

# PatternCity: A Serious Game to HCI Design Patterns Discovery and Delivery

Ahmed Seffah

*TechCICO-ICD STMR-CNRS 8229  
Troyes University of Technology,  
France  
ahmed.seffah@utt.fr*

Halima Habieb-Mammar

*Department of Computer Science and Software  
Engineering  
Concordia University, Montreal QC Canada  
mammar@cs.concordia.ca*

## ABSTRACT

A huge frustration is observed among the pattern users as they need to spend a significant amount of time just to find the right pattern for their use from the very large and heterogeneous collection of patterns that are available via different websites and databases. Since software developers are used to work under a tight schedule in various environments with different needs, they should find patterns in a clear and understandable format or ready-to-go state along with a proper organization. The paper is an effort to overcome such drawback while closing the communication gap between different professional groups who are interested in patterns and pattern languages. The goal is to introduce a methodical approach, in general, and a generalized format, in particular for pattern discovery, representation and delivery. We first overview the current drawbacks of HCI design patterns which motivate the search for a new tool to document and deliver patterns. Then, we detail the ideas that lead the concept of *PatternCity*, a serious game in which HCI design patterns are represented as a building in a virtual world, and where the players can collaboratively build and improve these buildings.

**Key words:** Patterns, User interface, Design, Prototype, artefacts.

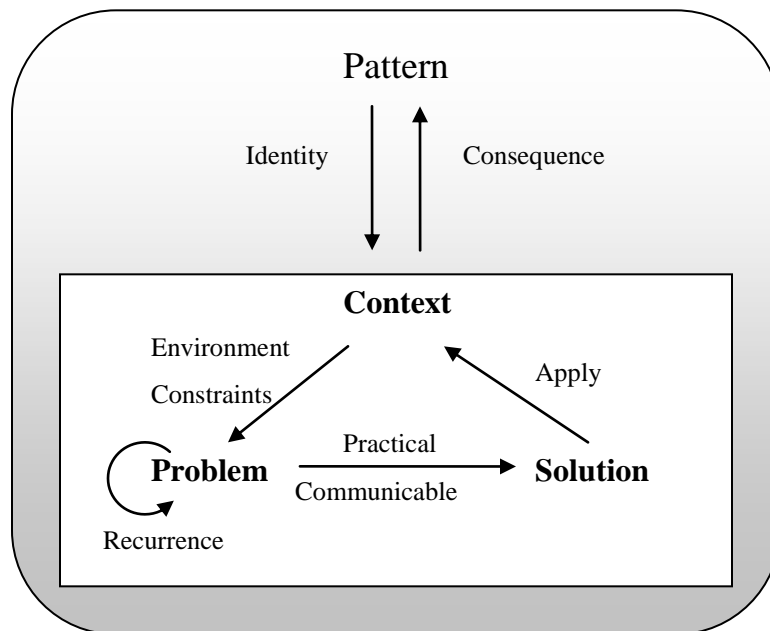
## 1- Introduction

A design pattern can be defined as “a general repeatable solution to a commonly-occurring usability problem in user interface design or interaction design” [Gamma et al., 1995]. It summarizes knowledge about best-practices that have been acquired taking into account different scientific perspectives and methodologies (from psychology, usability, or ergonomics), by describing at least the *context* in which it can be applied, the *problem* it can solve, and the proposed *solution*. Other possible elements include *examples* where this pattern has been used, a *design rationale* to explain the reasoning behind the pattern, as well as a mention of its *forces*, i.e. the – often contradictory – goals and constraints that need to be balanced out. The pattern is described at a level of detail where it is specific enough to allow straightforward application, yet abstract enough to be toolkit-independent and applicable to different situations.

Unlike in architecture [Alexander et al, 1977] and software engineering [Gamma et al, 1995], patterns are not well-established among the practitioners of user interface Design [Gaffar, 2005]. While they may be known to researchers, designers and developers are mostly oblivious to them. A part of the problem may be that there is no definitive pattern collection: many collections are published [van Duyne, 2002, Graham 2003, Borchers 2001, Welie et al., 2000], each using its own terminology and format, making comparisons difficult. A common pattern library would help defining a language between the different stakeholders in a software project, essentially playing the role of boundary objects [Star and Griesemer, 1998]. Thus, our goal is to increase the overall awareness of User Interface Patterns for these actors.

In this direction, the most important defining characteristics or basic terminologies of a pattern are: **identification** (of the problem in context and with imposed constraints), **existence** (of the

solution), **recurrence** (of the problem), **invariance** (of aspects of the solution), **practicality** (of the solution, which needs to strike a balance between **optimality** and **objectivity**), and **communicability** (of the problem and the process of arriving at the solution to the user). Some of these characteristics are illustrated in Figure 1.



**Figure 1:** The Pattern Anatomy and LifeCycle

Most often, pattern collections stored in a database may have the following advantages:

- First of all, as a consistent set of attributes is used to describe a pattern, the differences between two patterns are evident, so that one pattern can be chosen over another in an informed manner.
- Second, the collection can make interrelations between patterns explicit by categorizing and inter-linking them.
- Thirdly, a digital database may also produce data about the access frequency of specific patterns, which can be used to estimate its popularity among the user base of this database (as patterns are only patterns if they are re-used in similar context) this may reveal the need of re-formulation or dismissal of a pattern.

The fundamental question that has not been addressed correctly is that how can a designer access the content of this database while finding the right pattern at the right time? As, by definition, he doesn't know the name of the pattern that he searches for the first time, a pure text search may not be enough. Categorizing the patterns and adding keywords to them could help to alleviate this problem. But the discovery of previously unknown concepts remains a hard task, due to their textual form.

When building a platform for these different stakeholders, various problems arise. To begin with, they may come with different expectations, styles of communication and goals. To give a quick overview over the broad range of possibilities, we defined three personas (User eXperience UX Researcher, UI Designer and Front-end Developer). The user of the platform may also have different roles: a pattern author, seeking to spread knowledge about his insights, may try to publish and promote on this platform; while others simply enjoy the social and fun aspect of this community.

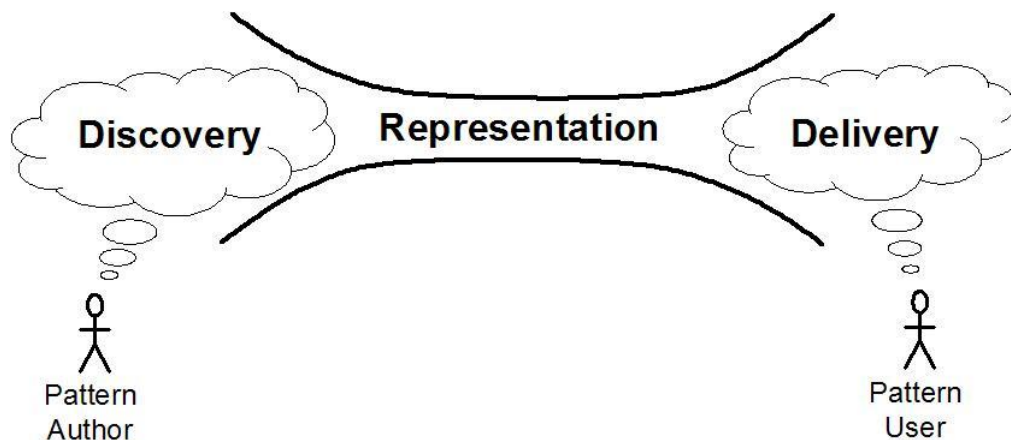
A related work that need to be mentioned is the Pattern Almanac [Rising, 2000] which is an attempt to make accessible (via a unifying user interface) a very large collection of all existing patterns and patterns languages. Several databases accessible via Internet also have been proposed. However, these attempts fail in increasing the “ease to use and learning patterns” while making the pattern user experience pleasant and enjoyable activity. As it will be discussed, the Pattern City project aims to overcome such goal while bridging the gap between patterns discovery and dissemination.

## 2- The problem of representing and delivering HCI design patterns

HCI design pattern writers, who are usability expert with background in psychology, focus on usability and human aspects of the user interface design. They generally prefer to use narrative formats to convey solutions to common user problems with supporting theories and concepts of interaction design and human factors. On the other hand, user interface designers are typically software developers who need concise and pragmatic guidance through their design and coding activities. They often find it hard to translate text pattern knowledge into concrete design [Lin et al, 2000]. Moreover, with the plethora of patterns available today, mainstream developers get inundated with huge number of pattern literature and links in many books and on the Internet. They have to manually read and sift through piles of texts looking for some useful patterns to apply [Gaffar, 2005].

In this regard, besides focusing on **what** should be presented in terms of information contents within patterns, a fundamental challenge is **how** it should be packaged and offered to developers in an appropriate way to help understand and apply them correctly and efficiently. As will be discussed in this paper, more flexible notations and views are required to represent different types of pattern information. The pattern lifecycle needs to be investigated, defined and modeled to allow for better understanding of pattern dynamics as an emerging design technique. In what follows we will explore the multi-faceted problem of pattern representations.

In this paper, we can look at patterns in general as artifacts that have three main milestones, organized from a user perspective (Figure 2).



**Figure 2:** Major milestones and users of patterns

### Pattern delivery and dissemination

On the pattern user side, we can say that patterns are harvested and represented with the main goal of being delivered to other users who implement them as solutions. A delivery paradigm is essential in the pattern approach because it indicates that patterns arrived effectively to potential users; a knowledge dissemination view. This means that patterns should be represented in a way that software developers can learn, master and apply easily and effectively in their context. This implementation highlights the main role of patterns, promoting effective reuse. If patterns were

harvested and written down just for the sake of archiving them then we have missed on the great benefits of patterns.

### Pattern discovery and collection

On the pattern writer side, the discovery of a pattern is only the beginning. Harvesting is a carefully selected metaphor that indicates the hard work associated with patterns. By observing existing artifacts and problems that have been solved successfully, we can detect a repeated structure or behavior that is worth recording. By asserting its importance, we can write down the essential components and –if possible- analyze them. An expert can provide insight as to why this combination is good or why it works well and in what context. Finally guidance of how to reuse this solution can be added to assist in modifying and reapplying the solution

### Pattern representation and documentation

Representation of patterns can be seen as a vehicle –a medium or an infrastructure –to bridge the gap between the two main activities; delivery and discovery. This representation is essentially about how to format the solution in a way that allows it to mature from its solution-format into a pattern. In essence, a pattern is a solution alongside other information that supports it. The reason is that in order for a solution to be used by others, they have to be convinced that this is a good solution. Part of this come by annotating pattern solution with expert analysis and comments, listing of some cases where the solution has been applied and the “success indicators”, and possibly some code examples. Bearing in mind that no two systems are exactly the same, and that every new software is a new adventure, patterns are typically annotated with important guidance on how to apply them in different contexts and situations. Some details are left out to allow the end user to rematerialize an abstract pattern back into a concrete solution that is adapted to the new design.

## 3- The founding ideas of the PatternCity concept

Build around a database, PatternCity combines a serious game and a social network combined into a 3D virtual world (Figure 3).

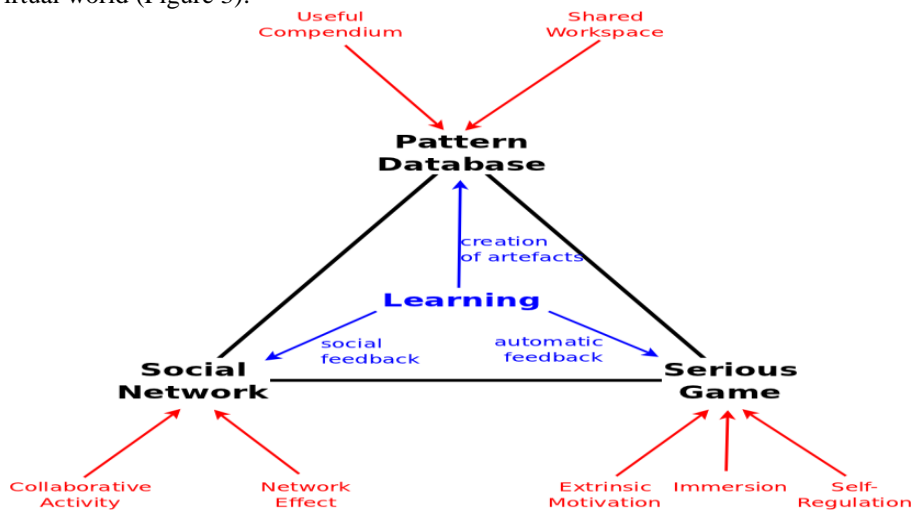


Figure 3: Components of proposed platform

The **database** containing information about HCI Patterns is first of all meant to be a shared workspace: the information is not only consumed, but actively produced by the participants. Unlike a book, many authors need to contribute their respective perspective on a specific topic in order to find a representation of knowledge that is universally accepted. Ultimately, this may lead to a state of maturity where the organically grown content gets an authoritative character (similar to a Body of Knowledge).

In order to support this collaborative activity, we added aspects of a **social network**, emphasizing the possibilities to communicate freely, to build up a circle of friends, and to stay aware of activity of friends or related events. We also use the process of group identification as an attractive element to invite new participants: being invited by pre-existing acquaintances may help new participants to familiarize and engage with this system.

These features are embedded within the context of a **serious game**: active participants can earn credits and other awards, increasing their motivation. This playful approach to a serious topic also encourages immersion, potentially leading to a more effective learning. Finally, this context allows us to establish rules, allowing this socio-technological system to regulate itself.

Ultimately, we can frame our solution as artifact-based, **constructionistic learning**. The core of learning here is not memorizing details, but rather creating knowledge representations and evolving along with them. Social feedback by other participants and system feedback of the game rules can induce the participants to engage with the topic in a deeper manner than mere passive reception (such as reading or classroom activities).

The following **goals** could be established (similar to the cycle of knowledge building [Stahl, 2004]):

- a. Where do I already use patterns, without being aware of its name and its concept?
- b. What do others say about a specific pattern?
- c. How do I express my internal knowledge about patterns so that different stakeholders can agree?
- d. Which patterns are suited in my specific context?
- e. How can I apply my knowledge about patterns in my current project?

#### **4- Early prototype**

The most recent result of works done in this topic was a clickable prototype (Figure 4). It contains all essential elements: it refers to the concept of social network (user has profiles, friends and can communicate with them), establishes a database of patterns (each user can create patterns), and brings in components of a serious game (it is possible to buy patterns, see the high-score of designers and patterns, and other statistics).

However, the interface doesn't state clearly how this game is functioning, and thus, why it would be enjoyable to join. Even if there was a page explaining the rules of the game, it would be quite difficult to convince a new user to engage with this system: the overall design doesn't look like a game any more than a blog or promotional website. Additionally, he would have to register before seeing any content.

There are big improvements to be made in the areas of story-telling (what does the site do? what is my role? which advantage does it give me? how can I be a part of this community?), wording (what is the concept behind this term?), consistent use of terms and motivation (why should I participate?).

In addition, some important aspects of the problem stated above were not addressed.

First of all, it doesn't take into account the diversity of the audience: as it proposes only one way of describing patterns (narrative, text-based form), it does not encourage discussions where different actors can contribute their respective perspectives.

Next, the list-based browsing, even if combined with a keyword search, doesn't not help the user find the applicable pattern if they don't know its terminology beforehand. The interface should invite the discovery of new or previously unknown patterns. Most importantly, the site does not fulfill the characteristics of a (serious) game [Ang and Zaphiris, 2005]: neither does it frame the action by a narrative, nor does it suggest explicit or implicit goals and rules.

It is for these reasons that we have decided to explore new approaches and ideas for this project, instead of iteratively evaluating and refining this prototype.



Figure 4: Early Prototype: User Dashboard

## 5- Exploration phase

The exploration that followed was led by the question: *how can we design an interface that is playful, expressive and suited to our needs?* Our main approach here was to try different metaphors and narratives, applying them to patterns and online communities, with the goal that the purpose of the interface should be clear at first glance.

### Pattern of minesweeping

The pattern of **Minesweeping** [Welie, 2008] inspired us to try out a playful discovery of patterns (Figure 5). The size of the square carries meaning, such as representing its popularity; positioning and color can show related patterns. Additionally, filters allow the usage of this interface for different tasks: giving an overview of patterns, patterns of other friends or a specific category. This interface would motivate mainly through its aesthetics: animations and colors encourage the use to “do something” with it, for the pleasure of manipulating it.

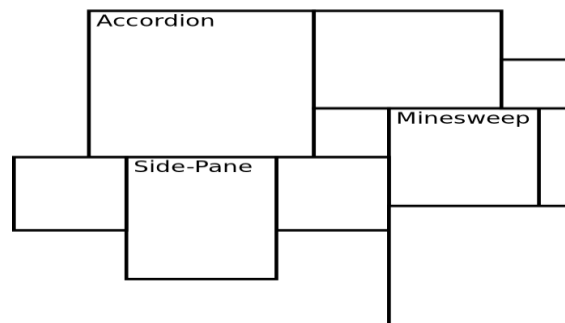
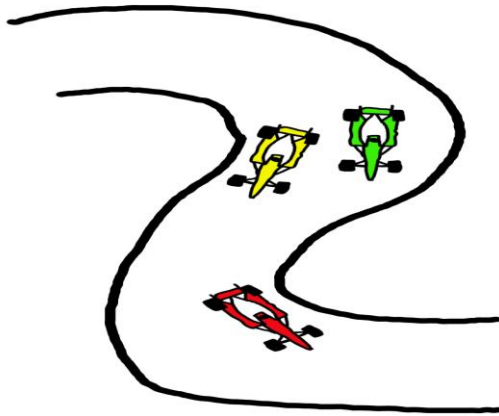


Figure 5: Playful discovery of Patterns

### Metaphors for learning patterns

Another method of motivating would be a system of **reputation**: points and badges reward positive behavior, as seen at stackexchange [http://ux.stackexchange.com] and other sites following the philosophy of gamification [Deterding et al., 2011]. This would be very direct and behavioristic feedback, and wouldn't suffice in itself to keep the user on site. However, it may be used to reinforce originally intrinsic motivation (“I want to learn about patterns” or “I want to document patterns”) or narratively induced goals.

As an example of such a narrative, the metaphor **Formula One** was studied (Figure 6). This game is divided in three phases: To become a member of a team, the participant needs to show an example where he used the corresponding pattern at least once. Then every team tries to improve their pattern (“tuning”). Finally, all patterns are published and voted by the community; the racing car moves forward as the popularity of the pattern increases. These phases are repeated in fixed time intervals; existing patterns may be improved further, or new patterns can be created. This relatively simple game would combine “serious” work with fun, leveraging the spirit of competition as a motivator, and fostering collaborative activity around the topic of interface patterns.



**Figure 6:** Native Context.

### **Metaphors for recognizing patterns**

Another metaphor would be that of the card game **Pairs**: the participant needs to find “two of the same kind”, either two different examples of the same pattern, or alternatively an example and the name of the pattern. By this, we could train the ability, recognize patterns in their real-life context, and induce curiosity about patterns the participant didn’t know before. Here we transfer the common knowledge of how to play Pairs to a different context of application, a method commonly used at language learning software.

## **6- Concept of PatternCity**

Based on these initial conceptual ideas, the concept of *PatternCity* evolved. Its core idea consists of 2 parts: every pattern is represented as a building in a virtual world, and the players (represented by their avatars) can collaboratively build and improve these buildings. The pattern City includes also the following buildings:

- Forum, a place for meetings where designers can discuss and share experiences regarding the use of patterns
- City Hall, the place where to register a pattern and get the approval to add it to the pattern city
- Trade Centre, the place where to buy and sell patterns
- Pattern Academy, the museum where the most innovative patterns and patterns authors are featured
- Tourist Information Centre, a place where to go to get information, advice and directions to visit and explore the pattern city
- The designer house is the place where all patterns proposed by a distinguished designer are featured

The two main characteristics of the PatternCity components are:

### **A. Pattern buildings:**

A building can be viewed from the inside as well as from the outside. When viewed from the outside, only the most important aspects of it are shown inside of the windows. It is only when he entered the building that the user can explore the details of the pattern.

Each of these details is hidden behind an object metaphor (e.g. a guest book to read and write comments, a vitrine to show the examples). In this way, we represent concepts as objects (an approach similar to the learning environment *Pattern Park* [Franke et al. 2007], where (software) patterns are represented by different rides in an amusement park.

These buildings are grouped into streets and townships, forming a digital city that can be shown on a map. In this city, some buildings have a special function: the Academy nominates and awards prizes to the best patterns, pattern collections can be shown in a Gallery, and the Stock Market displays the current popularity of each pattern. There is also a Town hall for signing up, filing bug reports, giving feedback, and reading the archives of the city newspaper. In the Information Center the player can view the current statistics about the city, and he may also found a Forum specific to a topic for news, blog entries and chat sessions.

There are different ways of navigation proposed: a map displays patterns by author, category, or personal bookmarks. Once he has chosen his destination, a navigation system leads him the way through the different streets. The navigation bar also shows his current position (on the map, as well as textual) and its surroundings. When navigating to a place he was before, the player can go there quicker.

#### **B. Players:**

The described environment serves as a shared space where all players can work together: the common goal is to “build up” the city. This happens not only by creating new buildings, but also by improving existing ones: the size of a building reflects its popularity. To break down this goal into smaller, individual goals, the game automatically gives feedback by giving reputation points for positive behavior. This encourages the player to try out the actions he can do, and continues to reinforce his motivation.

The players in the city also form a virtual community: direct communication between players is possible through a chat (synchronous medium) and messages (asynchronous medium). These tools are integrated into this game to foster social connections as well as collaborative work between real players.

To stay aware of modifications, prizes and new buildings, the player can consult the recent activities or events concerning the player or one of his friends. Additionally, the game leader, represented by “the Architect”, sends messages to inform about status informations when trying to build a new pattern, or participating in an Academy contest. For city-wide information, the Architect can publish a newspaper, containing statistics, new features or upcoming community events.

Most importantly, players do not only actively contribute text and images to the pattern database, but have many design choices while doing so. As an example, the interior decoration of a building can be re-arranged and modified, giving each building a particular flair. The position of the building is also a matter of choice: in this way, the structure of the city grows organically. Even the names of streets and townships may change, if a poll shows that many players agree on it. It is also possible to include elements of End-User-Programming: in this way, the player could even customize the interactive behavior of objects or add completely new objects.

In the same spirit, access to moderator functions is not limited to the administrator, but can be delegated to motivated and experienced users. A reputation system may keep track of the overall activity of a player. The amount of reputation of a player also helps other players to distinguish between beginners and experienced users (similar to Forums where the number of posts of a user is shown).

Players can also buy pattern shares at the Stock Market. Unlike the real stock market, the total number of shares is unlimited; the value/ “popularity” of a pattern is calculated as a function of its



number of visitors during a certain time period, its average ratings, and its current completeness. Buying a pattern thus means “speculating that it will increase in popularity”, and as it will be shown on the user profile as well, it also means “endorsing and recommending this pattern to others”.

Unregistered users can watch the city and its contents, but cannot interact with other players, modify buildings or buy at the Stock Market. If they want to participate in this game, then they will need to register at the Townhall.

In case of inappropriate behavior, that is, behavior that violates the “Charter of PatternCity” posted at the Townhall, users with the role of police officers may warn and, if needed, restrict the possibilities of a player (no communication, no edits) or even block his account temporarily.

The player can enter and leave the city by taking a bus; that is the reason why players don’t have their own houses in the city. In fact, PatternCity is a city where Pattern live.

## 7- Implementation

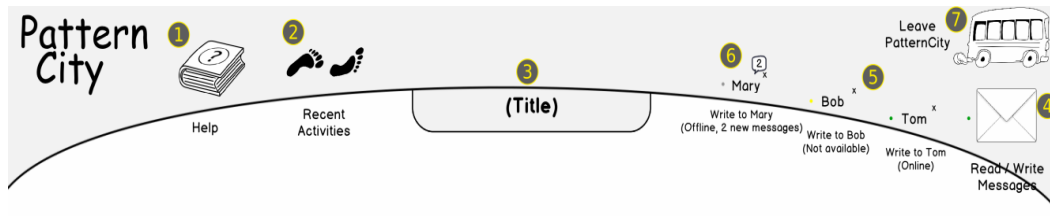
In order to realize this concept and define a **visual language**, a paper prototype was developed then transformed into a Balsamiq [<http://www.balsamiq.com>] wire-frame. The environment resembles *Second Life* [<http://www.balsamiq.com>] and *Visimmo3d V3D events* [<http://www.visimmo3d.com>].

The interface resembles adventure games in many ways: characters move in a (pseudo)-3D-Space, within a predefined area of the screen, by following the mouse clicks. Doors or streets indicate the possibility to go to another scene (interior of a building, or street view).

Objects, as they represent concepts and data, allow interaction with them: they show their name (or function) when the mouse hovers over them, and reveal their details in a pop-up window when clicked.

The setting is dominated by architectural design: the special buildings each have a unique expressive architectural style differentiating them from normal patterns.

In the following paragraphs, we will add some comments to the wire-frame prototype. There are two versions of it, with and without the label names. As the last sketch explains, these label names are only visible when the user hovers over the corresponding object. These “tool-tips” help new users to playfully discover their action possibilities, but without cluttering the interface with too much text.



**Figure 7:** Top Bar

### A- The top bar

The **top bar** (Figure 7) allows user communication and general game functionality. At the top left (1) a lexicon offers help. The user can type a question, and the system tries to find a similar “frequently-asked question”. These questions are logged, so that questions that *are* frequently asked can be added to the help. The window that opens when clicking on the foot steps (2) gives an overview over recent activity during the user’s absence or while he was doing something else. A title (3) indicates the current location of the showed scene. At the right side, communication tools are offered: the envelope (4) allows to begin a new conversation with a person in the current scene, related to the content of the current scene, or to the user’s friends. For every user, its current status is shown (online / not available / offline). Recent conversations (5) and incoming messages

(6) are shown on a per-user basis. (To communicate within a bigger group, the group can meet in a Forum.) Finally, a click on the bus (7) quits the game.

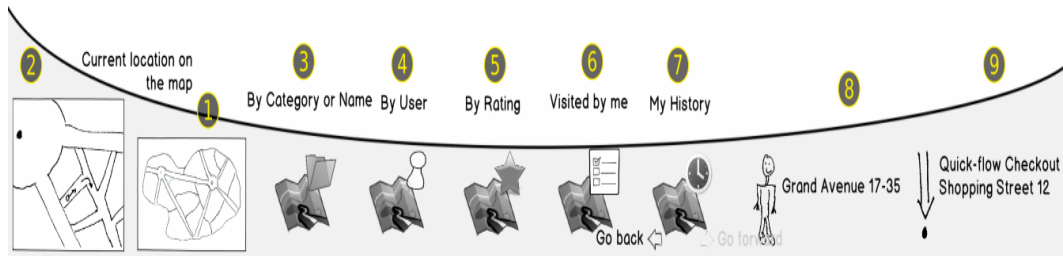


Figure 8: Bottom Bar

### B. The bottom bar

The **bottom bar** (Figure 8) assists navigation, at the left, a map of the city (1) along with a detail view (2) shows the current position of the player (Note that this map is updated as the city grows.) The map icons show the buildings on the city map filtered by different aspects: depending on the navigational strategy of the user, he may search for a pattern with a certain name or within a certain category (3), for patterns by a specific user or author (4), highly-rated and popular patterns (5) or patterns that the player already viewed or has the intention to view (6). He also can go back to the buildings he was before (a history of visits similar to web browsers) (5), or see the chronological order of his visits (7). At the right, the current location is displayed in a text form (8), as well as the current destination (9).

### C. The street view

The following sketch shows the **street view** (Figure 9). Arrows indicate the possibility to go to other scenes: the player may enter a pattern (1) or advance to the next street segment (2). Some doors are open, indicating the presence of other players (3). A big arrow (4) indicates the navigational direction, in this case, how to get to Quick-flow Checkout (5). The city may also contain green spaces (6) or unfinished buildings (7) which haven't been published yet.

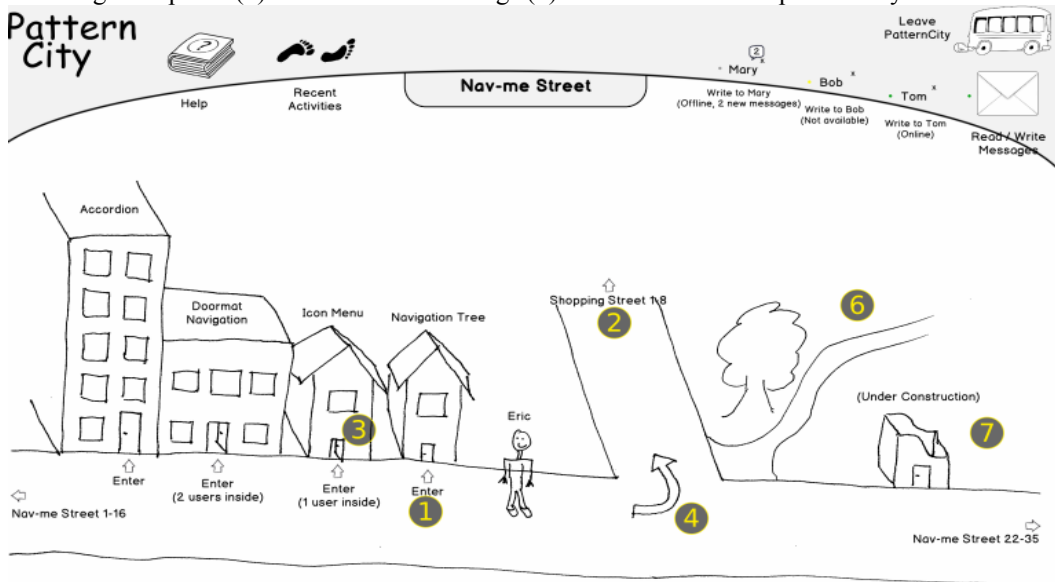


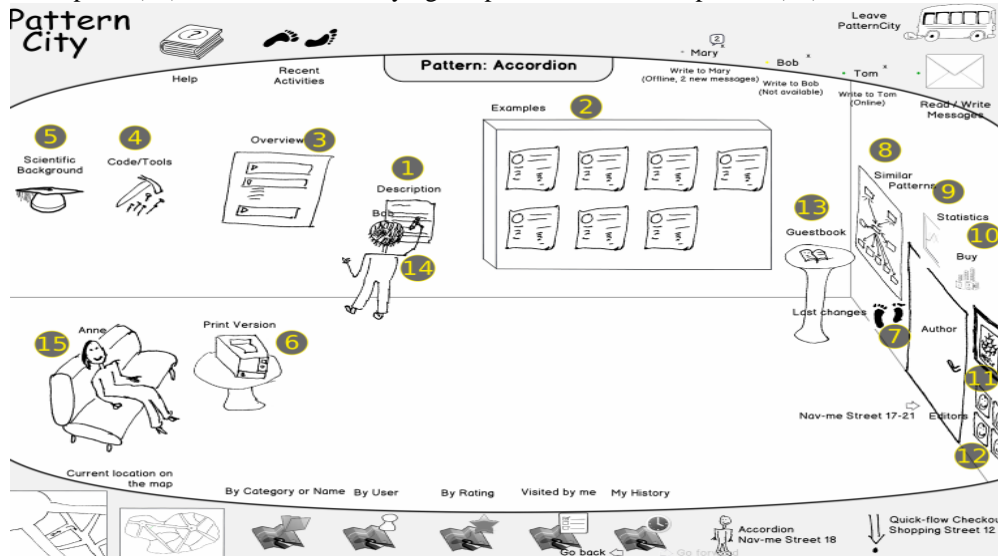
Figure 9: Street View

### D. The accordion pattern

As an example how pattern buildings can look like from the inside, the **accordion pattern** (Figure 10) has been depicted. Information that explains the pattern itself is posted on the back wall: A pattern description must contain a textual description (1) and at least one example (2) (screen-

shots, possibly annotated). Additionally, a schema that reduces the pattern to its core (3), implementation tips as well as links to existing libraries (4) and scientific background information (5) can be added. The player can also view these information as one document (e.g. for printing) (6). Meta-information can be found on the right wall: a list of recent changes by editors (7), a diagram shows the relations of this pattern to other patterns (can be used instead of, is not compatible with, is a part of, contains sub-patterns) (8). Statistics about the number of visitors over time (9), the possibility to buy a share of this pattern (10) and icons representing the author (11) and participating editors (12) are also provided. Finally, the player can leave his comments in the Guest-book of this pattern (13).

All of these objects open a pop-up when being clicked that reveals the complete information. Other players and their actions are visible in real-time: here, Bob is currently editing the description (14), while Anne is studying the print version of the pattern (15).



**Figure 10:** Inside a Pattern building

In which respects this concept is a game? Although there is no clear goal in order to “win the game”, it describes a system of rules in which some states are more desirable than others. This open-endedness allows the player to fix himself the goal that he wants to reach. To own many patterns, to be an editor of many patterns, to receive many awards, or to have many friends are all equally valid, intrinsically anchored goals. The aesthetic environment and the open question about the role of The Architect also contribute to make “work” “pleasurable”, a serious game.

Note that the descriptions from the above paragraphs are not completely identical with the scenario (which was written first). Also, the terminology needs to be more consistent: the name of the special buildings, the distinction between authors, editors and users and other concept need refinement and a clearer definition. For now, these inconsistencies remain to remind that this isn't the final concept yet, but rather a prototype of it.

## 8- Conclusion

The use of patterns in Interaction Design, or related fields such as web design and GUI design, is gaining momentum in practice. Many patterns collections are now publicly available in books or online. With the huge number of patterns, organization and classification is becoming a practical issue. No definitive pattern collection is available, terminology and formats are different for each pattern which makes comparisons difficult.

In this paper, our goal is to increase the overall awareness of user interface patterns for the different stakeholders. We suggested a platform of three components: Pattern Database which regroups helpful information about user interface that is produced by the participants. Their collaborative activities are supported by the social network component which emphasizes the

communication. The last component is serious game, a context in which the earlier features are embedded.

Based on these components, the concept of PatternCity evolved. It consists of two parts: building in a virtual world that represents a pattern and avatars which represent players who can collaboratively build and improve the buildings. Even if we do believe that the proposed PatternCity concept is applicable to all types of design patterns, we have focussed on HCI/UI design patterns, mainly for two reasons. First, because of our expertise in the field of HCI as designers, researchers and educators, we have developed a very large catalogue of patterns. Secondly, HCI patterns are most often developed by people who have their background in psychology. This makes it challenging to describe them in way that they are easily transferable and understandable by software developers, the end users of these patterns. To realize this concept, we developed a paper prototype which needs real implementation using Alice.

## Acknowledgment

The first author expressed his thanks to all the students from INF 10 classes (Fall 2010 and 2011) that developed the PatternCity prototypes especially Mourad Ayoub and Benjamin Pick.

## References

1. Alexander, Ishikawa and Silverstein: A Pattern Language: Towns, Buildings, Constructions. Oxford University Press publishing, New York, 1977.
2. Ang and Zaphiris, 2005. Developing enjoyable second language learning software tools: A computer game paradigm. Idea Group, 2005.
3. Borchers. 2001. A Pattern Approach to Interaction Design, John Wiley & Sons; ISBN: 0471498289.
4. Deterding, S., Sicart, M., Nacke, L., O'Hara, K., Dixon, D.: Gamification. using game-design elements in non-gaming contexts. ;In CHI Extended Abstracts(2011)2425-2428
5. Franke et al. 2007 Franke, Demian ; Freischlad, Stefan ; Friedrich, Lars ; Haug, Florian ; Klein, Benjamin ; Koslowski, Rudolf ; Stechert, Peer ; Ufer, Jonathan : Final report of the project Design Patterns. Department of Electrical Engineering and Computer Science, University of Siegen. 2007.
6. Gaffar: Studies on Pattern Dissemination and Reuse to Support Interaction Design. Concordia University, Montreal, 2005 (Appendix E).
7. Gamma et al.: Design Patterns: Elements of reusable object-oriented software. Addison-Wesley-Longman publishing, MA, USA, 1995
8. Graham, I. (2003), A pattern language for Web Usability, Addison-Wesley, Boston, US.
9. Lin J., Newman M. W., Hong J. I., Landay J.A.: DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In CHI Letters: Human Factors in Computing Systems, 2, 1(2000), 510-517.
10. Rising, L. (2000) The Pattern Almanac 2000. Addison Wesley Publishing Inc, 2000.
11. Stahl, 2004. Building Collaborative Knowing. In: Strijbos, Kirschner and Martens (Eds.): What we know about CSCL: And implementing it in higher education, pp. 53-86, Boston, MA: Kluwer Academic Publishers, 2004.
12. Star, S.L. Griesemer, J.R. 1998. Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology. Social Studies of Science, 1998.
13. Taleb, M, Javahery, H and Seffah, A. 2006, 'Pattern-Oriented Design Composition and Mapping for Cross-Platform Web Applications', The XIII International Workshop, DSVIS

2006, July 26-28 2006, Trinity College Dublin Ireland, DOI 10.1007/978-3-540-69554-7, ISBN 978-3-540-69553-0, Vol. 4323/2007, Publisher Springer VerlagBerlin Heidelberg, Germany.

14. Van Duyne, D.K., Landay, J.A, Hong, J. 2002, The design of sites, Addison-Wesley, Boston, US.

15. Welie: Patterns in Interaction Design. 2008

<http://welie.com/patterns/showPattern.php?patternID=minesweeping>.

16. Welie, M.V., van der Veer, G.C., 2003. Pattern Languages in Interaction Design: Structure and Organization. In: Proceedings of Interact '03, 1-5 September, Zürich, Switzerland, Eds: Rauterberg, Menozzi, Wesson, p527-534, ISBN 1-58603-363-8, IOS Press, Amsterdam, The Netherlands .

17. Welie, V.M., Van der Veer, G. & Eliëns, A. (2000), Patterns as Tools for User Interface Design, in 'International Workshop on Tools for Working with Guidelines', Biarritz, France, pp. 313-32.

**ANNEX:** New Prototype of PatternCity

