

Patterns of Innovation: Part I

JUNG-SING JWO, Tunghai University, Taichung, Taiwan
YU CHIN CHENG, Taipei Tech, Taiwan

This paper is the first in a series to present twenty-one patterns that are distilled by analyzing products and services that satisfy the following four criteria: (1) the innovation is something new whether it is a whole new existence or begins as an improvement of something already in existence; (2) the innovation brings betterment and improvement in the purpose it serves for its users; (3) the innovation is implementable; and (4) the innovation is value-creating. We make an attempt mainly to bring the concept of innovation patterns closer to the products and services that are found today. The patterns are presented using the Pattern 3.0 style. In particular, the examples in each pattern illustrate how the pattern is applied. This paper will cover three patterns: Assorted Variations to show how variations in a common use scenario can be integrated to individually satisfy each variation; Finding and Filling the Gaps to show how known gaps in an existing or new application can be eliminated by exploiting new solutions that become available; and Finding and Triggered Response to detect and initiate proper interaction with the user of an application. A case study of an application in retailing that involves all of the three patterns is presented.

Categories and Subject Descriptors: **Innovations, patterns, pattern language**

General Terms: Innovations

ACM Reference Format:

Jwo, J.-J. and Cheng, Y. C. 2017. Patterns of Innovation: Part I. Proc. of Asian Conf. on Pattern Lang. of Prog. 6 (March 2017), 8 pages.

1. INTRODUCTION

Originally created by Christopher Alexander and his colleagues for the design of buildings, towns and constructions [1], patterns and pattern languages have influenced many different areas, including software [2], human actions [3], and so on, where successful cases were analyzed and distilled into patterns. Thus, there are plenty of patterns from all areas that new comers as well experience practitioners find useful. Since innovation drives many of the products and services that we use today, patterns for innovation make sense, if we learn from the success of patterns in the various areas. This paper is the first in a series of papers to present the twenty-one patterns in innovation that we have identified. The twenty-one patterns are listed in the patlet format in the appendix. Indeed, there are good sources that discuss where innovation comes from, for example, Steven Johnson's *Where good ideas come from: a natural history of Innovation* gives seven patterns of innovation [4]. In comparison with the patterns treated in this and the subsequent papers, Johnson's patterns – *Adjacent Possible*, *Liquid Networks*, *Slow Hunch*, *Serendipity*, *Error*, *Exaptation*, and *Platforms* – are at the level of principles, much in the same way that *Open-Closed Principle*, *Dependency Inversion Principle* (and three others) are principles [5] whereas *Observer* and *Composite* (and many others) are patterns in software design [2].

In this paper, we make such an attempt mainly to bring the concept of patterns and pattern languages closer to the products and services that are in existence. To this end, the examples in the patterns will illustrate how the pattern has been applied in reality. In finding examples of the application of these innovation patterns, we have taken care to source products and services that simultaneously comprises four characteristics: (1) the result is something new whether it begins as a whole new existence or as an improvement of something already in existence; (2) it must bring betterment and improvement in the purpose it serves for its users; (3) it must be implementable; and (4) it must be value-creating. Meeting the four criteria ensures that we source these patterns only from practical value-adding innovations. Since innovation is a human centric experience, we shall present the patterns in the Pattern 3.0 format [3]. The character in the drawings is Archimedes having his Eureka! moment in the bathtub.

This work is supported in part by the Ministry of Science and Technology (MOST), Taiwan by Grant 105-2221-E-027-076. Author's address: Jung-sing Jwo, Department of Computer Science, 1727, Sec. 4, Taiwan Boulevard, Xitun District, Taichung 40704, Taiwan; email: jwo@thu.edu.tw; Yu Chin Cheng, Department of Computer Science and Information Engineering, 1, Sec. 3, Zhongxiao E. Rd., Taipei 10608 Taiwan; email: yccheng@ntut.edu.tw.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission. A preliminary version of this paper was presented in a writers' workshop at the 6th Conference on Asian Pattern Languages of Programs (AsianPloP). AsianPloP'17, March 12-13, Tokyo, Japan. Copyright 2017 is held by the author(s). HILLSIDE 978-1-941652-03-9

A note to the reader: This series of papers grows of the keynote speech the first author J.-S. Jwo gave at AsianPloP 2016. The patterns are originally presented in the Chinese language, where each pattern has the name in the form of a four-character idiom that is well-known among speakers of the language. Therefore the pattern name has a powerful way on conjuring up the pattern's context, applicability, and solution. In translating into the English language, we have found it difficult to find equivalents of such four-character idioms that retain the same power. For this reason, we keep the original Chinese idiom after its English pattern name.

2. ASSORTED VARIATIONS (兼容並蓄)

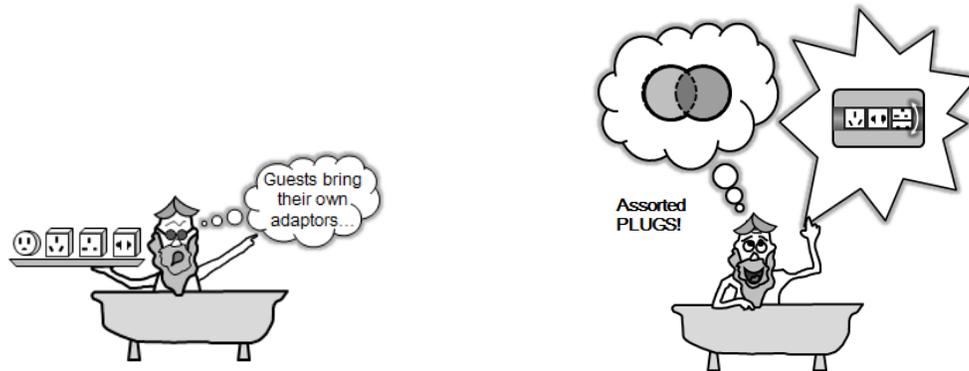


Figure 1. Assorted Variations

You are contriving a new product to meet all of the identified requirements. Several products that individually and partially meet the requirements already exist. Each product can be seen as a variation of the others in that it adapts to a specific context of use. **How do you create the new product given the knowledge of the existing products?**



Reusing the solutions in the existing products could save development time.

There could be incompatibilities in the solutions of the existing products.

Only a small number of variations are needed at the same time.



Create the new product to combine the existing products identified to cover all the variations. Identify the structures of solutions used in the existing products, making changes to resolve the incompatibilities where necessary, to come up with a unifying structure of the solution for the new product. The new product subsumes all of the existing products.

The Swiss army knife [6] is the quintessential example of this pattern. The main scenario is the use of pocket knife for cutting (therefore the name Swiss army knife) in all kinds of outdoor places and occasions. Adding bottle opener, screw drivers, scissors etc. into the same pocket unit obviously increases the utility of Swiss army knife. The tools are used mutual exclusively.

International guests bring and use their personal appliances such as smart phone, laptop, tablets, and so on for use while staying at a foreign hotel (See Figure 1.) The main scenario is the use of power by plugging the plugs of the traveler's appliances into the outlet sockets, which are incompatible by regions [7]. To the guests who forget to bring adaptors, the incompatibility becomes a nuisance. The spinning power outlet [8] integrates sockets that fit different plugs used in different regions of the world. The sockets are mutually exclusive: only sockets of the one type will be needed at any time. Thus, the sockets are assembled as a wheel that spins around an axle; the user spins the wheel until she/he sees the right socket, hiding the other sockets. Consequently, international guests need not bring their own adaptors.

The Wine Connoisseur in the case study is yet another example; see Section 5.

3. FINDING AND FILLING THE GAPS (拾遺補闕)

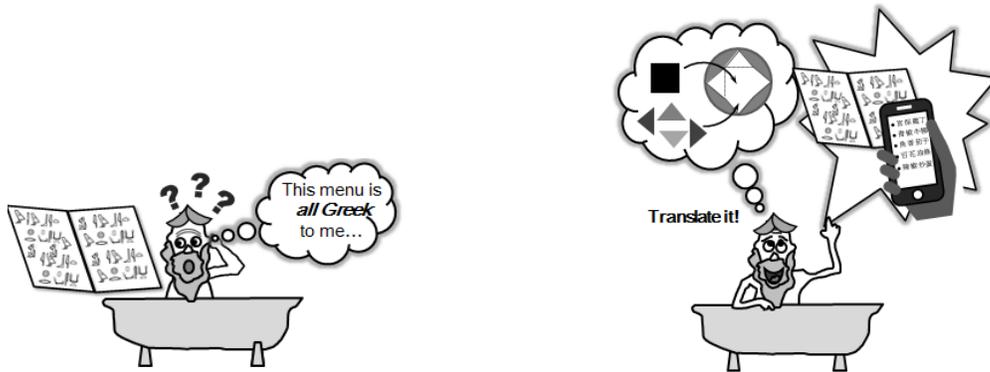


Figure 2. Finding and Filling the Gaps

Impediments could exist when the user interacts with a product, thus preventing the intended scenarios of the product’s use from playing out smoothly. While reexamining an existing product or developing a new one, e.g., by applying *Mash-up* (See Appendix) or *Assorted Variations* (Section 2), **how do you deal with such impediments, known or hidden?**



An existing product carries some kind of change-resisting inertia: people have gotten used to it; they know all about the small inconveniences and the circumventing workarounds.

Impediments may not be obvious to find – especially if the application of the product is complex, e.g., when leveraging solutions from multiple existing products to create a new product.

New technologies become available and could potentially fix the old impediments.



Therefore, perform an analysis to identify the impediments and missing links for the application and create designs to fill the gaps.

Examine the scenario (steps of interactions between the user and the product) where user faces impediments in using the product.

For each interaction where the user faces an impediment, extend or modify the existing structure and behavior of the product so that the impediment is removed. Often, this involves adapting new technology into the existing solution.

Iterate until all fillable gaps are filled and make note of those which cannot be filled yet.

Identifying gaps in an existing scenario opens up possibilities to leverage new technologies that are previously not available as shown in what follows (Figure 2.) You are traveling in a region that uses a language you don’t speak. Understanding what to order at a restaurant is a challenge for you. In the metropolitan, this is less of a problem since menus are often printed in multiple major languages. But if you venture into the countryside, reading a menu becomes a challenge because only the local language is used most of the time. Luckily, you have an app on your smart phone that scans texts in menus and translates it into the language of your choice [9]. The app could go beyond translation: it could provide you with the ingredients and how they are cooked. You therefore confidently know what to order.

The Wine Connoisseur case study of Section 5 is a new service that combines the patterns *Assorted Variations* (Section 2) and *Triggered Response* (Section 4). The customer has made a selection of wine, but where is it on the shelf? Leaving the customer to find it risks not completing sale if the customer could not find it. Since the Wine Connoisseur already knows the selection, the gap can be filled by adding to the shelf various indicators

(e.g., LEDs or e-Labels), which Wine Connoisseur lights up to indicate the selected wine. In the example, the gap is the customer's effort to find the wine and the missing link is the indicator and the mechanism to drive the indicator according to the customer's indication. The mechanism includes an identity model of wine selection, wine location, and the nearest indicator. Wine Connoisseur must enable the identity model to be built and maintained.

4. TRIGGERED RESPONSE (應運而生)

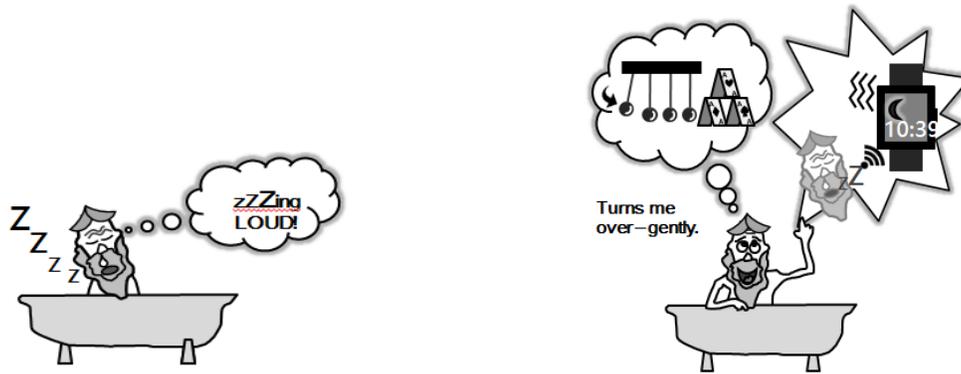


Figure 3. Triggered Response

A contrivance must respond to events that take place in an intended area of application. The responses to these events are known. **How do you ensure that such events are responded to?**



You know the ways for the contrivance to react to an event, but you don't know if and when an event occurs.

The detection of events and the reaction to them are simple or complex depending on the application.

A response can trigger further events and the contrivance needs to continuously respond to.



Therefore, devise an event detection and reaction mechanism into the contrivance to respond to an occurring event.

Identify the events the contrivance must react to. Some of such events including: user entering/leaving an area; user making a gesture or noise; occurrences of phenomena in the environment, and so on.

Identify or invent the means for detecting the events. This could involve sensor signal processing, computer vision, and so on.

Invoke event-specific behaviors of the contrivance to respond to the detected events.

Integrate the event detection and response mechanism so that they are continuously applied.

Here is an application of the pattern. Snoring is a problem for the couples sharing a room. Snoring can be stopped by a change of sleeping position, but the snorer lacks initiative to change position since he/she does not know it. As a result, the partner is awakened and suffers. The snoring watch detects a person's snoring and emits a minor electric sting that is just enough to stimulate the snorer into changing position and stopping the snore without awakening both the snorer and the partner (10).

Simple or complex reactions are needed depending on the application. In the case of the USB Mail Box Friends Alert (10), both the detection and reaction are easily built. Event detection amounts to registering a callback for new email arrivals. The callback lights up the USB Mailbox and raises its mail flag. After the user checks the emails, the USB Mailbox resets [11]. In contrast, in the Wine Connoisseur case study of Section 5, the crowd-sensing digital signage display plays the right promotion clips based on viewer types. It does so by first

detecting and categorizing the viewers, which requires a system that continuously analyzes live videos. The triggered response enables (but does not directly implement) further interactions from the viewer.

5. INNOVATION PATTERNS IN ACTION – THE WINE CONNOISSEUR

As a supermarket manager, you have observed that selecting wines can be a small challenge for customers. A multitude of criteria are at play: food-pairing, occasion, taste, body, color, price, and so on (12). Prior research is essential but not all customers do their homework before shopping. Reading wine labels is only marginally helpful because a label can carry only a limited amount of information. While Wine apps like Vivino (13) allow you to scan labels for information and rating, you have to know which label to scan first. Wine sale could increase if you could figure out a way to help in-store customers in selecting the right labels.

One proven solution is to bring in an expert to help customers right by the wine shelves. On the occasions that the store is able to do so, wine sale usually picks up as customers are able to ask questions and receive recommendations. In fact, you have observed that an on-site expert does much more than providing information. He or she is knowledgeable in interesting details that could stimulate purchasing interest of customers. When combined with promotions and discounts, wine sale could further increase.

However, since hiring an on-site wine expert is expensive, you could not have the arrangement as often as you would like. How do you design a regular and economically feasible way to increase the sale of wines as well as associated products?

Inspired by the success of the presence of on-site wine expert, you want a new contrivance to include all elements that make the presence of wine expert great. For instance, wine expert works well because he/she answers questions as well as proactively interacts with customers to guide customers in making a right selection. Therefore, you want the contrivance to interact with the customers at multiple levels (*Triggered Response*). At the basic level, the contrivance actively senses for any customer browsing the shelf. It displays the most appropriate information according to the customer’s detected demographic features (e.g., gender and age) and solicits interaction. At the next level, the customer is guided through a well-designed user interface for wine selection. In the process, whenever the customer selects a specific label, the contrivance lights up the indicator to point out the selected wine (*Filling the Gap*) as well as displays pairing food recommendations (*Assorted Variations*). With the contrivance, you will be able to serve more customers, sell more wines, and promote the sale of pairing foods

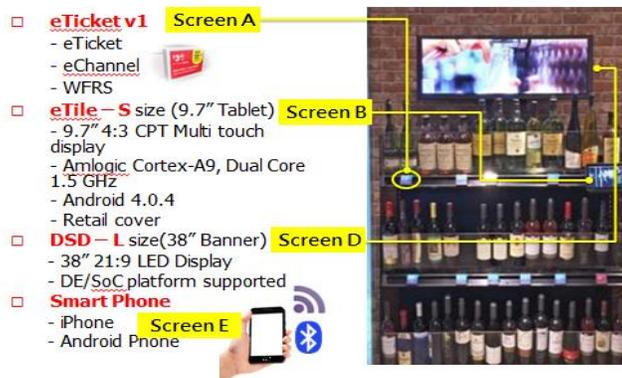


Fig. 4. Wine Connoisseur: A contrivance for wine selection help and food-pairing recommendation.

A contrivance called *Wine Connoisseur* is shown in Figure 4. Wine Connoisseur is a unit that integrates a digital signage system, IoT components, and a conventional shelf. Observing the customer through a camera, Wine Connoisseur detects customer presence, plays video to attract and retain the customer’s attention on the large display (Screen D), sets the default menu selection on Tablet B, and waits for the customer to navigate the selection process (*Triggered Response*). Through tablet B, Wine Connoisseur offers some categories of wine purchasing recommendations based on gender and age selection (*Assorted Variations*). Accepting a confirmed

selection, Wine Connoisseur plays a video clip of an expert elaborating the selection on screen D. When the customer makes the purchase selection, the small display A nearest the selected wine on shelf is lit to locate the selection (*Filling the Gap*). Additional information for recommended food-pairs is then displayed on tablet B (*Assorted Variations*), including the location and applicable discount of the food-pairs (*Filling the Gap*). In addition, customers with store app installed can interact with Wine Connoisseur with their smart phone E instead of tablet B. Wine Connoisseur serves continuously through the store hours.



Fig. 5. Recommend pairing food when wine is selected.

6. SUMMARY

We outline the patlets of 21 innovation patterns in this paper, and elaborate on three of them – Assorted Variations, Finding and Filling the Gap, and Triggered Response. We plan to write all of the 21 patterns in a series of subsequent papers. In the meantime, we plan to research how the twenty-one patterns are linked to Johnson’s patterns [4].

Acknowledgements

We thank our shepherd Hironori Washizaki for his valuable comments and feedback during the shepherding process and Nien-Lin Hsueh for observing this process. We also thank our 2017 AsianPloP Writers Workshop Group, Yuma Akado, Kei Ito, Joseph Yoder, Ayaka Yoshikawa, and Horng-Yi Yu for their valuable comments.

REFERENCES

- [1] Alexander, Christopher. A pattern language: towns, buildings, construction. Oxford University Press, 1977.
- [2] Gamma, Erich. Design patterns: elements of reusable object-oriented software. Pearson Education India, 1995.
- [3] Iba, Takashi. "Pattern Language 3.0: writing pattern languages for human actions." Invited Talk, the 19th Conference on Pattern Languages of Programs (PLoP2012), Arizona, USA. 2012.
- [4] Steven Johnson. 2010. Where good ideas come from: The natural history of innovation. Penguin UK.
- [5] Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.
- [6] Aditya Ghuwalewala, "The Swiss Knife Concept," Retrieved January 5, 2017, <https://adscopy.com/2007/07/16/the-swiss-knife-concept/>
- [7] Plug, socket & voltage by country. Retrieved April 5, 2016 from <http://www.worldstandards.eu/electricity/plug-voltage-by-country/>
- [8] U-Socket USB Wallplug. Retrieved April 5, 2016 from <http://www.thinkgeek.com/product/e81a/>
- [9] Waygo - Chinese, Japanese, and Korean Translator, Available from <https://itunes.apple.com/us/app/waygo-chinese-japanese-korean/id496038103?mt=8>
- [10] Robert Bates, "Zicac Stop Snoring Wrist Watch Review," Retrieved January 9, 2017 from <https://www.snoringdevicesthatwork.com/stop-snoring-wrist-watch-review/>

[11] USB Mail Box Friends Alert, Retrieved January 9, 2017 from <http://www.geekalerts.com/usb-mail-box-friends-alert/>

[12] How to Select a Bottle of Wine, Retrieved January 3, 2017 from <http://www.wikihow.com/Select-a-Bottle-of-Wine>

[13] The Vivino app, Retrieved January 3, 2017 from <https://www.vivino.com/app>

Received January 2017; revised March 2017; accepted March 2017

APPENDIX

The twenty-one patterns are summarized in the following patlets.

Pattern 1: Paint the tiger after the cat	You are creating a contrivance in its intended area of application. Another known object that is conceptually similar already exists in a different area of application. Imitate the known object and change its design if necessary to create the contrivance that satisfies the requirements of the intended area of application.
Pattern 2: Trade the lamb for the cow	You have identified a known object that could be used in the intended area of application. However, the intended area of application poses unique constraints that the known object could not satisfy, especially due to cost or inconvenience in use. Identify requirements that the known object meet but could be weakened or dropped. Simplify its design or use cheaper materials in light of the weakened requirements to create a replacement contrivance that meets the unique constraints of the intended area of application.
Pattern 3: Borrowed splash	You have a known object that is adequate for its intended area of application. The known object fails to reach the customers since its form is largely unknown or unattractive. Create a contrivance from the known object by blending elements of certain well-known forms to attract attention.
Pattern 4: Adorn to attract	You have an object in a known area of application. Although structurally and functionally adequate, the way it solicits a user action or its response to user action lacks amicability to the user or fails to intimate the user of the response. Add visual elements to alter solicitations and responses to user actions to increase its amicability.
Pattern 5: Add-on	You have a known object that fails to meet partial requirements. You don't want to replace the known object because of this partial failure. Although the partial requirements could be met by another object, introducing it causes inconvenience in the known object's use. Create an add-on object as an extension to the known object to satisfied the partial requirements with minimal or no alteration to the known object.
Pattern 6: Mash-up	You have a set of related requirements each of which requires a different known object. Create a contrivance to meet requirements that spans the different areas of application. Work to assimilate the requirements into a coherent whole.
Pattern 7: Inside and outside collaboration	You have a highly integrated, known object that satisfies numerous requirements in an intended area of application. The high degree of integration makes it inflexible, difficult to use, or costly to produce. Alternative, you have two known objects that are closely associated but difficult to use together. Factor the integrated object into components that offer internal mechanism for integration and external mechanism for its use. In the case of closely associated objects, develop the internal and external mechanisms for integration.
Pattern 8: Self as an aggregation of smaller selves	You have a known object offers similar functions as a large aggregation, but the aggregation causes waste, costs more to produce, or is difficult to use. Disintegrate the known object into smaller but functionally identical units that can either be used individually or be reintegrated into a larger aggregate without causing difficulty in use.
Pattern 9: Assorted variations	Create a new product that combines the existing products identified to cover all the variations.
Pattern 10: Stay lean	You have a known object that satisfies the requirements of an intended area of application. However, the known object has unneeded features or structures. Remove

	the extraneous features to let the key features stand out.
Pattern 11: Symmetry and asymmetry	Symmetry is a common feature of form and structure commonly seen in known objects for use in a wide variety of areas of application. Although symmetry is aesthetically appealing and easier to craft, it could cause difficulties of use in the known objects. Reversely, a known object could be difficult to use or costly to craft because it fails to consider symmetry in its design. Consider both symmetry and asymmetry in the design of contrivances in light of ease of use and craftability.
Pattern 12: Variation within commonality	You are creating a contrivance for an intended area of application. In individual use, some variations are required but these variations are often ignored as the cost to produce a large number of slightly different objects cannot be justified. Create the contrivance based on the common requirements, but provide mechanism for plugging in different kinds of variations.
Pattern 13: Adapting to application environment	You have a known object whose function, structure or operation is limited by the application environment. Develop different design of a novel object that adapts to the environment to replace the known object.
Pattern 14: Restructure	You have a known object for use in an area of application, but its structure causes some impediments in its use. Change the structure to eliminate the impediments.
Pattern 15: Virtual as real	The contrivance is difficult to create due to physical limitations, environmental constraints, high cost, or design difficulties. Per limitation and constraints, make up for the inadequacy by using modern technology to create a virtual reality that imitates the physical reality.
Pattern 16: Downstream push	You have a known object that functions well in a known area of application without draining all of its resources. Make use of the surplus resources to meet additional requirements to create a contrivance that widens the area of application.
Pattern 17: Favor a fundamental solution	A critical problem with serious ramifications occurs frequently in a predictable way. Further, rectifying the consequences is difficult or costly. Rather than wasting resources to rectify consequences, circumnavigate to avoid the problem altogether.
Pattern 18: Cutting corner	You have a known object that is structurally or functionally too complex to support streamlined operations. Simplify structure or features to streamline operations.
Pattern 19: Add to Simplify	You have a known object that supports complex operations that involves a steep learning curve. Further, the operations cannot be simplified. Provide an alternative guide or interface to help users cope with the operational difficulties.
Pattern 20: Finding and Filling the Gaps	Perform an analysis to identify the gaps and missing links for the application and create designs to fill them.
Pattern 21: Triggered response	Devise an event detection and reaction mechanism into the contrivance to respond to an occurring event.