Pattern Language and the Future of Education in Light of Constructivist Learning Theories, Part 4: Consideration with Constructionism of Seymour Papert

TAKASHI IBA, Faculty of Policy Management, Keio University
KARIN IWATA, Graduate School of Media and Governance, Keio University

This paper discusses how pattern language contribute to formation of knowledge and creation based on constructivism learning theory. From the point of view of constructivism, human construct their knowledge internally from oneself, and cannot acquire knowledge from external inputs. From the viewpoint, this study aims to pursue what pattern language is, and how it could support learning and practicing. This study is written in series of four papers; following the previous papers on constructivist learning theories by Jean Piaget (Iba and Munakata 2019), Lev Vygotsky (Iba and Burgoyne, 2019a), and John Dewey (Iba and Burgoyne, 2019b). This paper focuses on Constructionism proposed and discussed by Seymour Papert and his successor, Mitchel Resnick, and clarifies how pattern language supports learning by making, debug, and collaboration.

ACM Reference Format:

Iba, T. and Iwata, K. 2019. Pattern Language and the Future of Education in Light of Constructivist Learning Theories, Part 4: Consideration with Constructionism of Seymour Papert. HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. 26 (October 2019), 10 pages.

1. INTRODUCTION

Pattern language is a language which verbalizes rules of thumb for creative acts and supports people in acquiring it in high quality. Then, how can the knowledge and improvement be achieved? In this paper, we focus on how pattern language supports learning and practicing from the viewpoint of "Constructionism" proposed by Seymour Papert. Furthermore, this paper is a part of four series paper that aims to look for role of pattern language through the learning theory of Constructivism.

The reason why we became interested in theory of Constructivism is related to Dewey's thought which includes both constructivism and pragmatism. When we start thinking about what pattern language is and how it affects people, we must consider how it relates to learning and practice. About learning, there is a difference between knowing as information and understanding it by yourself. So, we started to focus on constructivist learning theory that it explains knowledge is structured not from outside but one's own experience. Pattern language support people's practice and enable new experiences so that they can compose and learn knowledge internally. Patterns are different from simple instructions or manuals so you must think and carry out the concrete actions by yourself, and this is the point where pragmatism start to relate. Pragmatism takes a knowledge as true when it actually has an effect and emphasis on experiments. Pattern written in the form of Context-Problem-Solution-Consequence is exactly the expression of knowledge in the form of "IF-Then…" in pragmatism, and you will experiment to see if it actually occurs and how to materialize and practice the abstract solution. So, the result is reflected and forms a belief. So, this connect to the theory of Constructivism.

Constructivist learning theories, which was originally started by Jean Piaget, emphasize that knowledge is never just transferred from the external world, but it is always constructed within individuals, as we introduced in Iba and Munakata (2019). In this paper, we take up Seymour Papert (1928 - 2016), who follows Piaget's theory partly and also emphasizes on *learning by making*. Papert studied under Piaget for five years and developed that thoughts in MIT Media Lab by working on research and development of tools for thinking, and how it turns out in children's learning process. In what follows, we first introduce the thought by Seymour Paper and then discuss how we look at pattern language from his point of view.

2. CONSTRUCTIONISM BY SEYMOUR PAPERT

2.1 Learning by Making

In the preface of his book, *Mindstorms: Children, Computers, and Powerful Ideas* (Papert, 1993a), Papert introduces his episode in his childhood. He recalls that he used to play with gear toys because he had a keen interest in cars from an early age. As a result, he was able to understand mechanics of combination and rotation in gear in an abstract level. He grasped the mathematical functions by calculating the systems through

arranging the gears in his head. From this experience, Papert was able to get an inspiration to find his idea on fundamental fact about learning.

> "Anything is easy if you can assimilate it to your collection of models. If you can't, anything can be painfully difficult." (Papert, 1993a, p. xix)

In this remark, the word "assimilate" is used here, which Piaget uses to explain recognition and construction of knowledge. Note that *assimilation* is the application of appropriate *schéma* to represent a perceived situation (Fig 1); It occurs only when there is something assimilable; Piaget stated that humans only assimilate things which are assimilable to the cognitive structures they have at that moment; Things which could not be assimilated cannot be recognized, and are just missed out.

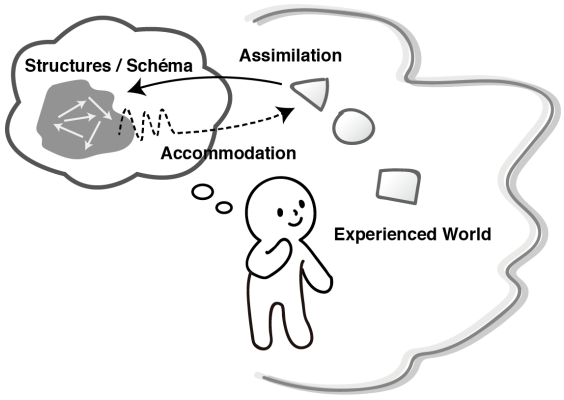

Fig. 1. Recognition is constructed by "assimilation" and "accommodation" of cognitive structures (Iba & Munakata, 2019)

Although Papert is a successor of Piaget, he has made further developments in Piaget's thoughts. The route to the development is integrated in this next question.

> "*The understanding of learning must be genetic.* It must refer to the genesis of knowledge. What an individual can learn, and how he learns it, depends on what models he has available. This raises, recursively, the question of how he learned these models." (Papert, 1993a, p. xix)

It is true that assimilation to structure is needed so that subject can cognition and understand, but he wondered how the structure is made. The answer by Piaget was by experiences one has so far. Papert agreed to that, however he pursued from that point further and focused on *tools* that contribute to construct that structure.

Remember that the boy Seymour became able to construct the mathematical structure by treating and operating the gear. Though Piaget studied about relationships of things around us like rock and tree, blanket basically, Papert have interest in more advanced device, the tool that can be composed and made by themselves (Fig. 2). Thus, he developed the programming environment "LOGO," which users can write program that direct turtle's movement and draw the trajectory as graphics. This LOGO environment is a tool for thinking, which was proposed by Papert.

Furthermore, He called his position as "Constructionism" in order to emphasize the significance of activity of construction for learning. As you can see, this term "Constructionism" is punned on the term "Constructivism" coined by Jean Piaget, but, in addition, it emphasizes "construction" (Papert, 1993b); it also shows the position against "Instructionism" (Papert, 1993c), which means education by instruction, in conventional education.

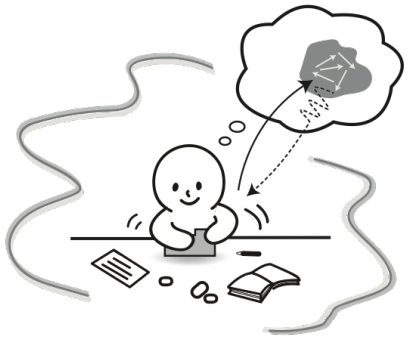

Fig. 2. learning by construction, which is emphasized in the Constructionism

In the LOGO environment, children write program and give instructions of action to a virtual turtle in computer world (Fig.3,4). This is the way which Papert mentioned "The Child programs the computer. And in teaching the computer how to think, children embark on an exploration about how they themselves think" (Papert, 1993a, p. 19). In this process, children would think and understand deeper. Papert describes this point as follows.

> "Even the simplest Turtle work can open new opportunities for sharpening one's thinking about thinking: Programming the Turtle starts by making one reflect on how one does oneself what one would like the Turtle to do. Thus teaching the Turtle to act or to 'think' can lead one to reflect on one's own actions and thinking. And as children move on, they program the computer to make more complex decisions and find themselves engaged in reflecting on more complex aspects of their own thinking." (Papert, 1993a, p. 28)

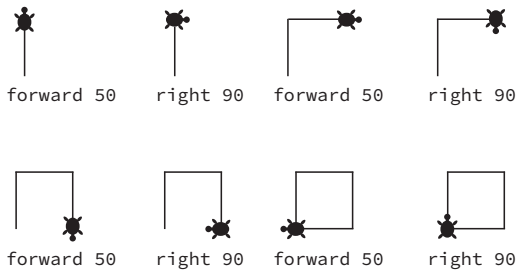

Fig. 3. The instructions of turtle's movement on LOGO (LOGO Foundation, created from 2000)

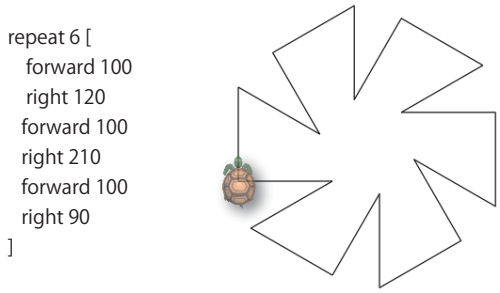

Fig. 4. An example of program in LOGO (Left: Program code, written by Takashi Iba, Right: Trajectory of the turtle [1])

2.2 Debug

Next, we move to "debug" that Papert mentioned as one of the important skills in programming and learning. The word "debug" is a term in computer field which means "remove bug" (de- means remove). In this context, "bug" is a mistake that hide somewhere in the program that one made; and the program won't behave properly because of that hiding bug. Papert emphasized that the debug skill is very important in learning as well as programming.

> "But when you learn to program a computer you almost never get it right the first time. Learning to be a master programmer is learning to become highly skilled at isolating and correcting 'bugs,' the parts that keep the program from working. The question to ask about the program is not whether it is right or wrong, but if it is fixable. If this way of looking at intellectual products were generalized to how the larger culture thinks about knowledge and its acquisition, we all might be less intimidated by our fears of 'being wrong.' This potential influence of the computer on changing our notion of a black and white version of our successes and failures is an example of using the computer as an 'object-to-think-with.'" (Papert, 1993a, p.23)

The whole process of finding problem, solving it, improving it is valued in the LOGO environment for learning by making. Even though debug is a wording in the field of computer, Papert thought that the fundamental concept of this is that "Surely 'debugging' strategies were developed by successful learners long before computers existed." (Papert, 1993a, p. 23). We often make problem finding and problem solving, repairing, improving things in our everyday life, not only in programming. Moreover, it seems the topic of debug makes clear the problem of present education and puts light to new direction that education should go.

> "One does not expect anything to work at the first try. One does not judge by standards like 'right --- you get a good grade' and 'wrong --- you get a bad grade.' Rather one asks the question: 'How can I fit it?' and to fix it one has first to understand what in its own terms. Only then can we make it happen on our terms." (Papert, 1993a, p.101-102)

Are schools in our society now able to become such a place? On one hand, schools and educators say that they will focus on "trial and error," on the other hand, isn't it required to tell right answer at first time? Don't they give students opportunities to correcting their mistakes and improving the quality of answer? Papert illuminated the problem of present education with the following episode.

> "I have seen this in many children's first sessions in a LOGO environment. The child plans to make the Turtle draw a certain figure, such as a house or stick man. A program is quickly written and tried. It doesn't work. Instead of being debugged, it is erased. Sometimes the whole project is abandoned. Sometimes the child tries again and again and again with admirable

[1] This graphics generated by running the program on ACSLogo For Mac OS X (https://www.alancsmith.co.uk/logo/).

> persistence but always starting from scratch in an apparent attempt to do the thing 'correctly' in one shot. The child might fail or might succeed in making the computer draw the picture. But this child has not yet succeeded in acquiring the strategy of debugging." (Papert, 1993a, p. 113-114)

> "The ethic of school has rubbed off too well. What we see as a good program with a small bug, the child sees as 'wrong,' 'bad,' 'a mistake.' School teaches that errors are bad; the last thing one wants to do is to pore over them, dwell on them, or think about them. The child is glad to take advantage of the computer's ability to erase it all without any trace for anyone to see. The debugging philosophy suggests an opposite attitude. Errors benefit us because they lead us to study what happened, to understand what went wrong, and, through understanding, to fit it. Experience with computer programming leads children more effectively than any other activity to 'believe in' debugging." (Papert, 1993a, p. 114)

We can imagine the situation that their everyday school life brought to them. But debug could bring a new way for present situation of school. Also, in learning by making based on Constructionism, earning arts becomes important as it enables people to actually make something. Papert said, "As in a good art class, the child is learning technical knowledge as a means to get to a creative and personally defined end." (Papert, 1993a, p. 134) Also, Alexander said repairing is the important process in making something with wholeness. " In the commonplace use of the word repair, we assume that when we repair something, we are essentially trying to get it back to its original state. This kind of repair is patching, conservative, static. But in this new use of the word repair, we assume, instead, that every entity is changing constantly: and that at every moment we use the defects of the present state as the starting point for the definition of the new state." (Alexander, 1979, p485)

In addition, in learning based on Constructionism, it enables us to make something through collaboration with others. If it's a new thing which is made actually for the first time in that place, it can collaborate beyond the experience and levels of knowledge. Papert noted that it enables students to collaborate even with their teacher as follows.

> "In traditional schoolrooms, teachers do try to work collaboratively with children, but usually the material itself does not spontaneously generate research problems. Can an adult and a child genuinely collaborate on elementary school arithmetic? A very important feature of work with computers is that the teacher and the learner can be engaged in a real intellectual collaboration; together they can try to get the computer to do this or that and understand what it actually does. New situations that neither teacher nor learner has been before come up frequently and so the teacher does not have to pretend not to know. Sharing the problem and the experience of solving it allows a child to learn from an adult not 'by doing what teacher says' but 'by doing what teacher does.'" (Papert, 1993a, p.115)

In this way, we can achieve the situation that "the teacher as well as the child can be genuinely excited by it" (Papert, 1993a, p. 134). This can be possible because this activity is for *making*, not because it is about using computer. Project of making enables teacher to learn by making with students.

3. SOME EXTENSIONS BY MITCHEL RESNICK

In this section, we overview about extended cases of constructionism by Mitchel Resnick (1956 -), who collaborated with Papert at MIT Media Lab (Kafai and Resnick, 1996; Resnick, 2017). Resnick developed the programming environment "Scratch" and also established online community for it. When Papert made LOGO, which was fifty years ago from now, a personal computer didn't exist and there were only computers that are quite large machines. At that time LOGO was extremely advanced, but it cannot be helped that people might consider the LOGO is too simple and old-fashioned because now they use a variety of media environments. In contrast, Scratch is a colorful and visual programming environment which attract children's interest now.

However, the most important point in Rick's development is that he established an online community where individual users can share and interact with their own outcome. Thanks to this online community, users become possible to learn from others and make collaboration with each other. Resnick introduces many

episodes about how children experience and have fun in this online community in his book, *Lifelong Kindergarte*n (Resnick, 2017). Users also can obtain players who watch and play with what they made, and then, they will deliberate what is good product which everyone can enjoy and feel happy.

LOGO environment supplies tools which focused on interaction between maker and the object, Scratch wider it socially and supplies a platform where they can interact and learn from each other. In that place, they could make a collaboration beyond the geographical limitation. In addition, even if they work on creative activities alone, it won't mean loneliness, because they feel there are many people who have a similar interest in the online community.

Also, Resnick focused on Constructionism proposed by Papert along with learning style of *Kindergarten* which Friedrich Froebel proposed and spread around the world in the 19th century. He thought learning style of Kindergarten has a good point which students of all ages could learn from it. So, he named his project "Lifelong Kindergarten" at MIT Media Lab and published his book with it as the title (Resnick, 2017). Resnick focused on "Froebel's Gifts" and positioned it as an origin of learning by "making" something.

> "Froebel wanted his kindergarten children to gain a better understanding of the world around them. One of the best ways to do that, he realized, was for children to create models of the world --- to 're-create' the world through their own eyes, with their own hands. That was the ultimate goal of Froebel's Gifts: understanding through 're-creation.'" (Resnick, 2017, p. 8)

In addition, he thought it is "recreation" as well as "re-creation" and picked up "kindergarten children are most likely to create and build when they are engaged in playful, imaginative activities" (Resnick, 2017, p.8) and say it is a creative learner. Resnick expressed the activity process which have done in Kindergarten as "Creative Learning Spiral" (Fig. 5).

The first step is "Imagine," where you will begin by imagining a world you want make. Next, you will move to "Create" step, where you make ideas into form. The next step is "Play," where you will experience it, and as a result you may sometimes realize that your product should be improved. Then, you will "Share" it with others and thus get participation and ideas of others. After that, you will "Reflect" the products and the points you couldn't make well. Furthermore, based on the experience, you start to new cycle from "Imagine" step. By repeating this Creative Learning Spiral, you enhance the power of creative thinking. Resnick thought project-based learning based on this Creative Learning Spiral is actually carried out and important not only for children at Kindergarten, but also for adults, including researchers at MIT Media Lab.

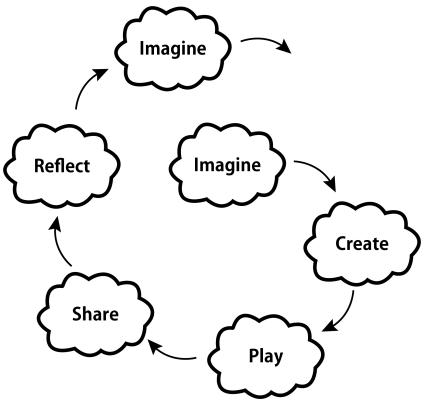

Fig. 5. Creative Learning Spiral (Resnick, 2017)

Moreover, Resnick proposes "four P's of creative learning" as four fundamental principles to become a creative thinker by these learning experiences. These four P's are: "Project", "Passion," "Peers," and "Play." The first P means launch a "project." In Scratch environment, you make a "project" unit to set a goal to aim what you want to make. In a project, you can work on by alone or with multiple people, or with someone in online community who physically apart from. The second P means we work on project by having "Passion." It is important to work on what we could have passion because there are various theme and styles of project. The

third P means the existence of "Peers." In making, it is important because existence of peers enables us to cooperate and share it and stimulate each other by showing works. The last P means "Play," which it is important to work on experimental challenge in a playful and fun way. You will see that this four P of creative learning is overlapping with what Creative Learning Spiral focused on.

What is said here is overlapping to not only community of Scratch but also to making in the sense of "Maker Movement," where the concept of "learning by making" or "learning through making" is spreading gradually in recent years (Martinez & Stager, 2013).

4. PATTERN LANGUAGE IN THE LIGHT OF CONSTRUCTIONISM

4.1 Patterns in a Pattern Language

In this section, we discuss how pattern language can be considered based on the Constructionism. First of all, pattern language is a language which describes the collection of rules of thumb in the field of designing or organizing some practices. Each practical knowledge in the language are called "pattern," which includes wisdoms and arts within a domain of expertise. Each pattern is basically structured in four parts: context, problem, solution, and consequence; it shows that, in what kind of "context", what kind of "problem" usually occurs, what good practitioner in the domain do as a "solution" to overcome the problem, and finally the "consequence" of the solution.

In addition, this is very important, "pattern name" is given to each pattern (Fig 6). More than as tips and techniques to improve the situation, pattern language provides a new *vocabulary* which can be used to think and communicate on good practice. With using words for practices, it becomes easier to think and communicate how to improve their situation. Note that while each pattern is written to improve action in the specific situation, the whole language is intended to improve the quality as a whole (Fig 7).

4.2 Learning Patterns from the viewpoint of Constructivism

When we look at pattern language from the viewpoint of Constructivism, it is clear that patterns cannot be directedly imported into the people's understanding, because recognition and knowledge must be constructed internally (Iba and Munakata, 2019). So, as shown in Fig. 8, learning patterns should be considered in the following way: First, patterns guide actions in certain situations, and improve the action to consequences. Through this experience, schéma and structure are constructed internally rather than importing the pattern description into the knowledge (Fig.8.).

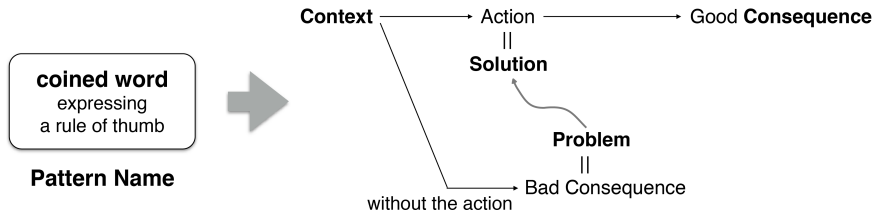

Fig. 6. Give words to context, problem, solution and consequence

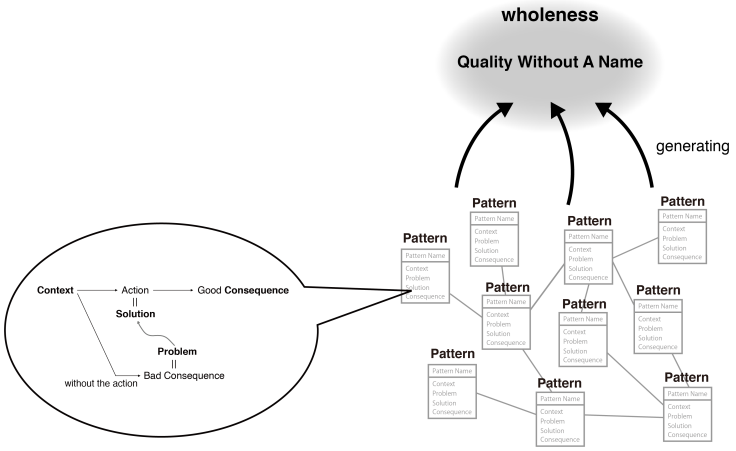

Fig. 7. Multiple patterns cooperate to enhance quality of practice in the whole

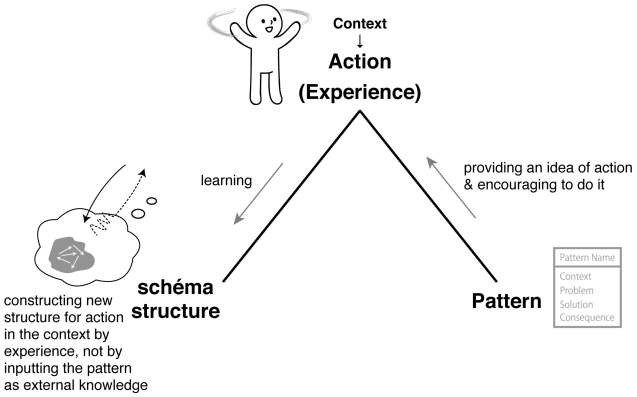

Fig. 8. How pattern languages contribute to constructing schéma and structure (Iba & Munakata, 2019)

4.3 Functions of Pattern Language from the viewpoint of Constructionism

We here discuss how pattern language functions from the viewpoint of Constructionism proposed by Papert. Firstly, by defining the good way of designing and practicing, pattern language support practice of making (Fig.9). For example, design patterns, like software design pattern (Gamma, *et al.*, 1995), provides the knowledge how to make good design and therefore people can actually do it with using them as hints. If using a pattern language for human actions (Iba, 2016) such as Project Design Patterns (Iba and Kajiwara, 2019) and Collaboration Patterns (Iba and Iba Lab, 2018), it provides hints to improve your practice itself, such as making something or collaborating with others.

Secondly, related to this, pattern language is also useful in debugging in process of making. Pattern language provides wisdoms of how we can improve the situation when the problem happens. According to Christopher Alexander, pattern language was devised to achieve piecemeal growth by "diagnosis and repair" (Alexander et al., 1975, Alexander, 1979) and for this reason, this is important function of pattern language.

Thirdly, when we make a collaboration with multiple people, pattern language become a common language with the members (Fig.10). This situation can be with people in the same space as Papert imagined, or with people in online separated geographically as Resnick imagined. In such a collaboration team, pattern language supports communication on design and practices by providing vocabulary.

In these ways, through empowered practices by pattern language, people get rich experiences and learn from that. This is an explanation on functions of pattern languages from the viewpoint from the Constructivism, especially Constructionism.

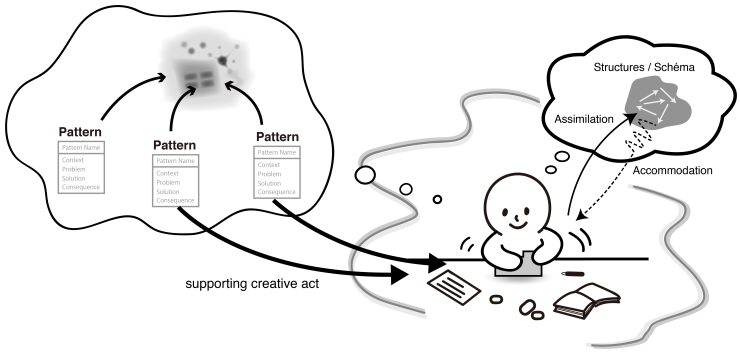

Fig. 9. Patterns support making and debugging

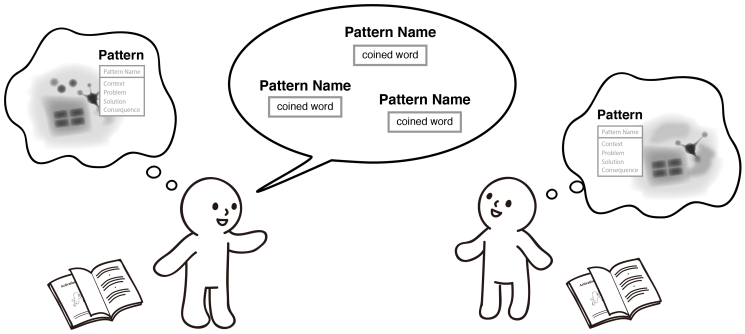

Fig. 10. Pattern Language functions as Vocabulary for Communication on design and practices

5. CONCLUSION

In this paper, as one of Constructivism theories, Constructionism proposed by Seymour Papert, especially learning by making and debugging. Then, we discussed how pattern language support practice and learning from the viewpoint. Note that in other papers of this series, we already discussed about the theories by Jean Piaget (Iba and Munakata 2019), Lev Vygotsky (Iba and Burgoyne, 2019a) and John Dewey (Iba and Burgoyne, 2019b) as Constructivist Learning Theories. Fig 11 shows an overview of the history of the constructivist learning theories, we discussed in this series of papers. Please also see these papers.

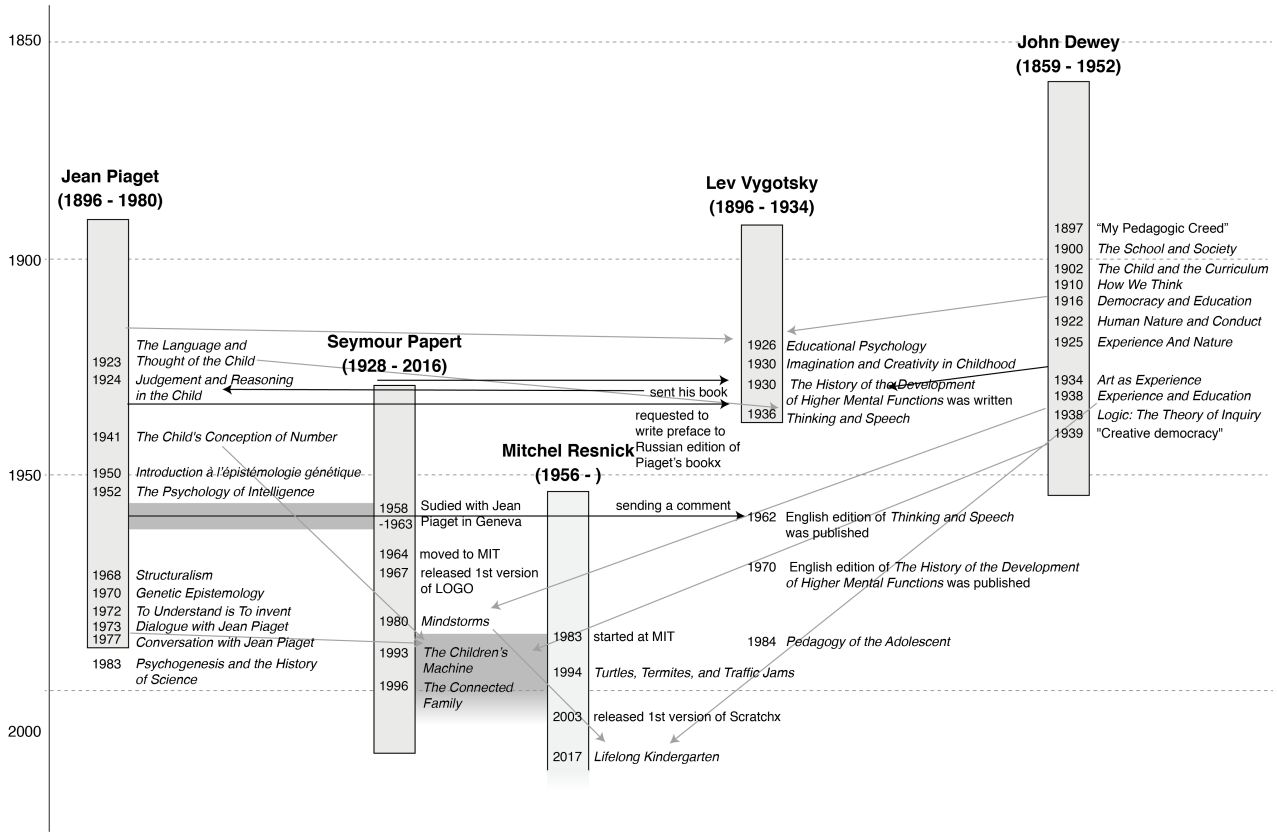

Fig. 11. An Overview of the History of Constructivist Learning Theories

REFERENCES

Alexander, C., Silverstein, M., Angel, S., Ishikawa, S., & Abrams, D. (1975) *The Oregon Experiment*, Oxford University Press.

Alexander, C. (1979) The Timeless Way of Building, Oxford University Press.

Martinez, S. L & Stager, G., (2013) *Invent to Learn: Making, Tinkering, and Engineering in the Classroom*, Constructing Modern Knowledge Press.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Iba, T. (2016) "Pattern Language 3.0 and Fundamental Behavioral Properties", in *Pursuit of Pattern Languages for Societal Change. Designing Lively Scenarios in Various Fields*, Peter Baumgartner, Tina Gruber-Muecke, Richard Sickinger (eds.), Berlin: epubli, pp.200-233

Iba, T. and Burgoyne, A. (2019a) "Pattern Language and the Future of Education in Light of Constructivist Learning Theories, Part 2: Consideration with Social Constructivism of Lev Vygotsky," *24th European Conference on Pattern Languages of Programs (EuroPLoP2019).*

Iba, T. and Burgoyne, A. (2019b) "Pattern Language and the Future of Education in Light of Constructivist Learning Theories, Part 3: Consideration with Pragmatism of John Dewey," *26th Conference on Pattern Languages of Programs (PLoP2019).*

Iba, T. and Iba Lab (2018) *Collaboration Patterns: A Pattern Language for Creative Collaborations*, CreativeShift.

Iba, T., Kajiwara, F. (2019) *Project Design Patterns: 32 Patterns of Practical Knowledge for Producers, Project Managers, and Those Involved in Launching New Businesses*, CreativeShift.

Iba, T. and Munakata, K. (2019) "Pattern Language and the Future of Education in Light of Constructivist Learning Theories, Part 1: Consideration with Genetic Epistemology by Jean Piaget," *24th European Conference on Pattern Languages of Programs (EuroPLoP2019).*

Kafai, Y. and Resnick, M. (eds.), (1996) *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*, Lawrence Erlbaum Associate, Publishers.

Papert, S. A. (1993a) *Mindstorms: Children, Computers, And Powerful Ideas*, 2nd edition, Basic Books.

Papert, S. A. (1993b) "Situating Constructionism", Idit Harel and Seymour Papert (eds), *Constructionism: Research Reports and Essays, 1985-1990 by the Epistemology & Learning Research Group,* The Media Laboratory, Massachusetts Institute of Technology, Ablex Publishing.

Papert, S. A. (1993c) *The Children's Machine: Rethinking School in the Age of the Computer*, Basic Books.

Resnick, M. (2017) *Lifelong Kindergarten: Cultivating Creativity through Projects*, Passion, Peers, and Play, The MIT Press.