

# Pattern Diagnostic System

## A Diagnostic Approach to Pattern Applications

Taichi Isaku

Faculty of Environment and Information Studies, Keio University

Koji Yamazaki

Faculty of Environment and Information Studies, Keio University

Takashi Iba

Faculty of Policy Management, Keio University

### Abstract

In this paper we propose the idea of a Pattern Diagnostic System. This is an online system that diagnoses users through a series of questions based on a pattern language, and visualizes what parts of the pattern language they are able to do now, and what kinds of patterns could be taken in for improvement. The system guides users through user-set projects – for example: “Learning Spanish” – in terms of patterns, and helps them gain growth. Users of the system will enter short episodes of experiences with the pattern, which the system will systematically synthesize into a visual map which shows successful and unsuccessful parts of the project. By engaging in many projects with the system, the user will naturally acquire skills that are embodied in a pattern language. This is a new approach to the practical use of pattern languages, and is based on the *principle of diagnosis* stated by Christopher Alexander in his book *The Oregon Experiment*. Users of the system will be able to visually see and how their ideal self and the current self changes over time, and this repeated cycle of diagnosis and repair will help the person achieve piecemeal growth to improve.

### 1. Introduction

Here we introduce the concept of the Pattern Diagnostic System. This is an approach to the practical application of pattern languages. The online system diagnoses users through a series of questions based on a pattern language, and provides results of the questionnaire visually mapped out. The system gives users a new perspective on how well they are performing the specific

actions the pattern language states, and provides them with findings on how they can improve. Through the repetitive use of the diagnostic system, users will be able to gain piecemeal growth, and improve their skills based on the pattern language. The system in this paper will use the *Learning Patterns* [ 1 ] [ 2 ], a pattern language for supporting learning, as the first prototype<sup>1</sup> of the system, but keep note that the general idea of the Pattern Diagnostics System is flexible and can be applied to any other pattern languages.

## 2. Background

Christopher Alexander in his book *The Oregon Experiment* has emphasized the importance of diagnosis using patterns when designing a building or a town [ 3 ]. In the book, six fundamental principles for implementation of patterns when designing a town are introduced. For a town to have the liveliness that meets the needs of its inhabitants, it must have what Alexander calls an organic order. This order is achieved not with the existence of a master plan that maps out the future image of the community, but is emergently created by a series of small projects guided by a communally founded pattern language. Citizens participate in this building process using patterns by making all decisions about the design for themselves.

### 2.1 The Principle of Diagnosis

One indicator of progress for the design of a town is found by looking at how well patterns are implemented. The *principle of diagnosis*, one of the six principles of implementation explained in the book, gives an idea of how this could be done. Alexander describes:

*The well being of the whole will be protected by an annual diagnosis which explains, in detail, which spaces are alive and which ones dead, at any given moment in the history of the community. To this end, the planning staff, working together with the people who use individual spaces, shall prepare annual diagnostic map for the entire community...*

The pattern maps show a map of the town colored on a red-yellow scale based on how successfully a pattern is applied to each area. Those areas colored yellow are areas where the pattern successfully exists and solves its problem. As the color turns orange and nears red, areas where the pattern exists but still has problems that are unsolved are indicated. These pattern maps are made for each pattern in the language, and the maps are then composited into one single map showing the health state of the whole community. By looking at this map, users can visually see

---

<sup>1</sup> As a pre-prototype of a diagnostic system using pattern languages for human actions, the Iba Laboratory has released the Generative Beauty Diagnostic system to diagnose users based on the Generative Beauty Patterns [ 7, ]. Details of the system and its outcomes can be read in the Appendix of this paper.

what parts of the community are healthy, and much more importantly, what parts need improvement. Once these parts are noticed, then the community can work to repair the parts that need work. This process is similar to that of an organism's internal system:

*The organism, from the very beginnings of its life, is constantly monitoring its internal state. In particular, those parts of the organism where critical variables have gone beyond their allowable limits are identified. We may call this the diagnosis. In response to the diagnosis, the organism sets in motion growth processes to repair this situation.*

It is this cycle of diagnosis and repair using patterns that keeps a community of people under control and healthy. Through the repeated process, the pattern map becomes renewed and updated every time a diagnosis is done.

## **2.2 The Spread Usage of Pattern Languages**

Recently, pattern languages have shown expansion in the field it covers. From Alexander's original theory of pattern languages to design a community, the idea has spread to be used for writing patterns found in software design. In more recent times, its application has extended even further to cover more abstract ideas: human behavior and actions. *Fearless Change* by Mary Lynn Manns and Linda Rising presents patterns for introducing new ideas into a community [ 4 ]. It is a well-known publication of a pattern language despite its unique theme of ideas it covers. Many of the pedagogical patterns that have recently been written are also good examples of such patterns[ 5 ].

The spread of the idea of pattern languages allowed patterns to be written in many different areas of profession. Since many of these pattern languages are easier to read and to start creating than those patterns for architecture or software, patterns can now be created and updated by a wide range of people at a rather fast rate. In such a situation, philosophical backgrounds such as those mentioned in *The Oregon Experiment* can be easily forgotten. Neglecting these principles may not necessarily be harmful, but bringing back such original principles of patterns into modern contexts will sure bring up user experience with the patterns.

## **3. Philosophy of the Pattern Diagnostic System**

The Pattern Diagnostic System is an effort of a new approach to the practical use of pattern languages. Though most of the pattern languages written about human behaviors and actions are complete by themselves, there are still struggles for the practical applications of them. This is especially true with patterns written about abstract ideas where in some cases the patterns are made, but no further action is taken. To tackle this situation, the system provides users with ways

to practically use a pattern language by diagnosing the user's habits and tendencies with a set of patterns.

When a person is trying to improve on something, having an abstract and flexible system to guide his growth will be helpful. Former improvement plans tended to be set around a clear goal, where we were made to calculate backwards from the goal to plan out a schedule to achieve it. This is exactly what Alexander criticized as a master plan. It is impossible to predict how things will change in the future and setting clear goals based on the uncertainty is foolish. As Alexander's patterns did with architecture, all other patterns too should be able to adapt to this kind of fast changing environment if the *principle of diagnosis* is effectively applied.

The best of the plans are the ones that are constantly improvised. Though thoughtful planning and systematic scheduling are effective, they must not become too rigid that they cannot accommodate for unpredicted changes in circumstances and become obsolete. Plans must give a person a direction to head for, but still have the flexibility to accommodate for changes in circumstances. The purpose of the Pattern Diagnostic System is to help actualize such a process.

Like the organic order in an organism, order in a person's life is formed by a cycle of diagnosis and repair. We can imagine ourselves being successful and becoming an ideal person in the future. These images may at most bring up our motivation, but in most cases, they are too vague and unrealistic. Specifics on what the ideal self can do and what the current self cannot do are hardly clear. These are exactly what the Pattern Diagnostic System tells its users. Users will be able to understand specifically what patterns they are successfully utilizing, and what patterns they could take in to improve. Based on these diagnostic results, the users will find pattern recommendations by the system to repair their habits. Through its repetitive use, the users would be guided through a process of diagnosis and repair to achieve piecemeal growth to become closer to their goals.

A work by Richard P. Gabriel suggests another perspective of looking at the system. In his 2008 paper *Designed as Designer*, he describes how the designed object self-referentially becomes its own designer [ 6 ]. His work gives poetry as an example where the first draft of a poem itself shows points that could be refactored, which then the new refactored version again gives more places that could be fixed. The same process could go on with humans. Introspection is usually difficult to do just alone, but the Pattern Diagnostics System gives its users a visual representation of themselves in terms of the patterns. By looking at this visual, the user will be able to notice by themselves places that could be improved. This way of designing will bring *Conceptual Integrity* - wholeness and seamlessness in design - to the product, or, in this case, the self.

## 4. System Details

We here will describe the specifics of the Pattern Diagnostic System. Its systematic overview can be seen in Figure 1. The system helps the users capture themselves in terms of a pattern language, and to see how they change and develop in performance over time. Users will enter short episodes about their experiences with the patterns into the system, which will be organized systematically to keep track of their growth. The user will also be able to see episodes by other users to gain hints for their own growth.

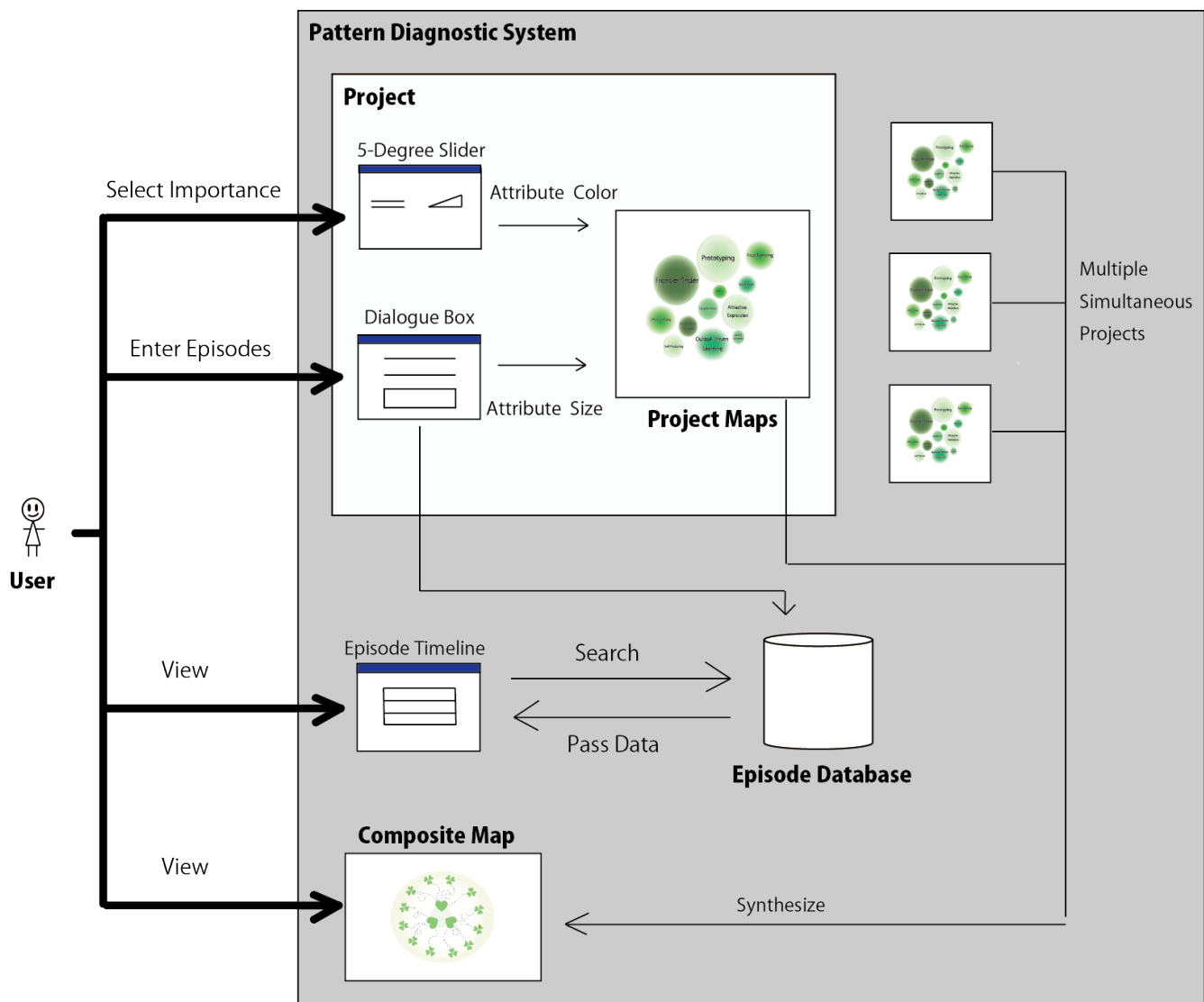


Figure 1. System Overview – summary of user interactions with the system

## 4.1 Creating a Project

When the users first create an account, they are asked to create a project they would like to use the patterns with. For example, when using the system with the *Learning Patterns*, they could make a project named "Learning Spanish".

After the project is named, the user then will be guided through each pattern in the pattern language to diagnose what they are already able to do. In the diagnosis, they are shown the patterns - mostly the solutions - paraphrased shortly. Under the sentence they will see a slider that has five degrees of volume. After reading the sentence, they would decide how important they think the pattern is for the project on the 5-degree scale: 5 representing “very important”, and a 1 representing “not important”. This selection is completely subjective and can be- in most cases, will be – changed later.

Pattern #4 - Jump in

You can jump into an new environment to learn without hesitating.

1 2 3 4 5 6

Not Important Important

Episode

Enter your experience here

Characters left: 100

**Figure 2. Diagnostic questions based on the patterns**  
**Episodes of past experiences can be typed into the dialogue box**

Then they would decide if they already have experience concerning the pattern. If they do, they can click on the dialogue box next to the slider to enter a brief episode concerning the pattern. Episodes are limited to 100 letters in length, so they should be concise but take little effort to type in. The system will then add two tags to the episode before saving it: one tag representing which pattern the episode is about and another which tells which project the episode is about. This phase exists to help the users get started with their collection of episodes, and also lets the users become familiar with the patterns.

Episode

I couldn' t speak any Spanish but I joined the spanish club at my school.

→

Episode

I couldn' t speak any Spanish but I joined the spanish club at my school.

#JumpIn

#LearningSpanish

**Figure 3. Patten tags and project tags are added to posted episodes**

## 4.2 Top Page

After going through the preliminary diagnosis, the user is brought to the top page. This is also the first page the user would see from the second time logging on. Here the user can see a couple of different information.

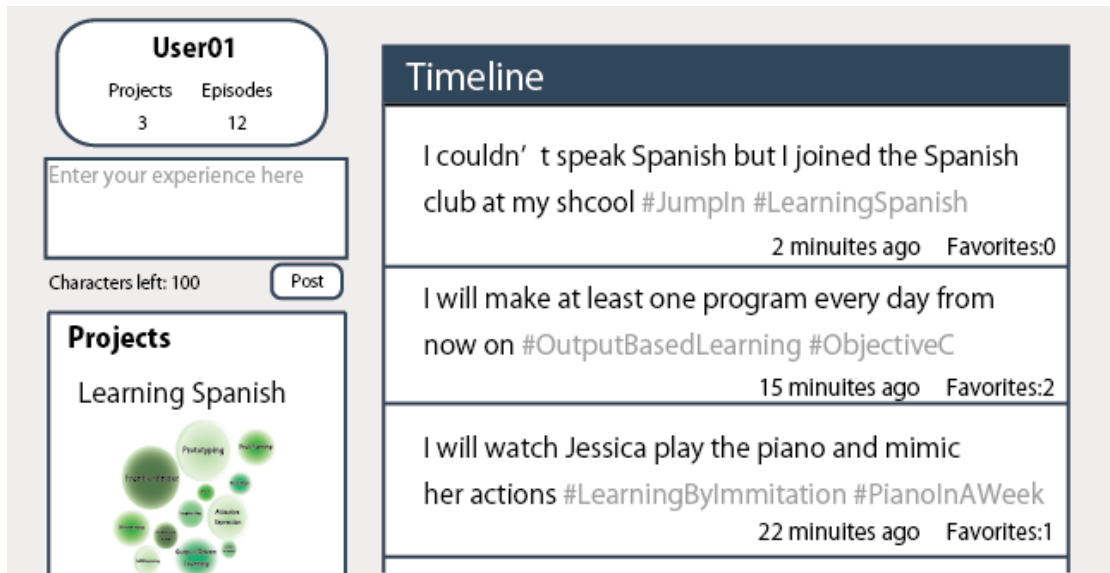


Figure 4. Mock image of the top page

### 4.2.1 Episode Timeline

In the center and the main part of the screen, a timeline of recent episodes is displayed - all episodes from all users. Users have the option to make their episodes private when they are posted, but those marked public are all shown here. The episodes are each marked with a tag showing which project the episode is a part of, along with another representing which pattern the episode is about. Episodes can be narrowed down and displayed based on the tags, allowing users to see episodes about a certain project, or about a specific pattern. Tags can be selected by directly clicking a tag within an episode, or by choosing from a list of tags shown along the sidebar of the screen.

Keep in mind here that the episodes are all anonymous, meaning that user names are not displayed with the episodes. Only some metadata such as the time posted can be seen. The tags on the episodes have clickable links, where when a user clicks on a tag, all public episodes with the same tag attributed are searched and shown in a single timeline for easy browsing.

### 4.2.2 User Division

The timeline occupies most of the top page, but some other actions can also be taken here. On the left sidebar of the screen, the user information is displayed. They can see the pattern maps (described in specific in section 4.3) of their projects that are currently activated, along with

recent episode posts by themselves. When the user name is clicked, the user is brought to a more specific user information page. The pattern maps of the projects are also clickable, bringing the user to the individual project's page.

### 4.2.3 Episode Dialogue Box

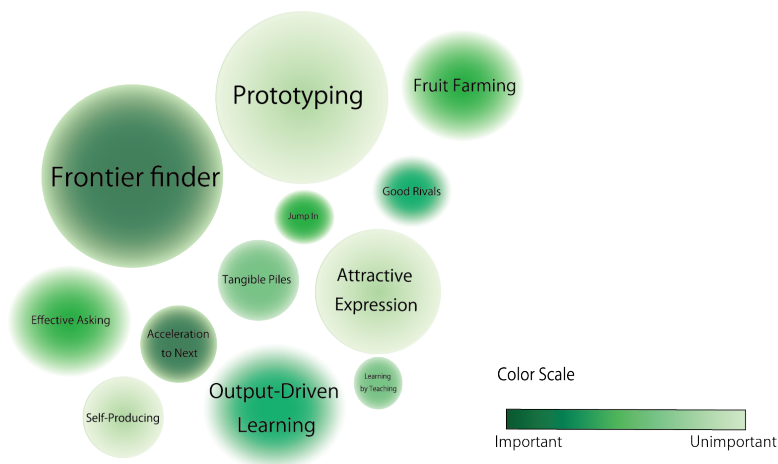
Below the user information, a dialogue box for posting a new episode exists. Here the user can enter an episode they have recently experienced, which can then be posted to become part of the timeline of episodes. Tags specifying project names or patterns can be easily added using a clickable selection system and an autofill support for entering hash tags.

## 4.3 Project Page

In most cases, users will likely have multiple projects going on simultaneously - for example having "Learning Spanish" and " Learning Objective C Programming" progressing at the same time – a separate page exists for managing each project. The progress of projects can be seen at the project's individual pages. Here the pattern maps for the specific project - the Project Maps - is shown. Alongside, a dialogue box exists on the page where the user can post an experience for the project. When done so, a hash tag for the specific project will be added automatically.

### 4.3.1 Mapping out the Pattern List

The Project Maps shown on the project page is created based on the diagnostic results of the user. To keep the system applicable to any pattern language, the pattern maps are kept simple with each pattern represented by a circle. For each pattern in the pattern language, a corresponding circle exists, colored based on the 5-degree importance scale the user answered in the diagnosis. Patterns that the user thinks of most importance will be colored a bright and bold green, and as the degree lessens, the color becomes light and nears a faint green almost close to clear.



**Figure 5. An sample Project Map – Patterns are shown in a green scale in different sizes**



At first all circles are shown the same size. As the user accumulates experiences and posts more episodes, the pattern map will start to change its appearance. The size of the circles will be proportional to the number of episodes that the user posts. As more episodes are posted for a pattern, its corresponding circle will grow in size.

Through this pattern map, the users can visually see in a color scale what parts they are able to do, and what parts can be worked on. This pattern map can be rearranged in different orders so the user can see the diagnostic results from different perspectives. For example the circles can be placed in order of ascending brightness of green (order of importance) or in order of size (order of number of experiences). The circles representing the patterns are all clickable to jump to a pattern page specific for each pattern. Details are described in section 4.5.

### **4.3.2 Past Project Maps**

Since the Project Maps are decided by the user's data, its appearance will change constantly. The system will check weekly the data of the pattern maps and store them in a database unique to the user in data form, so past pattern maps can be regenerated. On the project page where the user can see their current Project Map, a list of past Project Maps is also found. Users can scroll through the list to see its transition. By looking at how its color and form has transitioned over time, the user can notice how they have improved and changed over the repeated use of the system. A sample screenshot of the system in Figure 5 above shows how users can see pattern maps from past diagnoses and compare its transition.

## **4.4 User Page**

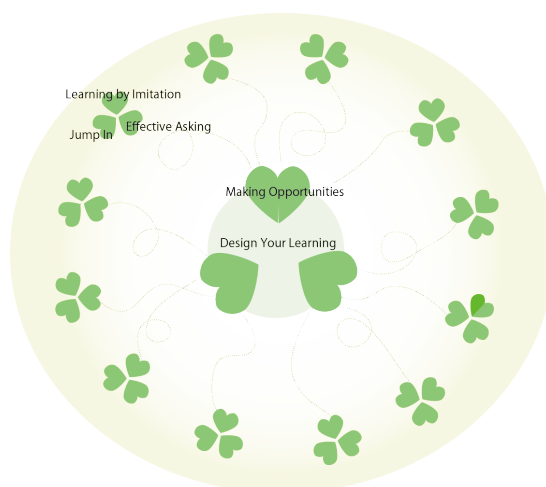
The user page shows user information that is not specific to any single project going on. This page allows the user to see all of their past posts, along with their "Composite Map" - a map which puts together all Project Maps of the user.

### **4.4.1 Creating a Composite Map**

Through the multiple projects that the user diagnoses, the user will accumulate many experiences with the patterns. Synthetic information about the patterns throughout all projects are collected into a single pattern map – the Composite Map. The number of experiences for all projects is counted up for each pattern to measure its level of expertise. The Composite Map, like the Project Maps, is shown in a green scale according to the importance for the user, with those in most importance shown in bright bold green. Its size too, grows larger as the user posts more episodes about the corresponding pattern.

There is one more aspect the Composite Map takes into account that the Project Maps do not. Some patterns in a pattern language are by themselves too abstract to measure its progress. For example, pattern #0 *Design Your Learning*, #1 *Making Opportunities*, #2 *Creative Project*, and

#3 *Open-Process Learning* from the *Learning Patterns*[ 1 ] cover a very wide topic and is too ambiguous by itself. This is because these patterns refer to life-long learning skills that are not easily met with immediate small actions. However, when its relationships with the other patterns are considered, it can be noticed that these patterns each contain other patterns within the area of concepts they cover. Pattern #1 *Making Opportunities*, for example, contain #4 *Jump In* and #6 *Effective Asking* along with many other patterns within its category. If the patterns within the categories are achieved, it can be safely said that the mother pattern is also well achieved. Hence, the Composite Map will collect experiences from all user projects to measure how effective these ambiguous mother patterns are taken in. Therefore the composite map will have a somewhat different appearance than the Project Maps, with some patterns placed within another. The composite map for the *Learning Patterns* is shown in the figure below. It can be seen how *Jump In*, *Learning by Imitation*, and *Effective Asking* are a part of *Making Opportunities*, which then is a part of *Design Your Learning*.



**Figure 6. Composite Map for the Pattern Diagnostic System using the *Learning Patterns* – It can be seen how some patterns are nested within another.**

According to this Composite Map, the user can see and decide what specific parts they can put effort in as a whole. Since some patterns are hard to apply in some projects, the Project Maps may contain circles which never turn a bold green. This might suggest the user that they need to work more on the specific pattern, but that may just be for the specific project. Efforts for the same pattern may be plentiful in another project. The Composite Map gives the user a more synthetic perspective to suggest a pattern they can work on. The Composite Map is in a sense a representation of the user's current state. By looking at the pattern map, the user will notice parts of the map that are still light in color, which are places where they could work on. In a future diagnosis, the user will see an updated version of the pattern map and notice how the colors have changed.

#### 4.4.2 Past Posts by the User

In addition to the composite pattern map, the user page will also show all past posts by the user in a timeline format. These posts can be displayed by project or by pattern according to its need. Also, if needed, episodes can be edited or deleted from here.

#### 4.5 Pattern Page

All pattern maps - both the Project Maps and the Composite Map - are clickable to bring the user to the pattern pages. A pattern page exists for all patterns in the pattern language, and the circles on the pattern maps representing the patterns will be linked to bring the user to the corresponding pattern's pattern page. Here they can see the contents of the pattern, along with all episodes that users has posted about the pattern. Some statistical information, such as the dates they were posted and how its numbers transitioned can also be seen. A dialogue box for posting an episode is also found on the page. When posted from here, a tag for the pattern on the current pattern page will be added on automatically.

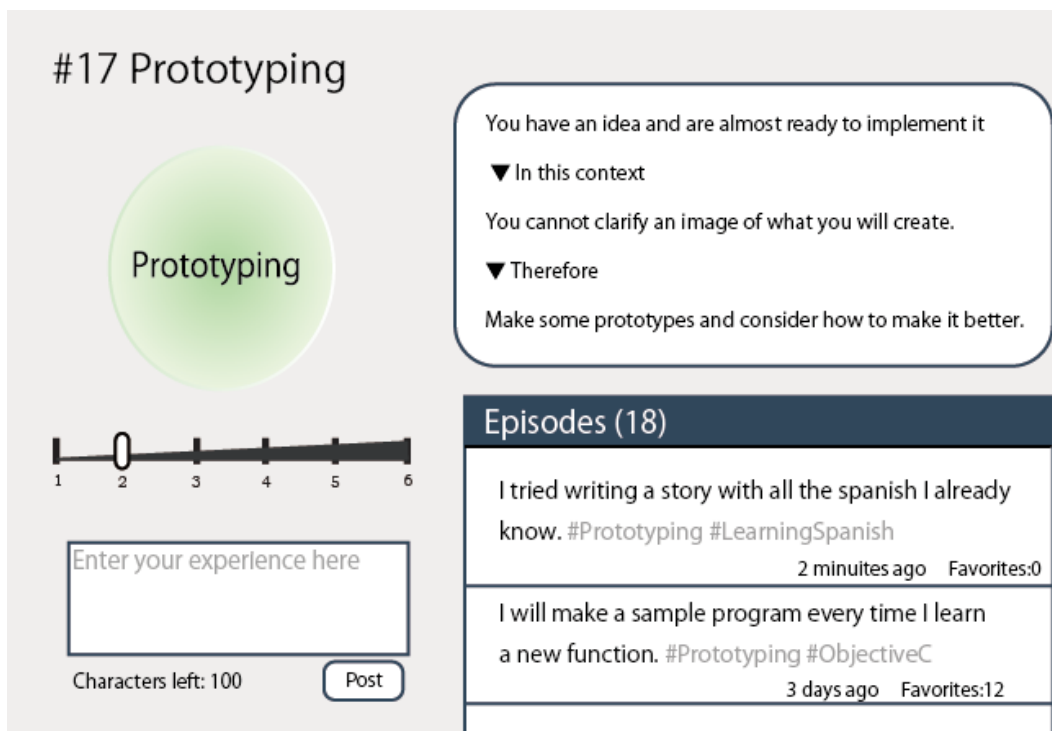


Figure 7. Mock image of the pattern page

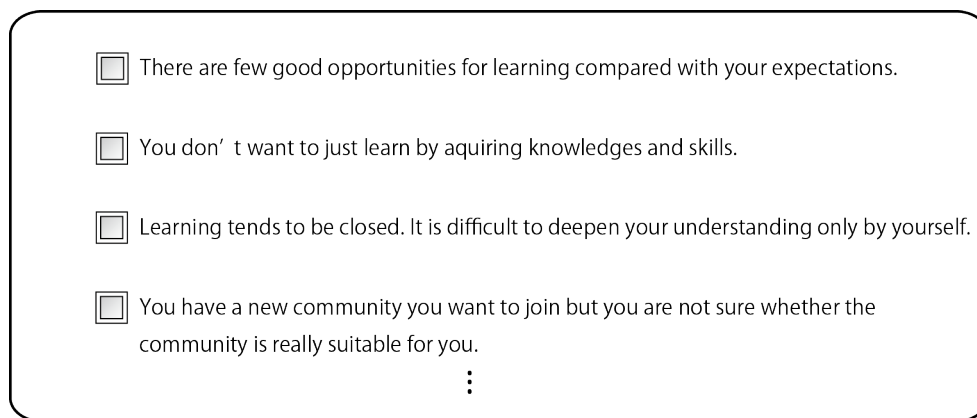
#### 4.6 Miscellaneous Functions of the Pattern Diagnostic System

In this section, we will describe miscellaneous functions of the Pattern Diagnostic System that were not described in the previous sections. Although these functions are supplementary, they add important contributions to user friendliness of the system.

#### 4.6.1 Supplementary Diagnosis for Finding Patterns to Work on

When just looking at the patterns as a list, it may be difficult to choose and exit which patterns are important for you. Especially when there are many patterns in the language, it is tiring and effortful to look through and decide which patterns are needed right now. In such cases, this supplementary diagnosis will be helpful.

The diagnosis provides users with a checklist that they would check if the sentence written applies to them. This is a list of the problems from the pattern language collected. As if a patient would check the symptoms they have before seeing the doctor, the user would click the checkbox next to the problems that apply to them.



- ☐ There are few good opportunities for learning compared with your expectations.
- ☐ You don't want to just learn by acquiring knowledges and skills.
- ☐ Learning tends to be closed. It is difficult to deepen your understanding only by yourself.
- ☐ You have a new community you want to join but you are not sure whether the community is really suitable for you.

⋮

**Figure 8. A checklist of problems to be checked for diagnosis**

After submitting the checklist from to the system, the system will give back the users a list of patterns which they had placed a check in. If they are having the problem written in the pattern, it is most likely that the pattern will be helpful for them. The user can then check to see the solution of the pattern will be helpful for them. If they decide that they need the pattern, they can change the position of the importance slider.

#### 4.6.2 Favoriting

If when the user is browsing through the timeline of episodes, and finds an episode that is interesting or helpful, they have the option to "favorite" the episode. Favorited episodes can be seen later on in a single place for reference. Although episodes are all anonymous, when an episode is favorited, the user who posted the episode receives a notification that their episode has been favorited (they will not know which user has favorited it). Each episode will have metadata attributed to it showing how many favorites it has received.

#### **4.6.3 Notifications / Setting Goals**

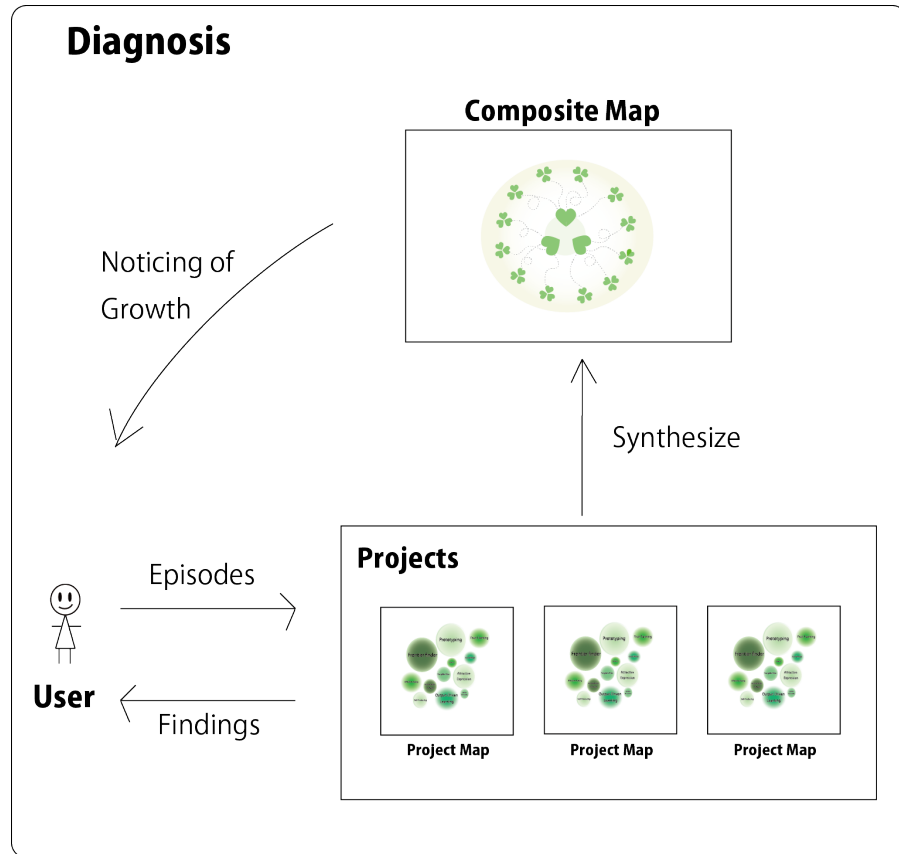
When looking at the transition of the pattern maps, it is sometimes hard to notice the patterns that haven't changed colors for a while. We get distracted with the changes, and tend to become blinded from the parts that don't change. This function of the Pattern Diagnosis System gives notifications when patterns are left without having any episodes posted about it for a long time. It may simply be that the user has forgotten about the pattern, or the pattern may be at a position hard to notice within the pattern map. This function will also suggest users to change the importance of a pattern, if there exists a pattern with many episodes (has a large circle on the pattern maps) but has a low importance (its color on the pattern map is light). If so many episodes arise about the pattern, it is most likely that the pattern takes up an important part of the user's thoughts and actions. These changes are suggested to users once in a while in hope to give the user such a finding.

The system also lets users set a time span for achieving a pattern. For example, a user can set a goal of posting 3 episodes about a pattern within 2 weeks. The notification system will support the user achieve these goals.

## **5. Summary**

Summarizing the Pattern Diagnostics System, its simplest diagnosis process can be represented by the cycle in figure10.

The user works on the process of repairing bad habits with the individual Project Maps. They would look at a Project Map, and notice places that could have some improvements done. When the improvements are made, the user would have gained new experiences with the patterns. These can be posted as episodes, which then would update the Project Maps. Here the cycle of diagnosis and repair occur for the individual projects. As the Project Maps continually becomes updated, the Composite Map which synthesizes all of the Project Maps into a single view will also become updated. It is this growth that the user finds on the Composite Map, that shows how much the user has grown, since the Composite Maps are a mere representations of the users themselves. The growth they find should become motivation for the users to start a new project, or to work harder on an existing one. The positive cycle will help keep a healthy attitude towards learning.



**Figure 9. Summary of the function of the Pattern Diagnostics System**

## 6. Conclusion

The best of the plans are the ones that are constantly improvised. The Pattern Diagnostic System is an effort to support people to go through an improvement process by providing a way to use patterns in a natural and convenient way. In the current state the Pattern Diagnostic System, the website only covers the Learning Patterns. Our view for the future of the project is to add a diagnostic system generator function where users can make a diagnostic system with their own pattern language on the site.

## Acknowledgements

Our greatest thanks goes to our Shepherd Michael Weiss for providing thoughtful comments on the paper insightful ideas for the Pattern Diagnostic System. We would also like to acknowledge the Pattern Diagnostic System team members Natsumi Noguchi and Satomi Oi for their many contributions to the Pattern Diagnostic System.

## References

- [ 1 ] Takashi Iba, Toko Miyake, Miyuko Naruse, and Natsumi Yotsumoto, "Learning Patterns: A Pattern Language for Active Learners", 16th Conference on Pattern Languages of Programs (PLoP2009), 2009
- [ 2 ] Takashi Iba and Toko Miyake "Learning Patterns: A Pattern Language for Creative Learners II", 1st Asian Conference of Pattern Language of Programs, Mar. 2010
- [ 3 ] Christopher Alexander, Murray Silverstein, Shlomo Angel, Sara Ishikawa, and Denny Abrams, *The Oregon Experiment*, Oxford University Press, 1975
- [ 4 ] Mary Lynn Manns and Linda Rising, *Fearless Change: Patterns for Introducing new Ideas*, Pearson Education, Inc., 2005
- [ 5 ] Pedagogical Patterns Editorial Board, *Pedagogical Patterns: Advice for Educators*, Joseph Bergin Software Tools, 2012
- [ 6 ] Richard Gabriel, "Designed as Designer", Object-oriented Programming, Systems, Languages, and Applications (OOPSLA'08), Nashville, Tennessee, USA, Oct., 2008
- [ 7 ] Rinko Arao, Ayano Tamefusa, Megumi Kadotani, Kaori Harasawa, Shingo Sakai, Keishi Saruwatari, and Takashi Iba, "Generative Beauty Patterns: A Pattern Language for Living Lively and Beautiful," The 19th Conference on Pattern Languages of Programs (PLoP2012), Arizona, USA, Oct., 2012

# Appendix: The Generative Beauty Pattern Diagnostic System

In November of 2012, the Iba Laboratory developed and released a trial diagnostic system using the Generative Beauty Patterns. The Generative Beauty Patterns is a pattern language written for people to achieve a life lived both lively and beautifully, and was made by the collaborative work of the Iba Laboratory and Kanebo Cosmetics Inc. This diagnostic system was released in Japanese for the 2012 Open Research Forum, hosted by the Keio Research Institute at SFC every fall to present the fruits of the research conducted on the campus to a wider audience. The event attracted thousands, and over 800 visitors were diagnosed.



**Over 800 visitors were diagnosed during the 2-day event in 2012**

This prototype version of the diagnostic system diagnosed users based on rather questions based on the 48 patterns in the *Generative Beauty Patterns*. The users were guided through the patterns in order, and presented with a short question asking if they were able to use the pattern in their lives. The questions are asked by presenting the solution of a pattern in a rather straightforward tone. For example, the pattern *Role Model* has the solution written: “choose a person who you can admire, and find parts of their philosophy or lifestyle that you can take in and learn from”. The question for this pattern will say: “Do you have a person who you admire, and are taking in and learning from parts of their philosophy or lifestyle?”. The user will then reflect on their lifestyle and choose one of three options: “yes”, “no”, or “not important”. This diagnosis screen can be seen in the image below.

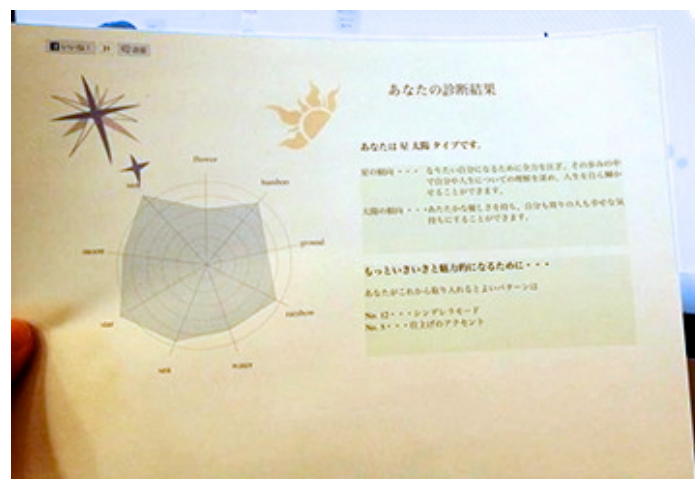
Based on these questions, a pattern map for the user is created. For this specific pattern language, we adopted a radar chart to represent the pattern map. Since the patterns in the *Generative Beauty Patterns* are categorized into 9 different groups based on their meanings, the





**The diagnosis screen presents the user with a question asking if they are usually doing the solution of a pattern. The user will then select “yes”, “no”, or “not important” to answer the question.**

radar chart held 9 points of evaluation. For each category, the interval increased by one for each pattern the user answered a yes to. The radar chart was made so the user can see what categories they are good at doing and, more importantly, what parts need more work. They would notice dents in the radar chart, and would go look at the patterns in that section to see if they could use any of them to improve their life. To help stimulate the process, the system tells its users what kind of a person they are based on the category of patterns they are currently able to do the most. In addition, recommendations for patterns were made from the category they had the most trouble with, excluding the patterns they had answered “not important” to. The recommendation was made to urge the next actions in the users so they would not stop with just the diagnosis in the diagnosis-repair cycle.



**The diagnosis result radar chart was printed out were handed out to each person.**

Though this system was somewhat successful, it contained many issues for its function as a Pattern Diagnostic System. First, the diagnosis questions were not asked and answered in the best way. For example, when asked “Are you eating healthily?”, the user would most likely hesitate to answer “not important” to the question. Similarly, since all of the patterns suggest some good tip to life, clicking the “not important” button would become a rare event. Subsequently, taking in all 48 patterns in the language would likely become the ideal for everyone. This would give the impression that there is a master plan for beauty, which would conflict with the purpose of pattern languages.

Expressing the pattern map as a radar chart also would cause the same problem. The radar chart used in this system cannot accommodate for the patterns which the user answered not important. This consequently hints the users that the ultimate goal is to fill the radar chart completely, which brings back the problem of the whole language becoming the master plan.

Another problem with this particular diagnostic system lurks in the fact that it is made specifically for the *Generative Beauty Patterns*. It takes advantage of the 9 categories and the images used specifically for the language. Problems would arise on the attempt to make the system applicable to other pattern languages.

The system proposed in this problem inherits the idea of a Pattern Diagnostic System from the Generative Beauty Pattern Diagnostic System, and at the same time solves the problems it had.