SICE Annual Conference 2014
September 9-12, 2014, Hokkaido University, Sapporo, Japan

Co-creative Project Management with Project Language

Masanari Motohashi[1†], Ken-ichiro Ogawa[1], Yoshihiro Miyake[1]

[1]Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Yokohama, Japan
(Tel: +81-45-924-5656; E-mail: motohasi@myk.dis.titech.ac.jp, ogawa@dis.titech.ac.jp, miyake@dis.titech.ac.jp)

Abstract: Why do original projects seem to be lack of creative feeling? Why does working together sometime seem to be lack of fairness and smartness? To answer the questions, project language based on well-known pattern language is proposed in this study. The project language will be useful for realizing co-creative working in real societies. We consider co-creation in terms of the project language, which is built by real users for users with pattern forms.

Keywords: Co-creation, Project Management, Pattern Language, Project Language

INTRODUCTION

Even though the standard of project management [1] is a common knowledge to precede projects in business and information technology, only 32% of projects got succeeded [2]. Key reasons of success on these projects are "user involvements" and "executive management support. [2]" Due to both the reasons being related to communication and relationship, it is meant that project management doesn't work on these areas.

The fundamental strategy of project management is "divide and conquer," the projects are separated and assigned each member under managing by only one or few person called project manager. This strategy makes a structure formed by trees, and the elements of projects hardly communicate or relate each without a project manager. Sharing knowledge is rare, and creativeness to prevent these problems is spoiled.

Co-working, or co-creation, style works very well to resolve these problems, because these work together and communicate frequently. Due to prefix "co-" means together, co-working is literally working together with others, and co-creation is the same. For example, WAIGAYA and SCRUM are famous in business and software development. The literal meaning of WAIGAYA is noisy with optimistic mood. Members with elective and manager talk essential problems during three days and nights. SCRUM is based on sharing knowledge and experience on a project [3].

These styles also improve creativeness, which is improving the original idea to get new design and insights. Working assumption to creativeness is provided from observation phenomena and culture to build models. Here, co-creation style has much potential of awareness from observing phenomena and culture rather than alone, because it has many views and thinking about it. However, the reliability of co-creation style is big mouth and shallow thinking in ordinary spaces. When one has a big mouth, observations and models are limited. For example, government, experts and enthusiastic persons tend to forces their opinion to their workers, citizens, students and so on.

We proposed Project Language that has an ability to build own language with the pattern format. Project Language is a collection of patterns to realizing concrete a project. C. Alexander, a famous architect, introduced the concept of pattern and pattern language to improve quality in life space [4] [5]. A pattern has semi-formed structure that contains of context, problem, and its solution. A pattern uses "context" to problem-solving situation for choosing a proper solution to the problem. Pattern Language is a model of the system for the communication that combines representational patterns [6]. The idea of patterns and pattern languages has been spread in various areas, especially IT and software. Although Pattern Language based on typical and common patterns can be used as an approximate solution, a gap occurs between special environment and subject. Here, we provide an idea of Project Language, which is Pattern Language with the special purpose for realizing a concrete project. [6] Participants and stakeholders co-create their own language as a project language.

Our working assumptions for co-creation are that stakeholders should have sharing scenarios and future vision. Indicating meta-model of Project Language: 1) own words of participants voices in context and environment, 2) pattern for the future to reveal problems that context have and find out possible solutions, 3) scenarios as language describe the future visions, and 4) center and centering process for adjusting between language and environment [6]. For pointing meta-process on structural-constructionism, this idea has a potential to solve cultural conflicts, build communities and markets and realize other democracies. In this paper, therefore, we consider co-creative working in terms of Project Language.

PROJECT MANAGEMENT

Standard of Project Management

In business, services and/or products are provided through various information technologies. There are many bodies of knowledge and standard, which are Project Management Body of Knowledge (PMBOK) [1], Business Analysis Body of Knowledge (BABOK) [7] Capability Maturity Model Integration (CMMI) [8] and so on. Many companies standardized their process and knowledge based on the knowledge. These standardized process and knowledge help business.

PMBOK is defined as a body of knowledge that a project is temporary in that it has a defined beginning and end in time, and therefore defined scope and resources [1]. Main idea of project management is "divide and conquer" strategy for tackling to complex and large size problem. The size of project is very important factor to success for the project. When you start your project, you and your stakeholders consider their purpose and scope of the project. A project manager, who manages the project, divides the project into tasks on working breaking structure (WBS), and assigns each task to members of the project. The members will complete these tasks that they assigned. The project manager is only people controlling these tasks that are separated.

Project management processes under PMBOK fall into five-steps:

1. Initiating;
2. Planning;
3. Executing;
4. Monitoring and Controlling; and
5. Closing.

At initiating step, purpose and structure of the project are built. At the planning step, a plan and design is made to finish the project for the purpose, and it is executed. The monitoring and controlling step is like an operation to adjusting. And the project ends at the closing step.

Success and failure of project

The above-described processes of project management seem to be natural and clear. However, 32% of all projects were succeeded (delivered on time, on budget, with required features and functions); 44% were challenged (late, over budget, and/or with less than the required features and functions); and 24% failed (cancelled prior to completion or delivered and never used) (see Figure 1).

The project manager hears the status and situation of progress and share knowledge on the project. And the project manager keeps the WBS. Even though the members report their own tasks on the progress meeting, the communication path is very narrow. The role of project manager easily becomes bottleneck of communication and sharing knowledge among their members and their users. The Standish Group researched the success factor over the year and reported key factors are "User Involvement" and "Executive Management Support. [2]" Although it means that keeping relationship and communication among stakeholders is important, the standard doesn't support these things.

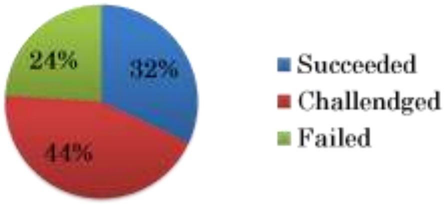


Figure 1: Resolution of project

Creativity in a standard project management

A project manager can be a hub of communication and knowledge. Task has tree structure, and process also has tree structure (Figure 2). Tree structure is good for easy understanding, but each node hardly makes relationship. For example, when node C and E are in same level, the node C have to pass node B, A, D, and E. This pass works in logic or computation, but a barrier among nodes is very large in real.

The member can finish the task without background of the project and sharing their knowledge. Even though the members can read purpose and background, knowledge of members is separated, and learning about the project and its background are limited.

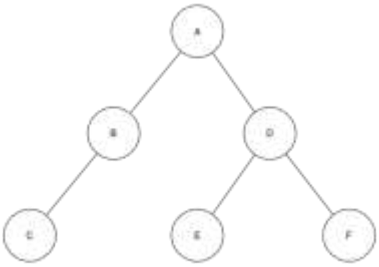


Figure 2: Tree structure of tasks in a project

Creativity means adapting plan and execution of a project. Sometimes creativity changes purpose or goal of the project in their situation and background. To get the creativity, somebody finds some hint or phenomena under their culture and environment, because these are hint under behavior or culture that be not aware. So, getting creativity needs deep observation and insights in their background.

However, standard project management tends to spoil creativity of member in a project because of 1) gap of plan and environment 2) lack of sharing knowledge 3) mismatch both other contexts 4) less understanding of the environment. The reasons are that creativity is rare, and if the creative chances are occurred, a project hardly applies it. A project is clearly divided to tasks, and knowledge sharing is very rare, so members cannot learning the real status, not by reported. Most places to get hint and insight are in execution process, because stakeholders and project members can see the situation and phenomena. If some creativity occurred in execution process, the plan is hardly to be changed.

CO-CREATION

Co-creation style working is to work together, because the prefix "co-" means "with" and "together." The literal meaning of co-creation is that create something together.

Creativeness is improving the original idea to get new design and insights. An working assumption to creativeness is provided from observing phenomena and culture to build models.

For instance, WAIGAYA meeting is talking and talking during 3days plus 3nights. WAIGAYA is an abbreviated form of the word "WAIWAI GAYAGAYA" in Japanese that means noisy with happy and creative mood. WAIGAYA meeting is famous from Honda

Motor Co., Ltd. This is one of good ways for co-creation. In software development, SCRUM framework is based on sharing knowledge and building teamwork. Therefore, co-working style helps stakeholders to get many viewpoints and opinions and discussing about topics to build model.

Issues on Co-creation

However, co-creative style working cannot match situation and human resources. When implementing SCRUM to software developers, there are some cases to reject persons, because they cannot synchronize framework of SCRUM although they have very high skill and experts. And, since business environment is also matched the SCRUM framework, a leader cannot see it.

When getting “own words” from disaster victim in TOHOKU area Japan, they are hardly talking their own mind in open spaces or workshops. Another example, in ordinal meeting and workplace, co-creation is depended on toughness and big voices. If someone has good idea or opinion, a person who has large voice tend to dominate the meeting, and may vanish the good idea. Although these problems should be occurred in standard project management, human factor is getting important in co-creation.

In general situation, co-creation style working is often not enough efficient way. Some people prefer deeply insights and thinking individually. In designing and planning, meeting and workshop style don’t work efficiently, because of “large voice” issue.

PATTERNS AND PROJECT LANGUAGE

Piecemeal growth

The standard of project management may make gaps between plan and environment in especially large size project. Large size project needs time and cost from planning to executions. If original purpose set wrong to the environment, the project is difficult to be changed in the standard. So, to proceed in proper way, solutions are getting feedback and change the project. Getting feedback from environment bridges gaps between plan and environment in a project.

On one hand, the processes of planning and execution are repeated quickly in one project. In this way, the project keeps the purpose and goal it has. This way uses iterative process to get feedback from other stakeholders or environment. Because this way is similar to the project management standard, this process is famous in software development as agile development.

On other hand, the project cannot be established in more complex environment. In software development, requirement is written down as a ticket, and it throw in the queue with priority. A project focus key futures and project to plan and execution (Figure 3).

To prevent from avoiding creativity, iterative process works, because a project can get feedback from real world. These ways are balance on scope and creativity. If you want to complete the project, you should not choose piecemeal growth way. Piecemeal growth is one of fundamental solutions for the issues of project management.

Figure 3: Task board on the wall

Background of pattern and pattern language

In the ordinary project, these are gaps between other environments. These different of environment occurs various scenes: planning and executing; teachers and students; consultant and workers; government and citizen. Role of teacher has their own success experience or their knowledge from books or papers.

However, when these knowledge getting in environment A imply the idea to environment B, gaps and conflicts are occurred in environment B. This way makes their environment wrong, because context in each environments are different. This gap is not small things. It likes colonization, because of neglect another environment. (Figure 4)

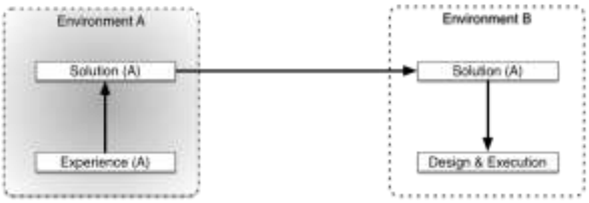


Figure 4: Standard project as two environments

Patterns

To prevent from these troubles, a pattern contains not only solutions but also context and problems in it (Figure 5). In addition, a pattern has consequence that is both benefit and reliabilities. These elements and form support users to choose patterns in the similar context.

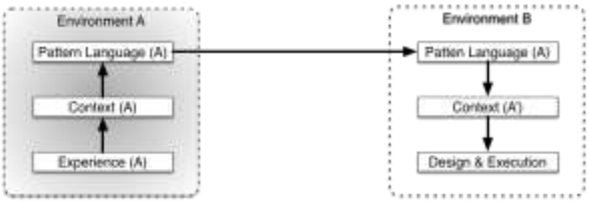


Figure 5: Patterns in two environments

Although the environment A and B are different, readers in B can understand context and problem of context. The lack of sharing knowledge increases risk of a project. To describe the knowledge, writing pattern has been succeeded in some areas, software development, learning, and social knowledge.

C. Alexander introduced 253 patterns by the book, A Pattern Language. Alexander defined pattern that "each pattern describes a problem that occurs over and over again in our environment and then describes the core of the solution to that problem in such a way that you can use this solution a million times over without ever doing it the same way twice."

Ward Cunningham and Kent Beck introduced Alexander's "pattern" to software development and information technology, and pattern prevails in related area [10]. For example, knowledge of building object-oriented program is shared as patterns. And patterns are used in various areas like education, organization, and spreading ideas. Agile development, which is light weighted software framework like eXtreme Programing and SCRUM, is originally written as patterns, and it is still growth.

In pattern community of information technologies, pattern contains Pattern Name, Context, Problem, Forces, and Solution as mandatory elements [11]. Mandatory elements are:

- Name: A name by which this problem/solution pairing can be referenced;
- Context: The circumstance in which the problem is being solved imposes constraints on the solution;
- Problem: The specific problem that needs to be solved;
- Forces: The often-contradictory considerations that must be taken into account when choosing a solution to a problem. The relative importance of the forces (those that need to be optimized at the expense of others) is implied by the context; and
- Solution: The proposed solution to the problem.

Each element of a pattern has roughly three parts: Context, Problem domain and solution domain. And force connects problem and domain as contradicts.

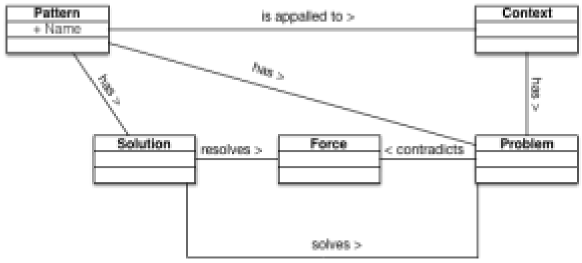


Figure 6: Elements of a pattern [6]

Figure 6 shows relationship among the elements in a pattern. In this figure,

- Pattern is applied to Context,
- Pattern has Problem and Solution,
- Context has Problem,
- Problem contradicts Force,
- Solution solves Problem,
- Solution resolves Force.

Two simple examples of potential pattern are presented below.

(Ex. 1) Pattern Name: Desktop light

Context: Knowledge workers like computer programmer work in an office together.

Problem: Brightness in office environment is variable.

Forces:

- In a dark office, workers hardly work such as reading documents and so on.
- Bright office is hardly worked in
- Separated partitions and rooms prevent the smooth communication among workers.

Solutions: Use a desktop light when it's needed.

(Ex. 2) Pattern Name: Carrying Short Literature

Context: You want to enrich your knowledge in an additional and unfamiliar area by reading some literatures.

Problem: You are too busy to make time for studying at your home and office.

Forces: You have to balance the following forces:

- Making time specific for study will sacrifice your family considerations and business.
- There are a number of discrete short times during your commuting, such as a time waiting for trains.
- There are various knowledge sources including short papers, articles and books.
- These short literatures can be good starting points to learn about unfamiliar area.

Solutions: Select short literatures and carry them at all times so that you could read them even in short time during commuting.

Current usual purpose of pattern is to communicate wisdom and insight that experts or teachers have. Some leaders in a domain wrote patterns and improve them. And readers consume these patterns as knowledge.

Pattern Languages

The word of pattern language has several meanings, 1) Patterns that introduced Alexander's Book "A Pattern Language", 2) Design systems originally for urban, 3) Generative systems and system.

Individual patterns are useful, but they are most powerful when combined into a language. "Each pattern then, depends both on the smaller patterns it contains, and on the larger patterns within which it is contained." To address real sized problems, languages are really essential.

A language is a collection of patterns that work together (Figure 7). Within a pattern language the patterns build upon each other, resolving unbalanced forces from previous patterns. Patterns and its relation build a language in a domain. A pattern capsules meaning that consists of name, context, problem, and solution as an episode. Both of users and providers uses pattern language.

Figure 8 shows an example of pattern language. Figure 9 shows the mapping language of shepherd to the meta-model. The language of shepherd is the collection of each pattern.

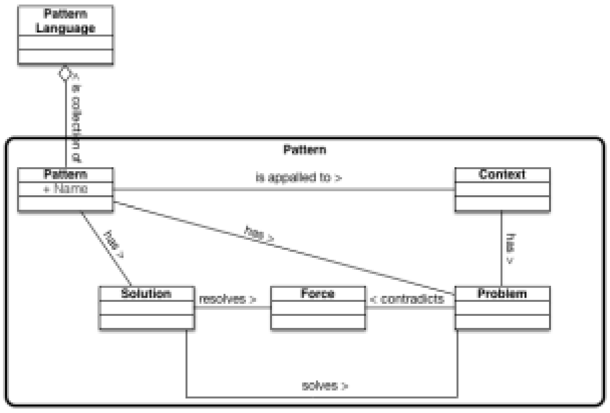


Figure 7: Meta-model of pattern language

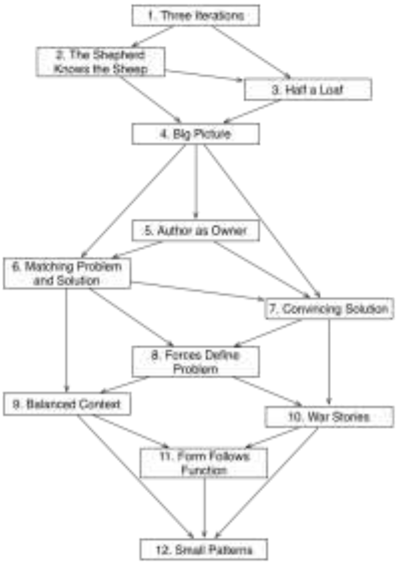


Figure 8: Language of shepherd

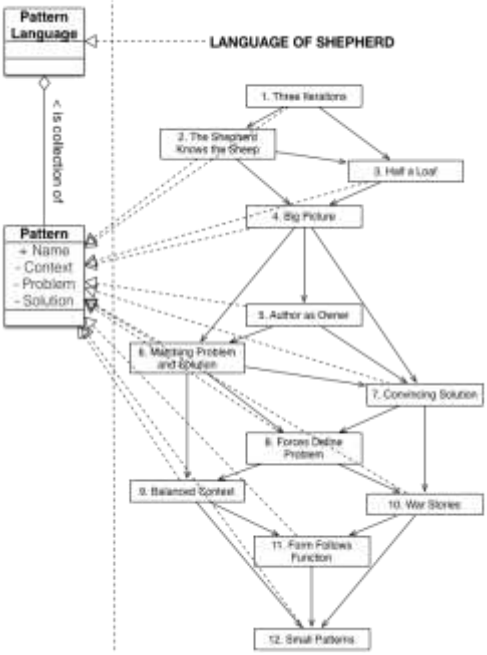


Figure 9: Mapping language of shepherd to the meta-model

PROJECT LANGUAGE

Pattern and pattern language is a good way to get creativity, thereby generating and sharing language. Even though patterns by proper experience are written in one environment, mismatch of context is occurred in another environment (Figure 5). This mismatch is smaller than standard project management. However this mismatch should still remain in the environment B.

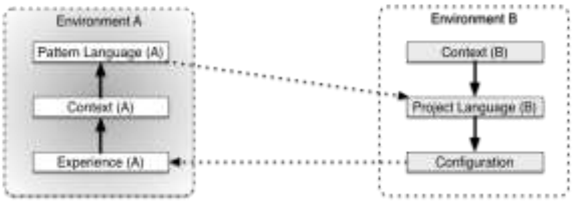


Figure 10: Project Language in two environments

To reduced mismatch, a language is built as a pattern, and another language is only referred (Figure 10). Teacher or consultant facilitates building language instead of "teaching" and "consulting."

Project Language is a pattern language with the special purpose for realizing a concrete project [6]. Figure 11, meta-model of Project Language, indicates elements and relationship of them:

- Pattern Language is collection of Patterns;
- Project Language describes Project;
- Pattern is embedded in Scenario;
- Future Vision is collection of Scenarios; and
- Project realizes Future Vision.

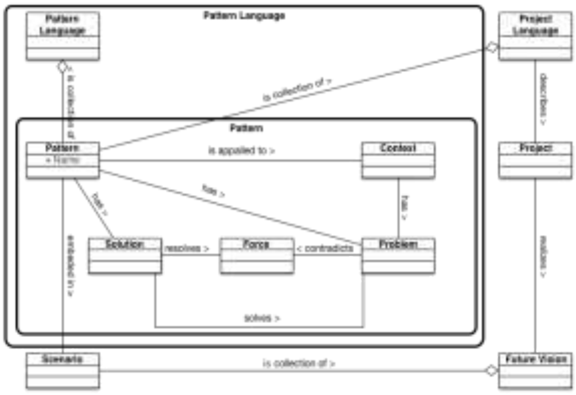


Figure 11: Meta-model of Project Language

A typical process of Project Language at Figure 12:

1. Own words: As many people as possible will participate, and tell your context, problem, and dream in your own words;
2. Pattern for the future: Build patterns from the context, problems, expected consequences towards the future using your words. This stage is built-in problem-solving techniques;
3. A language of vision: Picture visions and imaginations as pattern stories (Figure 13); and
4. Center and centering process: Use "center" and

"centering process" for adjusting two systems of the language and the environment. Centering is effective way to match two other systems.

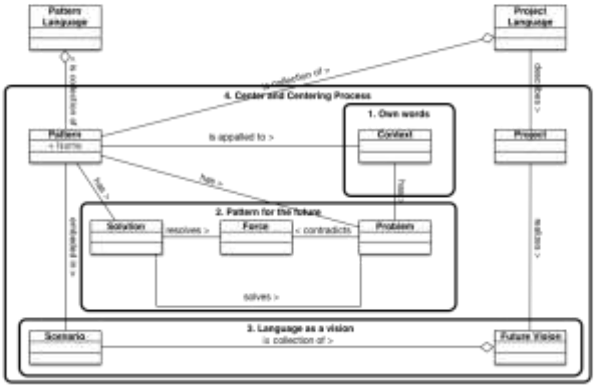
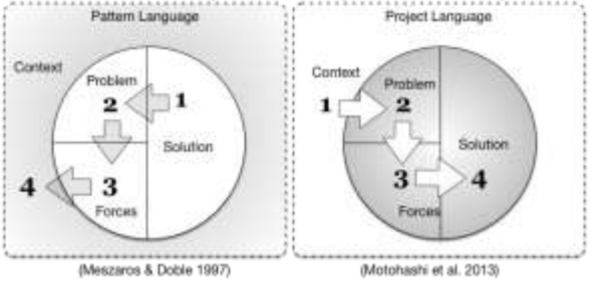


Figure 12: Meta-model of Project Language

Circled forms of a pattern are based on (Wellhausen & Fießer 1997)



(Meszaros & Doble 1997) (Motohashi et al. 2013)

Figure 13: Difference of building patterns

Project Language contains of problem-solving stage at 2, and configuring stage at 3 and 4 based on participants' voices at 1. So, this process works so for conflict management toward subletting among their problems. Project Language works well in building community and software development.

Co-creation with project language

Project Language has a meta-model and meta-process. By organizing patterns and a project management as patterns, these ideas and methods can be selected from a proper environment. For example, if you prefer to precede a project in a project management, you can plan and design in other methods after building scenarios.

For preventing a project from abusing "large voices" issue, you can choose interview in building own words. Project Language therefore has a potential to help both project and co-creation. Facilitator or leaders can choose a project language.

The reliability of building Project Language is 1) much cost and 2) out of expectation. Even though a project manager decides and controls the project in the standard project management, this way to get participants voices. However, consensus are easier, due to participants talk their opinion. In addition, teachers or government has their expectation, they can hardly comply it with students or citizen. However, the teachers or government can support and facilitate them.

CONCLUSION

In this paper, we show that Project Languages is useful for co-creative working in in-house and public workshops. However, Project Language is an experimental idea in workshop and business scenes, and therefore has not been analyzed by numerical value. We will build a concrete model for analyzing it.

REFERENCES

1 PROJECT MANAGEMENT INSTITUTE, INC. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)—Fifth Edition*. 2013.

2 THE STANDISH GROUP INTERNATIONAL, INCORPORATED. *he CHAOS Manifesto*. 2010.

3 Takeuchi, Hirotaka and Nonaka, Ikujiro. *The New Product Development Game*. Harvard Business Review, 1986.

4 Alexander, Christopher. *The Timeless Way of Building*. 1979.

5 Alexander, Christopher. *The Oregon Experiment*. 1975.

6 Motohashi, Masanari, Hanyuda, Eiiti, Nakano, Hiroshi. *From Pattern Language to Project Language*. Pattern Language on Programs, 2013.

7 INTERNATIONAL INSTITUTE OF BUSINESS ANALYSIS. *A Guide to the Business Analysis Body of Knowledge*. 2009.

8 SOFTWARE ENGINEERING INSTITUTE. *CMMI Version 1.3*. 2011.

9 Eto, Koichiro. *Pattern、Wiki、XP ~ Timeless way of principles of creativeness [In Japanese]*. Gijutsu Hyoronshsya, 2009.

10 Meszaros, Gerard and Doble, Jim. *A Pattern Language for Pattern Writing*. 1997. The Pattern Languages of Program Design Third Book.