

Interaction Design Patterns in Recommender Systems

Paolo Cremonesi

Dept. of Electronics, Information
and Bioengineering
Politecnico di Milano
Via Ponzio 34/5,
20133 Milan, Italy
paolo.cremonesi@polimi.it

Mehdi Elahi

Dept. of Electronics, Information
and Bioengineering
Politecnico di Milano
Via Ponzio 34/5,
20133 Milan, Italy
mehdi.elahi@polimi.it

Franca Garzotto

Dept. of Electronics, Information
and Bioengineering
Politecnico di Milano
Via Ponzio 34/5,
20133 Milan, Italy
franca.garzotto@polimi.it

ABSTRACT

Design patterns are acknowledged as powerful conceptual tools to improve design quality and to reduce the time and cost of design by effect of the *reuse* of “good” solutions. In many fields such as software engineering, web engineering, and interface design, patterns are widely used by practitioners and are also investigated from a research perspective. Still, the concept of design pattern has received marginal attention in the arena of user interfaces (UIs) for Recommender Systems (RSs). To our knowledge, a little is known about the use of patterns in this specific class of applications, in spite of their increasing popularity, and no *RS specific* interface pattern is available in existing pattern languages. We have performed a systematic analysis of 28 real-world RSs in a variety of sectors, in order to: (i) discover occurrences of existing general (i.e., domain independent) UI patterns; (ii) identify recurrent UI design solutions for RS specific features; (iii) elicit a set of new UI patterns for RS interfaces. The analysis of patterns occurrences highlights the degree at which “good” UI design solutions are adopted in RSs for the different sectors. The new patterns can be used by UI designers of RSs to improve the UX of their systems.

CCS Concepts

• Software and its engineering • Design patterns • Information systems • Recommender systems • Human-centered computing • Interaction design.

Keywords

Design, Human Factors, Standardization.

1. INTRODUCTION

A design pattern “... describes a problem which occurs over and over again in our environment and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice” [3]. It provides a descriptive structure for integrating the analysis and the solution of a recurring design problem, in a way that is sensitive to context, is informed by theory and evidence and is re-usable with a minimum degree of customization.

In many fields such as software engineering [6, 7, 10], web engineering [11, 16], and Interface Design [20, 96, 9], patterns are

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CHItaly 2015, September 28-30, 2015, Rome, Italy
© 2015 ACM. ISBN 978-1-4503-3684-0/15/09...\$15.00
DOI: <http://dx.doi.org/10.1145/2808435.2808442>

acknowledged as powerful conceptual tools to improve design quality and to reduce the time and cost of design by effect of reuse of “good” solutions. As such, they are not only investigated from a research perspective but are also widely used by practitioners in software design or UI design. Still, the concept of design pattern has received marginal attention in the arena of Recommender Systems (RSs).

RSs play an increasingly important role in online applications characterized by a very large amount of data - e.g., multimedia catalogs of music, products, news, images, movies, or tourism services. Their goal is to filter information and to recommend to users only the items that are likely to be of interest for them.

To our knowledge, in spite of the relevance of RSs in many domains, little is known about the use of design patterns in this specific class of applications and no RS specific pattern is available in existing pattern languages. In our research, we explore patterns for User Interface (UI) design of RSs. We have performed a systematic analysis of 28 real-world RSs in a variety of sectors, in order to: i) discover occurrences of existing general (i.e., domain independent) UI patterns; ii) identify recurrent UI design solutions for RS specific features and elicit a set of 30 new UI patterns for RS interfaces (5 of which are reported in the following sections).

The analysis comprises several steps including: (a) selecting real-world RSs and clustering them by “application domain” or “business sector”; (b) systematically inspecting the selected RSs using a pre-defined set of user scenarios; (c) identifying recurring design solutions; (d) matching these solutions against existing UI patterns available in a well-established pattern library, or articulating the description of these solutions in terms of new patterns.

Why are these results interesting from a research and practical perspective? UI design is progressively acknowledged as one of the crucial aspects to be considered for a “good” RS. While algorithmic features certainly plays an important role in RS quality, recent studies have highlighted a mismatch between the quality of algorithms and user perceived quality, which is largely mediated by the quality of the UI [28]. As UI patterns define “good” UI design solutions, the degree of adoption of UI patterns can be regarded as an indicator of UI design quality. Hence the process followed in our study suggests a method for heuristically inspecting, measuring, and comparing the UI design quality of RSs in different domain. In addition, the new UI patterns we have elicited provide RS specific “standard reference points” [1] that can benefit both novice and experienced user interface designers, helping them to create more usable RS UIs in a more efficient way.

2. RELATED WORK

The idea behind design pattern began in late eighties where with an early work of C. Alexander [3] in software engineering. At that moment, the design patterns were defined and used as means to capture design experience, that might help effectively to achieve the economies of scale for developing affordable and usable software applications [6, 7, 10]. Later on, the notion of design patterns has drawn very much the attention of the researchers in other disciplines as well, and a variety of design pattern repositories have been established [1, 23, 24, 26], such as Hypermedia and Web Engineering [11, 16], Human-Computer Interaction (HCI) [5, 8, 19, 21], Computer Supported Cooperative Work (CSCW) [4, 18, 17], and e-learning [12, 13, 14].

In the light of that, a set of digital online catalogues for design patterns has been established, including, the Hypermedia Design Patterns Repository [2], the design patterns for Web, GUI and Mobile Interfaces by M. van Welie [20], design patterns for game interfaces and social applications called Ericksons Interaction Design Patterns [36], the ninety web design patterns by K. Van Duyne [9], the thirty user interface patterns by J. Tidwell, with a range of various real-life examples from desktop applications to web sites and web applications for mobile devices [22], and the Pattern Language for CSCW in [15].

Despite such attention in variety of application domains, the notion of design pattern has received a little attention in the RSs research area: To our knowledge, there is a very limited research on this topic. The existing studies have focused only on building RSs for design patterns [25, 29, 30], addressing the problem of selecting and recommending design patterns during the software engineering process.

Our research takes a different perspective. We focus on design patterns adopted in RS UIs, investigating the degree at which *existing* design patterns are applied in the UIs of popular real-world RSs. In addition, we propose a set of brand *new* design patterns that are specific for RS UIs.

3. DEFINITION OF DESIGN PATTERN FORMAT

Generally, a design pattern is defined as a general reusable solution to a commonly occurring problem within a given context in software design. There is a set of main properties that the design pattern hold, listed in the following:

1. Solve a problem: Patterns capture solutions, not just abstract principles or strategies.
2. Be a proven concept: Patterns capture solutions with a track record, not theories or speculation.
3. Provide a not obvious solution
4. Have a significant human component

Moreover, there is a formulation of design patterns that conforms the conceptual structure proposed by Alexander [3], described below:

- **Name:** The pattern name is a meaningful and memorable identifier that succinctly grasps the essence of a problem in such a way as to be clearly understood by all members of a design community and a sudden association with the core feature of the referred design solution can be easily made.
- **Problem:** By knowing the problem, the designer should be able to evaluate the relevance and the applicability of

that pattern to the situation s/he is coping with and to achieve a better understanding of the potential effectiveness of the pattern.

- **Context (or Usage):** From a clear definition of the environment and context of use in which the problem and the solution are likely to recur, a designer can understand the preconditions under which s/he will probably meet the problem, thus improving the problem-matching process.
- **Forces:** Forces define the constraints, relationships, contrast and conflicts permeating the scene in which the pattern acts. Explaining forces may help to realize which tradeoff must be considered while adapting the pattern to a specific design situation.
- **Solution:** The solution is the essence of the design experience the pattern wants to convey. A solution is composed of a list of rules or easy-to-remember guidelines describing how to shape the desired artifact, to help the designer while implementing the pattern in a concrete analogue situation.
- **Rationale:** The rationale should explain the key factors that make the pattern solution really useful, effective and valuable. The rationale tells us why the patterns provide a good solution to the stated design problem. The actual basic strategies by which forces and constraints are managed in order to achieve a certain task are also described here. While the pattern solution can be viewed as the body of the pattern that operates, the rationale is the soul of the pattern, its inner motivation of behaving.
- **Examples:** To help the designer understand the use of a pattern and its applicability, it could be especially illuminating to provide one or more sample examples and known uses of the pattern in specific contexts.
- **Related Patterns:** Relationships among patterns can be established for different reasons. A pattern can accomplish a specific task within a larger design strategy and its synergy with other patterns can more effectively achieve the goal of supporting design. Two or more patterns can be related because they try to solve a (portion of a) similar design problem, or because they can be considered as slightly different variants of the same design solution. Different patterns applicable in different contexts can share key factors or design elements, which is another reason why a relationship may arise.

It worth noting that this is the most complete specification of design pattern formulation, which subsumes the formats, adopted by most of the existing pattern languages.

For simplicity and lack of space, in this paper, the new patterns discovered in our research are reported using a simpler, more compact version, which includes Name, Problem, Usage, Solution, and Examples.

4. REASEARCH METHODOLOGY

The first step toward conducting the analysis was the selection of 28 web-based RSs.

Afterward, each RS has been categorized based on the application domain, according to the six categories described in [27]: Online Dating, Photo Sharing, Social Bookmarking, Social Network, Social News, and Tourism services.

The complete list of RSs analyzed in this study is shown in Table 1.

Table 1. Inspected RSs and Considered Usage Scenarios

Domain	RS Title	Recommended Items
Online Dating	Meetic, Badoo, PerfectMatch	User Profiles
Photo Sharing	Flickr, DeviantArt, Imgur, Photobucket	Photos
Social Bookmarking	StumbleUpon, Pinterest, WeHeartIt	Online content
Social Network	Facebook, LinkedIn, Twitter, MySpace, Google+, FourSquare	User Profiles, Posts, Offers, POIs
Social News	Reddit, 9GAG, Digg	Online content
Tourism Services	Booking, AirBnB, TripAdvisor, Holiday, Watchdog, Gogobot, Volagratis, Trivago, Yelp	POIs

For each of the selected websites, we have defined a use case scenario. For each scenario, we considered a first-time user and we imagined a flow starting from the homepage, going through the registration process, and ultimately landing to the recommendation page. The scenarios used in our inspection are described in Table 2.

Table 2. Considered Usage Scenarios

Domain	Considered Usage Scenario
Online Dating	A single man wants to find his soulmate, and for that, he registers to an online dating website. The website asks him to create a profile by entering his personal information. Based on his profile, a set of women, who has similar interests and living close to his place, is recommended to him.
Photo Sharing	A young girl whose hobby is photography, wants to share some of her taken photos. She registers to a photo sharing website, creates an album, and uploads the photos. She also browses a set of photos, recommended by the website, and notices that a number of them are quite interesting. She clicks on some of the recommended photos, and reads the details about them.

Social Bookmarking	A girl wants to find some pictures about some of her interesting topics, and for that, tries a website advised to her. In order to use the website, she is asked to register by entering basic information. The website tries to learn her preferences by presenting her a set of pictures and by asking her to select the most interesting ones. Finally, the website recommends some pictures she may be interested in and asks her to tag them.
Social Network	A mid-aged woman wants to find her childhood friend whom she has not met for many years. She is advised looking on a social network, while she is not even familiar with properly using internet. She tries to register by entering some information. She discovers some friends and adds them to her friend list. Then she searches her childhood friend by name, and easily finds her and adds her to the friend list. Later on, the website, presents to her some recommendations of people who attended the same school.
Social News	A young adult is reading an online newspaper and finds an article interesting to him. He wants to share the article and eventually open up a discussion. He proceeds with fast registration to a social news website, finds the relevant news channel recommended by the system. He later posts the article by inserting the article link and writing his opinion about it.
Tourism Services	A young couple wants to spend their holiday in London. They would prefer to make an online reservation of their accommodation and they register to a website for online booking of hotels and bed & breakfast. They enter some information requested by the service and then receive a list of recommendations ordered by price. They choose one of them and make reservation.

5. RESULTS AND DISCUSSIONS

In this section, we present and discuss the results of our analysis. We split this section into two parts: discussion on generic design patterns (i.e., design patters not specific to RSs) and discussion on design patterns specific to RSs,

5.1 Generic UI Design Patterns for RSs

The first part of the analysis is focused on measuring how many and which generic UI design patterns are used in each of the six categories of recommender systems.

Figure 1 shows the percentage of generic UI design patterns used in the different categories of RSs. The categories with the greatest usage of designs pattern are Tourism Services and Online Dating. The categories with the lowest usage of design pattern are Social Bookmarking and Photo Sharing. Comparing these results, we observe that the categories with the greater usage of design pattern represent more complex types of RSs compared to the other categories and, as such, they likely require further exploitation of design patterns. In fact, the functionalities offered by Tourism Services and Online Dating systems, as well as the interaction

models, are more complex in comparison to the other categories. Accordingly, the RSs within the categories with the lowest usage of design patterns do typically follow a standard interaction design with a more straightforward user interaction model.

DP Usage over Categories

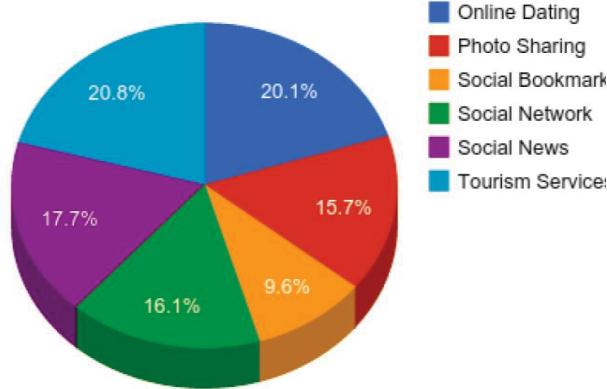


Figure 1 UI Design Pattern usage for different categories of RSs

Figure 2 shows the 11 most used generic design patterns, the first being Pagination and Input Prompt. The Pagination pattern addresses the user's need to view a list of data items that is not easily displayed in a single page. The Input Prompt pattern addresses the need of the user to provide data to the system [1]. Both of these patterns are important in RSs as they address two essential tasks in RSs, i.e., collecting data from users to generate recommendations, and, displaying the recommendations to the users. Hence, expectedly, these design patterns have been the most adopted ones in the RSs, we inspected.

Most used General Design Patterns

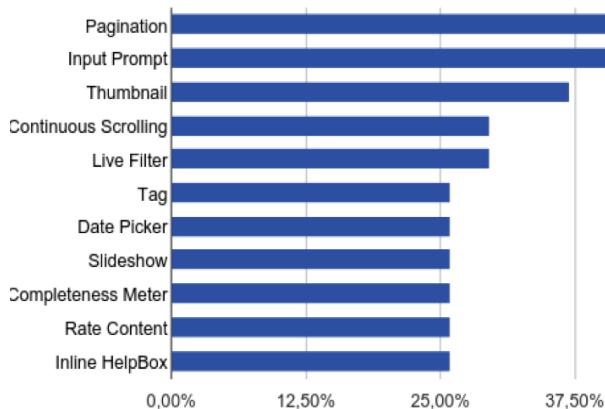


Figure 2 The most applied design patterns in RS UIs

5.2 Specific UI Design Patterns for RSs

In addition to the occurrences of generic UI design patterns, we have elicited 30 UI design patterns that we identified as *new and specific to RSs*. In this section, we report 5 of them, which in our opinion are more relevant for RSs UI design. The full set of patterns defined in our research can be obtained by contacting the authors.

5.2.1 Design Pattern 1: Social Connection

Problem: A user knows some other users and would like to connect to them through the system. The RS can use some already established connections among all the users, in order to recommend to that user whom to connect with.

Usage:

- Use when user connections are the core part of the system's experience.
- Use when relationships will be confirmed providing a two-way reciprocal relationship.
- Use when ignoring a connection request is allowed.
- Use when the system needs social connection among users to generate recommendations.
- Use when the system generates recommendation for a user based on the tastes and preferences of her friends.

Solution: Provide a button or a link to add a person, whom the user knows her as a friend. Once the person has been added as a friend, clearly indicate to the user that this person is now a friend. The system can now also make an analysis of common interests in order to provide recommendations.

Rationale: Indeed, people are interested in doing things with their friends or with others who have similar tastes or interests, and allowing them to connect, strengthens these ties. It also provides a better estimation about what could be the users' interests or tastes. Allowing the users to connect to others, encourages them to make conversation, and to share information amongst themselves, which improves the social aspect of the system and makes the system viral.

Examples: Google Plus, Facebook



Figure 3 An example of design pattern 1: Social Connection in Facebook

5.2.2 Design Pattern 2: Add Comments

Problem: The users would like to provide their opinion on certain items.

Usage:

- Use in systems that use comments/reviews to recommend items to the users (Content-Based RSs).
- Use in systems that allow the user to write opinions about items.

Solution: The system recommends items based on the content of reviews/comments given. The comments to the items are analyzed in order to improve recommendation quality. The RS considers the comments, as description of items, and exploits them to generate recommendations to users. For example, for a user who is more interested in movies with positive comments, the system may analyze the comments of movies and recommend those movies, which received positive comments.

Rationale: The systems that use this pattern assume that people are more prone to give opinion directly to other users using natural language, instead of using other interaction tools, such as like/dislike or star ratings. The items with a large number of comments are supposed to be somehow interesting for the community, and they can be, for example, shown on the top of the activity stream of such systems. This is true in an equal form for items with positive reviews, negative reviews, and controversial reviews.

Examples: DeviantArt, Facebook, Twitter



Figure 4 An example of design pattern 4: added comments in Deviantart

5.2.3 Design Pattern 3: Social Login

Problem: The RS initially requires some information about the user. Such information can be elicited implicitly from user's existing online profiles, such as her profiles in social networks.

Usage:

- Use when the user is required to create a profile before being able to use the RS.
- Use when the RS requires collecting information about the user (e.g., demographics, social interactions, friend list, colleagues list, followers, etc.) in order to generate relevant recommendations for her.
- Use when the RS provides easy and quick registration method for the new users.

Solution: To properly adopt this design pattern the RS should implement a particular registration method. This method allows the user to login to the system by simply entering the login information of other (mainly social networks) websites. This allows the system to collect her profile information in those websites. Such websites may range from social networks, such as Facebook, Google+, or Twitter, to even email service providers such as Gmail. In practice, the RSs provide a button, typically called *Connect With*, placed in the homepage or registration form, which allows the user to establish a secure connection with a third party system and login with her credentials.

Rationale: The RS that provide this registration method bases its design choice on the fact that some users may prefer to use their existing online profiles to avoid repeatedly entering their information in several websites or to speed-up registration process. Some may even want to allow the system to improve the recommendation quality, by letting the system to implicitly mine their social interactions, rather than explicitly requesting them to provide further information.

Hence, they can register using their login data, stored in another website. It worth noting that, sometimes, the users may be concerned with the security level of the system, and opt for a trusted third party login process that offers a better level of

security. In such a case, the profile information of the user can be obtained from that third party that provides the secure registration service.

Examples: WeHeartIt, Facebook, TripAdvisor

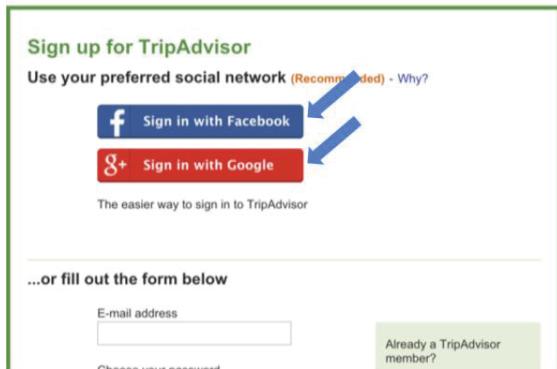


Figure 5 An example of design pattern 3: social login in TripAdvisor

5.2.4 Design Pattern 4: Similar Content

Problem: The user wants to explore content similar to what she is viewing.

Usage:

- Use when the user wants to see content similar to what she reads.
- Use when the system wants to recommend the content that the user can be familiar with.
- Use when items are provided a graphical representation, such as an image.

Solution: Content similar to what the user is currently viewing is recommended and shown.

Rationale: In order to better describe this design pattern, we can refer Gestalt psychology, i.e., a theory of mind, which tries to understand the laws of our ability to acquire and maintain meaningful perceptions in an apparently chaotic world. In particular, a major aspect of Gestalt psychology is that it implies that the mind understands external stimuli as a whole rather than the sum of their parts. The wholes are structured and organized using grouping laws. We focus on one law that fits perfectly to the scope of this design pattern: the law of proximity states that when an individual perceives an assortment of objects, she perceives those objects that are close to each other as forming a group. Therefore, this design pattern shows similar content in a position close to the position of the currently displayed one, with the aim

to form a group. The user will likely perceive the elements as part of a group, and hence, will consider them similar.

Examples: Facebook, DeviantArt



Figure 4 An example of design pattern 4: similar content in Facebook

5.2.5 Design Pattern 5: Profile as Business Card

Problem: The system needs to display a link or a short summary of user profile.

Usage:

- Use when implementing social network functionality.
- Use when it is important to help the user recognize other users.
- Use when a compact representation of a profile is needed.

Solution: A user profile is represented as a classical Business Card, and it contains at least an image, the full name and some additional information, which helps people to identify that user.

Rationale: This pattern resembles the classical business card. Users may enjoy finding elements of UI that they may associate to their previous experiences, such as business card format that resembles a tangible object. This may make the proposed pattern a

straightforward and intuitive element in the UI design of user profiles.

Examples: Facebook, Google+



Figure 6 An example of design pattern 5: profile as business card in PerfectMatchbody.

6. CONCLUSION AND FUTURE WORK

In this paper we describe the results of a wide inspection of the design solutions adopted in the UIs of real-world Recommender System. We have performed a comprehensive analysis and identified a wide range of instances of existing “generic” UI design patterns, which can be adopted in any UI design. In addition, we have discovered a number of recurrent UI design problems and solutions that are specific to RSs. These have been distilled into the specification of 30 design patterns for RS user interfaces, examples of which are described in the paper.

To the best of our knowledge, this is a novel contribution. No previous study has analyzed UI for RSs from a design pattern perspective, nor has identified new UI patterns for this specific class of applications.

Our results can have benefit for RS practitioners, helping them to both improve the UX quality of RSs, and support a more systematic approach to RS interfaces.

In addition, our work can pave the ground for future research bridging RSs and Interface Design by means of design patterns: There are a number of interesting topics in this arena that have not been addressed yet in our work and can be considered for future studies.

Firstly, it is important to study the possible correlation between the design patterns adopted in RSs and the type of items recommended by them. For example, the UI design for music RSs would be different from computer products RSs. One difference can be that in music RSs the user can simply and immediately experience the recommended item (e.g., by listening to the recommended music items) while this is not feasible in computer RSs. Hence, there might be various differences in UI design of these two types of RSs, and hence, differences in the design patterns adopted.

Another line of research for future work is the investigation of design patterns adopted in RSs that fit different types of recommender algorithms. For example, the design patterns adopted in Collaborative Filtering RSs can be different from Knowledge-based RSs. In the former type, the user interaction with the system typically begins with rating elicitation, where the system obtains a number of ratings. However, in the later system, the user typically provides her preference in the form of the query

she makes. Hence, the design patterns might substantially differ in these two types of RSs.

7. REFERENCES

- [1] Bowman, Design patterns. ui-patterns.com/patterns. Accessed: 3 April 2015.
- [2] Hypermedia design patterns repository. designpattern.lu.unisi.ch/PatternsRepository.htm. Accessed: 17 December 2008.
- [3] C. Alexander. The timeless way of building, volume 1. Oxford University Press, 1979.
- [4] M. Arvola. Interaction design patterns for computers in sociable use. International Journal of Computer Applications in Technology, 25(2):128–139, 2006.
- [5] J. O. Borchers. A pattern approach to interaction design. *Ai & Society*, 15(4):359–376, 2001.
- [6] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. Pattern- oriented software architecture: A system of patterns. Addison Wesley, 1996.
- [7] J. O. Coplien and D. C. Schmidt. Pattern languages of program design, volume 58. Addison-Wesley Reading, 1995.
- [8] A. Dearden and J. Finlay. Pattern languages in HCI: A critical review. *Human-computer interaction*, 21(1):49–102, 2006.
- [9] D. K. V. Duyne, J. Landay, and J. I. Hong. The design of sites: patterns, principles, and processes for crafting a customer-centered Web experience. Addison-Wesley Longman Publishing Co., Inc., 2002.
- [10] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design patterns: elements of reusable object-oriented software. Pearson Education, 1994.
- [11] F. Garzotto, P. Paolini, D. Bolchini, and S. Valenti. modeling-by-patterns of web applications. In *Advances in Conceptual Modeling*, pages 293–306. Springer, 1999.
- [12] F. Garzotto and C. Poggi. Design patterns for collaborative learning in shared virtual worlds. *Technology-enhanced learning: design patterns and pattern languages*. IGI Global (2009to appear).
- [13] F. Garzotto and S. Retalis. Symposium on design patterns for e-learning. In *EDMEDIA 2004*, Lugano, Switzerland, 2004.
- [14] P. Goodyear, P. Avgeriou, R. Baggetun, S. Bartoluzzi, S. Retalis, F. Ronteltap, and E. Rusman. Towards a pattern language for networked learning. In *Networked learning*, pages 449–455, 2004.
- [15] D. Martin, T. Rodden, I. Sommerville, M. Rouncefield, and J. Hughes. Pointer: Patterns of interaction: A pattern language for cscw. comp.lancs.ac.uk/computing/research/cseg/projects/pointer/pointer.html, 2002. Accessed: 17 December 2008.
- [16] G. Rossi, D. Schwabe, and A. Garrido. Design reuse in hypermedia applications development. In *Proceedings of the eighth ACM conference on Hypertext*, pages 57–66. ACM, 1997.
- [17] T. Schümmel. Gama: A pattern language for computer supported dynamic collaboration. In *EuroPLoP*, pages 53–114, 2003.

- [18] T. Schummer. A Pattern Approach for End User Centered Groupware Development. FernUniversität in Hagen, 2005.
- [19] T. Schummer and S. Lukosch. Patterns for computer-mediated interaction. John Wiley & Sons, 2013.
- [20] M. van Welie. Design patterns for web, gui, and mobile interfaces. [welie.com](http://www.welie.com). Accessed: 17 December 2008.
- [21] M. Van Welie and G. C. Van der Veer. Pattern languages in interaction design: Structure and organization. In Proceedings of interact, volume 3, pages 1–5, 2003.
- [22] J. Tidwell. Designing interfaces. "O'Reilly Media, Inc.", 2010.
- [23] Interaction Design Patterns. https://www.interaction-design.org/encyclopedia/interaction_design_patterns.html. Accessed: 9 April 2015.
- [24] Build Your Perfect Interface with UI Design Patterns. <http://www.sitepoint.com/build-your-perfect-interface-with-ui-design-patterns>. Accessed: 9 April 2015.
- [25] P. Francis, H. Farzin, Y. G. Guéhéneuc, and N. Moha. "Recommendation system for design patterns in software development: An dpr overview." In Proceedings of the Third International Workshop on Recommendation Systems for Software Engineering, pp. 1-5. IEEE Press, 2012.
- [26] Interaction Design Patterns Library. <http://www.welie.com/patterns/index.php>. Accessed: 9 April 2015.
- [27] R. K. Nageswara. "Application domain and functional classification of recommender systems—a survey." DESIDOC Journal of Library & Information Technology 28, no. 3 (2010): 17-35.
- [28] P. Cremonesi, F. Garzotto, and R. Turrin. "Investigating the persuasion potential of recommender systems from a quality perspective: An empirical study." ACM Transactions on Interactive Intelligent Systems (TiiS) 2, no. 2 (2012): 11.
- [29] Y. G. Guéhéneuc, R. Mustapha. "A simple recommender system for design patterns." Proceedings of the 1st EuroPLoP Focus Group on Pattern Repositories (2007).
- [30] B. Nadia, A. Kouas, and H. Ben-Abdallah. "A design pattern recommendation approach." In Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on, pp. 590-593. IEEE, 2011.