected position with repetition.

2.5 Painting Program with Artificial Intelligence

AARON (McCORDUCK 1990) is a program designed and coded by a painter Harold Cohen. This program draws original paintings beyond the idea of its programmer about the composition of the paintings.



Fig. 1. Picture as output of AARON

At the early stage of AARON development, the program generated "abstract pictures" using random numbers. It was programmed to draw closed curves, avoiding overlaps of drawings. The detail of curve shapes are defined, one by one, using the parameters. Later, the program AARON, had added several models of predefined motifs. Then AARON has enabled to paint concrete motifs.

This is one of the examples of a system that generates unpredictable design. Even though the creator of AARON, he cannot expect what will be appeared as the next painting.

2.6 Algorithmic Architecture

Algorithmic architecture (Terzidis 2006) is a method to design architectural structures using computer. Architects define the structure on CAD system.

Architects write programs space for layouting by generating parts of buildings and specifying their positions in the space.

Architectural designers use program libraries based on object generation, modification, layout, etc. Those processes include trial and error based heuristic search. One of the important aspects of this approach is; computer sometimes generates shapes that the designer could not imagine.

3 Basic Concept for the Idea Generation

In this section, the concept for an idea generation by using random numbers will be discussed. Before the discussion, we will give a hypothesis.

3.1 Hypothesis on the Problem Solving

To solve a problem in a business environment, the way to find innovation is discussed. An innovation requires talent, ingenuity, and knowledge for a specific field. With these resources innovator need carefully observe the target and focus enough to the target problem (Drucker 1977). Or to make a good invention, there is a concrete procedure to solve target problem (Altshuller 1999). Those methods are used to solve important problems.

In this paper, the size of the expected target problem is smaller than that in previous paragraph. For example, if a creator wants to have small illustration. How does the creator create an idea? Perhaps, the creator may try to draw several illustrations and then choose the best one from them.

On the process, what is happening to the creator? One of the hypotheses is as follows; the creator already has a vague idea about the final requirement in mind. It is because the creator has experiences as specialist and enough knowledge about what is required. However, the final image is not clear enough, at stage. Those requirements may include contradictions. Or too many choices are there to convert the idea on paper.

On this hypotheses, before generating an idea,

- Target problem to solve is not well defined, problem is a set of requirements
- The set of problem may contain contradictions
- If any candidate solutions are shown it will be evaluated

In another words, when we are looking for the new idea, we are making a heuristic search of solution. If a creator is requested to give an illustration, then the

creator starts to find heuristic solution by generating solution candidates.

3.2 Generation and Evaluation of Target Idea

In some cases, designers try to create an idea, without the definition of what the ideal goal is. It is interesting that even in such an ambiguous cases, when a designer evaluate the idea, the person can easily classify whether the idea is valuable or not.

This phenomenon remembers the case that designers have tacit knowledge of what is a good idea, but cannot explicitly express the knowledge.

When designers use the approach to create innovative ideas using random numbers, it is easy to select only good patterns after generating many possible patterns, because the designer can distinguish good ideas from normal ones. If the speed of the automatic pattern generation is fast enough, it can be a productive process to choose good solutions from the automatically generated outputs. This can be a good collaboration between human and computer.

This system provides the idea generation schema by computer and checking by human.

3.3 Construction of the Concept by Observation

As mentioned before, the idea creators do not always have a concrete final concept or exact requirements.

In such a case, the creators will construct the concept or requirements for the idea in the process of finding solutions. For example, with the method of IDEO, observation of the target product and its user is important to construct the concept. By observing the target, we can find new points of views for evaluation.

But, in most cases observation is not easy, because it is not easy to find what we should be aware of. To overcome this problem, teams in IDEO design group includes ethnomethodologist as a specialist, because ethnomethodologist had a training to see things without serious prejudice in cultural point of view.

3.4 Construction of the Concept by Repetition

Construction of the concepts with repetitive constructive process is important, because in many cases, we can evaluate its value only after constructing target object and making it real.

A number of repetitive processes are important because we cannot make the creation process forward without realization and evaluation.

3.5 Repetitive Evaluation and Random Numbers

Use of the computer program can help us to check the idea by generating prototypes.

In many cases, at first, we define the system for creating objects. To create an object, a set of random numbers is defined to use as parameters. By changing the set of applications, different output will be created.

By using such a system, user can get new object just with simple operation such as clicking. As the result, user can try to create many objects and can evaluate them and choose good results. Required work is light enough for users, because they do not make any objects but just doing evaluation.

As the difference of objects has its origin from the value of the random number, those parameters provide the difference to the output. We have another factor of differences in output that depends on the nature of the program. From those factors, we sometimes feel the series of output has “the same tastes.”

3.6 Universality of Idea Generator

If an idea generator looks working well, we have another concern. If the range of output idea from the program is limited, soon or later the idea from the generator will be no more innovative.

We have several domain specific idea generators but, unfortunately, do not have a complete idea generator for universal domain. If a conceptual discussion is allowed, the computer system and programming language can be the lowest level description of idea generator; the second level description is a symbol generator, which effectively utilizes random value generator; the third level description will be universal domain independent descriptions if it can define; the final level will be the domain specific description.

We are not sure whether we can define the universal idea generation frameworks.

4 Experiments on Problem Solution

We made two systems for idea generation using random numbers. In addition to the cases in algorithmic architecture, from those experiences, we confirmed the use of random numbers are helpful to generate unknown useful solutions.

4.1 Visual Design Tool

ThinkingSketch (Mima 2001) is a system designed to generate graphical patterns by interacting with user. By using random numbers as parameter for positioning

and coloring. On this system rules for positioning is predefined. The value of generated random number is used as parameters for positioning and coloring.

ThinkingSketch is implemented on a graphic object editor, whose behavior looks like Adobe Illustrator or MacDraw. Object primitives such as line, rectangle, ellipse, free curve, image, etc. are predefined.

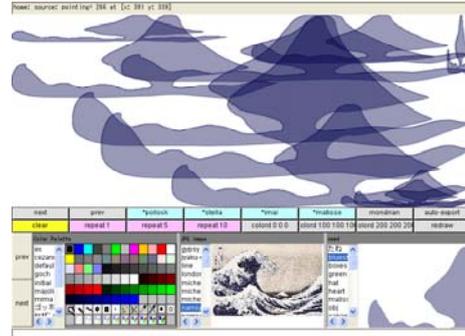


Fig. 2. ThinkingSketch with creator UI.

4.1.1 Definition: Graphic Pattern

The words “graphic patterns” are used to express a combination of the graphic objects.

Graphic patterns are made by repetitive operations of making a new graphic object from the prepared objects and placing it with randomly generated parameters.

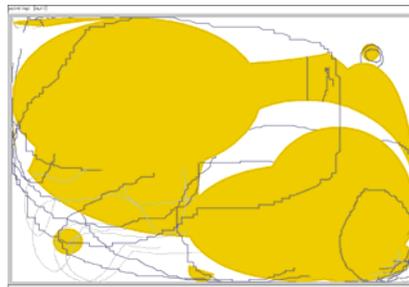


Fig. 3. Graphic pattern example generated using ThinkingSketch.

4.1.2 Graphic Pattern Generation

In order to create a graphic pattern, user makes registration of one or more graphical objects into a graphic seeds list.

ThinkingSketch simply makes a copy from the seeds list and places them to canvas area.

1. Copies one object from the top of the seed objects’.
2. Lists and changes the parameters of the target object, such as size, color, line-size etc. On

changing these parameters, random numbers are used.

3. Object will be placed to canvas by specifying the position using random number generator.

4.1.3 Control the Randomness

It is not enough to have only one “pure” random generator to generate. So we prepared different types of randomness.

- Change the distribution and average of target x and y coordinate
- Change the size of target object

Those random number generators can be controlled the pattern of distribution by specifying parameters. ThinkingSketch user can control the behavior of random number generator by adding additional parameters.

4.1.4 Other Feature for changing Graphic Attributes

The system has color assignment operation for changing the color of the object. User can modify the target object color when the system copies an object from the object list to canvas.

4.1.5 Description of pattern generation

In order to execute the same operation, ThinkingSketch has a simple scripting system. ThinkingSketch has a function to record user operations. Therefore, ThinkingSketch commands can be stored in a file and same operation sequences can be executed.

If the command is for layout operation, top object of the seed object list are copied, modified (if modification is specified), and will be pasted to the canvas area one by one with randomly generated parameters.

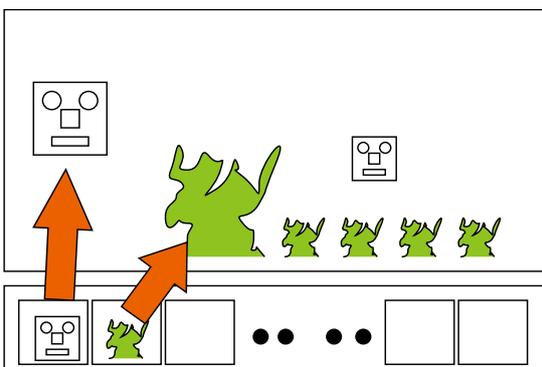


Fig. 4. Seeds Graphics Objects are composed by user specified mapping rules with randomly generated parameters.



Fig. 5. Parts of seeds for the Graphic Pattern Figure 6. each object is not generated by computer but traced with hand using functions for trace on ThinkingSketch.

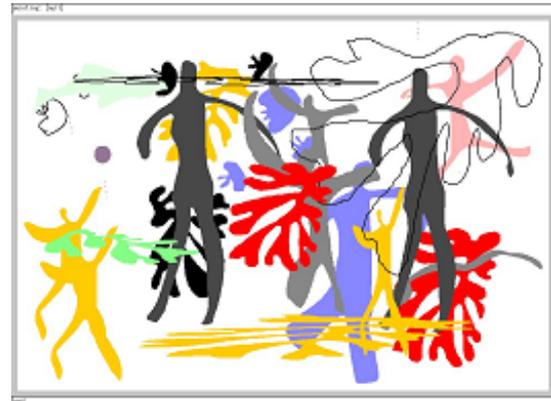


Fig. 6. Sample Output of the Visual Design Tool. Graphic primitives generated by tracing the shapes from the artworks painted by Matisse.

4.1.6 Graphic Pattern Generation

ThinkingSketch has simple software structure. As a preparation user simply draw some graphic objects and saves them as “seeds for pictures.” Then specify some rules for copying the objects. Usually, user operates ThinkingSketch until they find a good patterns.

If a user prefers some graphic patterns, the user can get the process for generating preferable patterns from the operation history. Then they re-execute the same operation scenario. At this stage, a user has

- a set of the graphical seeds
- a series of command to generate graphic pattern

a set of output graphics data will be generated from the system. If the number of object is large enough (the number depends on the situation) the output result seems to be made with the same drawing taste. (We found that this system can be useful to generate series of pictures for book covers, etc.)

4.1.7 Self Reflection on Seeds and Commands

At the initial stage, ThinkingSketch was designed to help beginner of design. One useful nature of ThinkingSketch is as follows; if a user finds a series of good graphic patterns generated from the same rule, the person can investigate the rules expressed as sequence of commands. It will help to analyze the tacit knowledge about the graphic patterns.

User can repeatedly generate the graphic patterns, which ThinkingSketch generates. Then, consider which seeds or commands are essential. User can change the seeds or commands and check how effective those elements are.

Another important nature is, even such a simple system, it is hard for user to expect what kind of impression will be appeared when user changes seed objects or commands. Therefore, making trial (with quick response) has important meaning to users.

4.1.8 Evaluation from Users

As ThinkingSketch do not make evaluation with statistical method. ThinkingSketch was used in six hours education for 40 students in a university, three workshops for about 20 children (younger than 14 years old), museum exhibition for about one year.

After using this system, we got free style format evaluation. From the users voice in common, the most impressed feature was user could easily reach unexpected but preferable figures by repeatedly generating the graphic patterns. Another voice was “the use of this system induces concentration.”

4.2 Tool for Japanese Typesetting

ThinkingTypography is a experimental software for Japanese typesetting tool, which allows assigning individual font attributes for each character. Attributes such as the size of the font, typeface and many other parameters are defined automatically by using this system.

4.2.1 Expected Goal

The purpose of this system is to add impressions to the typographical text expression. By changing the size, type-faces, and so on, extra information to textual expressions will be given.

For example, randomness in choosing the typeface or font-size on each character provides strange impressions for us. If we make a poem book and the impressions of the text layout matches to the contents with such a typesetter, it can make sense as publication.

4.2.2 Tuning the Randomness

However, it is difficult to assign appropriate attributes to each character. As the first step, designer of the typesetting tool decided to assign randomly generated attributes to each characters.

At the initial stage of system design, it is not realistic to directly apply the same concept from drawing software to this typesetting software. If the random number is directly assigned to each character, the output does not fit for comfortable reading. As the result, a mechanism for tuning the font attributes is added. On this process, rules for introducing randomness has made with heuristic approach.

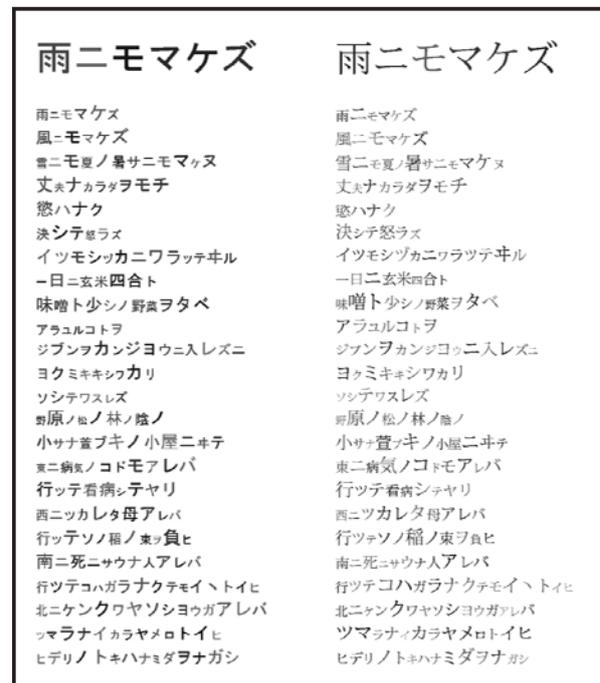


Fig. 7. Japanese Type Setting Tool Output Examples. The contents represents a Japanese poem. The designer tries to add more expression to normal textual expression.

4.2.3 Domain Specific Concept

On developing ThinkingTypography, aspects, which relates to the nature of natural language, had appeared as important features. In the early stage, a small set of “low level random pattern control commands” such as maximum or minimum font size, specification of distribution.

For example, Japanese language includes Kanji as ideogram, and two sets of phonogram, Hiragana and Katakana. For example, Kanji is usually chosen to be equal or bigger than other phonogram. Or, it is not good to change the size of symbols, which is used to separate sentences. If the fluctuation of the font-size is

not so sensitive, it is better to give the same rhythm pattern of the modification for the proper name.

To keep the balance of appearance the system must be aware about language specific matters. And thus, the language to control the randomness will be defined.

5 Discussion

From the studies including system prototyping, the control of random number and the mapping to the final expressions was considered to work as an approach for idea creation. In this section, the reasons, why this method is effective, are discussed.

5.1 Unpredictability in Randomness

We discussed that the use of random number is essential because it induces unpredictable results. After using unpredictable idea generator, we found more ideas that are useful for design.

On ThinkingSketch, another unpredictability exists. User prepares some seeds as initiator of process and applies a series of operations. By doing repetitive modifications, the output will be unpredictable.

5.2 Easiness on Filtering

Those two sample prototypes has important nature as interactive application. As final evaluation is made by creators, the generated patterns should be evaluated quickly. By repeating the operation (1) generation and (2) evaluation, the better solutions will be found.

In order to get preferable result: if the system presents any output, the output must be evaluated quickly. In above two example cases, the final output is shown visually.

5.3 Reproducibility

Those outputs are controlled with multiple parameters. By generating many outputs, user will find some preferable outputs. (In another words, they continue search, until they find preferable patterns.)

When the initial patterns and sequences of commands are fixed, the observed output will be predictable. Then, by picking up the sequence of commands from the “champion data” generated, user will re-evaluate the meaning of the initial patterns and sequence of commands.

5.4 Language and Impression

This is the last but the most important. The tool for pattern generation as idea generator should have a language to control the generation.

The language is important both for the clarification of target structure and the meaning of data modification.

6 Concluding Remarks

From the studies of idea generation software including two prototype systems, we found that the stochastic methods can be used to get new ideas. Though this is a kind of trial and error based system, the speed of idea generations can be accelerated by language specification. Randomization has extended the range of the outputs (findings).

As the next step, construction of a prototype system for generating a user interface is planned. One of the final issues on this area will be the establishment of a meta-framework for idea generation tool.

Acknowledgements

The author give special thanks to Ken-ichi KIMURA and Hidekatsu YANAGI, who have contributed to ThinkingSketch project.

References

- Altshuller G, (1999) The Innovation Algorithm: TRIZ, Systematic Innovation and Technical Creativity, Technical Innovation Center Inc., Worcester
- Drucker P, (1977) The Discipline of Innovation, Harvard Business Review on Breakthrough Thinking,
- Kawakita J, (1967) Hassouhou (abduction), Tyuuou Kouron
- Kelly T, (2001) The art of Innovation, Doubleday
- McCORDER P, (1990) AARON’S CODE, W. H. Freeman and Company
- Mima Y, Kimura K, (2001) ThinkingSketch: A reflection tool for drawing pictures on computer, Asia Design Conference
- Nakashima H, Suwa M, and Fujii H, (2008) Innovation in the Framework of Methodology for Constructive Informatics, IPSJ Journal, Vol.49, No.4, 1508–1514
- Osborn A, (1953) Applied Imagination: Principles and Procedures of Creative Problem Solving. New York, New York: Charles Scribner’s Sons
- Terzidis K, (2006) Algorithmic Architecture, Elsevier Ltd.