Multimed Tools Appl
DOI 10.1007/s11042-016-3946-5

User interface patterns in recommendation-empowered content intensive multimedia applications

Paolo Cremonesi [1] **· Mehdi Elahi** [1] **· Franca Garzotto** [1]

Received: 15 December 2015 / Revised: 5 September 2016 / Accepted: 6 September 2016

Abstract Design Patterns (DPs) are acknowledged as powerful conceptual tools to improve design quality and to reduce time and cost of the development process by effect of the reuse of "good" design solutions. In many fields (e.g., software engineering, web engineering, interface design) patterns are widely used by practitioners and are also investigated from a research perspective. Still, they have been seldom explored in the arena of Recommender Systems (RSs). RSs provide suggestions ("recommendations") for items that are likely to be appropriate for the user profile, and are increasingly adopted in content-intensive multimedia applications to complement traditional forms of search in large information spaces. This paper explores RSs through the lens of User Interface (UI) Design Patterns. We have performed a systematic analysis of 54 recommendation-empowered content-intensive multimedia applications, in order to: (i) discover the occurrences of existing domain independent UI patterns; (ii) identify frequently adopted UI solutions that are not modelled by existing patterns, and define a set of new UI patterns, some of which are specific of the interfaces for recommendation features while others can be useful also in a broader context. The results of our inspection have been discussed with and evaluated by a team of experts, leading to a consolidated set of 14 new patterns that are reported in the paper. Reusing pattern-based design solutions instead of building new solutions from scratch enables novice and expert designers to build good UIs for Recommendation-empowered content intensive multimedia applications more effectively, and ultimately can improve the UX experience in this class of systems. From a broader perspective,

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

✉ Mehdi Elahi
mehdi.elahi@polimi.it

Paolo Cremonesi
paolo.cremonesi@polimi.it

Franca Garzotto
franca.garzotto@polimi.it

[1] Department of Electronics, Information and Bioengineering, Politecnico di Milano, Via Ponzio 34/5, 20133 Milan, Italy

Published online: 24 September 2016

our work can stimulate future research bridging Recommender Systems, Web Engineering and Interface Design by means of Design Patterns, and highlights new research directions also discussed in the paper.

Keywords Multimedia · Recommender Systems · Design Patterns · Human Factors · HCI · Standardization

1 Introduction

A design pattern (DP) distills a portion of design know-how in a specific domain, and presents in a compact form a design solution that has been proved to be effective to address a commonly occurring problem in that domain.

In many fields such as software engineering [19, 20, 43], web engineering [44, 67], and HCI (Human-Computer Interaction) [38, 54, 81], design patterns are acknowledged as powerful conceptual tools to improve the design quality of software architectures, on-line applications, or user interfaces, to prevent issues that can cause major problems in the following phases, and to reduce the time and cost of the development process by providing tested solutions that can be reused [75]. Design patterns can improve the communication and reporting activities; They enable designers and developers who are familiar with the patterns to discuss, document and share design specifications or alternatives using well-known, well understood names for design solutions instead of long descriptions. In the HCI domain, the adoption of popular, widely used interface design solutions makes the UX more intuitive and predictable, capitalizing on the user's familiarity.

Despite their success in various academic and industry contexts, design patterns have received marginal attention in the arena of Recommender Systems (RSs) [22].

A RS provides the user with suggestions ("recommendations") of items that are likely to be appropriate to his/her profile, characteristics, or intentions [33, 55]. RSs are widely used in what we referred to as "content intensive applications", characterized by a very large amount of online multimedia information, either made available by the service provider or user generated (think of the 3 billion videos uploaded on YouTube by late 2012 [57] or the 20 million songs on Spotify [66]). In these contexts, the dimension of the multimedia search space and the availability of an enormous set of choices may slow down the user's exploration, reduce the visibility of some potentially interesting items, and increase the complexity of the decision making process [8, 28, 29, 59, 71, 79]. Complementing (and in some cases even replacing) free navigation and traditional query-based paradigms, recommendations can alleviate the above problems, and reduce the information overload by focusing the search space and orienting the user's decisions [9].

An increasing number of online services (e.g., multimedia catalogs of music, news, images, movies, physical products, or tourism services) integrate today recommendation features; the design and implementation of which are becoming regular tasks within the development process of content-intensive multimedia applications.

To our knowledge no publically available documentation exists that describes patterns for recommendations design or reports about the use of patterns in the RS arena [22]. This paper focuses on the User Interface (UI) design dimension and explores UI patterns in relationship to RSs [21]. Starting from "general" UI patterns, which address the design of domain-independent interfaces and are described in an existing pattern library, we explore the degree to which the corresponding design solutions are adopted in content-intensive online applications that involve recommendation features. In addition, we identify a set of novel UI design patterns; some of them are specific for recommendation-empowered data-intensive multimedia applications, and others have a broader applicability.

Our research has comprised several steps including:

(a) selecting and clustering (by "application domain" or "business sector") 54 data-intensive multimedia applications that include recommendation features;
(b) systematically inspecting the interfaces of all selected applications using a pre-defined set of user scenarios;
(c) identifying recurring UI design solutions;
(d) matching these solutions against the UI patterns available in a well-established pattern library, or, when no match is found, articulating the description of these recurrent solutions in terms of new design patterns;
(e) validating the set of new patterns with a panel of RS experts and UI designers.

The rest of this paper is organized as follows. Section 2 provides an overview of the main concepts and results related to recommender systems and design patterns, to help the reader better contextualize our research. This section also discusses the formulation of patterns and introduces the format we have adopted. Section 3 describes the research methodology and procedures. Section 4 presents the main results concerning the use of *existing* UI patterns in recommendation-empowered content-intensive applications and describes 14 new patterns. Section 5 draws the conclusions and outlines future research directions.

2 Related work

2.1 Recommender system

The term "Recommender System" (RS) denotes a functionality (often integrated with other information access features e.g., navigation and query-based search) that suggests items estimated to be "desirable" or "interesting" for the user within a large set of items [2, 3, 65, 76]. Several techniques have been proposed and used for the generation of recommendations [4–6, 17, 41, 53, 64], the most popular of which are named Content-based Filtering (CBF) [11, 46], Collaborative Filtering (CF) [35, 56], and Hybrid [1, 18].

Content-based approaches exploit explicit features of items (e.g., metadata) or implicit ones (derived from the interpretation of non-structured data, e. g. video files [31, 32]. CBF suggests items that have characteristics similar to the ones the user has liked in previous experiences with the application, or has shown to like in the ongoing interaction with the application. For instance, news recommender systems consider the terms in the news as features and recommend the news articles that have features similar to the ones the user preferred before.

Collaborative filtering approaches "ignore" content and exploit collective preferences of the crowd, i.e., they generate recommendations using different users' rating profiles, suggesting items that other users with similar tastes "liked" in the past. The degree to which two user tastes are deemed similar is based on the similarity of their rating histories. Roughly speaking, the approach can be summarized as "people who watched this TV program also watched . . . " [25].

Each of the above techniques have pros and cons. To cope with their limitations, hybrid approaches implement a mix of content-based and collaborative techniques, and generate recommendations based on the item properties as well as community preferences.

The quality of a RS can be defined either in terms of system-centric metrics, evaluated offline, or with user-centric experiments, evaluated online. The system-centric approach evaluates the recommender system against a pre-built ground truth dataset of opinions. Users do not interact with the system under test but the evaluation, in terms of accuracy (e.g., recall, precision, mean squared error), is based on the comparison between the opinion of users on items as estimated by the recommender system and the judgments previously collected from real users on the same items. In user-centric evaluation users interact with a running recommender system and receive recommendations. Feedback from the users is then collected by either asking them explicitly (interviews, surveys) or by observing them and then subjecting system logs to various analyses (e.g., click through, conversion rate).

A prior research [25] has provided some empirical evidences that system-centric and user-centric quality methods may lead to inconsistent results, e.g., RSs that were the "best" according to system-centric measures were not the top ones according to user-centric measures. Also other works [13, 23, 24, 27, 61, 77] have pinpointed that system-centric quality might not always correlate with user-centric quality, as the latter may depend on factors that go beyond the characteristics of the recommendation algorithm itself.

2.2 Design patterns and their formulation

The idea behind design patterns began in late eighties with the early work of C. Alexander [7] in the field of architecture. Alexander defined design patterns as "generic, not obvious, reusable solutions to commonly occurring problems within a given context" that have a "practical" nature and a significant human component, being grounded on design experience rather than abstract principles or theories.

In the nineties, the idea of design pattern was enthusiastically embraced by several researchers and practitioners in software engineering, who defined and used patterns as a means to capture software design experience, to support quality, standardization, reusability and maintainability of software artifacts, and to achieve the economies of scale for developing affordable and usable software applications [19, 20, 43, 49]. Later on, the notion of design patterns has drawn the attention other communities and various design pattern collections have been created [16, 34, 51, 52], such as in the fields of Hypermedia and Web Engineering [44, 67], Human-Computer Interaction (HCI) [14, 30, 74, 80] Computer Supported Cooperative Work (CSCW) [10, 72, 73] and e-learning [36, 45, 47, 58]. A set of digital online catalogues for design patterns have been appeared, including the Hypermedia Design Patterns Repository [50], the "Design patterns for Web, GUI and Mobile Interfaces" by M. van Welie [81], design patterns for game interfaces and social applications called "Ericksons Interaction Design Patterns" [37], the 90 web design patterns by K. Van Duyne [38], the "30 user interface patterns" by J. Tidwell (including a range of various real-life examples from desktop applications to websites and web applications for mobile devices [78]), and the Pattern Language for CSCW in [60].

Despite such attention in many disciplines, the notion of design pattern is largely unexplored in the RSs research area and to our knowledge there are limited results on this topic. The existing studies have focused only on building RSs *for* design patterns [42, 48, 62], addressing the problem of selecting and recommending appropriate design patterns during the software engineering process.

In our research we focus on design patterns for the interfaces of the recommendation features of content intensive multimedia applications. We investigate the degree at which *existing* UI design patterns are applied in this context and we propose a set of *new* design patterns.

In the original formulation proposed by Alexander, patterns were described through natural language and their conceptual structure (see Fig. 1) comprised the following components:

- **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. It should be easy to make an association of the pattern name with the core feature of the referred design solution.
- **Problem** helps the designer 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**) is the clear definition of the environment and the context of use in which the problem and the solution are likely to recur. By knowing the context, a designer can understand the preconditions under which s/he will probably meet the problem, thus improving the problem-matching process.
- **Forces** define the constraints, relationships, contrasts 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** is the essence of the design experience the pattern wants to convey. A solution is composed of easy-to-remember rules and guidelines that describe how to shape the desired artifact, to help the designer while implementing the pattern in a concrete analogue situation.
- **Rationale** explains the key factors that make the pattern solution really useful, effective and valuable. The rationale manifests the reason a pattern provides a good solution to the

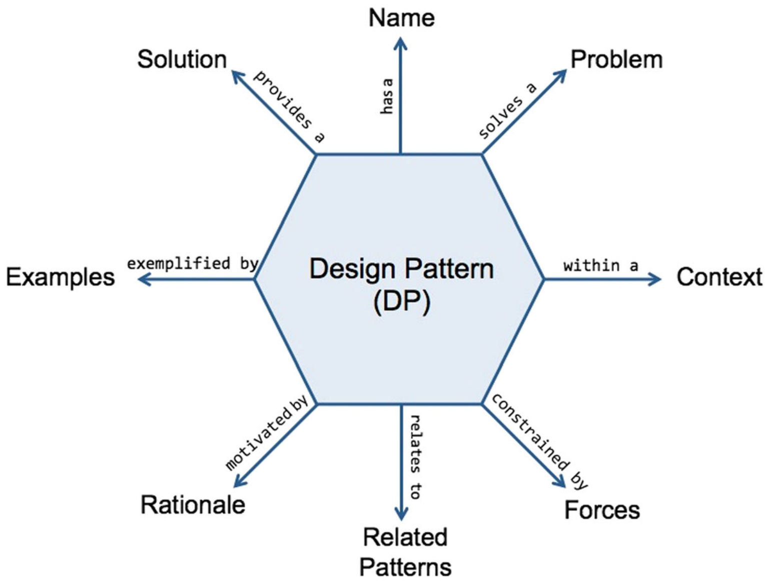

Fig. 1 Conceptual structure of a design pattern [58]

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** help the designer understand the use of a pattern and its applicability; it could be useful to provide one or more sample examples and known uses of the pattern in specific contexts.
- **Related Patterns** indicate relationships among patterns that 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.

Some authors (Bottoni et al. [15]) claim that the full realisation of the power of design patterns, at least in the field of software engineering, is hindered by the lack of a standard, generic formalization of the notion of pattern. Therefore they propose a formal, visual approach to the specification of software design patterns to facilitate pattern instantiation, identification, and composition, as well as analysis of pattern conflicts. Still, they also acknowledge that a natural language specification is necessary and useful for pattern documentation and communication: defining patterns using a formal approach, and understanding them, require expert knowledge of the used underlying formalism, which is rarely found in the average software engineer. As a matter of fact, most existing patterns languages [10, 14, 16, 30, 34, 44, 51, 52, 67, 72–74, 80] adopt informal, natural language formulations only; in most cases the pattern structure includes a subset of the components of Alexander's template comprising Name, Problem, Usage, Solution, and Examples. We will use this style and simplified formulations to describe the patterns reported in the rest of the paper.

3 Research methodology

3.1 General Procedure

In the design pattern community there are two main approaches to define patterns. Some authors claim that patterns can only be created and validated by expert designers, by reflecting on their own experience, selecting the best possible solutions, and comparing them with other designers' proposals. Other authors propose that patterns are "discovered", not "invented" [68, 69], implicitly suggesting that not only the identification but also the validation of a pattern is largely founded on the frequency of use of a given design solution. This approach has been extensively applied in the field of software engineering, mining the instances of design solutions from the source code or the design specifications of "good" software systems [37].

Our procedure of pattern definition has implemented a mix of the two approaches: *pattern mining* and *expert validation*. Figure 2 schematically illustrates the steps progressively performed:

1. *selecting* the recommendations-empowered content intensive applications for the study

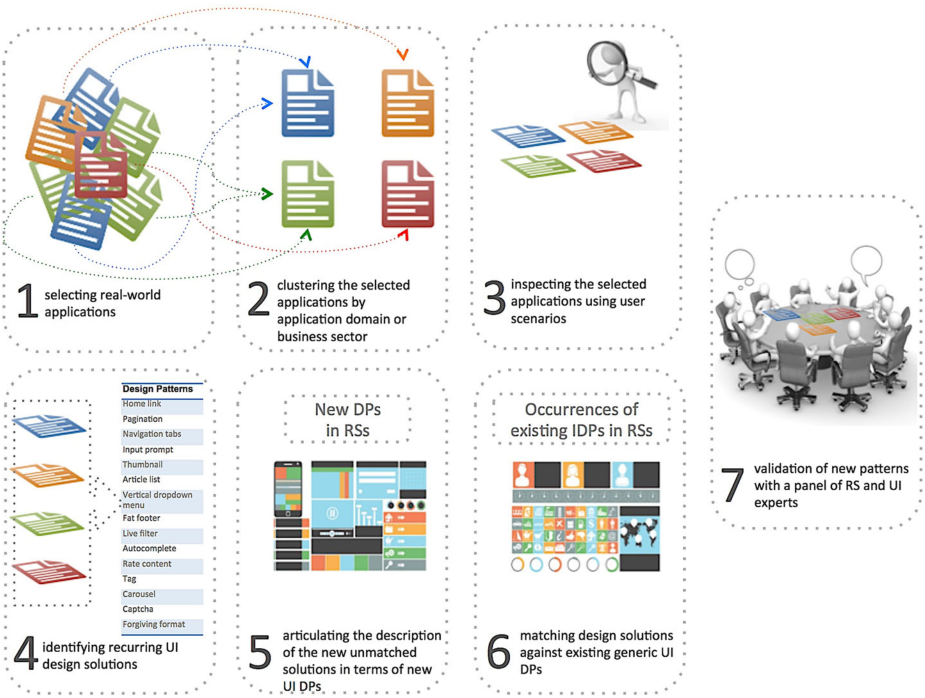

Fig. 2 Flowchart of the research procedure

2. *clustering* them by "application domain" or "business sector";
3. systematically *inspecting* the selected applications using a set of *user scenarios* defined for each cluster
4. *identifying* recurring UI design solutions;
5. *matching* these solutions against *existing UI patterns*;
6. articulating the description of most frequent *unmatched* solutions in terms of *new UI patterns*;
7. submitting of the new patterns to *public scrutiny* for validation, involving a panel of RS experts in RS and interface design.

A team composed by the authors and 6 experts in HCI and RSs performed steps 1–6; the public scrutiny (step 7) involved the authors and a panel of external experts (9 in the RS field and 4 in HCI).

3.2 Selecting Applications

To select of the applications to be used for the purpose of our study, each team member proposed a set of candidates and listed them in a shared Google Drive document. After removing duplications, the list included *54* applications, which belong to variety of domains. Each inspector proposed a classification of each candidate using a slightly modified version of the RS taxonomy described in [63]: Video Sharing / Multimedia Data Base, Music Streaming, News, Photo Sharing, Online Bookstore / Book Digital Library, Online Dating, Social Bookmarking, Social Network, Social News, and Tourism Services. The classifications were discussed and agreed among all group members and the applications were clustered by domain, as shown in Table 1.

3.3 Scenario-based Inspection and Pattern Mining

During pattern mining, the team inspected the selected applications and identified the most frequent UI design solutions. Some solutions concerned generic UI features and matched some existing patterns. As a baseline, we used the Pattern Library described in [34], which was chosen because of its popularity, online availability, richness of patterns and clarity of their formulation, and wideness of examples. This matching process enabled us to assess the degree to which existing patterns are adopted in content intensive multimedia applications that involve recommendations features. The solutions not matching existing patterns were integrated in the formulation of *new* patterns.

The inspection followed a *scenario-based approach* inspired by the web inspection method described in [40]. In general terms, scenarios are "stories of use" of a system [12] and are well-known conceptual tool in HCI. They are used in different moments of the development process (most frequently requirement management, UX design and evaluation) and have different formulations and levels of abstraction depending on the intended use. For the purpose of our study, scenarios are meant to provide guidelines for inspectors that make individual pattern-mining more systematic and results less subjective. The team defined a scenario for each application cluster, i.e., domain (Table 2). Each scenario presents a sequence of tasks that a "typical" user would perform in the applications of the cluster, e.g., accessing the homepage, going through a registration process, and ultimately landing to a page where recommendations are shown.

An on-line form was used to document the analysis performed while execution the scenarios, and was filled by each inspector independently during application inspection. The

Table 1 Inspected RS-empowered information intensive applications

Classification (Domain)	Name	Recommended Items
Video sharing / Multimedia Data Base	Movielens, Netflix, What to Rent?, Youtube, Daily Motion, IMDB	Video
News	Boston Herald, Google News, IGN Italia, The Intercept, Tom's Hardware, Lexology, Reddit, 9GAG, Digg	Articles
Music streaming	Spotify, Deezer, Rdio, Last.fm	Music
Photo sharing	Flickr, DeviantArt, Imgur, Photobucket	Photos
Online bookstore / book digital library	Goodreads, aNobii, Amazon, What Should I Read Next?, Scribd	Book
Online dating	Meetic, Badoo, PerfectMatch	User Profiles
Social bookmarking	StumbleUpon, Pinterest, WeHeartIt	Online content
Social network	Facebook, LinkedIn, Twitter, MySpace, Google+, FourSquare	User Profiles, Posts, Offers, POIs
Tourism services	Booking, AirBnB, TripAdvisor, Holiday, Watchdog, Gogobot, Volagratis, Trivago, Yelp	POIs
Miscellaneous	Groupon, Tastekid, Google, Softsonic, SourceForg	Offers, Online content, Software

Table 2 Inspection Scenarios by Domain

Domain	Usage Scenario
Video sharing / Multimedia Data Base	A person is a member of a website that hosts a database of movies. He/she queries the database for a movie title in order to find the plot and rating of the movie. When typing few letters in the search box, a number of movie titles appear in a drop-down menu, allowing him/her to select one of them. He/she selects one of the movies, and visits the movie page, where he/she finds the full description of the plot, and the detailed ratings given by other members. In addition, he/she is provided with a list of recommended movies that are similar to the selected movie.
Music streaming	A (young) person is interested in creating and listening to his/her own personal music playlist. He/she registers to a music streaming website and builds a profile and adds his/her favorite songs and artists to the profile. The system analyzes the user profile, and generates automatically a personalized playlist, which contains a sequence of the songs predicted to be interesting to him/her.
News	A user of a news website finds a news article very attractive. After reading the article, he/she gets interested in reading more about the topic of the article. On the article page, the user notices a button (or a section) "*Who reads this article, also reads*", which shows a list of similar news articles. The user clicks on the first one and reads the associated article.
Photo Sharing	A young girl wants to share some of her photos with the other people. She registers to a photo sharing website, creates an album, and uploads her photos. She also browses a set of photos recommended by the website, and notices that a number of them are quite interesting. She selects some of the recommended photos and reads the detail description of them.
Online Bookstore/Book Digital Library	A young girl is interested in finding new books to read and to share with friends. She registers to an online bookstore and provides some information about her taste, by rating some familiar books, and adding her favorite genres. Later, the system generates a set of recommended books, displayed either in a separate page (i.e., recommendation page), or in the same pages of the viewed books.
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, his interests, and his photo. Based on the created profile, a set of user profiles, of the people with similar interests who live closely, is recommended to him.
Social Bookmarking	A girl is looking for pictures on a certain topic, and finds a website that hosts such pictures. She is asked to register and enter her topic preferences. The website builds her profile and presents her a set of pictures. She rates and tags the ones she likes more.
Social Network	A mid-aged woman wants to find her childhood friend, whom she has not met for several years. She is advised to check a social network, with hundred millions of members. She registers to the system by entering some basic information. She also adds some of her current and old friends into her friend list. Then she browses recommended friends and finally finds her childhood friend.
Social News	A person who often reads an online social newspaper finds one of the popular articles interesting to him. He/she wants to add a comment to the article. He/she proceeds with the "fast registration" to the website, and adds his/her comment. Later, he/she finds other news articles recommended by the system. He/she also makes a new post, by inserting a link to his/her weblog where he/she expresses his own opinion about the topic.
Tourism Services	A young couple are interested in spending their holiday in London. They register to an online hotel reservation service. They enter the requested information about their trip and accommodation preferences, and then receive a list of recommendations, including name, address, pictures, and reviews of the suggested hotels. They choose the cheapest option and proceed with the reservation.

form was structured as a 2D matrix listing the 54 applications (with their name and url) and a set of patterns from the Pattern Library described in [34]. This Pattern Library addresses user interface design in a broad number of domains, and includes over 100 UI patterns. We selected 63 of them, omitting the ones that apparently were not relevant for content intensive multimedia applications. Each pattern name in the matrix was linked to the corresponding online description in the Library.

For each application, each inspector executed the scenario defined for the application cluster and filled the online form as follows:

- When (s)he discovered the use of an existing pattern in the Library, (s)he mapped it to the application under inspection by marking the corresponding cell in the matrix
- When (s)he discovered a good interface design solution that did *not* correspond to any of the existing patterns in the Library:

  (s)he created a new item in the pattern list of the matrix, marking it as a "*would-be-pattern*"
  (s)he mapped this new item to the application under inspection
  (s)he created an online document that mentioned: i) the task the inspector was performing when the solution was noticed; ii) a preliminary description of the would-be pattern in terms of the design problem addressed, the discovered solution, and the links to the page(s) where such solution was found [1]
  (s)he linked the would-be-pattern to the document

- When (s)he discovered the use of an interface design solution that corresponded to a would-be-pattern, (s)he mapped it to the application under inspection by marking the corresponding cell in the matrix; in addition, (s)he updated the document describing the would-be pattern with a note about the task the inspector was performing when the solution was noticed and, in the example section, with links to the page(s) where the pattern solution was found.

At the end of individual inspections, all results were merged, organized by application, and finally compared and discussed within the group.

Concerning the occurrences of existing patterns, we initially measured an average 96 % agreement between the inspectors. The *percentage of agreement* was measured for each pattern as follows:

number of applications in which occurrences of the pattern were discovered/total number of applications

The average was calculated on the percentages of agreement of the set of existing or would be patterns considered in the study. A second round of individual inspections was performed on applications where inspectors had disagreements, focusing on the patterns where they had discrepant results, eventually reaching a 100 % of agreement.

The descriptions of "would-be" patterns were organized in clusters, each one grouping the ones addressing the same problem in spite of the potentially different formulations of

[1] In the pattern language literature, a description of this kind is sometimes referred to as "proto-pattern" [37]-something that is documented in a pattern-like form, but lacks of a refined formulation and enough supporting known uses.

the different inspectors. The clusters and their contents were submitted to a pruning, filtering, and reformulation process. According to the rule of thumb in pattern mining that claims that a pattern must be used at least 3 times before it can really be called a "pattern", a cluster that was not supported by at least 3 examples in 3 different applications was removed (pruning). In each remaining cluster, the solutions were discussed in terms of usability and design utility (i.e. effectiveness to solve the corresponding problem). Descriptions comprising solutions that were not considered fully satisfactory by all inspectors were removed (filtering). At this point, in each remaining filtered cluster, a revised formulation of the problem and the solution was collaboratively created and agreed among all inspectors. Finally, these two components were integrated with the definition of the "Usage" ("Context"), the "Rationale", and "Examples". At this stage, all examples discovered by all inspectors were included (they were filtered later, during the public scrutiny phase, as discussed in the next section). This process led to transform the would-be-patterns into a set of *30 new* patterns ready for public scrutiny, which were organized in 2 groups:

– **G1**: 20 new patterns addressing UI design requirements induced by the presence of recommendations.
– **G2**:10 new UI patterns that can be applied to a wide spectrum of content-intensive applications regardless of the presence of recommendations.

3.4 Public scrutiny

The 30 new patterns resulting from the inspection activity were subject to the review of experts in recommendation technology, HCI and UI design.

The first group of 20 patterns (G1, addressing UI design requirements related to recommendation features) was discussed and evaluated during an intense full-day workshop (part of a 3 days meeting of an EC funded project) that involved 9 researchers and developers in the recommender systems field (4 from academia and 5 from industry).

The organization of the workshop was inspired by the "Writers' Workshops" that are used by the pattern community to review, evaluate, and improve pattern descriptions and take place for example during PLOP (Pattern Languages Of Programs) international conferences. The general structure of a Writers Workshop has a group of "discussants" or "reviewers" read the patterns descriptions carefully before the session. During the workshop the discussants examine the strengths and weaknesses of each pattern, accentuating positive aspects and suggesting improvements in content and style.

In our case, approximately 4 weeks before the workshop date the experts received a report describing the 20 patterns. Experts were asked not only to read the document but also to score each pattern according to two parameters: the relevance of the design problem addressed and the quality of the proposed solution. Each measure ranged from 1 (lowest value) to 10 (highest value). From these scores we calculated a ranking among patterns, based on the values of [relevance + solution quality] averaged among all experts. The ranking allowed us to select the top-9 patterns that could be reasonably discussed during the time frame of the workshop, and to define the discussion schedule (from higher to lower ranked patterns). Finally, each pattern was assigned to an expert ("lead reviewer") to lead the pattern discussion during the evaluation sessions.

During the workshop, each patterns was discussed for approximately 1 h, as it normally happens in a Writers Workshop. At the beginning of each session, the authors briefly summarized the pattern, and then remained silent for most of the remaining time, taking notes about the ongoing discussion. The lead reviewer pinpointed what (s)he felt particularly important about the pattern from his/her personal viewpoint and what (s)he thought were the key points of the pattern. Coordinated by the lead reviewer, the group then discussed what they liked or disliked about the pattern, in terms of content and formulation style, to identify and praise the strengths of the pattern and to highlight its potential weaknesses if any. Quality was discussed along some main criteria: problem framing (how well the problem is defined and framed in its context of use); occurrences of pattern solution (the more the better); and number of examples that witness its use; effectiveness of the solution itself (not necessarily a popular solution is the best possible one): expressiveness of the reported examples.

After discussing the positive and negative aspects of the pattern, the group discussed how to improve its content and style, to give the authors constructive suggestions on how to make the pattern formulation more clear and its content more useful. The reviewers first focused on the problem, rationale and usage, then on the solution, and finally on the examples also suggesting which ones had to be included in the pattern formulation. At this stage, the authors re-entered in the loop and asked questions to the reviewers to clarify their statements, in order to better understand certain comments.

After the workshop, the authors performed the final reformulation of the 9 patterns evaluated during the workshop and sent the final version to the experts for final approval.

A similar procedure was followed for the 10 “generic” patterns (G2). Here we involved a team of 4 UI experts from the Design Department of our University, who were called for a half-day workshop at our lab. The outcome of the workshop led to the selection and reformulation of 5 new “generic” patterns.

The final versions of all patterns evaluated during the above discussed process are reported in sections 4.2 and 4.3.

4 Results

In this section, we present and discuss the main results of the inspection, data mining and validation process, presenting the analysis of the occurrence of *existing UI design patterns* in RS-empowered content-intensive web applications and the definition of 14 *new UI design patterns*.

4.1 Occurrence of existing UI design patterns

A total of **56** out of 63 patterns from the Pattern Library [34] occur at least once in the 54 applications analyzed. Figure 3 lists the 15 most frequently used patterns from the Pattern Library, and their *frequency*. Frequency is measured in terms of the number of applications using a particular pattern over the total number of inspected applications. For instance, the *Home link* pattern is used in 70 % of the inspected applications. The patterns not shown in the figure appear in less than 10 % of the applications, and are not considered as relevant to our study.

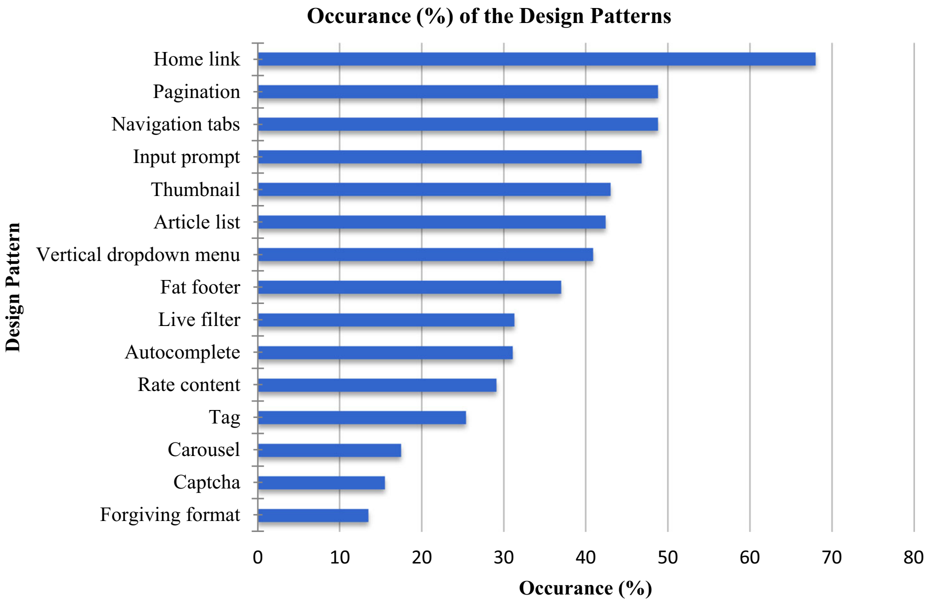

Fig. 3 Top 15 most applied UI design patterns in the inspected applications

Figure 3 pinpoints that the most used UI design pattern is *Home link*. The *Home link* design pattern addresses the user's need to come back to the home page of a website. This is a basic and important design pattern in any online application, and, as expected, it is adopted in any RS-empowered content-intensive applications.

The other patterns address some key elements that have to be taken into account when designing the UI of any multimedia RS: collecting preference data from users to generate recommendations, and, displaying to them the generated recommendations. *Pagination* and *Navigation tabs* design patterns address the user's need to view a list of items, which cannot be displayed in a single page, and has to be separated into pages (or sections). *Input prompt* design pattern addresses the need of the user to provide data to the system, as a recommender system can be viewed as a personalized search engine able to filter relevant items and rank them based on both explicit and implicit user needs and opinions.

4.2 New recommendation-specific UI patterns

The 9 UI new patterns that concern recommendation-specific design issues are summarized in Table 3 and described in the rest of this section.

- *Design Pattern 1*: *Similar Content* -

Problem: Users would like to explore the multimedia items that are similar to what they view.

Usage:

Use when:

- users are interested to view the multimedia content that is similar to what they view;
- the system recommends the multimedia content that the user is likely familiar with;

Table 3 List of the new RS-specific design patterns

Id	Pattern Name	Problem
1	Similar Content	The user would like to explore the multimedia items that are similar to what she views (e.g., directed by the same director, or with a similar plot).
2	Explanation of Recommendations	The user might be interested to get brief explanation about the recommendation process (e.g., which of the previously rated items have influenced the recommendations).
3	Rating Elicitation	The system needs to obtain the preferences of users in terms of ratings in order to generate relevant recommendations.
4	Recommendations from Different Categories	The user might be interested to get recommendations from different categories of multimedia items, in order to improve diversity of recommendations.
5	"Consumed Together" Recommendation	The user might be interested in consuming items grouped altogether (e.g., a series of movies).
6	Rating Group of Items	The system may want to elicit the rating of the user for a group of items.
7	Recommendation Criteria	The user might be interested to fine-tune the recommendation criteria to control what is recommended (e.g., blockbuster movies vs. niche movies).
8	Editors Picks	The user might be interested to view multimedia contents selected by experts.
9	Choice of Personalization	The user might be interested in switching between the choice of personalized recommendations and non-personalized recommendations.

- multimedia items are attributed with graphical information (e.g., geo-tagged images) and are recommend based on the vicinity to the user.

Solution: Multimedia 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 to *Gestalt psychology*, i.e., a theory of mind, which studies our ability to acquire and maintain meaningful perceptions in an apparently chaotic world. In particular, a major aspect of Gestalt psychology 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, i.e., *Law of Proximity*: 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 multimedia content in a position close to the position of the currently displayed one, with the aim of forming a group. Hence, the user likely perceives the elements as parts of a group, and, considers them similar.

Examples: Youtube, Amazon, Facebook, DeviantArt (Fig. 4)

- *Design Pattern 2*: *Explanation of Recommendation* -

Problem: The user might be interested to get a brief explanation about the recommendation process, i.e., why the recommended items have been proposed.

Usage:

Use when it is important to enforce the transparency of the recommendation process and improve the trust of the user into the recommender system.

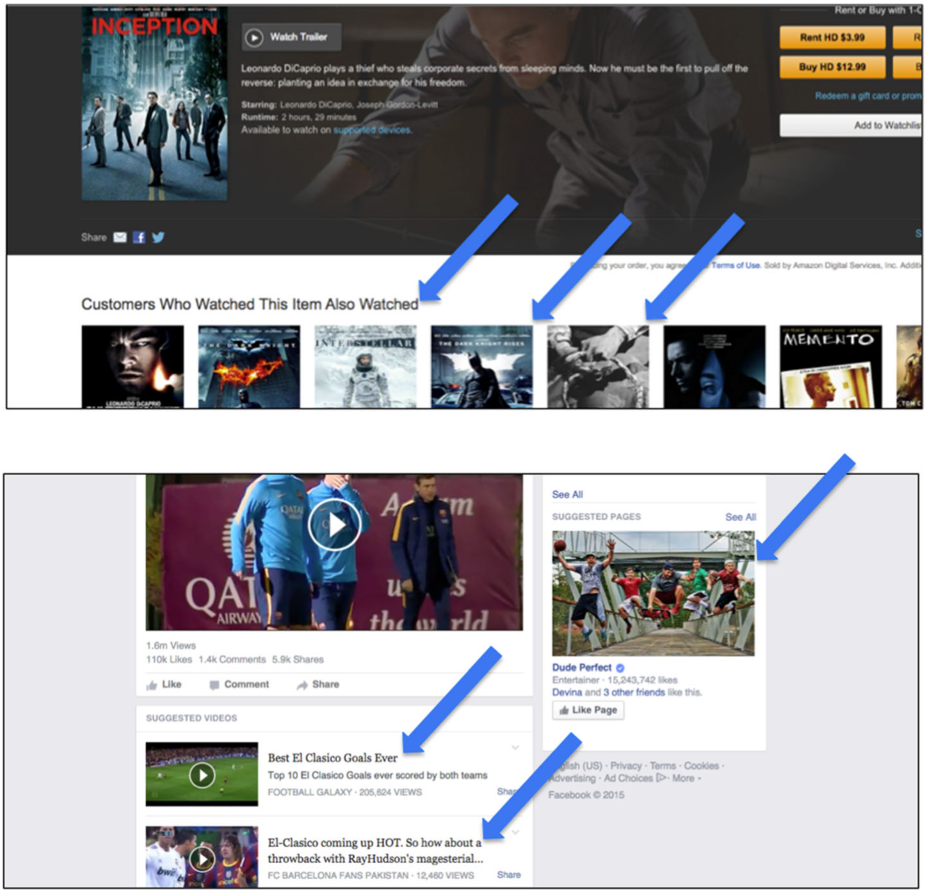

Fig. 4 Two examples of design pattern 1 *Similar Content*: (top) Amazon, (bottom) Facebook

Solution: The system displays a short explanation of the recommendation process, as an attempt to better serve the user and highlight why the proposed items may fit the user needs.

Rationale: when displaying a recommendation list, it is helpful to explain to the user why the items have been selected (and how) to improve user's trust and user's perception that the proposed items match with his/her interests.

Examples: Youtube, Amazon, Movielens, Spotify, Goodreads, Last.fm (Fig. 5)

- *Design Pattern 3*: *Rating Elicitation* -

Problem: The system needs to elicit user preferences explicitly, in terms of item ratings, in order to generate relevant recommendations.

Usage:

Use when:

- the system makes recommendations based on the user ratings [26];
- the user is new and the system does not have enough information to build the user profile and generate personalized recommendations;
- a multimedia item is new and no user has already rated that item, which may result in the system to be unable to accurately recommend that item to any users.

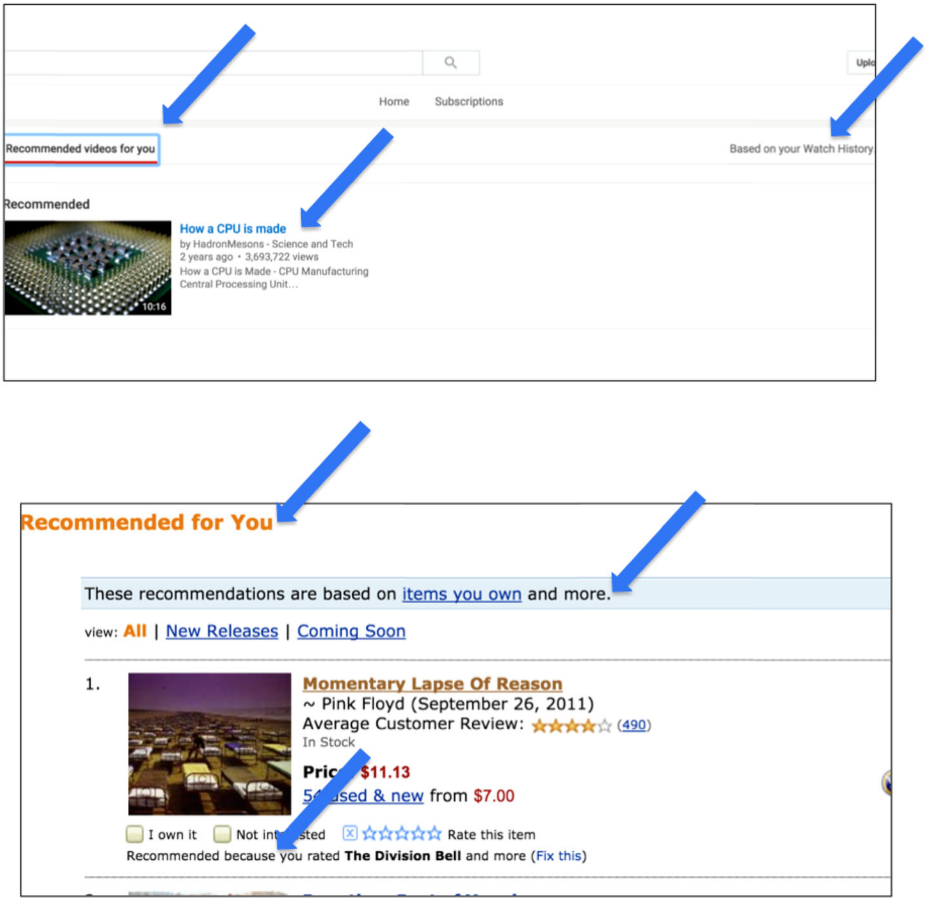

Fig. 5 Two examples of design pattern 2: *Explanation of Recommendations*: (top) Youtube, (bottom) Amazon

Solution: The system invites the user to rate a set of multimedia items that are selected as the most informative to reveal the preferences of the user to the system.

Rationale: RSs typically fail to generate relevant recommendations to users if the available information on users or items is not sufficient. In such a case, the system has to obtain a minimum amount of information by explicitly requesting the users to rate a set of items. This typically happens when a new user registers to the system and the system has no or very limited information about this user (new user problem [70]), or when a new item is added to the catalogue and the system has no opinions on that item (new item problem [39]).

Examples: Movielens, Netflix (Fig. 6)

- *Design Pattern 4*: *Recommendations from Different Categories* -

Problem: The user might be interested to get recommendations on different categories of items.

Usage:

Use to improve the diversity of recommendations and to make the recommendations more “serendipitous” (i.e., unexpected).

Solution: The system displays recommendations from different categories of multimedia items.

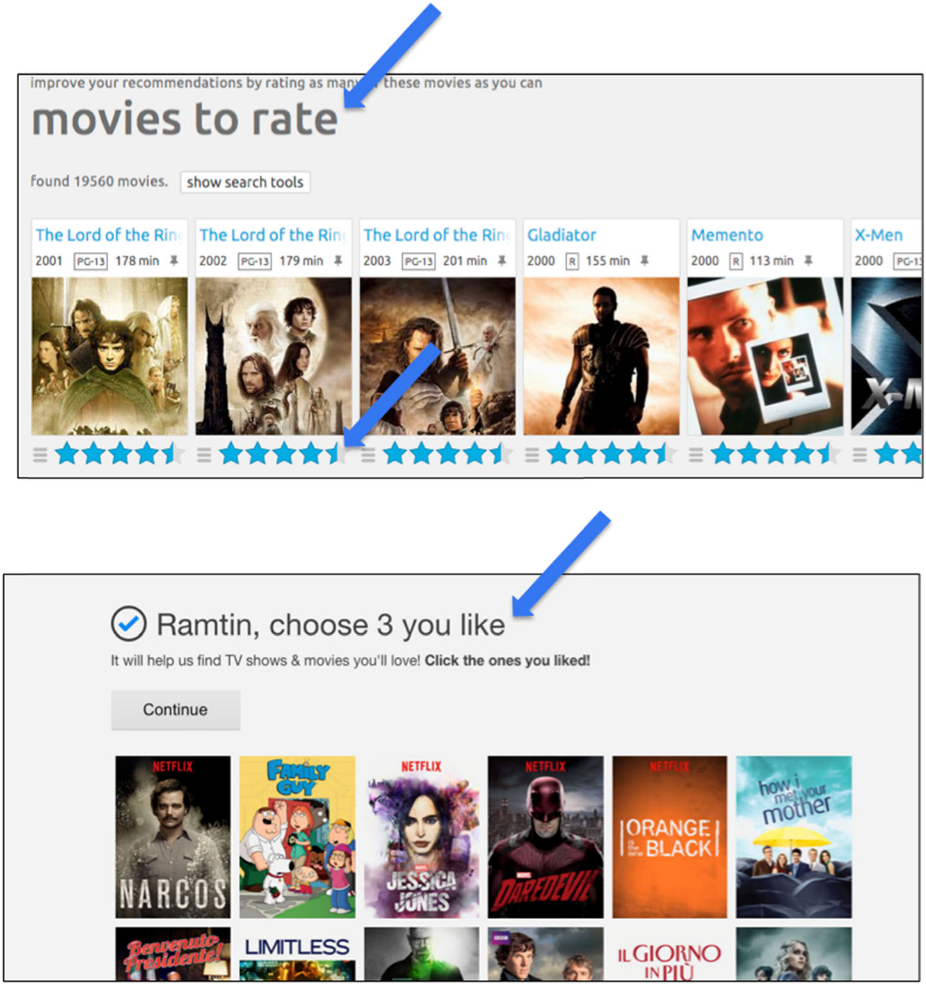

Fig. 6 Two examples of design pattern 3: *Rating Elicitation*: (top) Movielens, (bottom) Netflix

Rationale: In a recommender system, it is more likely that the user will find a suitable item if there is a certain degree of diversity among the recommended items. There is often no value in having recommendations for a restricted type of items, especially in the early stage of a decision making process, when the users want to explore new and diverse directions (e.g., if a user likes sci-fi movies, there is no value in recommending only sci-fi movies, as the user would be able to search for them without the need for recommendations; there is more value in recommending also dramatic movies that, unexpectedly, fit the taste of sci-fi lovers).

Examples: Goodreads, IMDB, Google, Youtube, DailyMotion, Rdio (Fig. 7)

- *Design Pattern 5*: "*Consumed Together*" *Recommendations* -

Problem: The user might be not so interested on a single specific item but rather in a set of items as a whole. For instance, a user might be interested in a playlist of songs together rather than a single song.

Usage:

Use when it is expected that some items are "consumed" together as a group rather than individually.

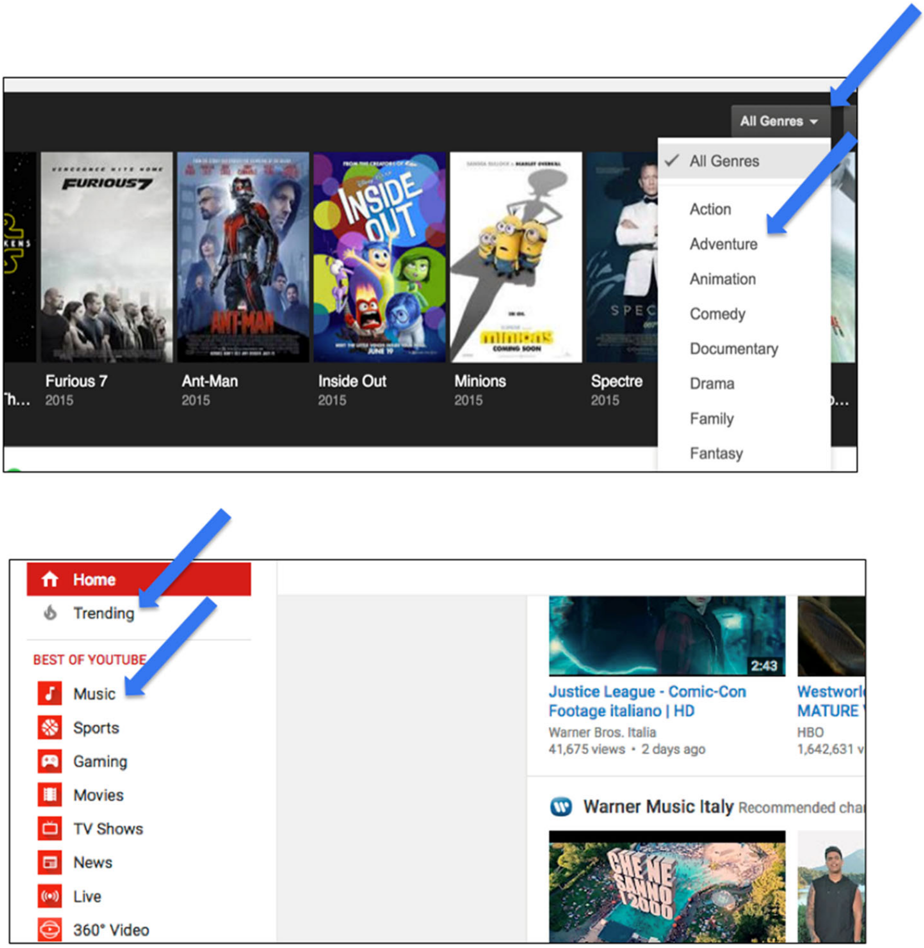

Fig. 7 Two examples of design pattern 4: *Recommendations from Different Categories*: (top) Google, (bottom) Youtube

Solution: The system groups a set of recommended items together and displays them as a consistent collection.

Rationale: The users are not always interested in consuming items individually, but they might be interested in consuming them as a group or package. In such a case, the user is not looking for a specific product, but for a set of products.

Examples: Goodreads, aNobii, Scribd, IMDb, Amazon, Youtube, Dailymotion, Spotify (Fig. 8)

- *Design Pattern 6*: *Rating Group of Items* -

Problem: The system may want to elicit the rating of the user for a group of items.

Usage:

Use to allow the user to rate a group of items rather than individual items, so that the user effort in the rating process is lower, or when recommended items can be organized as package of individual elements (see pattern 5).

Solution: The system allows the user to rate a set of items that are grouped together using a single action (rather than multiple actions, one for each element). For instance, a user might rate an album of songs rather than the single tracks.

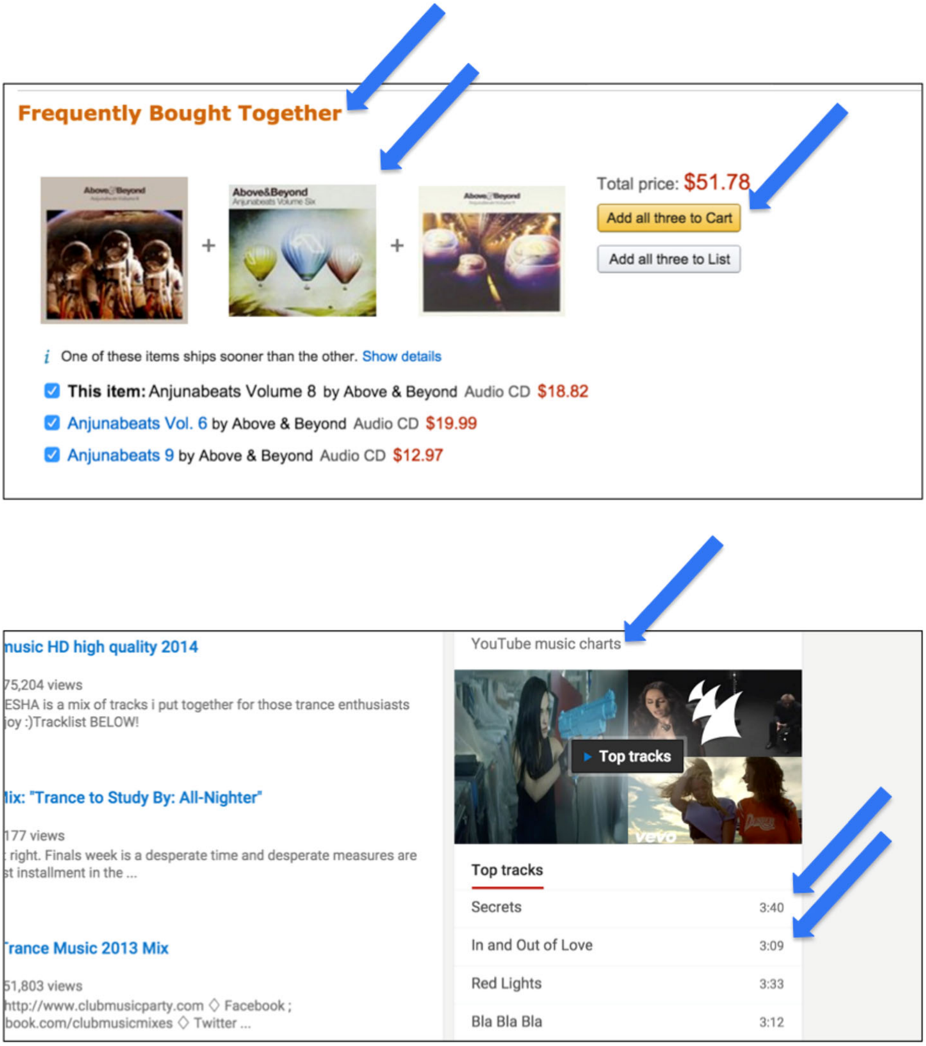

Fig. 8 Two examples of design pattern 5: "*Consumed Together*" *Recommendations*: (top) Amazon, (bottom) Youtube

Rationale: items that can be grouped as a package and consumed together, can be also rated as a group rather than as individual elements. Hence, the system can offer such a possibility to the users.

Examples: Youtube, Amazon, Groupon, Netflix (Fig. 9)

- *Design Pattern 7*: *Recommendation Criteria* -

Problem: Users might be interested to fine-tune the recommendation criteria to control what is recommended to them or refine the recommendation results.

Usage:

Use to:

- empower the user with some control of the recommendation process;

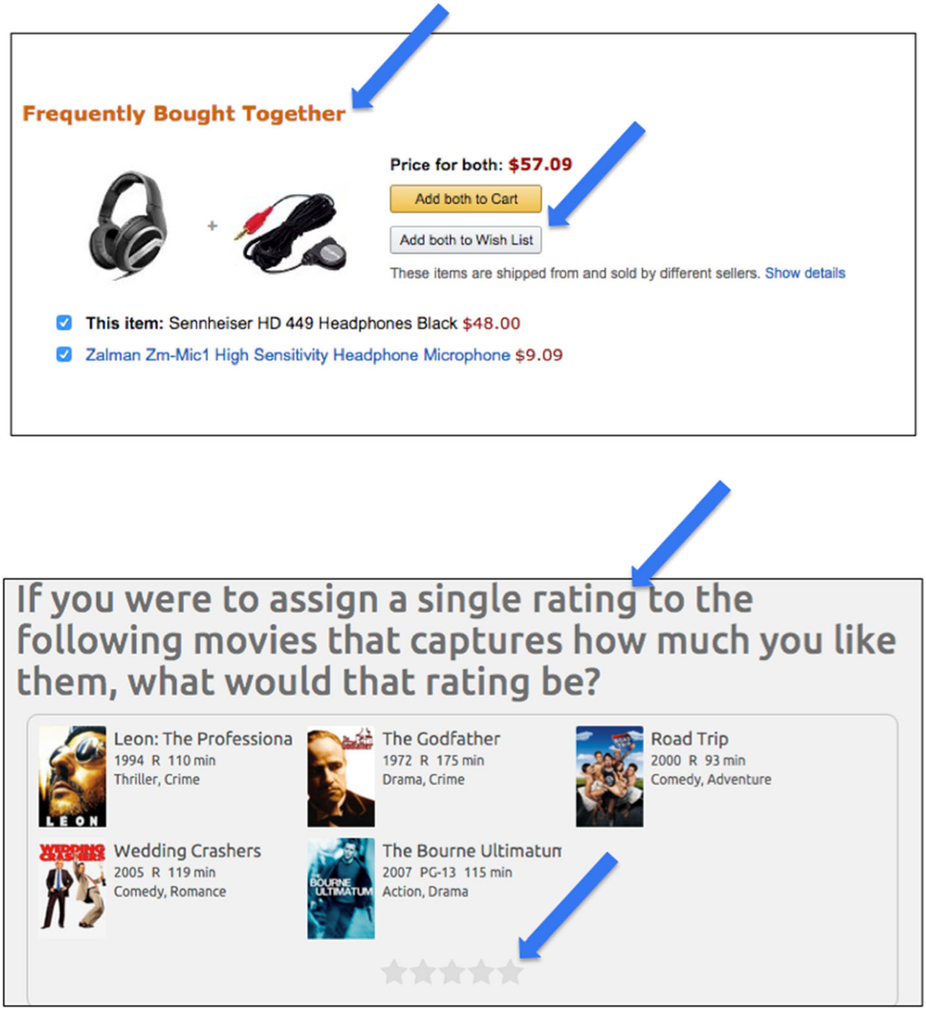

Fig. 9 Two examples of design pattern 6: *Rating a group of items*: (top) Amazon, (bottom) Movielens

- improve the transparency of the system and increase the user trust;
- learn the preferences of user implicitly (analyzing the customization criteria adopted by the user).

Solution: The system allows users to set their recommendation criteria and to filter out what they might be not interested.

Rationale: The system makes some assumptions about the users' preferences on item features, giving control over the recommendation criteria, and helping users to tune the recommendations according to their preferences (based on users' need and mood).

Examples: Youtube, Amazon, Groupon, Netflix (Fig. 10)

- *Design Pattern 8*: *Editors Picks* -

Problem: The user might be interested in viewing multimedia contents selected by trusted experts.

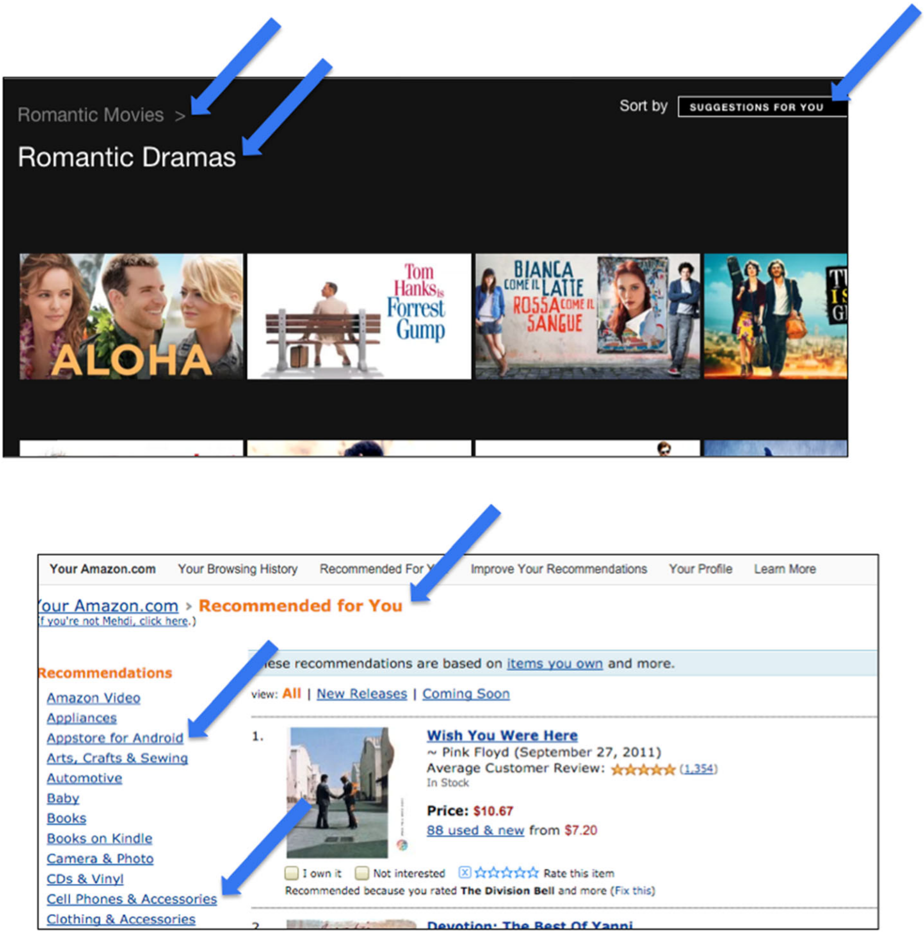

Fig. 10 Two examples of design pattern 7: *Recommendation Criteria*: (top) Netflix, (bottom) Amazon

Usage:

Use when the system wants to promote certain multimedia items, and the expert reviews or ratings are available.

Solution: The system displays a recommendation together with a positive review from an expert (e.g. a famous person).

Rationale: Providing reviews from experts motivates the consumers to consume the recommended item. Indeed, the opinion of experts improves the trust of the users into the recommendation.

Examples: Goodreads, Groupon, Sourceforge, Google, Google News, Deezer (Fig. 11)

- *Design Pattern 9*: *Choice of Personalization* -

Problem: The user might be interested in switching between different forms of recommendations, e.g., personalized and non-personalized recommendations.

Usage:

Use when:

- the system supports different generation techniques, for personalized and non-personalized recommendations;

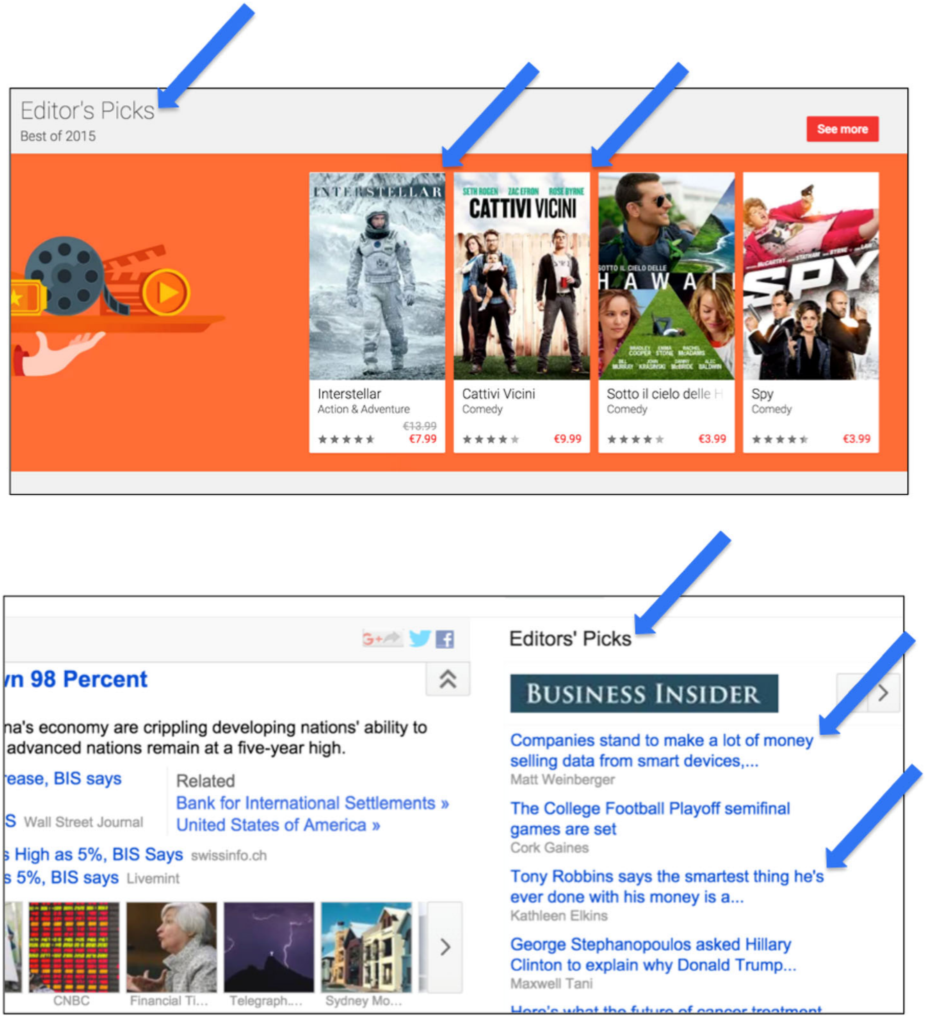

Fig. 11 Two examples of design pattern 8: *Editors Picks*: (top) Google, (bottom) Google news

- users need to browse and explore the item catalogues “as it is”, without any filter and regardless of their personal profiles;
- users are new and there is no information about their profiles, so that personalized recommendations cannot be offered, but could become available after some interaction steps (new user problem [70]).

Solution: The system offers different versions of the list of recommended items and allows the user to choose among them.

Rationale: Users are not necessarily always interested in receiving personalized recommendations. Indeed, they are sometimes interested in browsing and exploring items that are not recommended. For example, they may be interested in viewing items that are popular and mostly seen by the user community. Moreover, there are situations in which the system is not able to generate recommendations and hence can present only non-personalized recommendations.

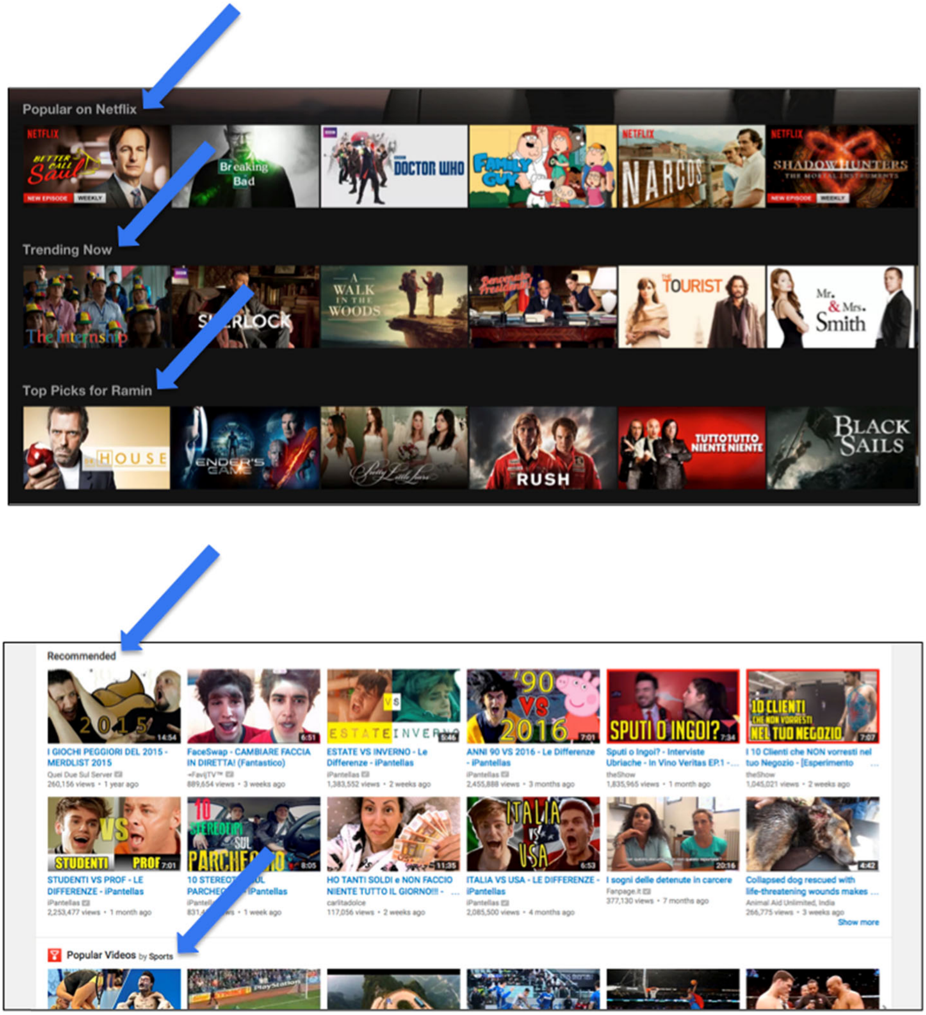

Fig. 12 Two examples of design pattern 9: *Choice of Personalization*: (top) Netflix, (bottom) Youtube

Examples: Goodreads, aNobii, Softonic, Lexology, Youtube, Netflix, Google News, Spotify, Deezer, Rdio (Fig. 12).

4.3 New UI patterns

In this section, we discuss the *new* UI design patterns that meet the UI design requirements of Recommendation-empowered applications but have a wider utility and can be used in a wide spectrum of content-intensive applications, regardless the presence of recommendations. The complete list of these news design patterns is presented in Table 4.

- *Design Pattern 10*: *Social Connection* -

Problem: A user would like to find and connect to other users (e.g., friends, colleagues). The multimedia RS analyzes connections that are already established among the entire network of users, and recommend to that user whom to connect with.

Table 4 List of the new design patterns

Id	Pattern Name	Problem
10	Social Connection	A user would like to connect to other users whom she may know. The multimedia RS analyzes connections that are already established among entire network of users, and recommend to that user whom to connect with.
11	Add Comments	The users would like to provide their opinion on multimedia items.
12	Social Login	RSs require some information about the users' tastes and preferences, before serving them with relevant recommendations. Such information can be elicited implicitly from users' existing online profiles, such as the profiles in social networks.
13	Profile as Business Card	The system needs to display a link or a short summary of user profile.
14	Binary Rating	The system needs to elicit the rating in the simplest form.

Usage:

Use when:

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

Solution: Provide a button or a link to add a person as friend or "trusted" person. Once the person has been added as a friend, clearly indicate to the user that this person is now a friend. The recommender system can later analyze common interests in order to provide multimedia recommendations.

Rationale: People are interested in sharing their experiences with their friends, whom likely have similar tastes or interests, and allowing them to connect and strengthen these ties. This also provides a better estimation on what could be the users' interests and 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 multimedia recommender system and makes the system viral.

Examples: MySpace, Flickr, Google Plus, Facebook, LinkedIn (Fig. 13)

- *Design Pattern 11*: *Add Comments* -

Problem: The users would like to provide their written opinions (i.e., reviews) on multimedia items.

Usage:

Use:

- when the recommender system generates recommendations based on reviews of users on multimedia items;
- in recommender systems that allow the users to write opinions on multimedia items.

Solution: Recommender systems can generate recommendations based on reviews that users provide on items. For example, if a user is interested in controversial movies, the system may analyze the reviews of movies and recommend those movies that are identified as controversial.

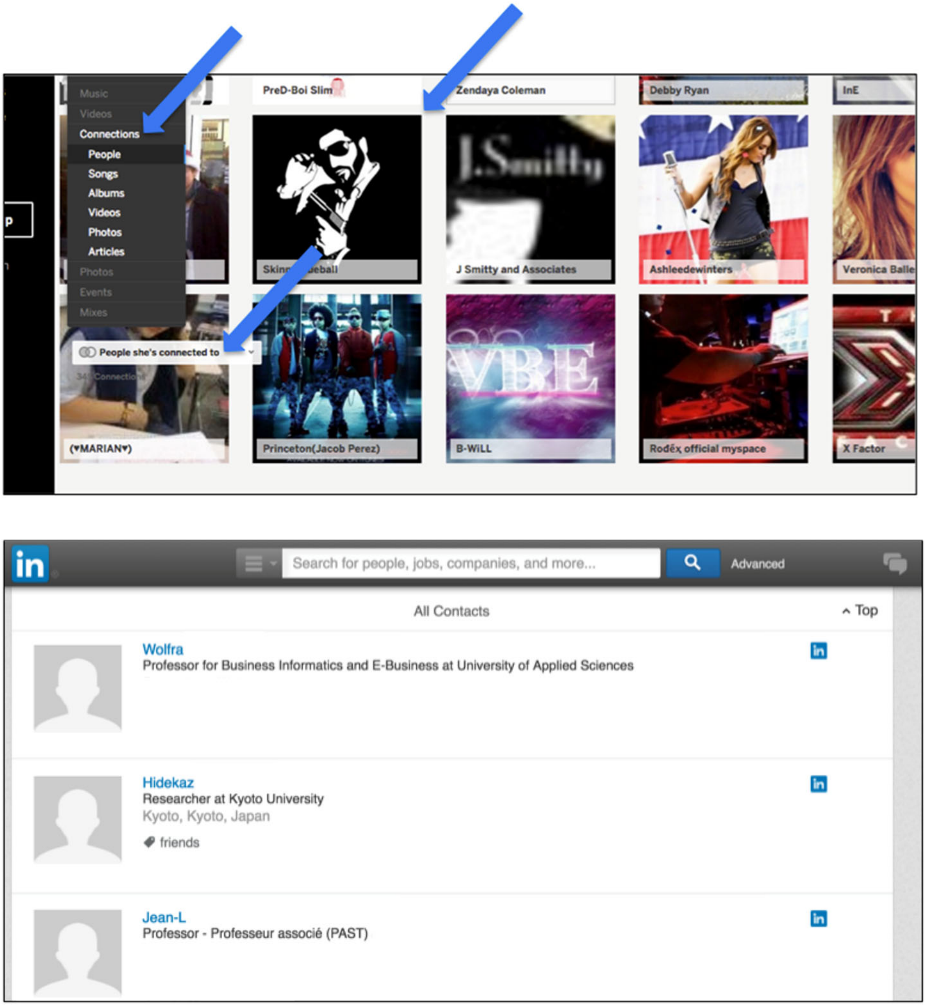

Fig. 13 Two examples of design pattern 10: *Social Connection*: (top) MySpace, (bottom) LinkedIn. For privacy reasons, persons images are removed

Rationale: Recommender systems using this design pattern assume that people are more prone to give opinions using natural language, instead of using other interaction tools, such as like/dislike or ratings. The multimedia items with a large number of comments are typically interesting for the community of users.

Examples: Last.fm, Youtube, DeviantArt, Facebook, Twitter (Fig. 14)

- *Design Pattern 12*: *Social Login* -

Problem: RSs require some information about the users' tastes and preferences, before serving them with relevant recommendations. Such information can be elicited implicitly from users' existing social profiles.

Usage:

Use when:

- the user is required to create a profile before being able to use the RS;

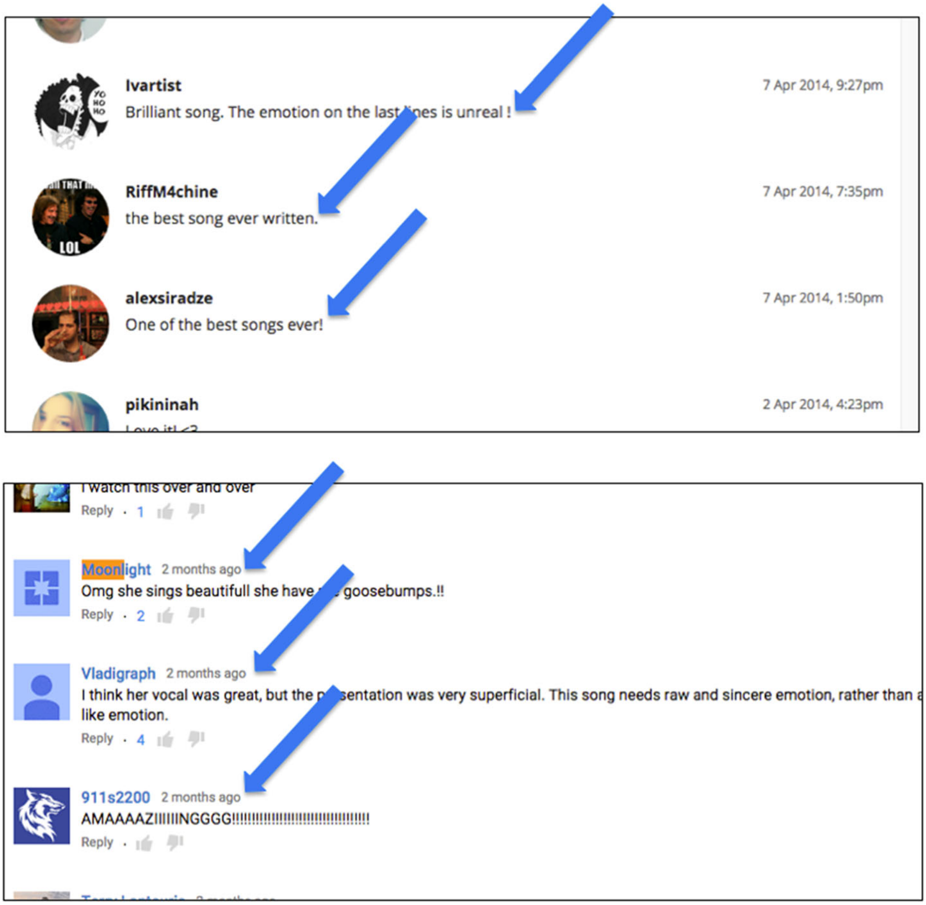

Fig. 14 Two examples of design pattern 11: *Added Comments*: (top) Last.fm, (bottom) Youtube

- RS requires collecting information about the user (e.g., demographics, social interactions, friend list, colleagues list, followers, etc.) in order to generate relevant recommendations;
- RS provides easy and quick registration method for the new users.

Solution: To properly adopt this design pattern, the RS implements a particular registration method. This method allows the user to login to the system by simply entering the login information of other online services (mainly social networks). This allows the system to import the user profile from those online services. Such online services 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, placed in the homepage or registration form, which allows the user to establish a secure connection with a third party system and login with their credentials.

Rationale: Some users may prefer to use their existing online profiles to speed-up the registration process. Some may even want to allow the RS to improve the

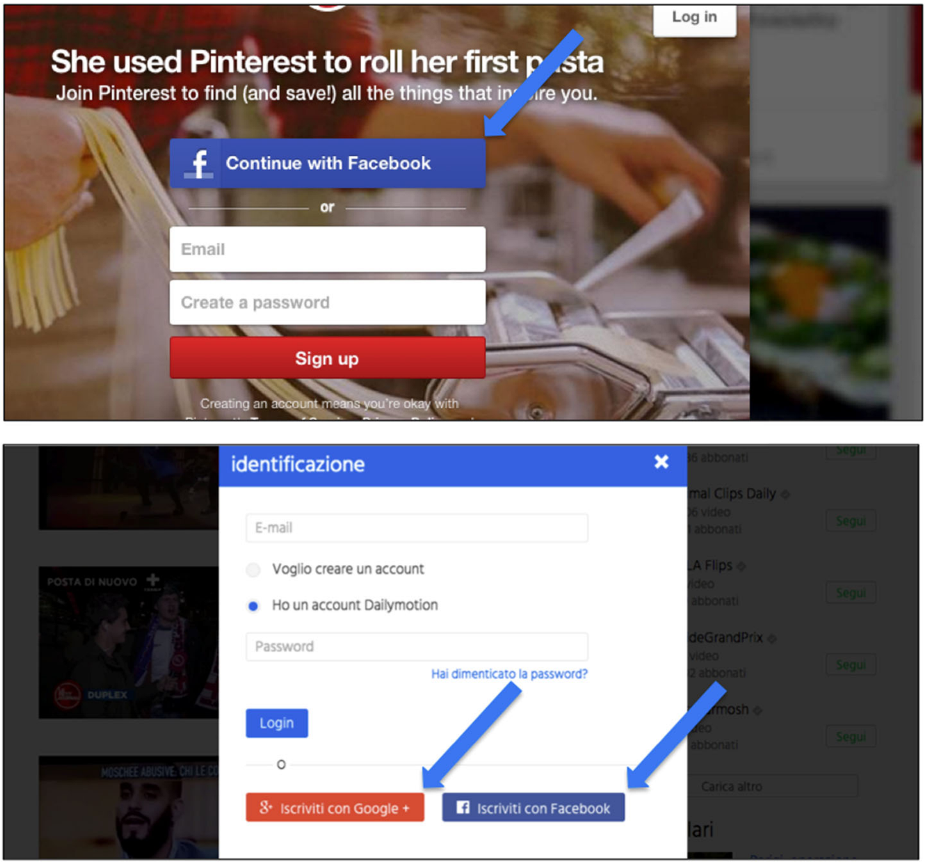

Fig. 15 Two examples of design pattern 12: *Social Login*: (top) Pinterest, (bottom) Dailymotion

recommendation quality, by letting the system to extensively mine their social interactions.

Other users may be concerned with the security level of the recommender 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: Pinterest, WeHeartIt, Facebook, TripAdvisor, Dailymotion (Fig. 15)

- *Design Pattern 13*: *Profile as Business Card* -

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

Usage:

Use when:

- implementing social network functionality;
- helping the user to recognize other users.

Solution: A user profile is represented as a classical Business Card, and as such it contains name, address, picture, and other additional information on the user in a compact format, which helps people to identify that user.

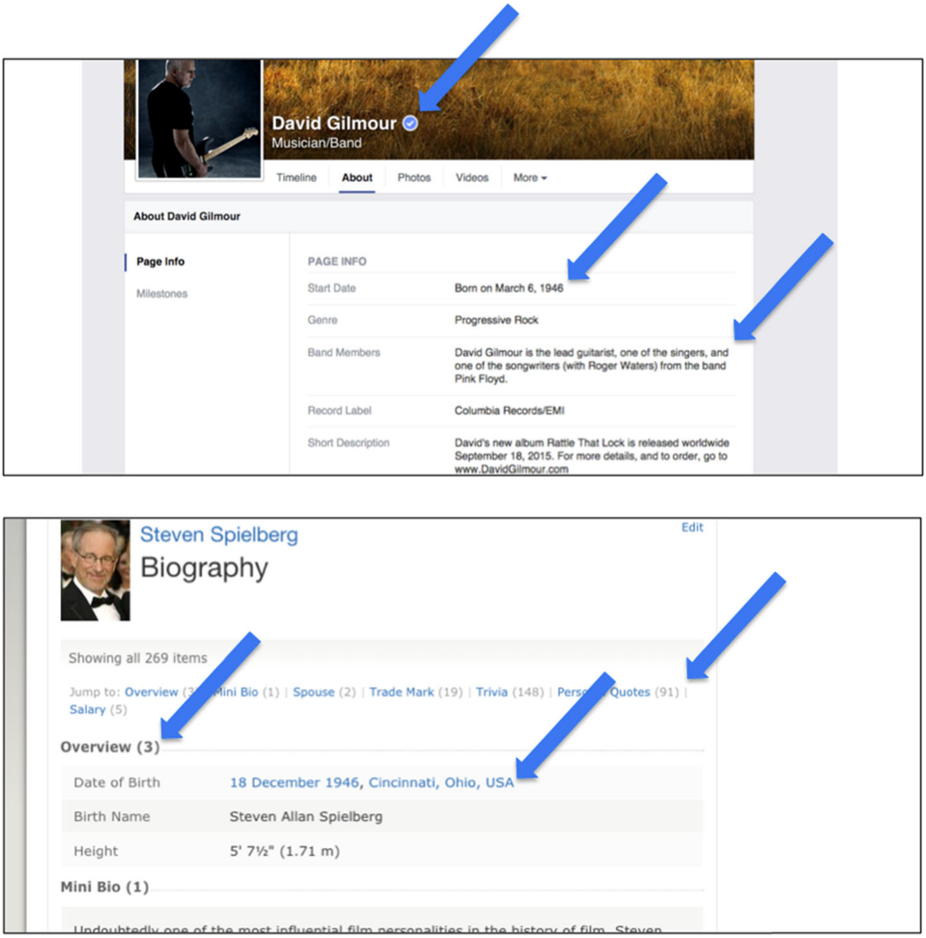

Fig. 16 Two examples of design pattern 13: *Profile as Business card*: (top) Facebook, (bottom) IMDB

Rationale: This pattern resembles the paper-based classical business card. Users may enjoy finding elements of UI that they associate to their previous experiences. This make the proposed pattern a straightforward and intuitive element in the UI designs of the user profiles.

Examples: Facebook, IMDB, Google+ (Fig. 16)

- *Design Pattern 14*: *Binary Rating* -

Problem: The system needs to elicit the ratings in the simplest form.

Usage:

Use when it is important to obtain the rating of the users in a binary scale, i.e., when the rating is given in the form of either positive or negative feedback (e.g., Like / Dislike).

Solution: Users are given the possibility to provide their feedback on a binary scale.

Rationale: It can be confusing for users to provide their feedback in the form of Likert scale (e.g., Likert 1–5 scale). Indeed, it could be unclear to the users what is their precise feedback and hence the feedback may include unwanted biases. Instead, binary feedback poses a clear distinction between positive and negative feedback showing clearly the people's opinion on the products.

Examples: Facebook, Google+ (Fig. 17)

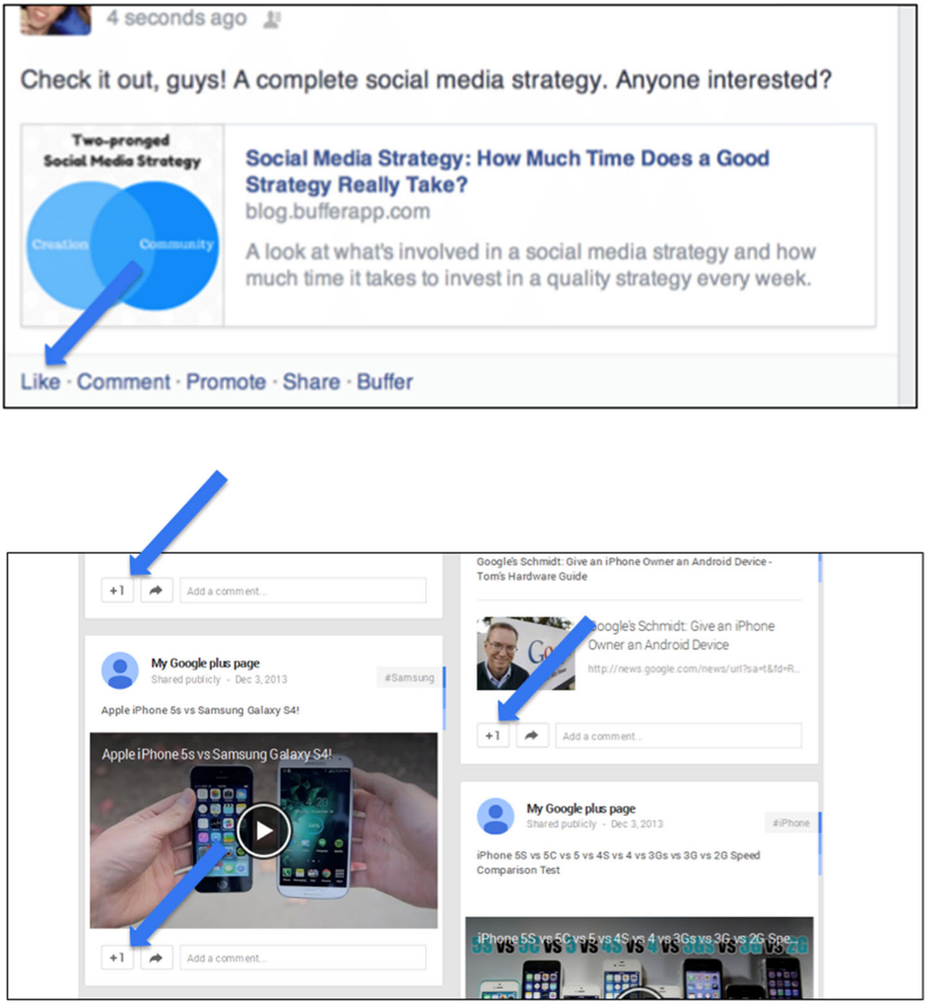

Fig. 17 Two examples of design pattern 14: *Binary Rating*: (top) Google+, (bottom) Facebook

5 Conclusion and future work

Our work looks at recommendations in content-intensive online applications through the lens of Design Patterns. We report the results of a wide systematic inspection of the UIs of 54 recommendation-empowered content-intensive applications. Our in-depth analysis has led to the identification of the frequency of use of existing UI patterns in the inspected applications and the definition of 30 new patterns; some of them are specific of recommendation interfaces while others can be useful for a broader set of UI components. These has been discussed with and evaluated by a team of experts, leading to a filtering and reformulation of 14 new patterns, which have been reported in the paper.

Our work sheds a light on the use of UI design patterns in recommendation-empowered content-intensive applications and offers some novel contributions.

No previous study has analyzed UI for recommendations from a design pattern perspective, nor has identified UI patterns for this specific component of content-intensive online applications. As such, our work provides a contribution, in terms of method and practice, to various communities: Design Patterns, Web Engineering, and Recommender Systems.

The research reported in this paper increases our understanding of the potential of design patterns in a domain (recommendation-empowered content intensive online applications) where patterns are largely unexplored, and extends the set of design patterns currently available in the existing Design Pattern libraries with a new set of UI patterns.

Some of our new patterns can be useful for designing the UI of a wide spectrum of content-intensive online application, including those that do not offer recommendation services. This can benefit the Web Engineering community in broad terms.

To highlight the potential impact of our work from the RS perspective, it is worth mentioning that the quality of user experience has been progressively acknowledged as one of the crucial aspects for a "good" recommendation service. Algorithmic features, dominating RS research in the past, will continue to play an important role; still, the mismatch between the quality of algorithms and user perceived quality, highlighted in several studies [25], suggests the need for more research on UIs for recommender systems. Our results are in line with this approach. As UI patterns propose "good" UI design solutions, the degree of adoption of UI patterns can be regarded as one of the possible metrics for assessing UI design quality in RSs; the analysis process performed in our study can be used as a method for heuristically inspecting and measuring this quality indicator.

From a practitioner's perspective, the set of existing UI patterns that we have identified as most frequently used, as well as the new UI patterns we have defined, can benefit both novice and experienced UX designers of RS interfaces, helping them to create more usable interfaces for recommendations in a more efficient way, by reusing pattern-based solutions instead of building new designs from scratch.

Finally, our work can stimulate future research bridging RSs, Web Engineering and Interface Design by means of Design Patterns, and highlight new topics that can be considered for future studies.

A first stream of research may explore the possible correlation between the UI design patterns adopted and the type of items recommended. For example, the UI design for music recommendations would be different from tourism services recommendations: in the music domain the user could simply and immediately "try" the recommended items (e.g., by listening to the suggested music) while tourism recommendations would involve other ways of experiencing the suggested items, which in turn may involve different design solutions and different patterns.

A second stream of research may investigate the correlations between UI design solutions and the recommendation algorithms used for recommendations, leading to the discovery of new algorithm-specific UI design patterns. UIs for recommendations generated using collaborative filtering techniques may have different UI requirements w.r.t. those generated by knowledge-based recommendation engines [17]. In the former case, the user interaction with the system typically begins with rating elicitation, where the system obtains a number of ratings. In the latter case, the user typically expresses her preferences in the form of a query.

A third opportunity for new research bridging Design Patterns and Recommender Systems is induced by the new interaction paradigms that are progressively become more and more popular (e.g., based on smart objects, touchless and wearable devices) and offer new ways of experiencing recommendations. This scenario opens new research directions for RSs, but increases the difficulty of UX design, calling for conceptual tools like patterns to master such complexity.

Acknowledgments This work has been partially supported by the European Institute of Technology (EIT) - grant EIT DIGITAL # 15008 – 2015.

References

1. Adomavicius G, Kwon Y (2015) Multi-criteria recommender systems, in: Recommender Systems Handbook, Springer, pp. 847–880
2. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749
3. Aggarwal CC (2016) An Introduction to Recommender Systems. In Recommender Systems 2016, pp. 1–28, Springer International Publishing
4. Aggarwal CC (2016) Content-Based Recommender Systems. In Recommender Systems, pp. 139–166. Springer International Publishing.
5. Aggarwal CC (2016) Model-Based Collaborative Filtering. In Recommender Systems, pp. 71–138, Springer International Publishing
6. Aggarwal CC (2016) Neighborhood-Based Collaborative Filtering. In Recommender Systems, pp. 29–70, Springer International Publishing
7. Alexander C (1979) The timeless way of building, vol 1. Oxford University Press, New York
8. Amatriain X, Basilico J (2015) Recommender Systems in Industry: A Netflix Case Study. In Recommender Systems Handbook, pp. 385–419. Springer US
9. Anthony J, Willemsen M C, Felfernig A, Gemmis M D, Lops P, Semeraro G, Chen L (2015) Human decision making and recommender systems. In Recommender Systems Handbook, pp. 611–648. Springer US.
10. Arvola M (2006) Interaction design patterns for computers in sociable use. Int J Comput Appl Technol 25(2–3):128–139. doi:10.1504/IJCAT.2006.009063
11. Balabanovic M, Shoham Y (1997) Fab: content-based, collaborative recommendation. Commun ACM 40(3):66–72
12. Blas ND, Garzotto F, Guermandi MP (2002) It works! A systematic method to evaluate the features of museum Web-sites, Palazzo dei Congressi, Bibliocom 2002, Rome
13. Bollen D, Knijnenburg BP, Willemsen MC, Graus M (2010) Understanding choice overload in recommender systems. In Proceedings of the fourth ACM conference on Recommender Systems, pp. 63–70. ACM
14. Borchers JO (2001) A pattern approach to interaction design. Ai Soc 15(4):359–376. doi:10.1007/BF01206115
15. Bottoni P, Guerra E, de Lara J (2010) A language-independent and formal approach to pattern-based modelling with support for composition and analysis. Inf Softw Technol 52(8):821–844
16. Build Your Perfect Interface with UI Design Patterns. http://www.sitepoint.com/build-your-perfect-interface-with-ui-design-patterns. Accessed: 9 April 2015
17. Burke R (2000) Knowledge-based recommender systems
18. Burke R (2002) Hybrid recommender systems: survey and experiments. User Model User-Adapt Interact 12(4):331–370
19. Buschmann F, Meunier R, Rohnert H, Sommerlad P, Stal M (1996) Pattern-oriented software architecture: a system of patterns. Addison Wesley
20. Coplien JO, Schmidt DC (1995) Pattern languages of program design. ACM Press/Addison-Wesley Publishing Co
21. Cosley D, Lam S K, Albert I, Konstan J A, Riedl J (2003) Is seeing believing?: how recommender system interfaces affect users' opinions. In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 585–592. ACM
22. Cremonesi P, Elahi M, Garzotto F (2015) Interaction design patterns in recommender systems, Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter, pp. 66–73, ACM.
23. Cremonesi P, Garzotto F, Negro S, Papadopoulos A, Turrin R (2011) Comparative evaluation of recommender system quality. In CHI'11 Extended Abstracts on Human Factors in Computing Systems, pp. 1927–1932. ACM
24. Cremonesi P, Garzotto F, Negro S, Papadopoulos A, Turrin R (2011) Looking for "good" recommendations: A comparative evaluation of recommender systems. In IFIP Conference on Human-Computer Interaction, pp. 152–168. Springer Berlin Heidelberg

25. Cremonesi P, Garzotto F, Turrin R (2012) Investigating the persuasion potential of Recommender Systems from a quality perspective: An empirical study. ACM Transactions on Interactive Intelligent Systems (TiiS) 2, no. 2 pp. 11
26. Cremonesi P, Garzotto F, Turrin R (2012) User effort vs. accuracy in rating-based elicitation. In Proceedings of the sixth ACM conference on Recommender Systems, pp. 27–34. ACM
27. Cremonesi P, Garzotto F, Turrin R (2013) User-centric vs. System-centric evaluation of recommender systems. Human-computer interaction–INTERACT 2013. Springer, Berlin Heidelberg, pp 334–351
28. Cremonesi P, Turrin R (2010) Recommender systems for interactive TV
29. Cremonesi P, Turrin R (2010) Time-evolution of IPTV recommender systems. In Proceedings of the 8th international interactive conference on Interactive TV&Video, pp. 105–114. ACM
30. Dearden A, Finlay J (2006) Pattern languages in HCI: a critical review. Hum–Comput Interact 21(1):49–102
31. Deldjoo Y, Elahi M, Cremonesi P, Garzotto F, Piazzolla P (2016) Recommending Movies Based on Mise-en-Scene Design. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, pp. 1540–1547. ACM
32. Deldjoo Y, Elahi M, Cremonesi P, Garzotto F, Piazzolla P, Quadrana M (2016) Content-Based Video Recommendation System Based on Stylistic Visual Features, Journal on Data Semantics, pp. 1–15, Springer
33. Deshpande M, Karypis G (2004) Item-based top-n recommendation algorithms. ACM Trans Inf Syst (TOIS) 22(1):143–177
34. Design patterns. ui-patterns.com/patterns. Accessed: 3 April 2015
35. Desrosiers C, Karypis G (2011) A comprehensive survey of neighborhood-based recommendation methods. In: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender systems handbook, Springer, pp. 107–144
36. Di Blas N, Garzotto F, Poggi C (2009) Web engineering at the frontier of the Web 2.0: design patterns for online 3D shared spaces. World Wide Web 12(4):345–379. doi:10.1007/s11280-009-0065-5
37. Dong J, Zhao Y, Peng T (2009) A review of design pattern mining techniques. Int J Softw Eng Knowl Eng 19(06):823–855
38. Duyne DKV, Landay J, Hong JI (2002) The design of sites: patterns, principles, and processes for crafting a customer-centered Web experience. Addison-Wesley Longman Publishing Co., Inc
39. Elahi M, Ricci F, Rubens N (2016) A survey of active learning in collaborative filtering recommender systems, Computer Science Review, Elsevier
40. Garzotto F, Matera M, Paolini P, (1998). To use or not to use? Evaluating usability of museum web sites. In Proceedings of Museums and the Web International Conference, Toronto, Canada, Archives & Museum Informatics, 1998; http://www.museumsandtheweb.com/mw98/papers/garzotto/garzotto_paper.html
41. Felfernig A, Friedrich G, Jannach D, Zanker M (2015) Constrain-tbased recommender systems, in: Recommender Systems Handbook, Springer, pp. 161–190
42. Francis P, Farzin H, Guéhéneuc YG, Moha N (2012) Recommendation system for design patterns in software development: An dpr overview. In Proceedings of the Third International Workshop on Recommendation Systems for Software Engineering, .. 1–5. IEEE Press
43. Gamma E, Helm R, Johnson R, Vlissides J (1994) Design patterns: elements of reusable object-oriented software. Pearson Education
44. Garzotto F, Paolini P, Bolchini D, Valenti S (1999) "Modeling-by-Patterns" of Web A.lications. In International Conference on Conceptual Modeling, .. 293–306. Springer Berlin Heidelberg. doi: 10.1007/3-540-48054-4_24
45. Garzotto F, Retalis S (2004) Symposium on design patterns for e-learning. In EDMEDIA 2004, Lugano, Switzerland
46. Gemmis MD, Lops P, Musto C, Narducci F, Semeraro G (2015) Semantics-aware content-based recommender systems, in: Recommender Systems Handbook, Springer, pp. 119–159
47. Goodyear P, Avgeriou P, Baggetun R, Bartoluzzi S, Retalis S, Ronteltap F, and Rusman E (2004) Towards a pattern language for networked learning. In Networked learning, pp. 449–455
48. Guéhéneuc YG, Mustapha R (2007) A simple recommender system for design patterns. Proceedings of the 1st EuroPLoP Focus Group on Pattern Repositories
49. Harrison R, Flood D, Duce D (2013) Usability of mobile applications: literature review and rationale for a new usability model. J Interact Sci 1(1):1–16
50. Hypermedia design patterns repository. designpattern.lu.unisi.ch/PatternsRepository.htm
51. Interaction Design Patterns Library. http://www.welie.com/patterns/index.php. Accessed: 9 April 2015
52. Interaction Design Patterns. https://www.interaction-design.org/encyclopedia/interaction_design_patterns.html. Accessed: 9 April 2015
53. Jannach D, Zanker M, Felfernig A, Friedrich G (2010) Recommender systems: an introduction, 1st edn. Cambridge University Press, New York
54. Jézéquel J M, Train M, Mingins C (1999) Design Patterns with Contracts. Addison-Wesley Longman Publishing Co., Inc
55. Kantor PB, Rokach L, Ricci F, Shapira B (2011) Recommender systems handbook. Springer

56. Koren Y, Bell R (2015) Advances in collaborative filtering, in: Recommender Systems Handbook, Springer, pp. 77–118
57. Liikkanen LA, Salovaara A (2015) Music on YouTube: user engagement with traditional, user-appropriated and derivative videos. Comput Hum Behav 50:108–124
58. Lockyer L, Bennett S, Agostinho S, Harper B (2009) Handbook of research on learning design and learning objects: issues, applications, and technologies (2 volumes). IGI Global, Hershey
59. Manzato D, Fonseca NLS (2013) A survey of channel switching schemes for IPTV. Commun Mag, IEEE 51(8):120–127
60. Martin D, Rodden T, Sommerville I, Rouncefield M, Hughes J (2002) Pointer: Patterns of interaction: A pattern language for CSCW .comp.lancs.ac.uk/computing/research/cseg/projects/pointer/pointer.html. Accessed: 17 December 2008
61. McNee S M, Riedl J, Konstan J A (2006) Being accurate is not enough: how accuracy metrics have hurt recommender systems. In CHI'06 extended abstracts on Human factors in computing systems, pp. 1097–1101. ACM
62. Nadia B, Kouas A, Ben-Abdallah H (2011) A design pattern recommendation approach. In Software Engineering and Service Science (ICSESS), 2011 I.E. 2nd International Conference on, pp. 590–593. IEEE
63. Nageswara RK (2010) Application domain and functional classification of recommender systems—a survey. DESIDOC J Libr Inf Technol 28(3):17–35
64. Resnick P, Varian HR (1997) Recommender systems. Commun ACM 40(3):56–58
65. Ricci F, Rokach L, Shapira B (2011) Introduction to recommender systems handbook. Springer, US, pp 1–35
66. Richardson JH (2014) The Spotify Paradox: how the creation of a compulsory license scheme for streaming on-demand music platforms can save the music industry. Entertainment Law Review 22, no. 1
67. Rossi G, Schwabe D, Garrido A (1997) Design reuse in hypermedia a.lications development. In Proceedings of the eighth ACM conference on Hypertext, pp. 57–66. ACM
68. Rossi G, Schwabe D, Garrido A (1997) Design Reuse in Hypermedia Applications Development. In Proc. of the ACM International Conference on Hypertext '97, ACM Press, pp. 57–66
69. Rossi G, Schwabe D, Lyardet F (1999) Improving Web Information Systems with Design Patterns. In Proc. of the 8th International World Wide Web Conference, Toronto (CA), Elsevier Science
70. Rubens N, Elahi M, Sugiyama M, Kaplan D (2015) Active Learning in recommender systems, Recommender Systems Handbook - chapter 24: Recommending Active Learning, pp. 809–846, Springer US
71. Schedl M, Knees P, McFee B, Bogdanov D, Kaminskas M (2015) Music Recommender Systems. In Recommender Systems Handbook, pp. 453–492. Springer US
72. Schummer T (2003) Gama: A pattern language for computer supported dynamic collaboration. In EuroPLoP, pp. 53–114
73. Schummer T (2005) A pattern approach for end user centered groupware development. Fern Universitat in Hagen
74. Schummer T, Lukosch S (2013) Patterns for computer-mediated interaction. Wiley
75. Shvets A, Frey G, M Pavlova (2016) Proxy Design Pattern from Design Patterns Explained Simply, sourcemaking.com/design_patterns/proxy, SourceMaking.com
76. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Advances in artificial intelligence, pp. 4
77. Swearingen K, Sinha R (2001) Beyond algorithms: an HCI perspective on recommender systems. In ACM SIGIR 2001 Workshop on Recommender Systems, vol. 13, no. 5–6, pp. 1–11
78. Tidwell J (2010) Designing interfaces. O'Reilly Media, Inc
79. Véras D, Prota T, Bispo A, Prudêncio R, Ferraz C (2015) A literature review of recommender systems in the television domain. Expert Syst Appl 42(22):9046–9076
80. Van Welie M, Van der Veer GC (2003) Pattern languages in interaction design: structure and organization. Proc Interact 3:1–5
81. Welie, M Van (2008) Design patterns for web, gui, and mobile interfaces. welie.com. Accessed: 17 Dec 2008

Paolo Cremonesi Is an associated professor at the Politecnico di Milano where he teaches courses on Computer System Architectures, Performance and Reliability. Since joining the Politecnico di Milano he has been involved in the performance assessment and optimization of computer system architectures and applications, ranging from real-time embedded systems up to enterprise-wide datacenters. He's author of various publications concerning distributed and parallel computation, performance evaluation and capacity planning. Paolo Cremonesi was born in Bergamo (Italy) in 1967. In 1991 he joined the "Von Karman Institute for Fluid Dynamics" in Brussels for a 4-month fellowship on supercomputing. In 1992 he received his master degree in Aerospace Engineering, summa cum laude, from the Politecnico di Milano. From 1992 until 1993 he worked with the "Italian Electric Power Company (ENEL) on the realization of a parallel real-time supercomputer. He received his PhD in Computer Science from the Politecnico di Milano in 1996. From 2001 until 2006 he has been Editor of the Euromicro Journal of Systems Architecture, published by Elsevier. In 2001 he joined the Computer Science Departement of the Politecnico di Milano as an assistant professor and, in 2005, as associate professor.

Mehdi Elahi Received M.Sc. degree in Electrical Engineering (Sweden), and Ph.D. degree in Computer Science (Italy), under the supervision of Prof. Francesco Ricci. During the course of Ph.D., he has researched on Recommender Systems, mainly focused on the cold start problem. He has designed, developed, and evaluated (offline/online) several personalized techniques for Active Learning in recommender systems. These techniques were integrated in a mobile context-aware recommender system for tourism, called "South Tyrol Suggests". As a result of his graduate research work, Mehdi Elahi served as a primary author or co-author on several publications in AI, ML, RS, IR, HCI, and UM related conferences and journals. He was also given the opportunity to publish his research findings in a chapter of the Recommender Systems Handbook (2nd edition). He is currently working at Politecnico di Milano as a postdoctoral researcher, researching on recommender systems, and coordinating the work of Ph.D. students.

Franca Garzotto Is Associate Professor of Computer Engineering at the Department of Electronics and Information, Politecnico di Milano. She has a M. Sc. Degree in Mathematics from the University of Padova (Italy) and a Ph.D. in Computer Engineering from the Politecnico di Milano. Her main research interests have span from hypermedia modeling to usability evaluation, interaction design, advanced user interfaces. She has been project coordinator of over 10 EC funded projects in the domains of e-learning, e- tourism, e-commerce, and e-health. She has served as European Chair of SIG Web and Italian Chair of SIGCHI. She is member of the Review Panel of European Research Council (ERC) Grants and of the Italian Management Board of EIT Digital (European Institute of Innovation and Technology)