Syntax and Semantics of Pattern Language

Ichiro Nagasaka*

*Kobe University
Kobe, Japan, nagakasa@kobe-u.ac.jp

Abstract: In this paper, we describe the overall picture of the pattern language by examining the literature in the 1960s based on the correspondence to the formal systems of mathematics. First, we briefly show the outline of pattern language. Second, we analyze the syntactical structure of the pattern language and show that the generating system and the patterns correspond to the syntactical system and inference rules in the formal system, respectively. Third, we examine the semantics of pattern language and show that Alexander’s definition of the design problems gives the semantic framework of the language. After describing the pattern language can be regarded as a syntactical object like proofs in the formal systems, we discuss the limit of the pattern language and give one possible reason why he needed to explore “geometric features” of forms generated by the patterns.

Keywords: Christopher Alexander, pattern language, syntax, semantics, formal systems

1. Introduction

Christopher Alexander is considered one of the founders of theory of design programming. His theory called “pattern language” has been influential in various domains including architecture and urban planning as well as software science.

The pattern language mainly consists of two concepts, namely, pattern and combination process (generating system). There are many literature and application about the pattern. However, there is comparatively little research on the combination process. Since the combination process is considered the grammar of the language, without understanding the nature of the combination process, the pattern language cannot be a “language”.

In Systems Generating Systems [1], Alexander showed the examples of generating systems, such as, formal systems of mathematics, system of natural language, genetic system, as well as the structure of the building system. As he noted, the genetic system is the best example of the generating system, but, since the system is extremely complex — maybe more complex than the generating system of the pattern language itself — it is not suitable to analyze the generating system of the pattern language based on the genetic system. Much the same is true on the system of natural language. On the other hand, the formal systems of mathematics are the simplest and the most clearly defined among the examples. Addition to this fact, Alexander once studied mathematics and often referred to mathematical systems especially in his literature of the 1960s, it would be beneficial to analyze the correspondence between pattern language and formal system of mathematics to understand the combination process.

Therefore, in this study, we will describe the overall picture of the pattern language by examining the literature in the 1960s based on the correspondence to the formal systems of mathematics. We first summarize the framework of pattern language, then, introduce the two faces of the formal systems, semantics and syntax, and identify the corresponding parts in the pattern language. Here, his definition of the design problems gives the semantics of the
language, and the generating system and the patterns plays the role of its syntactical system and inference rules, respectively. After describing how the pattern language can be regarded as a syntactical object, i.e., proofs in the formal systems, we finish by briefly discussing the limit of the pattern language caused by this framework.

2. Brief Introduction of Framework of Pattern Language

2.1 Definition of Design Problem

In [2, p.76] Alexander showed Figure 1 to explain the role of the designer. The context C1 and the form F1 in the figure are the real context of the actual world, and the physical form of an artifact in the context. C2 and F2 are the mental picture of them in the designer’s mind, and C3 and F3 are the system of the formal picture of the mental pictures, respectively. Here, he described C3 in the formal language of set theory and made his design methodology based on it.

In [3], he defined the design problem as follows [3, p.178]:

The ultimate object of design is form.

Every design problem begins with an effort to achieve fitness between two entities: the form in question and its context.

Thus, the design problem in his methodology is to find the way in which we acquire the form (F3) that fits in the context (C3) in the formal picture.

2.2 Definition of Force

To archive such fitness between them, the first thing to do is to define what Alexander called force. In [4, p.109], he states:

The first step in the process of design involves an explicit statement of the forces at work and the pressure pattern the form is to reflect. The designer’s tasks to create order: to organize conflicting material and to make a form.

To define such forces, we must define:

1. The exact circumstances under which the force arises.
2. The exact conditions which the force is seeking [5, p.96].

These forces generate form.

When a certain context is given, forces are always present. Alexander showed the wind ripples as an example of the form generated by such forces (Figure 2)[5, p.96]. After describing five forces (or tendencies) such as windward slope tends to “catch” the grains or the wind picks up more grains on a windward slope than on a leeward slope, and so on, he explains how the forces generate form.
These five forces make any level surface or any unevenly spaced pattern of bumps unstable. The slightest bump will grow into a ripple; and the ripples will repeat at regular intervals downwind, so that gradually a “wavelike” pattern of ripples is built up. With the wind blowing, the level sand surface is an unstable form because it gives rise to forces which ultimately destroy it. The rippled form is stable because the forces which it gives rise to maintain the form.

Thus, when there are conflicts between forces and their context, the forces are elicited. In above example, when the sand surface is level, there are conflicts between the form of sand and the forces around it. Consequently, the force destroys the level surface. Contrary, when the sand surface has rippled form, there is no clear conflict between the form of sand and the force. In this case, the form is stable and the forces are not elicited, in other words, the form fit in the context.

2.3 Combination of Forces and Invention of Forms

After defining the forces (tendencies), they are combined in the following way:

1. We try to determine, as abstractly as possible, the physical relation which each individual tendency is seeking.

2. We try to combine these individual abstract relational implications, by fusion, to generate the form [5, p.101]. Alexander insisted “once a conflict between tendencies is clearly stated, it is then possible to define the geometrical relation which is required to prevent the conflict” [6, p.2]. Therefore, the form which generated by the combination of such geometrical relations should not conflict with its context.

Pattern language is essentially the individual abstract relational implications with the rule of combining them. The rules are defined based on the definition of the force, i.e., if the circumstance under which the force is arise is present, then the condition which the force is seeking is formed. Alexander called the rule *pattern*.

Alexander stated in [1, p.605]:

*A generating system is not a view of a single thing. It is a kit of parts, with rules about the way these parts may be combined.*

and in [7, p.17],

The pattern language, is a system which shows how the patterns fit together, and helps the designer make a whole of them. The cascade drawing (Figure 3) is a rudimentary picture of the language for the 64 multi-service center patterns.
Summary: From Context to Form

Above explained framework of pattern language would be summarise as follows:

1. First, a certain context is given.
2. In the given context, there are always forces.
3. To define forces, we must define:
   (a) the exact circumstances under which the force arises,
   (b) and the exact conditions which the force is seeking.
4. These forces are combined by following the patterns and generate a form.
5. The generated form is always fit in its context.

3. Syntax of Pattern Language

In this section, we shall now examine the syntax of the pattern language based on the correspondence to the formal systems of mathematics. To describe the overall picture of the pattern language as abstractly and clearly possible, we investigate it based on a formal system of mathematics, namely, a system of predicate logic called natural deduction system (NK). Since natural deduction was presented as abstract system of predicate logic based on inference rules which are intended to capture ordinary human reasoning, and has more inference rules as compare to so called Hilbert-style deduction system, it is suitable to investigate the systems like the pattern language that has many rules.

3.1 Formal Systems of Mathematics as Generating Systems

In [1, p.605], he gave several examples of the generating system, such as, betting system, Montessori system, democratic system, formal systems of mathematics, system of language, genetic system, and building system. The formal systems of mathematics are the simplest and the most clearly defined systems among these examples. Here, he explained the generalized notion of a generating system based on the formal systems of mathematics:

We may generalize the notion of a generative system. Such a system will usually consist of a kit of a parts (or elements) together with rules for combining them to form allowable ‘things’. The formal systems of mathematics are systems in this sense. The parts are numbers, variables, and signs like + and =1. The rules specify ways of combining these parts to form expressions2, ways of forming

---

1 Specification of symbols used in the formal system.
2 Definition of expressions in the system
expressions from other expressions, ways of forming true sentences from expressions, and ways of forming true sentences from other true sentences. The combinations of parts, generated by such a system, are the true sentences, hence theorems, of mathematics. Any combination of parts which is not formed according to the rules is either meaningless or false.

As showing in the footnotes, this explanation contains almost all the aspects of the formal system. This means, as being a mathematician, Alexander had the extensive knowledge of the formal systems of mathematics.

The first part of a formal system is its language. We usually create artificial languages so that we can use and study the language in abstract and precise way. To specify a language, we have to specify its symbols first. In the case of English, the symbols are the alphabets, numbers, punctuation marks, and so on. In natural deduction, infinitely many symbols, such as variables \( x, y, \ldots \), function symbols \( f, g, \ldots \) and predicate symbols \( P, Q, R, \ldots \), are provided. Any finite sequence of symbols of a language is called an expression. In each language, certain expressions of the language are designated as the well-formed formulas of the language; it is intended that these be the expressions which assert some fact. Formulas without logical connectives or quantifiers are called atomic formulas which do not contain any simpler formula in them. When its symbols and formulas are specified, a language is considered completely specified.

The next part of a formal system consists of its axioms. The only requirement on axioms is that each axiom must be a formula of the language. However, it is generally required that the axioms assert facts which are always true. The third part of a formal system is the rules of inference. Each rule of inference states that under certain conditions, one formula — called the conclusion — can be inferred from certain other formulas — called the hypotheses. When a formula can be inferred from the finite chains of inference from the axioms, the formula is called a theorem and the finite chains of inference is called a proof of the theorem. When its language, axioms and rules of inference are specified, we consider a formal system totally specified.

3.2 Syntactical Structure of Pattern Language

Symbols and Expressions: Since the pattern language written not in artificial language but in natural language, the symbols used in the language are not specified.

Atomic Formulas (Atoms of Environmental Structure): In [6], Alexander explained the atoms of environmental structure as follows:

The atoms of environmental structure are relations. Relations are geometrical patterns. They are the simplest geometrical patterns in a building which can be functionally right or wrong [6, p.1].

As mentioned above, atomic formulas are the simplest formulas which assert some fact that can be true or false. In natural deduction, when \( P^i_n (i = 0, 1, 2, \ldots; n = 1, 2, \ldots) \) is a predicate symbol and \((t_1, \ldots, t_n)\) is a term, then \( P(t_1, \ldots, t_n) \) is an atomic formula. If formulas do not contain any free variables, the formulas can be true or false. In the case of the pattern language, the relation is \( n = 2 \) in above predicate symbols, such as:

1. Near(check_out_counter, exit_door).
2. Insideof(stack_of_baskets, entrance).

---

1. Definition of well-formed formulas.
4. Definition of the proofs in the system.
5. When the combined expression is not well-formed formulas.
6. When the combined expression is not outcome of the rules of inference.
7. Explanation of formal systems in this section is largely based on [8].
The relation can be true or false when the variables like “check_out_counter” are translated into real objects in the context.

**Axioms:** There is no clearly defined axiom in the pattern language.\(^{10}\)

Natural deduction also has no axioms. The proofs of natural deduction starts from a hypothesis and generally involve the making and “discharging” of hypothesis. For instance, in the case of the following proof:

\[
\begin{align*}
\frac{P \land Q}{Q} & \quad \text{\& E} \\
\frac{P}{P} & \quad \text{\& E} \\
\frac{P \rightarrow (Q \rightarrow R)}{Q \rightarrow R} & \quad \rightarrow E \\
\frac{R}{(P \land Q) \rightarrow R} & \quad \rightarrow I_1 \\
\frac{(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \land Q) \rightarrow R)}{(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \land Q) \rightarrow R)} & \quad \rightarrow I_2 \\
\end{align*}
\]

while the proof starts from hypotheses “\([P \land Q]\)” and “\([P \rightarrow (Q \rightarrow R)]\)” to the conclusion “\((P \rightarrow (Q \rightarrow R)) \rightarrow ((P \land Q) \rightarrow R)\),” these hypotheses are discharged at the application of the rule “\(\rightarrow I_1\)” and “\(\rightarrow I_2\)”\(^{12}\), respectively. Therefore, above figure is, in fact, the proof of the conclusion “\((P \rightarrow (Q \rightarrow R)) \rightarrow ((P \land Q) \rightarrow R)\).”

In the pattern language, by analogy with the proof of natural deduction, as far as all relations that contain conflicts are discharged at the application of certain rules, we do not need axioms.

**Inference Rules:** In [9, p.8], Alexander explains the process of obtaining relations.

We have described a process which has two steps:

1. Identifying a conflict.
2. Deriving a relation from it.

This process for obtaining a relation is objective in the sense that each of its steps is based on a hypothesis which can be tested. The two hypothesis are: 1. Under certain specific conditions, such and such potential conflicting tendencies occur. 2. Under these conditions, the relation \(R\) is both necessary and sufficient to prevent the conflict.

Then, he made derivation rules based on the process of obtaining relations in the way that “if you can identify a conflict then you can derive a relation from it” and called the rule “pattern”. He explained it as follows[7, p.15]:

\[\text{. . . each pattern has two part: the PATTERN statement itself, and a PROBLEM statement. The PATTERN state is itself broken down into two further parts, an IF part, and a THEN part. In full the statement of each pattern reads like this:}\]

\[\text{IF: X THEN: Z / PROBLEM: Y}\]

\(X\) defines a set of conditions. \(Y\) defines some problem which is always liable to occur under the conditions \(X\). \(Z\) defines some abstract spatial relation which needs to be present under the conditions \(X\), in order to solve the problem \(Y\).

In short, if the condition \(X\) occur, then we should do \(Z\), in order to solve the problem \(Y\).

In natural deduction, there are a number of inference (derivation) rules. For example, the inference rule called “\(\land I\)” (\(\land\)-Introduction) is expressed as follows:

\(^{10}\)The word “axiom” appeared in [1, p.610], but this axiom: “to ensure the holistic system properties of buildings and cities, we must invent generating system, ...” is axiom about his attitude toward design problems in general and not the axiom of pattern language.

\(^{12}\)Superscript figures indicate that these are the hypotheses of the proof.
This means that if sentences “\(P\)" and “\(Q\)" are provided at the same time, we can derive a new sentence “\(P \land Q\)”. Here, the logical connective “\(\land\)" that was not present before the application of the rule is introduced. That is why this rule is called \(^\land\)-Introduction.

The PATTERN statement would be expressed in similar style of natural deduction as follows:

\[
\frac{\text{set of condition } X}{\text{spatial relation } Z} \quad (\text{problem } Y)
\]

Using the terminologies of [5], it would be:

\[
\frac{\text{Conflicts}}{\text{Relations}} \quad (\text{Forces})
\]

### 3.3 Combination Process as Proving Process

In [10, p.38], Alexander explains the combination process of patterns:

The combination process is not unlike the process by which the leaves on a tree are formed. All the leaves are defined by the same morphogenetic rules: the individual leaves are formed by the interaction between these rules and the local conditions which the leaves are subject to. [...] The combination process works in the same way.

He also stressed that this process can be carried out by any trained draftsman since the rules of the combination process are almost mechanical.

And in [7, p.17],

The pattern language, is a system which shows how the patterns fit together, and helps the designer make a whole of them. The cascade drawing (Figure 3) is a rudimentary picture of the language for the 64 multi-service center patterns.

In showing examples of the application of patterns, he state:

For each example, the steps are presented in sequence (\(A, B, C, D, \ldots\)). Each step introduces new patterns into the design [7, p.19].

This sequence corresponds to the process of proving a formula where the formula is proved by a certain sequence of application of inference rules\(^{13}\). In this sense, proofs in natural deduction can be seen as a cascade of the inference rules which generate a conclusion “\((P \rightarrow (Q \rightarrow R)) \rightarrow ((P \land Q) \rightarrow R)\)” as shown in Figure 4(b).

The correspondence between combination process of pattern language and proving process of natural deduction would be clearly shown if you read a pattern in Figure 3 as an inference rule (Figure 4(a)) and compare it to the proof figure of natural deduction. Based on the correspondence, the pattern language can be regarded as a proof in the formal systems. This correspondence is indicated in the epilogue of [2]:

Man’s feeling for mathematical form was able to develop only from his feeling for the processes of proof. I believe that our feeling for architectural form can never reach a comparable order of development, until we too have first learned a comparable feeling for the process of design.

\(^{13}\)However, after mentioning this, Alexander noted:

One point must be heavily underlined. Although the evolution of these designs is presented in a step-wise sequential manner, this is merely for convenience of presentation. It does not imply that the design process generated by the language, is, in any but the most general sense, itself sequential.

Is is also applied to the case of the formal system. In text books, proofs are always presented in a step-wise sequential manner, but, mathematicians usually does not prove theorems in that way. However, since “natural” deduction system intended to reflect ordinary human reasoning, in the most general sense we prove theorem in the manner of proof in natural deduction.
4. Semantics of Pattern Language

Semantics of formal systems are generally defined in the following two steps:

1. First, define mathematical representations of the configuration of the world (context).
2. Second, define the conditions under which a sentence (formula) of the formal language is true in these mathematical configurations. This condition is called a truth condition.

The world, together with a truth condition, is sometime called a model.

As mentioned above, Alexander defined the design problem as follows:

The ultimate object of design is form.

Every design problem begins with an effort to achieve fitness between two entities: the form in question and its context [3, p.178].

and, the word “form”, “context” and “fitness” in the definition are explained as follows:

The form, then, is that part of the world which we decide to shape, while leaving the rest of the world as it is. The context is that part of the world which puts demands on this form; anything in the world that makes demands of the form is the context. In other words, the form is the solution to the problem; the context defines the problem. Fitness is the relation of mutual acceptability between these two. In a problem of design we want to satisfy the mutual demands which the two make on one another[3, p.179].

Here, in the light of semantics of formal systems, the words “form”, “context” and “fitness” correspond to the formal expression\(^{14}\), the mathematical configurations and the truth-value\(^{15}\), respectively. In the case of the pattern language, the real world is taken as the configuration and the truth conditions are defined as the circumstance in which no tendencies conflict, that is, a spatial relation (form) is true (fit) in the configuration (context).

Note that, in his development of design theory from [4] to [7], Alexander did not explicitly show the evaluation stage where the fitness of the spatial relation obtained by the combination process is evaluated, that is, there is no systematic way to tell whether the form in question and its context actually fits in his theory of design programming. This might imply that Alexander assumed some kinds of soundness of pattern language in the sense that classical first order predicate logic is sound, which means that the rules of inference of a formal system will never lead a false sentence from a true sentence. In other words, spatial relations that are obtained by patterns are always fit.

\(^{14}\)In [2, p.134], Alexander stated:

The shapes of mathematics are abstract, of course, and the shapes of architecture concrete and human. But that difference is inessential. The crucial quality of shape, no matter of what kind, lies in its organization, and when we think of it this way we call it form.

\(^{15}\)In [2, p.99], he noted: “A design problem is not an optimization problem. [...] It is a strictly binary situation.”
in the context just like sentences that have proofs are always true in any given models (world) in sound formal systems.

5. Overall Picture of Pattern Language

The following is a summary of the proving process of natural deduction (in the same way as we summarise the framework of pattern language in Section 2.):

1. First, a formal system and its world is given.
2. In the given formal system, there are a number of inference rules.
3. To define the rules of inference, we must define:
   (a) formula schemata to which a rule can be applied,
   (b) and a formula schema which can be inferred from above formula schemata.
4. These rules of inference are combined and generate a theorem.
5. The theorem is always true in the world.

This proving process is depicted in Figure 5.

By following the Figure 5, the syntax and semantics of pattern language is shown in Figure 6.

So far we have analyzed the framework of the pattern language. The analysis sheds light on some limitations of the pattern language. First, as there are many choices of rules of inference for mathematicians at certain stages of proving process, there are a number of choices of patterns for designers at some stages of combination process of pattern language and there is no clear criteria presented to figure out if designers are in the right courses in the combination process. Second, even if each form generated by a pattern fits in the context, the aggregation of the forms generated by fusion of spatial relations does not necessarily fit in the context. Therefore, it is not guaranteed that the pattern language is actually sound system.
These limitations suggest that there should be some evaluation process in combination process in the pattern language. In fact, Alexander realized this limitation after publishing *Pattern Language* and said in [11, p.127] “Up until that time, I assumed that if you did the patterns correctly, from a social point of view, and you put together the overall layout of the building in terms of those patterns, it should be quite alright to build it in whatever contemporary way that was considered normal. But then I began to realize that it was not going to work that way.” and he proceeded to an exploration of the criterion called “geometric features”. As he noted in [11] “You can use the geometry as a sort of litmus test, because the geometry will indeed change as a result of the life coming into it”. This test by the geometric features could be seen as the evaluation process that lacked in the pattern language.

6. Concluding Remarks
In this study, we described the overall picture of the pattern language by examining the literature in the 1960s based on the correspondence to the formal systems of mathematics. First, we analyzed the syntactical structure of the pattern language and showed that the generating system and the patterns corresponded to the syntactical system and inference rules in the formal system. As a result of this analysis, we showed that the pattern language can be regarded as a proof in the formal systems. Second, we examined the semantics of pattern language and showed that Alexander’s definition of the design problems gives the semantic framework of the language. After describing the pattern language a proof, we finish by briefly discussing the limitation of the pattern language and gave one possible reason why he needed to explore “geometric features” of the forms generated by the patterns.

Acknowledgements
This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 19560618, 2006.

References