



**HAL**  
open science

# CM-DIR: A Method to Support the Specification of the User's Dynamic Behavior in Recommender Systems

Carla A Martins, Carina Dorneles, Ankica Barišić, Thiago Rocha Silva, Marco Winckler

## ► To cite this version:

Carla A Martins, Carina Dorneles, Ankica Barišić, Thiago Rocha Silva, Marco Winckler. CM-DIR: A Method to Support the Specification of the User's Dynamic Behavior in Recommender Systems. HCSE 2024 - International Conference on Human-Centred Software Engineering, Jul 2024, Reyjavik, Iceland. pp.26 - 46, <10.1007/978-3-031-64576-1\_2>. <hal-04829247>

**HAL Id: hal-04829247**

**<https://inria.hal.science/hal-04829247v1>**

Submitted on 10 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

# CM-DIR: A Method to Support the Specification of the User’s Dynamic Behavior in Recommender Systems

Carla A. Martins<sup>1</sup>, Carina Dorneles<sup>1</sup>, Ankica Barišić<sup>2</sup>[0000–0001–7513–7907],  
Thiago Rocha Silva<sup>3</sup>[0000–0001–8961–4663], and Marco  
Winckler<sup>2</sup>[0000–0002–0756–6934]

<sup>1</sup> Federal University of Santa Catarina, Brazil

<sup>2</sup> Université Côte d’Azur, CNRS, Inria, I3S, France

{Marco.Winckler, Ankica.Barisic}@univ-cotedazur.fr

<sup>3</sup> The Maersk M. Moller Institute, University of Southern Denmark, Denmark  
thiago@mmmi.sdu.dk

**Abstract.** This paper introduces CM-DIR (Conceptual Model - Dynamicity and Interaction in Recommender Systems), a method that we have originally devised to help with the description of dynamic user behavior in Recommender Systems (RS). The identification of user behavior change is paramount to optimize RSs to face new contexts of use and adapt their user interface accordingly. Capturing dynamic user behavior is, however, not an easy task and most RS systems do not take this information into account during the requirements engineering process. The proposed method leverages user and Behaviour-Driven Development (BDD) stories and extends them to describe dynamicity. The method has been evaluated with a panel of experts through semi-structured interviews and the results suggest that the CM-DIR is necessary and comprehensible and that the proposed extension allows flexibility and adaptability while ensuring informative specifications.

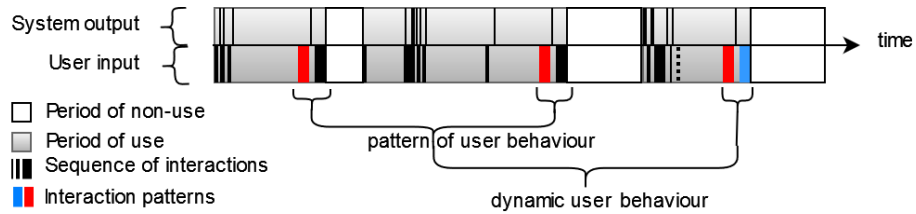
**Keywords:** User Stories · Behaviour-Driven Development (BDD) · Requirements Specification · Recommender Systems · User Behavior

## 1 Introduction

A Recommender Systems (RS) is a type of adaptive systems that uses information about the user interaction (such as clicking on items, browsing history, page revisiting, purchase history, likes, etc.) as an input to an algorithm that recommends a set of items that might match users’ interest. In that, RSs system might help users to discover new items and/or to find again items previously seem that are still relevant. RSs are largely employed nowadays, from reading content to web services [2, 15, 32], and play an essential role in helping users with decision making processes, such as what items to buy, what music to listen to, what news to read, etc. In all these processes, user interaction is a key component for the system to respond satisfactorily in recommending items [1]. While the accuracy

of the predictions is a core research question, the design of user interaction and the user interaction with RSs also have a strong impact on users' assessment of the recommendations and their overall experience of using the system [42].

Time of interactions is a core factor for understanding the users preferences. As we shall see in Figure 1, user interactions are characterized by periods of use (the so-called sessions) alternated by periods of non-use. While the analysis of *sequences of interactions* in a user session reveals the dialog between the user and the interactive system, the analysis of interactions across sessions allows the identification of *patterns of user behaviour*. In a longer run, it is also possible to observe the evolution of these patterns and then characterize the *dynamic user behavior* [34].



**Fig. 1.** General scheme of user interaction and behaviour changing overtime.

There may be countless reasons for a behavior change, and not all are worthy of interest. However, discovering behavior changes that make sense to the user allows the development of more flexible and valuable RSs. For recommendation algorithms, the problem is neither collecting information about the user interactions or detecting patterns, but rather to make sense of changing characteristics of the users.

A clear understanding of users needs along the time might be used to improve algorithms, thus preventing useless recommendations such as illustrated by Figure 2 where the RS proposes (at T2) the very same product that users just bought (at T1).

Researchers have shown that taking into account the temporal properties of datasets leads to improvements in rating prediction and ranking. Hence, analyzing the contextual nature of user feedback shows promise for performance improvement [11]. In this scenario, the system could, for example, adapt more effectively to the dynamic behavior of the user, which can be identified in some situations that consider the user behavior, the external context, or even periods of time.

Whilst many RS's algorithms value information about interaction as a mean to determine the recommendations, little is known about how to identify, specify, and use dynamic behaviors as part of the users requirements for tuning the recommendations. Indeed, as we shall see in section 2, there is a gap in the RS literature in terms of methods for specifying how user interactions should affect recommendations; so that, quite often, the only way of understanding recommen-

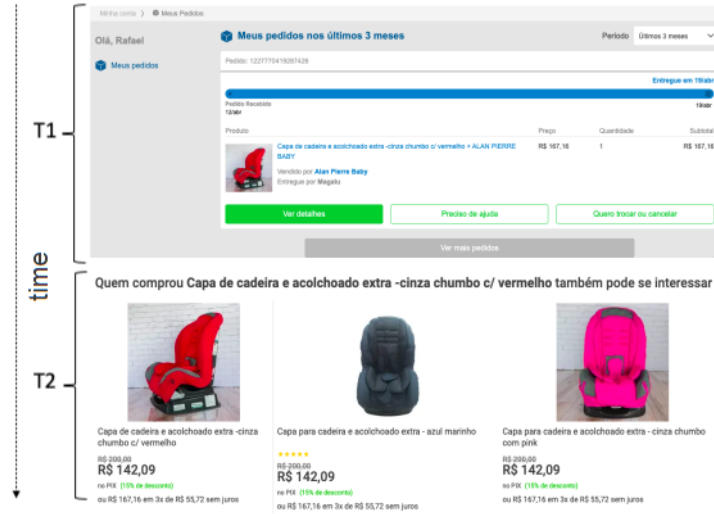


Fig. 2. Recommendation after a product purchase.

dations is to look at the code source documentation (if any) of RS algorithms. For that, this paper presents a method called CM-DIR (Conceptual Model – Dynamicity and Interaction in Recommender Systems). The method CM-DIR aims to assist requirements analysts and software developers with the specification of *user dynamic behavior* and their use for creating recommendations prior to the implementation of RS algorithms. The CM-DIR method proposes a few extensions to Behaviour-Driven Development (BDD) stories [24][29] to support the specification of *user dynamic behavior*. It was specially devised to address the following premises:

- To detect and to cope with ambiguous and/or conflicting recommendations;
- To inform which users' actions on the user interface, and more specifically *patterns of user behaviour*, must be taken into account to formulate recommendations;
- To prevent code source reading (of RS algorithms) as the sole way of understanding recommendations;
- To cope with of the evolution of all aspects surrounding recommendation systems (such as requirements, recommendations, user interface design, user behaviour, and system technology) that are prone to evolve over time.

The rest of the paper is organized as follows: section 2 presents the review of the literature. The section 3 introduces the theoretical framework of the method as well as the extension proposed for the user and BDD stories. Then, preliminary results of an evaluation of the method based on structure interviews with domain experts are presented in section 4 and discussed with respect to the related work in section 5. Lately, section 6 concludes the paper and discusses future work.

## 2 Overview of User Interaction and RSs

The literature often divide RSs into three main approaches based on filtering methods: content-based (CB), collaborative filtering (CF), and hybrid (Hybrid). Content-based methods generate recommendations by relying on attributes of items and users, such as age, sex, job type, and other personal information. On the other hand, item attributes are descriptive information that distinguishes individual items. Collaborative filtering methods focus on the historical interactions of users with the items to generate recommendations. Historical interactions of users with items can be a product purchase, listening to a song, rating a product, watching a movie, clicking on a news article, and similar. Finally, the hybrid approach combines previous ones to obtain a recommendation with fewer drawbacks than the others, which most large platforms use [27].

Users can interact with a RS either implicitly or explicitly. The action is explicit when the user selects or rates an item on the interface [12]. On the other hand, the interaction is implicit when the relevance of an item is based on the user history, such as the number of clicks, number of page visits, the number of times some user played a song, and so on, i.e., the number and time in which the implicit interaction occurs may represent the interest of users [2, 39]. Compared to the explicit interaction, the implicit interaction is much more expressive as hardly users provide explicit *feedback*, only 15% of users according to [39].

Some works consider time as a crucial factor in defining user preferences [14, 25, 9, 7, 22], as it serves as a basis for calculating values of variables used as measures to define preferences, such as time of permanence in certain items and navigation time and the number of actions repetitions over time [14]. Over time, the relative order of the clicked items can also reflect the detailed trajectories of the change of interest of a given user [9]. An SR also needs to consider the evolution of the user based on their needs, not just their personal preferences that are variable over time [25]. A RS can work on generating logs that mark the frequency and duration of occurrence of user interactions [7]. The logs are available on the server, browser, or proxy, and scores are assigned, which can be used to calculate the subsequent recommendations [22].

Other works in the literature propose metrics to measure user interaction with the interface in several dimensions [7]. The metrics assess user interaction, and the values measured serve as a parameter to filter and eliminate unnecessary information and recommend what is important [13, 35]. The metrics defined are simple, such as permanence time, number of visits, purchases made, the favorite topic, favorite content, and session duration [6]. Misztal-Radecka et al. [20] propose another set of metrics, and the idea is to build archetypes that serve as a structure in predicting new interactions, influencing the SR. In [6], the authors define the following items for this measurement: (a) total interaction time (in minutes); (b) favorite topic (ordered list of topics); (c) favorite content (ordered list of contents); (d) session duration (ordered list of content/duration pairs); (e) comments (number); (f) direct feedback (number of “starts”); (g) repetitive readings (number of iterations); (h) session length/content length.

In addition to the time variable, user interaction actions are influenced by the context in which the user is inserted. The context may restrict the [21] item selection actions. For example, in [26], the authors mention that depending on the environment in which they are, the user selects a style of music, or in [37], the authors show that depending on the context, the explanatory text of a document recommendation can be better detailed. In the analyzed studies, few studies considered the user’s context at the moment of interaction.

It is possible to identify the user interaction that contributes to the enhancement of the interface for using the recommendations, investing in the placement of the items to be chosen, and explaining the recommended items [37]. For this, a structure can be defined in the interface/document body, and attributes can be used to understand user interaction by reviewing the text or selecting the item. Feedback on such information begins to influence future review propositions [38].

User interaction may also influence RSs through the sequential interaction of the user in the selection (or not) of items, with the entire interaction being stored over time. After the interaction is captured, it is modeled with complex relationships for understanding implicit data recommendation [23]. In this case, the idea is to use the dynamic elements of interaction and time that contribute to the possibility of variation in the result of the recommendations.

When the contexts influence item selection, there is a personalization of the recommendations for which such interactions are mapped [26]. In this case, the dynamic elements worked are interaction and context, which may significantly influence the variation of selection results of the recommendations. Some studies [19, 40] deepen the specialization of user interaction learning to feedback to the RS and, thus, contribute to the precision of future item propositions.

Regarding user interaction, [23] exemplified a way called Sequential Implicit To Explicit (SITE), which directly models both types, item sequence and sequential user interactions with an item, to capture user interaction. The SITE method uses two continuous phases in a single task to model the several types of interactions as action sequences. Therefore, it is possible to model complex relationships among the user interactions with an item. It is also possible to capture the evolution of user preferences over time, exploring sequence items with which they interacted in the past.

### 3 The CM-DIR Method

The CM-DIR method was conceived to capture scenarios describing temporal aspects of the context of use, to specify the user dynamic behaviour, and to show which aspects of RSs should be modified (specially the user interface) to address user needs. The method also aims at capturing in the process the various influences of variables internal (desires and needs) and external to the user. In general, user interaction is directly connected to the variations in the decision to choose the recommended item, such as time, context, and the change of interests that impact interaction.

### 3.1 A Motivating Example

To illustrate one of the dynamic behaviors that our method aims to capture, consider the following motivating example related to a searched item. Julia majors in psychology and, for four semesters, have bought books and read articles in this topic. However, she changed her degree to dentistry and the subjects of her interest therefore changed. This dynamic change in behavior cannot be predicted by the current system, which keeps recommending the old subjects. In order to reduce the learning curve, the description of scenarios formalising *user dynamic behavior* is built using and extensions of the method Behaviour-Driven Development (BDD) [24][29]. This behavior change could be easily specified by our method as (the proposed extensions are highlighted in **lime**):

As a student, I want to remove the course of psychology <interaction 1>  
**Context:** study  
 OR dentistry <interaction 2>, OR physiotherapy <interaction 3>,  
 OR computing <interaction 4>  
**At time 1:** <sequence>  
 So that I can keep information of interest <interaction 5>

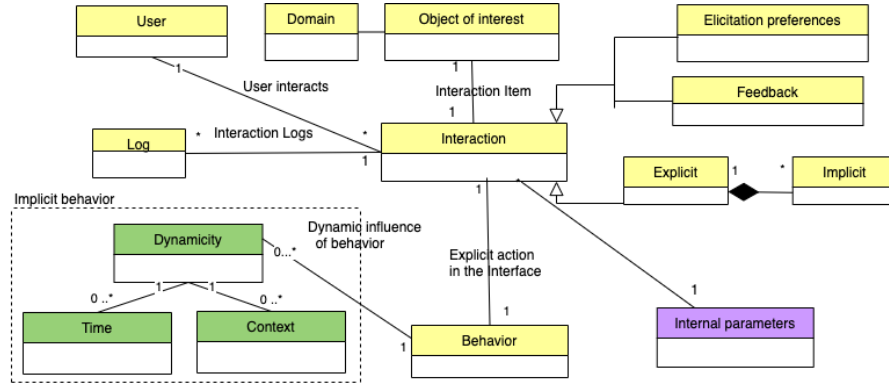
**Scenario:** Course change  
**Given** I do not use the course of psychology <interaction 1>  
**Context:** study  
 OR dentistry <interaction 2>, OR physiotherapy <interaction 3>,  
 OR computing <interaction 4>  
**At time 1:** <sequence>  
**When** I select dentistry <interaction 5>  
**And** I present a new course <interaction 6>  
**Then** the system relates it to the available content <interaction 7>

As we shall see, the use of BDD contemplates the indication of time, context, the user's specific object of desire, and the implicit interaction. A dictionary is then provided to specify the interactions on the user interface that capture such behaviors.

**Dictionary:**  
**interaction 1** : <unchecked> do not send information about the unchecked course.  
**interaction 2, 3, and 4** : <check> be able to select one of the options to receive information.  
**interaction 5** : <confirm> to present selected item to which content will be sent.  
**interaction 6** : <present check> selected component.  
**interaction 7** : <combo list> with information to be sent.

### 3.2 Conceptual Elements of CM-DIR

The method includes a conceptual model that describes the main concepts used for defining dynamic behavior variables and a stepwise process for mapping dynamic user behaviors to the (desired) adaptation of the user interface. The CM-DIR covers a set of concepts that determine the dynamic aspects of the user interactions with RSs. These concepts are presented in Figure 3.



**Fig. 3.** Core concepts in the conceptual model of dynamicity and interaction with RS.

The conceptual model used for describing the dynamic behavior is organized around the concept of user’s “Interaction”. This concept can be implemented either as “Elicitation preference” (when the user specify the preference prior to interacting with the system, like by defining preferred settings) or a “Feedback” (when users tell their preferences while interacting with the system, like by giving a positive mark for a recommendation). An interaction may be explicit or implicit. A user interaction is often counted as an explicit behavior, but some implicit behavior, such as waiting times, can be computed to determine the outcomes. The objects of interest of a user (item / product / feature) and the application domain are also specified as part of the conceptual model as they determine which objects will be affected by the recommendations. The interaction may represent dynamic behavioral issues related to the elapsed time and the context in which the user is.

The term “time” encapsulates the dynamics of the behavior which can be expressed as a sequence, a moment, and/or a period that describe the timeliness relevance of information for the user. Therefore, time is a core concept that will be used in the definition of the RS specifications. In addition, the user interaction is related to the internal system configuration parameters that process input variables to the system from the interface. With these concepts mapped, it is possible to provide conceptual support in identifying the dynamicity depicted by the user behavior with the system interface.

### 3.3 A 3-Stages Method

The CM-DIR method encompasses three stages (Figure 4), namely : (1) Requirements Gathering and Specification, (2) Requirements Analysis and Presentation, and (3) Interface Modeling. They are based on consolidated requirements processes such as CMMI [31] with an addition of dynamicity. Stage 1 (Requirements Gathering and Specification) focuses on identifying dynamic information about user behavior. Stages 2 and 3 are specifically designed to exploit the specification of requirements, respectively, for the presentation of requirements to the development team and for guiding the design of the user interface of RSs.

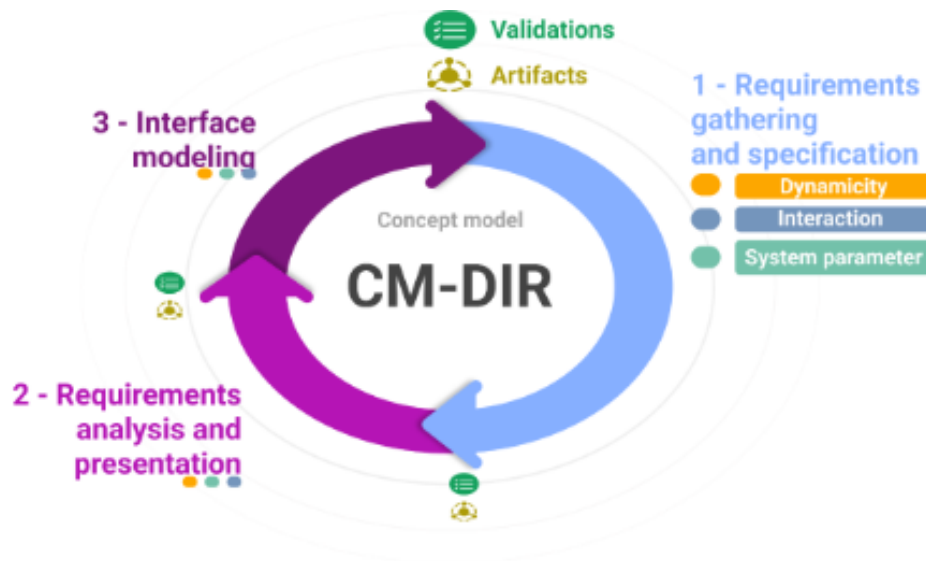


Fig. 4. Overview of the steps of the CM-DIR method.

The specification is key to prevent the ambiguities not only in terms of the description of the requirements but also in terms of how requirements are expected to be implemented in the user interface. Notice that the method (see Figure 4) also includes a validation step, which must be performed with the stakeholders at the completion of each stage to ensure that their needs and expectations are aligned with the generated artifacts. The following sections describe in detail the three stage of the method.

### 3.4 Requirements Gathering and Specification

**Requirements Gathering** is a step that aims at identifying requirements that might encompass changes in the user behaviors. For that we need to:

1. **Identify the dynamicity:** behaviors are identified through (explicit/implicit) interactions with objects of interest in a context of use. The variable interactions depend on the context of use. For the variable time, it tells the timely relevance of information. The following tasks are sought to help to identify these variables:
  - identify behaviors that generate explicit interactions on the interface;
  - note the time (moment/sequence/period) of events causing a behavior change;
  - check all context of use to identify a change in the users' interest with objects (ex. items, features, products).
2. **Identify interactions:** In this step, the temporal dynamicity (ex. sequence of interactions) associated with the behavior is also identified. Stakeholders must be consulted to determine the relevant associations, as it is not possible to generalize an interaction for all contexts of use. Then, for each interaction, the expected outcome on the user interface is described.
3. **Identify system parameters:** this cover internal system's requirements (and/or constraints) such as filtering algorithms, clustering techniques, and format of the system outputs (ex. ranking). The following elements must be considered:
  - parameters used in explicit / implicit interactions;
  - parameters used by time and context variables;
  - internal system requirements that meet the inputs/outputs of variables.

**Requirements Specification** is a step that focuses on practices to standardize the writing of the stakeholders' needs, as completely and generic as possible. These are recommendations for writing generic specifications:

- The user interactions with the interface should be generic and do not include terms that are specific to an application domain.
- Always assume that a user interface is the main source of information, where dynamic user behavior can be captured and the recommendations presented to the user.
- For each specification of behavior, we consider one object of interest at time.
- The attributes of the object interests must encompass information about the context. Hence, it is possible to relate items with the same context and differentiate contexts in which the user may choose.
- The variables time, context, and object of interest work together with behavior, but only time and context are optional in a specification.

Moreover, the notation to be used should be sufficiently flexible to accommodate the scenario with two or more options, express the context in which the item belongs, and work on the time variables and definition of the interaction that may add value with the defined implicit interaction. As for the formalism of specification, we extend two widely known techniques (user stories [5] and BDD scenarios [24]) in order to meet the use of the dynamic variables of time, context, and behavior. The proposed template with the dynamicity extensions (including a dictionary of controlled vocabulary) is presented below:

<p><b>As a</b> &lt;persona&gt;, <b>I want</b> &lt;feature&gt; &lt;interaction&gt;</p> <p><b>Context:</b> &lt;product attribute or item under specification&gt;</p> <p><b>AND/OR</b> &lt;resource&gt; &lt;interaction&gt;</p> <p><b>At time t:</b> &lt;sequence, period, moment&gt;</p> <p><b>So that</b> &lt;benefit&gt; &lt;interaction&gt;</p>
<p><b>Scenario:</b> &lt;title&gt;</p> <p><b>Given</b> &lt;condition c&gt; &lt;interaction&gt;</p> <p><b>Context:</b> &lt;product attribute or item under specification&gt;</p> <p><b>AND/OR</b> &lt;condition c+1&gt; &lt;interaction&gt;</p> <p><b>At time t:</b> &lt;sequence, period, moment&gt;</p> <p><b>When/And</b> &lt;action&gt; &lt;interaction&gt;</p> <p><b>Then/And</b> &lt;outcome&gt; &lt;interaction&gt;</p>
<p><b>Dictionary:</b></p> <p><b>interaction i</b> : &lt;interaction component on the user interface&gt;</p>

The following conventions apply to the template:

- Parameters of objects of interest as item/feature/product: (1) the name of the item, (2) the context, and (3) when the item will be shown again.
- Interaction dictionary: indicates which interaction component will capture the behavior that will be translated into explicit/implicit interaction. It must contain the identification and description of the meaning of the capture by the system.
- Re-redisplay settings are specified according to the operators presented in Table 1, including the options on the interface (symbolizes the options presented on the interface in which the user will take the action), the user actions (the attitude of the user towards the options available on the interface, which may be viewing, purchasing, evaluating, etc.), and the expected behavior of the system according to the specification (behavior expected by the system according to the defined specification of !OR, OR, !AND, and AND).
- Time: it is used to define the classification of time to be considered, as follows:
  - Sequence** – used when it is necessary to consider the order in which the items undergo an action. For example, when a user selects a sequence of items (such as the books), other users with similar profile will receive the same recommendation.
  - Period** – when a period of time for which an action is performed on an item. For example, in the case that a user selects a car seat cover and the RS waits for one or two years to provide a new recommendation of the same item.
  - Moment** – (progression) the exact moment in which an action is performed on an item. For example, at moment T1, a user goes onto a page to look at book options and does not execute anything on the page until moment T2. Depending on the time, different definitions may be considered, such as the user giving up or not finding what they need. These parameters may be used together or separately. It is possible to consider situations in which the information is provided through explicit interaction or as a result of processing an implicit interaction.

**Table 1.** Re-redisplay according to the action.

Operator	Options	User Actions	Behavior
!OR	1 .. n	2	Fetches what was not selected
OR	1 .. n	2	Fetches what was selected
!AND	1 .. n	Any option	Do not display anything
AND	1 .. n	Any option	Display everything

When a user story presents itself dynamically, several options are presented to the user, enabling the construction of various test scenarios. The importance of knowing if the story is dynamic is related to the number of necessary scenarios that must be created for testing, considering that there may be different times to understand the behavior. Each option in the interface is considered a test scenario. The keywords are repeated in BDD, but now with the aim of specifying all possible test conditions. The tests address one or more options selected through explicit interaction, which we may use in the dynamic capture of behavior by identifying implicit interactions (keyword interaction).

### 3.5 Requirements Analysis and Presentation

In this phase, all the specifications created by the RS analysts (including user stories and BDD templates) are organized and classified to facilitate the access to recommendations. The goal is to have a representation that all project participants may use to allow the joint analysis of the identified requirements, the possible existence of conflicts, duplicity, ambiguity, and other points that need to be unified or separated into new ones. During the analysis, the characteristics presented in the first stage about the dynamic user behavior in the face of systems are taken into account.

We suggest that the specifications must be grouped according to their purpose. If the requirements specified are too high level and need to be detailed, then organized by similarity. In addition, dependencies are identified if they are too complex and may also be prioritized according to the interest in delivery.

### 3.6 Interface Modeling

This stage foresees the construction of wireframes of the interfaces considering the behaviors identified and represented by the Interaction Components. Interaction components that capture the dynamicity of user behavior must be visible in the wireframe. The specification of the interaction and the use on the interface are described at the beginning of this section. At this step, the low-fidelity screens are validated with the stakeholders to ensure the achievement of satisfaction in the delivery of the developed product. Once the wireframe is created, we proceed to a second validation to unify requirements and eliminate possible ambiguities.

## 4 Evaluation

The method was assessed using semi-structured interviews of about 1 hour via video conference with seven independent experts (four from industry and three from academia), who evaluated the feasibility, ease of use, and completeness of the method. The analysis of interview data was inspired by Silva et al. [28]. The questions used, data colled, the analysis are available in a data repository [18].

### 4.1 Participants

Table 2 shows the profile of participants that were selected in a convenience sample to take part in the study. All participants had a higher education in computer science and/or software and are knowledgeable about BDD.

**Table 2.** Participant’s profile

ID	Gender	Education	Professional Experience	Role
P1	Female	Master’s	17y	Requirements Analyst
P2	Male	Postgraduate	14y	Solutions Architect
P3	Male	Postgraduate	18y	UX Designer
P4	Male	Postgraduate	20y	Engineer
P5	Female	PhD	22y	Research Engineer
P6	Male	PhD	20y	Researcher
P7	Male	Bachelor’s	6y	Full-Stack Developer

### 4.2 Procedure

The interviews followed the presentation of : (i) the goals of the study and overview of the method CM-DIR; (ii) concepts of interaction; (iii) concepts with dynamic characteristics (dynamic behaviour, time, context); (iv) review of user stories; (v) extensions for user stories; (vi) illustrative examples; (vii) the three stages of the CM-DIR method emphasizing the extensions; (viii) an example using the extended user and BDD stories; (ix) questions requesting participants to elaborate a user/BDD story including dynamicity using the proposed extension; (x) a post-questionnaire; and (xi) debriefing. The questions used in the post-questionnaire are: **(Q1)** *Is it clear why the conceptual model is needed? For the method as a whole? Why?*, **(Q2)** *Do the proposed concepts in the model reflect the reality? Why?*, **(Q3)** *Does it make sense for you that the conceptual model only covers the first step of the method?*, **(Q4)** *Do you consider that both user story and BDD appropriate techniques to address the dynamic aspects of recommender systems? Would you use another type of specification?*, **(Q5)** *How intuitive is the proposed specification?*, and **(Q6)** *Do you consider that the method encompasses all the necessary steps to address the presented problem? Do you think more specification is needed for implementation purposes?*

### 4.3 Results

**Proposal conformity and provided suggestions** We explored how participants perceived and used the various concepts (as described in the step ii and iii of the interview) to create an example using the extended user/BDD story (step ix). Figure 5 illustrates the participants' adherence to each item. Certain evaluation items exhibited deviations in adherence to issues and recommendations.

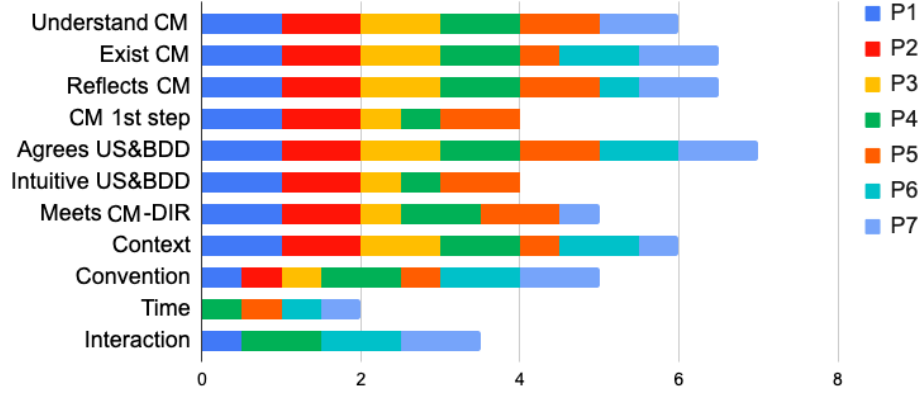


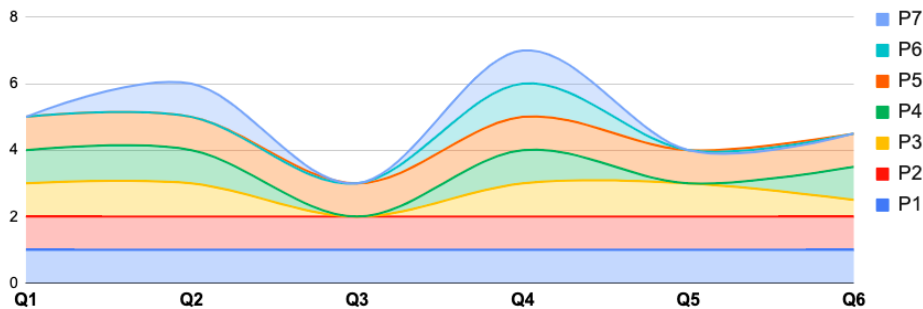
Fig. 5. Participants adherence to proposed conceptual model (CM).

Below, we elaborate the key findings, identifying areas for improvement and issues that could be addressed through adjustments to the proposed specification extension. The recommendations were detailed as follows:

1. Adequacy of a conceptual model (i,ii,iii)
  - *Understand CM* (Understanding of the presented model): It is not clear how the specified requirements are used in all the stages of the CM-DIR method
  - *Exist CM* (Enhancing a need for the existence of CM).
  - *Reflects CM* (The presented concepts in the model reflect the reality): More details would to facilitate design and testing, in particular how to concurrently combine time and context or to allow them to occur independently.
  - *CM 1st step* [4]: Two participants expressed the opinion that the conceptual model should be applied in the second phase, emphasizing its relevance in reflecting the subsequent steps of the CM-DIR. Additionally, its use in requirements validations was highlighted.
2. Standard US and BDD (iv, v)
  - *Agrees US and BDD* (The technique presented is suitable for use with user stories and BDD).
3. CM-DIR (vi, vii, viii)
  - *Intuitive US and BDD* (The proposal is intuitive): Enhance the comprehension of the connections between the conceptual model and the CM-DIR method. Refine the representation of e-commercial and E, as they symbolize the same concept and may lead to confusion. Improve the clarity of convention symbols. Propose an extension that integrates the use of US and BDD in tandem.

- *Meets CM-DIR* (The proposal is suitable to solve the problem raised about dynamism): Include a reference to tests in this context. Additionally, address the lack of detail regarding the utilization of UX design, as this information can impose limitations.
4. Use of the extended specification item (ix)
    - *Context* - All participants utilized the concept of context, with two employing phrases or expressions to define it. Enhance the guidance for defining context with a particular emphasis on using language suitable for the specification within a pre-defined scenario.
    - *Convention* - In one instance, no convention symbol was selected, and in another case, two symbols were employed. The latter scenario provides an opportunity to examine the feasibility of using multiple symbols simultaneously.
    - *Time* - In three specifications, the progression time and the time for re-displaying options on the interface were not indicated. Four other specifications provided partial information on time, encompassing aspects like the presentation of re-presentation time, definition of the moment, definition of the progression, and, in one case, the parameters (sequence, moment, and period) were left undefined.
    - *Interaction* - In three cases, there was no definition provided for interaction. Two cases partially outlined the interaction, while two other cases presented a comprehensive definition of interaction.

**Analysis of answers** Figure 6 summarizes the answers during the post-interview, step (x). Concerning *Q1* and *Q2*, five participants confirmed the need for a conceptual model, and they agreed on that the presented model accurately mirrors the reality. Two participants noted areas for potential enhancement in the conceptual model. Regarding *Q3*, four participants suggested alternatives for using the conceptual model into other phases of the CM-DIR method in order to facilitate validations across stages, fostering testing, and UX design. Conversely, three participants contend that confining the use of the conceptual model solely to the initial stage of the method is sufficient.



**Fig. 6.** Participants' answers to evaluation questions.

As for *Q4*, there was unanimous agreement among all participants asserting that BDD adequately captures the dynamic issues posed by the proposed model. Addressing *Q5*, unanimous positive feedback was received, with all interviewees expressing ease and intuitiveness in utilizing CM-DIR and its BDD extensions. Notably, three participants provided constructive suggestions aimed at refining the specification. Concerning *Q6*, the suggestions are towards going into more detail on UX design and testing.

**Problems while elaborating examples** During the interviews, participants were asked to write sample of requirements (step ix). Figure 7 shows a summary of problems detected on participants' specifications.

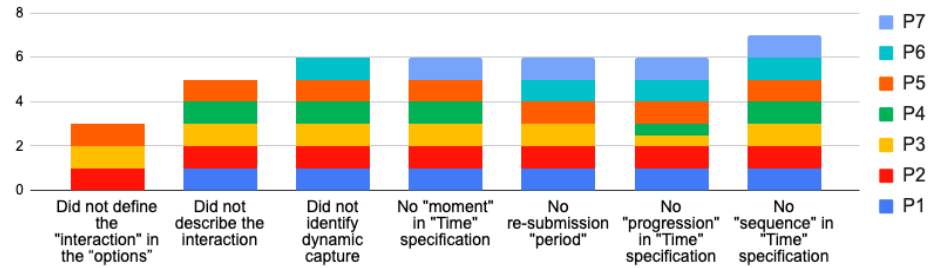


Fig. 7. Problem detected when using the proposed specification.

Hereafter we provide the list of the suggestions received to improve the CM-DIR method, as follows:

1. *Missing the definition of the 'interaction' in the 'options'* - Three participants did not define this parameter, possibly because explicit known interaction components are limited to clicks and selections, thus restricting the concept of dynamic behaviour captured through interaction.
2. *Did not describe the interaction* (Detailed explanation of how the interaction works - should include the explicit interaction used and how it is presented on the screen and received by the behavior screen) - Five participants did not provide descriptions of interactions, whether they used them or not. Some participants may have hastily included them directly into the specification without utilizing the dictionary.
3. *Did not identify dynamic capture* (Report on how the explicit dynamic user behavior will be interpreted by the explicit interaction) - Six participants did not address this capture aspect, indicating a need for more guidance on how to approach this task.
4. *Missing the option 'moment' in the 'Time' specification* (Information supporting the capture of information to generate implicit interaction) - Six participants did not define the moment of option selection. This was possibly due to the recommendation strategy outlined in the specification, which may not have required the use of this parameter. The moment parameter is not mandatory.

5. *No re-submission 'period'* (The definition of this time is linked to how the convention should function at each time) - Six participants did not utilize this specification, as their individual specification ideas did not pose a problem for the presentation of repeated items in a short period.
6. *No 'progression' in the 'Time' specification* (The progression time considers the situation of parameters defined at each moment of the specification) - Some participants defined T1, but the existence of Tn is necessary to identify dynamic variation. Based on the specifications and the conceptualization of projecting the same specification expressed at different times with different outcomes, it is assumed this might not be straightforward for those evaluating it.
7. *No 'sequence' in the 'Time' specification* (Considers the sequence in which items were selected or should be presented) - Although not a mandatory parameter, it provides information on implicit interaction to enhance the accuracy of recommending the next options.

**Summary of results** Overall, the findings from the interviews were quite encouraging. The majority of improvements suggested by participants are related to the use of the conceptual model. Participants suggested alternative examples that could be more intuitive for illustrating the method. Concerning the use specification, participants suggested clarifying when mandatory items (such as actions related to time progression and the period linked to conventions) are required. They also prompted the importance of defining interactions.

The proposed extension offers flexibility. For dynamic specifications, it must include behavior, options, convention, and at least one of the other parameters (interaction, context, or time). If any mandatory parameter is unspecified, the user story converts to the default use thus ensuring user-friendly specifications. Despite some participants' specifications lacking certain parameters, the dynamism is preserved, indicating the resilience and adaptability of the method.

In evaluating participant responses, a consensus among five participants underscores the need for a conceptual model reflecting behavior, context, time, and product. Some suggest integrating this model across CM-DIR phases for usability, while others find its application limited to the initial stage sufficient. Unanimous agreement asserts that BDD effectively addresses dynamic issues. While feedback on CM-DIR and BDD extensions was positive, there is an emphasized need for detailed explanations on UX design and testing.

In the evaluation of our approach, several limitations were also identified and should be discussed. First, the evaluation was constrained by the limited number of participants involved, which may impact the generalizability of the findings. Another notable limitation is the lack of experience of participants in handling dynamic requirements. Despite the flexibility of not requiring the simultaneous use of all parameters, there is a challenge in defining and managing the multitude of parameters involved. This complexity could potentially hinder the ease of implementation and usability of the approach.

## 5 Discussion and Related Work

The conception of our method was based on a prior systematic literature review (SLR) on requirements engineering techniques for RSs. During the stage of analysis of the works in the literature, we initially sought proposals that took into account user behavior in the face of RSs.

Most studies on RSs focus on the performance of recommendation algorithms. In general, only a few examples of recommendation algorithms use variables related to *sequence of interaction* over time [19, 20, 13, 23, 16] or *pattern of user behaviour* related to the context [37, 26]. *Dynamic user behavior* changes over time is not take into account [11].

We did not find any prior work with a proposal to formalize the requirements specification that reflected the dynamicity of user behavior. Therefore, we reviewed the scope of this study and sought to analyze some works that considered the user interaction actions in RS as these are systems in which the user interaction has a direct effect on the accuracy of the recommendations made. These works were classified into three groups: *(i)* preference elicitation, including works that captured interactions at the time when user preferences were not known; *(ii)* indication of preferences in the results, indicating user feedback and including works that considered capturing interactions when the system was already calibrated with some preferences; and *(iii)* hybrid, including works that used an approach that considered the two previous cases.

Works such as [10, 26, 35] require use preference to be elicited clearly, in which case the most mentioned action was “selecting”. Other important preferences, represented by interactions such as “view” [10], “visit” [13] and “check” [20] are related to the action of seeing something interesting, so it may be chosen and marked as a preference. As much as confirming what is observed through behavior is necessary, walking this path implicitly brings information of interest. Interactions of “editing” and “accessing” items appeared in only a few studies [20, 33], but have their functions in setting preferences. These interactions, combined with other ones, present relevant implicit information that may contribute to calibrating the systems, either with other interactions or with dynamic variables that support the flexibility of RSs functioning.

In the second group, concerning the indication of preferences in the results, the most recurring interaction actions were “rate product” or “categorize items” [17, 41, 30, 20] which indirectly make it possible to evaluate the purchased product. The “evaluate items” action [4, 17] also appeared as a user interaction. The “comment” [38] and “like” [3] actions were the least frequent from the preference results according to the works surveyed.

Finally, in the third group - the hybrid one [36, 16, 40, 3, 23], the interactions to be defined are implicitly captured at different moments and screens, and each moment will have its interaction.

Similar to other works, CM-DIR shares the overarching goal of addressing accuracy in recommendations. However, its distinctive feature lies in emphasizing precision improvement while building recommender systems.

This work is not alien to the problems reported in the Fogg Behavioral Model (FBM) [8]. The FBM asserts that three factors must occur at the same time for a person to perform a target behavior: the person must (1) be sufficiently motivated, (2) have the ability to perform the behavior, and (3) be triggered to perform the behavior. Despite the fact that the FBM has been demonstrated useful to support the analysis and design of recommendation systems by explaining the factors that affect the change in user behavior, it does not specify which elements of the user interface of RS systems are affected by the user behaviour change. In that sense, the CM-DIR method provides an complimentary notation that can finely describe the behaviour change on the system.

## 6 Conclusion

This paper introduced CM-DIR, a method that aims to guide requirements analysts and developers in building systems that must consider dynamic user behavior. Exploring the aspect of dynamism in user behavior corroborates studies already carried out in the literature, in which a deficiency was detected in investigating the problem [11]. Our proposed solution extends user and BDD stories by adding new elements to reflect the dynamism of user behavior in RSs, thus allowing interaction actions to be captured and treated more accurately. Our method was evaluated with a panel of experts, and the results proved promising because the participants responded positively to the approach, indicating that the problem is relevant and the solution is quite feasible.

In future work, the focus will be on refining the structure of user stories and BDD to enhance intuitiveness, with specific consideration of retaining only dynamism in BDD. The need for using certain parameters in conjunction, like Period and Convention, will be clarified. Additionally, the extension will involve the application of time parameters to each interface option. Further investigation will delve into the intricate details of defining interaction to better support the work of UX design. Furthermore, efforts will be directed towards improving the model to align with the requirements of phases 2 and 3 in the CM-DIR method.

**Acknowledgements** Partial support for this research was provided by the National Council for Scientific and Technological Development (CNPq) under Grant for graduate scholarship, Process Nr. 131014/2021-8 - GM GD 2020.

## References

1. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. *Knowledge-Based Systems* **46**, 109–132 (2013). <https://doi.org/10.1016/j.knosys.2013.03.012>
2. Cao, Z., Qiao, X., Jiang, S., Zhang, X.: An efficient knowledge-graph-based web service recommendation algorithm. *Symmetry* **11**(3) (2019). <https://doi.org/10.3390/sym11030392>

3. Chang, J., et al: Sequential recommendation with graph neural networks. p. 378–387. Association for Computing Machinery, New York, NY, USA (2021)
4. Chong, S., Abeliuk, A.: Quantifying the effects of recommendation systems. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 3008–3015 (2019)
5. Cohn, M.: User stories applied: For agile software development. Addison-Wesley Professional (2004)
6. Crespo, R.G., Martínez, O.S., Lovelle, J.M.C., García-Bustelo, B.C.P., Gayo, J.E.L., de Pablos, P.O.: Recommendation system based on user interaction data applied to intelligent electronic books. *Computers in Human Behavior* **27**(4), 1445–1449 (2011). <https://doi.org/10.1016/j.chb.2010.09.012>
7. Feng, M., Peck, E., Harrison, L.: Patterns and pace: Quantifying diverse exploration behavior with visualizations on the web. *IEEE Transactions on Visualization and Computer Graphics* **25**(1), 501–511 (2019). <https://doi.org/10.1109/TVCG.2018.2865117>
8. Fogg, B.: A behavior model for persuasive design. In: Proceedings of the 4th International Conference on Persuasive Technology. Persuasive '09, Association for Computing Machinery, New York, NY, USA (2009). <https://doi.org/10.1145/1541948.1541999>, <https://doi.org/10.1145/1541948.1541999>
9. Gan, M., Xiao, K.: R-rnn: Extracting user recent behavior sequence for click-through rate prediction. *IEEE Access* **7**, 111767–111777 (2019). <https://doi.org/10.1109/ACCESS.2019.2927717>
10. Guo, S., Li, C.: Hybrid recommendation algorithm based on user behavior. In: IEEE 9th Joint International Inf. Tec. and Artificial Intelligence Conf. (2020)
11. Jawaheer, G., Weller, P., Kostkova, P.: Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Trans. Interact. Intell. Syst.* **4**(2) (2014)
12. Jawaheer, G., Weller, P., Kostkova, P.: Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Trans. Interact. Intell. Syst.* **4**(2) (jun 2014). <https://doi.org/10.1145/2512208>
13. Jianjun, M.: Research on collaborative filtering recommendation algorithm based on user behavior characteristics. In: 2020 Int. Conf. on Big Data Artificial Intelligence Soft. Eng. (ICBASE) (2020)
14. Kang, S., Jeong, C., Chung, K.: Tree-based real-time advertisement recommendation system in online broadcasting. *IEEE Access* **8**, 192693–192702 (2020). <https://doi.org/10.1109/ACCESS.2020.3031925>
15. Kong, X., Mao, M., Wang, W., Liu, J., Xu, B.: Voprec: Vector representation learning of papers with text information and structural identity for recommendation. *IEEE Transactions on Emerging Topics in Computing* **9**(1), 226–237 (2021). <https://doi.org/10.1109/TETC.2018.2830698>
16. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
17. Liu, T., et al: Sorrrs: Social recommendation incorporating rating similarity and user relationships analysis. In: 2020 7<sup>th</sup> Int. Conf. on Inf., Cybernetics, and Comp. Social Syst. (ICCSS) (2020)
18. Martins, C.A.: Research evaluation information - cm-dir - mendeley data (2023). <https://doi.org/10.17632/ddmvrdc4sc>
19. Meshram, R., et al: Monte carlo rollout policy for recommendation systems with dynamic user behavior. In: Int. Conf. on COMMunic. Syst. NETWORKS (2021)

20. Misztal-Radecka, J., Indurkha, B.: Persona prototypes for improving the qualitative evaluation of recommendation systems. In: Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization. pp. 206–212 (2020)
21. Mo, Y., Chen, J., Xie, X., Luo, C., Yang, L.T.: Cloud-based mobile multimedia recommendation system with user behavior information. *IEEE Systems Journal* **8**(1), 184–193 (2014). <https://doi.org/10.1109/JSYST.2013.2279732>
22. Neelima, G., Rodda, S.: Predicting user behavior through sessions using the web log mining. In: 2016 International Conference on Advances in Human Machine Interaction (HMI). pp. 1–5 (2016). <https://doi.org/10.1109/HMI.2016.7449167>
23. Nguyen, T., Ngo Van, L., Than, K.: Modeling the sequential behaviors of online users in recommender systems. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II (2020)
24. North, D.: Introducing bdd. (2006), <http://dannorth.net/introducing-bdd/>
25. Ortiz Viso, B.: Evolutionary approach in recommendation systems for complex structured objects. In: Fourteenth ACM Conference on Recommender Systems. p. 776–781. RecSys '20, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3383313.3411455>
26. Qian Gao, P.M.: Graph neural network and context-aware user behavior prediction and recommendation system research. *Computational Intelligence and Neuroscience* (2021)
27. Ricci, F., Rokach, L., Shapira, B.: Recommender Systems: Techniques, Applications, and Challenges, pp. 1–35. Springer US, New York, NY (2022). [https://doi.org/10.1007/978-1-0716-2197-4\\_1](https://doi.org/10.1007/978-1-0716-2197-4_1)
28. Rocha Silva, T., Winckler, M., Bach, C.: Evaluating the usage of predefined interactive behaviors for writing user stories: an empirical study with potential product owners. *Cognition, Technology & Work* **22**, 437–457 (2020)
29. Rocha Silva, T.a.a.: Ensuring the consistency between user requirements and task models: A behavior-based automated approach. *Proc. ACM Hum.-Comput. Interact.* **4**(EICS) (jun 2020). <https://doi.org/10.1145/3394979>
30. Saranya, et al: Social recommendation system using network embedding and temporal information. In: 5<sup>o</sup> Internat. Conf. on Computing, Communic. and Security (ICCCS) (2020)
31. (SEI), S.E.I.: Cmmi® for development version 1.2 improving processes for better products. Pittsburgh, Pensilvânia, EUA: Carnegie Mellon University (2006)
32. Setty, V., Hose, K.: Event2vec: Neural embeddings for news events. SIGIR '18, Association for Computing Machinery, New York, NY, USA (2018). <https://doi.org/10.1145/3209978.3210136>
33. Shibamoto, E., et al: A recommendation system of sightseeing places based on user's behavior of taking and editing photos. In: IEEE Pacific Rim Conf. on Commun., Comp. and Signal Proc.(PACRIM) (2019)
34. Shin, D.: How do users interact with algorithm recommender systems? the interaction of users, algorithms, and performance. *Computers in Human Behavior* **109**, 106344 (2020)
35. Walek, B.: Creating adaptive web recommendation system based on user behavior. *Journal of Physics: Conference Series* **933**, 012014 (2018)
36. Widiyaningtyas, T., et al: User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system. *Journal of Big Data* (2021)
37. Xu, X., et al: Understanding user behavior for document recommendation. In: Proceedings of The Web Conference 2020. ACM (2020)

38. Yang, Z., Zhang, M.: Textog: A recommendation model for rating prediction based on heterogeneous fusion of review data. *IEEE Access* (2020)
39. Yi, P., et al: An opt. method for rec. sys. based on user implicit behavior. In: 2015 5<sup>o</sup> Int. Conf. on Instrument. and Measurement, Comp., Communic. and Control (2015). <https://doi.org/10.1109/IMCCC.2015.326>
40. Zheng, L., et al: Joint deep modeling of users and items using reviews for recommendation. In: 10th ACM Int. Conf. on Web Search and Data Mining (2017)
41. Zhou, D., et al: Novel sddm rating prediction models for recommendation systems. *IEEE Access* (2021)
42. Ziegler, J., Loepp, B.: *Interactive Recommendation Systems*, pp. 1–29. Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-319-27648-9\\_54-1](https://doi.org/10.1007/978-3-319-27648-9_54-1)