Noname manuscript No.
(will be inserted by the editor)

Situation Awareness for Recommender Systems

Christian Richthammer · Günther Pernul

Received: date / Accepted: date

Abstract One major shortcoming of traditional recommender systems is their inability to adjust to users' short-term preferences resulting from varying situation-specific factors. To address this, we propose the notion of situation-aware recommender systems, which are supposed to autonomously determine the users' current situation based on a multitude of contextual side information and generate truly personalized recommendations. In particular, we develop a situation awareness model for recommender systems, include it in a situation-aware recommendation process, and derive generic design steps for the design of situation-aware recommender systems. The feasibility of these concepts is demonstrated by directly employing them for the development and implementation of a music recommender system for everyday situations. Moreover, their meaningfulness is shown by means of an empirical user study. The outcomes of the evaluation indicate a significant increase in user satisfaction compared to traditional (i.e. non-situation-aware) recommendations.

Keywords Recommender systems · Situation awareness · Context awareness · Contextual side information · Situation-aware recommender systems

1 Introduction

The rise of the World Wide Web has made sharing and accessing various kinds of information easier and faster than ever before, resulting in considerable benefits for its users (including both content providers and consumers). Long since,

Christian Richthammer
University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany
Tel.: +49-941-943-5638
Fax: +49-941-943-2744
E-mail: christian.richthammer@ur.de

Günther Pernul
University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

however, this trend has reached a point where the increasing amounts of alternatives and information overwhelm the users in the course of their decision making processes [26]. Recommender systems are intended to solve this phenomenon of information overload by making users aware of only those items they are probably interested in [13, 14]. However, one basic assumption underlying most recommender systems still is that users' preferences do not change very fast [7]. Users' general interests may in fact be relatively stable but their preferences may also be influenced by many additional situation-specific factors, which are generally referred to as "context" [7]. Thus, users' preferences in a specific situation may differ greatly from their general interests. Being unable to take this into account (i.e. to adjust to short-term preferences) constitutes a major shortcoming of traditional recommendation techniques.

Against this background, the area of context-aware recommender systems has been established. Obviously, the main goal of the research on context-aware recommender systems is to increase the quality of recommendations, and thus the users' satisfaction with the service, by taking advantage of the correlations between contextual side information and rating data in order to adapt the recommendations to the users' current needs. In this paper, we pursue this goal as well. But instead of context awareness, we focus on situation awareness. Context awareness denotes the system's state of being aware of single pieces of contextual side information describing the users' current environment. Compared to this, we go one step further and blend the different kinds of contextual side information together into the concept of situation. It is important to note that the same set of contextual side information may refer to completely different situations for different users. For example, a high heart rate on a sunny Monday morning at work may mean preparing for an important presentation for one user while it may mean loading a van with heavy parcels for another user. These situations demand for completely different recommendations – calming music and energizing music, for instance. Thus, situation awareness denotes the system's state of being able to actually comprehend the users' current personal situation.

Following this motivation, the starting point of this paper is the development of a novel model of situation awareness for recommender systems. This model is then incorporated into a traditional recommendation process. Based on the proposed situation-aware recommendation process, we define three generic design steps for the design of situation-aware recommender systems. To practically illustrate the design steps, they are directly employed for the design of a music recommender system for everyday situations. First, we conduct an extensive user survey with 158 participants to determine the influence of different types of contextual side information on people's music preferences. Second, we develop a distance measure to enable our system to autonomously detect the situation users are currently in. And third, we select an existing recommendation technique that is compatible with the proposed situation-aware recommendation process. Implementing the developed design in the form of a prototypical software application demonstrates the feasibility of the concept of situation-aware recommender systems. Finally, we show in a second empirical study with 67 participants that users receiving situation-aware recommendations are significantly more satisfied than users receiving recommendations from the same recommendation technique but without taking their situation into consideration. In particular, situation-aware recommendations were rated 16 % better.

The remainder of the paper is organized according to the guidelines for conducting design science research by Hevner et al. [27] and Peffers et al. [36]. Sect. 2 introduces the basics of (context-aware) recommender systems as well as situation awareness and delimits the envisaged concept of situation-aware recommender systems from related work. In Sect. 3, a situation awareness model for recommender systems as well as a situation-aware recommendation process are introduced and, based on this, three generic design steps for the design of situation-aware recommender systems are derived. These are directly employed for the design and development of a music recommender system for everyday situations in Sect. 4. Sect. 5 discusses the prototypical software application implementing the developed design. In Sect. 6, the concept of situation-aware recommendations is evaluated. Sect. 7 concludes the paper by pointing out aspects for future work.

2 Background and Related Work

2.1 Recommender Systems

Recommender systems can be seen as a filtering mechanism and are supposed to provide people with only the most relevant information and only those alternatives that are worth trying [13, 14]. Over the years, many different ways of estimating users' preference values have been proposed. They are usually classified into different categories according to their general approach to recommendation generation, with collaborative filtering and content-based filtering being the originally distinguished ones [3, 6]. However, the main shortcoming of these approaches is their underlying assumption that users' preferences are stable over time or at least do not change very fast [7]. Hence, they are not designed for being able to adjust to short-term preferences.

A step towards focusing on users' short-term preferences can be seen in the knowledge-based approach. Knowledge-based recommender systems [12] allow users to specify their preferences in the form of explicit criteria and provide recommendations matching these criteria. Automatically considering users' short-term preferences, which may be influenced by varying situation-specific factors, is eventually addressed by context-aware recommender systems.

2.2 Context-aware Recommender Systems

The notion of context is frequently considered across different research disciplines. Therefore, it has been conceptualized through a wide variety of definitions or so-called "context models" over the years. A thorough analysis of the work on these two conceptualization approaches has been done by Bazire and Brézillon [11] and Bauer and Novotny [10], respectively. In this paper, we rely on the much-cited definition by Dey [19]:

> Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Moreover, we use the following terminology [9,44]:

- Contextual dimension: A type of contextual side information (e.g. location, weather)
- Contextual condition: A specific value of a contextual dimension (e.g. home / work, rainy / sunny)
- Contextual situation: A combination of several contextual conditions (e.g. {home, rainy})

On a meta level, there are two views of context: the representational and the interactional view [20]. The key assumptions of the representational view are that context is stable and independent from the underlying activity. By contrast, the interactional view assumes that the contextual dimensions and conditions are defined dynamically and that context and activity are in a cyclical relationship.

Regarding the incorporation of contextual information into the recommendation process, three basic paradigms can be distinguished [2]. In the contextual pre-filtering paradigm, only those ratings that have been provided in the given context are considered for recommendation generation. Conversely, in the contextual post-filtering paradigm contextual information is initially ignored and the intermediate set of recommendations is then filtered according to the given context. In the contextual modeling paradigm, contextual information is used directly in the rating prediction process.

2.3 Situation Awareness

In general, the concept of situation awareness has been around long before the interest in context-aware recommender systems [21]. According to Niu et al. [33], successful naturalistic decision making requires sufficient situation awareness of the decision makers. Similarly, Stanners and French [40] empirically validate the obvious but previously unconfirmed assumption that good situation awareness leads to good decisions. As decision making is especially critical in military combat and emergency situations, it is not surprising that most of the literature on situation awareness originates from these domains [1, 22,23,37].

In the respective papers, situation awareness is mainly viewed from the perspective of human decision makers because, despite the possible support by machines, the final decisions are taken by them. Due to the technological advances of the recent years, however, decisions are more and more made autonomously by machines. Therefore, we argue that situation awareness should also be viewed from the system perspective (cf. Sect. 3.1). So far, this has been done only briefly by only few scholars [24,41].

2.4 Delimitation from Related Work

Instead of providing a comprehensive list and detailed discussion of single related publications, we take the corresponding recommender system types as a whole and compare them to the envisaged situation-aware recommender system according to different stages of the recommendation process. Moreover, we do not discuss each specialization of context-aware recommender systems, i.e. focusing on only one specific type of contextual side information, separately. Instead, we add Spotify's recommender system to the comparison. It is summarized in Fig. 1.

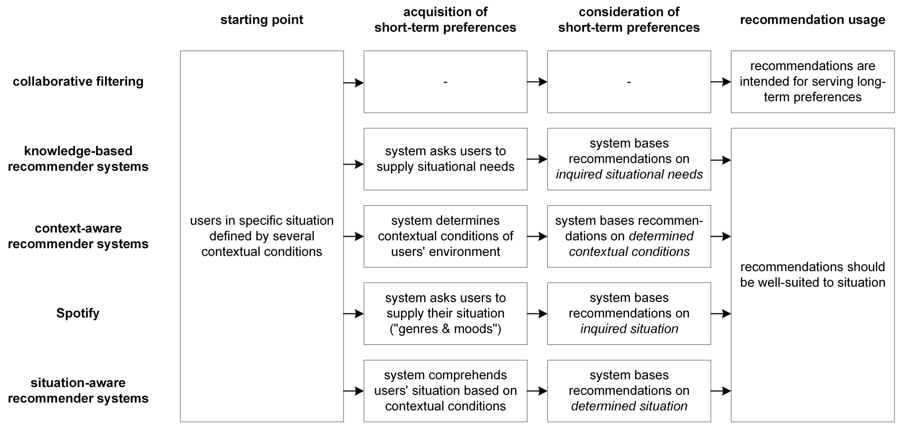
Collaborative filtering: Collaborative filtering approaches are taken as a baseline for the comparison. They are representative for all recommendation approaches that are intended for serving the users' long-term preferences and that do not take any specific short-term preferences into account.

Knowledge-based recommender systems: As introduced in Sect. 2.1, the knowledge-based approach can be seen as a step towards focusing on users' short-term preferences. Here, the system explicitly asks the users to supply their situational needs and considers them accordingly.

Context-aware recommender systems: The context-aware approach (cf. Sect. 2.2) determines the contextual conditions of the users' current environment and uses them as additional input data for the recommendation process. However, it does not actually comprehend the situations.

Spotify: On Spotify, users are able to supply their current situational needs in the form of labeled situations through a dedicated menu called "genres and moods". Spotify then displays several curated playlists fitting the selected situation. This constitutes a shift from context awareness to situation awareness. However, neither the contents of the playlists nor the choice of playlists are personalized.

Situation-aware recommender systems: As opposed to context-aware recommender systems, we propose a system that is not only aware of the contextual conditions describing the users' current environment (referred to as "context awareness") but that is able to actually comprehend the users' current situation (referred to as "situation awareness"). This is similar to the approach of Spotify. But instead of asking the users to explicitly indicate their current situation, our envisaged system determines the users' situation autonomously. Moreover, it generates truly personalized recommendations instead of only displaying several sets of recommendations that generally fit the determined situation.



	starting point	acquisition of short-term preferences	consideration of short-term preferences	recommendation usage
collaborative filtering	users in specific situation defined by several contextual conditions	-	-	recommendations are intended for serving long-term preferences
knowledge-based recommender systems		system asks users to supply situational needs	system bases recommendations on *inquired situational needs*	recommendations should be well-suited to situation
context-aware recommender systems		system determines contextual conditions of users' environment	system bases recommendations on *determined contextual conditions*	
Spotify		system asks users to supply their situation ("genres & moods")	system bases recommendations on *inquired situation*	
situation-aware recommender systems		system comprehends users' situation based on contextual conditions	system bases recommendations on *determined situation*	

Fig. 1 Situation-aware recommender systems compared to related approaches

3 Situation-aware Recommender Systems

3.1 Situation Awareness Model

As introduced in Sect. 2.3, we argue that situation awareness should also be viewed from the system perspective. In particular, we propose a situation awareness model specifically tailored to recommender systems.

In accordance with the original situation awareness model by Endsley [23], the heart of the model is the decision making process starting at the *state of the environment* and eventually leading to the *performance of actions* and the *feedback* of the accompanying effects to the environment. The key element of this process, in turn, is the achievement of *situation awareness*. In this regard, the three situation awareness levels of the original model are adapted to the mechanics of recommender systems. The first level is concerned with the mere perception of the elements of the users' current situation. In particular, low-level ("raw") environmental data have to be collected continuously and in real-time. On the second level, high-level contextual conditions are derived from the collected raw data. For example, the current time is mapped onto one of several discrete times of day. The set of derived contextual conditions adds up to a contextual situation (cf. Sect. 2.2). On the third level, the recommender system determines the situation the users are in based on the set of contextual conditions (i.e. based on the contextual situation). An example would be "rainy Monday morning at work". Following this, the system is able to adjust the generation of recommendations (i.e. the *decision*) to the users' current situation.

Both the overall decision making process and the achievement of situation awareness are influenced by several *system factors*. Following the original model, these are the *goals & objectives*, the *information processing mechanisms*, and the *memory stores* of the recommender system.

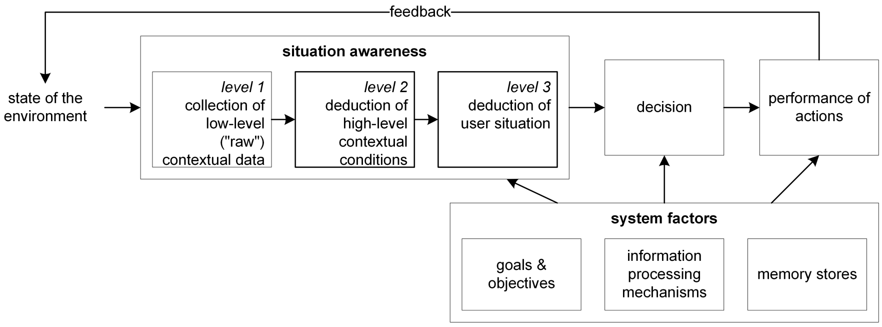

feedback
state of the environment
situation awareness
level 1 collection of low-level ("raw") contextual data
level 2 deduction of high-level contextual conditions
level 3 deduction of user situation
decision
performance of actions
system factors
goals & objectives
information processing mechanisms
memory stores

Fig. 2 Situation awareness model for recommender systems

3.2 Basic Design Decisions

Since the situation awareness model implies that only those preference data the users have provided in the identified situation are used for recommendation generation, contextual information is incorporated through *pre-filtering*. In particular, pre-filtering is realized in the form of user splitting [8]. As pre-filtering may drastically reduce the amount of feedback data available for recommendation generation [4], we focus on *content-based filtering* techniques because these suffer far less from data sparsity than collaborative filtering methods [32]. In particular, we rely on item recommendation based on seed items [16,31]. These are items that are representative of the users' preferences in a given situation and provided as *explicit feedback* by the users.

As most of the work on context-aware recommender systems [2], we focus on the *representational view*. Thus, we consider the contextual dimensions to be fully observable and static. In particular, this means that both the set of relevant contextual dimensions and their set of contextual conditions are known a priori and do not change over time. Hence, they have to be specified as part of the system design [2]. Furthermore, this means that all contextual conditions of an actual situation can be determined at the time of recommendation.

The decision of recommending *one item vs. a set of items* depends on the application scenario and does not have to be answered until the development of the system. The same applies to the decision of recommending *unknown vs. known items*. In the music domain, for example, only recommending unknown songs is not meaningful because songs are typically listened to many times [15]. By contrast, movie recommender systems can focus on unknown items because the same item is usually not consumed again and again in this domain.

3.3 Process Overview

The situation awareness model and the basic design decisions form the basis for the formulation of a situation-aware recommendation process (cf. Fig. 3).

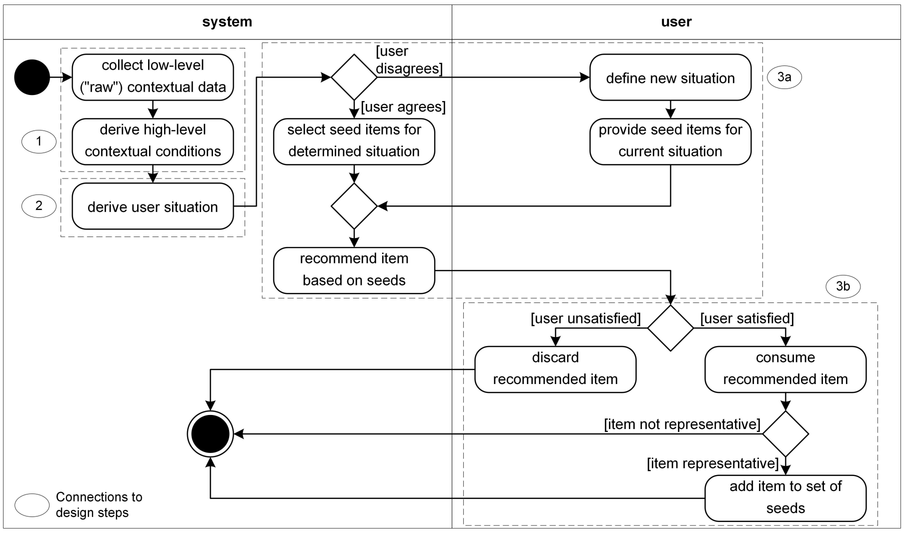

system
user
collect low-level ("raw") contextual data
derive high-level contextual conditions
1
2
derive user situation
[user disagrees]
[user agrees]
select seed items for determined situation
define new situation
3a
provide seed items for current situation
recommend item based on seeds
3b
[user unsatisfied]
[user satisfied]
discard recommended item
consume recommended item
[item not representative]
[item representative]
add item to set of seeds
Connections to design steps

Fig. 3 UML activity diagram of situation-aware recommender systems

At first, the system runs through the three levels of situation awareness as introduced in Sect. 3.1, which results in the deduction of the user's situation. If the user agrees with this situation, the system searches for seed items that the user has defined for the given situation in the past. If he disagrees, he is able to define a new situation in the system and provide some seed items for it. The next activity is the item recommendation based on the seed items that are available for the given situation. The user can then either discard the recommended item or consume it. If he decides to consume it, he may furthermore assess it as representative of his preferences in the given situation and add it to the corresponding set of seeds.

3.4 Generic Design Steps

Based on the activity diagram previously introduced (cf. Fig. 3), we define three generic design steps for situation-aware recommender systems.

In order for the system to be able to derive the contextual conditions of a given situation, the designer first has to specify which contextual dimensions the system is supposed to consider. Therefore, the first design step is the identification of the contextual dimensions that are most relevant to the envisaged application scenario (1). After that, the deduction of the user's situation requires the identification of the situation stored in the system that is most similar to the current contextual situation. Therefore, the second design step is the specification of a distance metric indicating the similarity between a pair of situations (2). The third and final design step is the definition of the recommendation technique (3a). In this regard, it also must be specified how the system is supposed to present the recommendations to the users and acquire corresponding feedback (3b).

Overall, the generic design steps are independent of any scenario and thus universally applicable. Nevertheless, discussing them in more detail is more meaningful if they are actually applied to a specific recommendation domain.

4 Music Recommender System for Everyday Situations

As required for situation-aware recommender systems, music recommendation is characterized by great amounts of user feedback on the same item but in different situations. The reasons for this are that listening to a song only takes a few minutes, that users may listen to the same song again and again, and that users' music preferences may vary greatly for different situations [15]. Moreover, the search space in music recommendation is extremely large [15], which makes the information overload problem particularly relevant in this domain.

4.1 Step 1: Identification of Most Relevant Contextual Dimensions

The most relevant contextual dimensions are identified in the following three steps. They are inspired by both Endsley's [22] methodology for determining the relevant elements in connection with situation awareness and Baltrunas et al.'s [9] approach on the acquisition of context relevance.

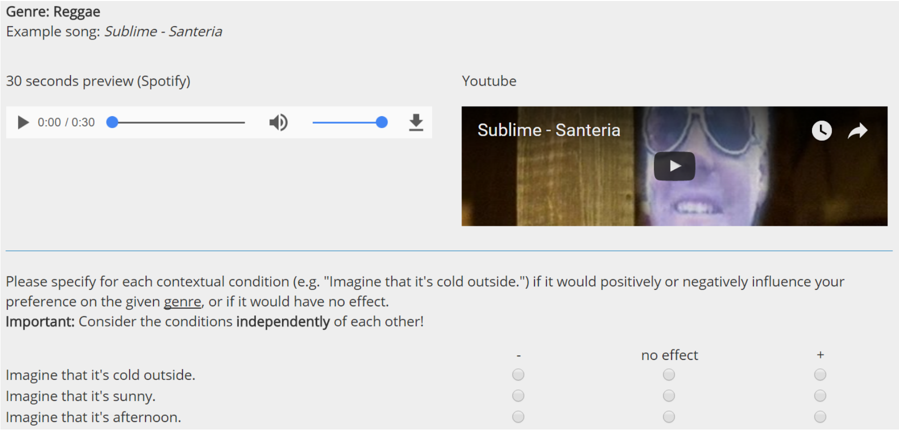
4.1.1 Survey of Contextual Dimensions Used in Existing Work

In order to come up with a comprehensive list of possible contextual dimensions, we conduct a systematic literature review according to the well-established guidelines by Webster and Watson [43] and Levy and Ellis [30].

In the literature review, the following digital libraries are used: ACM, AISeL, IEEE Xplore, and ScienceDirect. We arrive at our search phrases by combining each of the terms *recommend**, *collaborative filter**, and *content-based filter** with each of the terms *context** and *mobil**. *Mobil** is included because context awareness plays an important role in the area of mobile recommender systems. The asterisk (*) ensures that all variations of the terms (e.g. recommender, recommendation) are considered in our search.

4.1.2 Manual Pre-Selection of Meaningful Contextual Dimensions

The final list of contextual dimensions used in recommender systems literature contains more than 70 items. Many of them, however, are very specific and/or have only been used in one paper (e.g. wind direction [38]). Others require explicit user input (e.g. accompanying people), which would be in conflict with our goal of autonomously determining the users' situation. Overall, the utility



Genre: Reggae
Example song: *Sublime - Santeria*

30 seconds preview (Spotify)

Youtube

Please specify for each contextual condition (e.g. "Imagine that it's cold outside.") if it would positively or negatively influence your preference on the given genre, or if it would have no effect.
Important: Consider the conditions **independently** of each other!

	-	no effect	+
Imagine that it's cold outside.	○	○	○
Imagine that it's sunny.	○	○	○
Imagine that it's afternoon.	○	○	○

Fig. 4 Web survey to determine the practical relevance of different contextual dimensions

of the contextual dimensions varies greatly depending on the characteristics of the particular recommender system (e.g. application area, recommendation technique) [2].

Considering these criteria, we manually pre-select those we consider potentially meaningful in the light of the application area of music recommendation and the basic design decisions outlined in Sect. 3.2. Thus, the list is reduced to the following eight: season (S), day of the week (D), time of day (TD), temperature (T), heart rate (HR), place (P), weather (W), and movement (M).

4.1.3 Determination of Practical Relevance of Contextual Dimensions

Since too many contextual dimensions in connection with sparsely available feedback data lead to overfitting and decreased recommendation performance [4,35], it may be necessary to even further reduce the set of considered dimensions. Thus, we additionally conduct an in-depth user study to determine the practical relevance of the pre-selected dimensions.

The study is realized by means of a web application (cf. Fig. 4). After providing some optional data (e.g. age, gender), the participants are given one out of ten genres [42] that they are supposed to relate their subsequent answers to. For the sake of imaginability, the genre is represented by an example song. The participants' task is to indicate whether a certain contextual condition (e.g. "Imagine that it's cold outside.") would positively or negatively influence their preference on the given genre, or would have no effect. This has to be done for three contextual conditions. Note that each contextual condition belongs to one of the contextual dimensions that have been pre-selected before and that both contextual dimensions and conditions are drawn at random. The outlined procedure is repeated for the other nine genres (in random order), which adds up to 30 opinions per participant.

Table 1 Uncertainty coefficients $U(I|C_i)$ for different music genres

Blues		**Classical**		**Country**		**Disco**		**Hip Hop**	
HR	0.14486	P	0.14593	TD	0.12689	D	0.30724	TD	0.17222
TD	0.13577	S	0.13808	M	0.09144	M	0.23773	T	0.13571
M	0.07021	M	0.12523	D	0.08750	TD	0.14876	S	0.12551
P	0.06910	T	0.10254	HR	0.07954	W	0.09649	D	0.09508
W	0.06903	TD	0.08733	S	0.07050	P	0.08962	W	0.08729
D	0.04297	W	0.07263	P	0.05345	S	0.08876	M	0.07445
S	0.02257	HR	0.06478	W	0.03408	T	0.01997	HR	0.07408
T	0.01620	D	0.01765	T	0.01653	HR	0.01315	P	0.04977
Jazz		**Metal**		**Pop**		**Reggae**		**Rock**	
M	0.14912	TD	0.15282	S	0.10569	M	0.23967	HR	0.17009
HR	0.12451	HR	0.14399	HR	0.10562	S	0.23865	P	0.14795
D	0.10328	W	0.13198	M	0.10465	D	0.17205	M	0.11131
W	0.09178	M	0.08104	T	0.05478	W	0.16677	D	0.09786
P	0.04867	D	0.04622	TD	0.05371	HR	0.11659	TD	0.09710
S	0.04347	S	0.03632	P	0.04878	T	0.11593	W	0.04716
TD	0.03604	P	0.01649	D	0.02335	P	0.04835	S	0.03932
T	0.01305	T	0.01096	W	0.00680	TD	0.03298	T	0.03521

The web survey was completed by 158 participants (80 male, 76 female), who gave 4,718 opinions in total.[1] The median of the participants' age is 24 (SD = 6.898), which is due to the large number of students (109). Although the study may be criticized for being based on the participants' imagination, this approach has several advantages. First, people's decisions are more logical in virtual situations [5]. Second, virtual situations have a larger impact on people's ratings [5]. And third, it is impossible to create notably different real-world situations in a time frame that is reasonable for this kind of study.

With the data acquired through the web survey, we are able to model the influence of a contextual dimension C_i on the users' inclination I to listen to a given music genre G by means of the uncertainty coefficient. For a fixed G, the uncertainty coefficient is defined as:

$$U(I|C_i) = \frac{H(I) - H(I|C_i)}{H(I)}$$

where $H(I)$ is the entropy of the marginal distribution of I and $H(I|C_i)$ is the mean conditional entropy of I under C_i. $U(I|C_i)$ equals 0 if C_i does not have any influence on I, i.e. the two variables are stochastically independent. The higher $U(I|C_i)$ the more C_i helps reduce the error in predicting I. $U(I|C_i)$ equals 1 if the value of I is certain for each C_i.

Consequently, we calculate $U(I|C_i)$ for all contextual dimensions and genres. The results are depicted in Table 1. The mean values of the uncertainty coefficients of each contextual dimension are as follows (in descending order): M = 0.1285, TD = 0.1044, HR = 0.1037, D = 0.0993, S = 0.0909, W = 0.0804, P = 0.0718, and T = 0.0521. With respect to the second design step, these values can be used to reduce the set of contextual dimensions considered in the

[1] Responding was optional for all personal data as well as for all contextual conditions.

Table 2 Contextual dimensions and conditions of the envisaged recommender system

contextual dimension	**conditions**
season (S)	0 (January), ..., 11 (December)
day of the week (D)	0 (Sunday), ..., 6 (Saturday)
time of day (TD)	0 h, ..., 23 h
temperature (T)	-20 °C, ..., 29 °C
heart rate (HR)	40 bpm, ..., 200 bpm
place (P)	(-90.0..., -180.0...), ..., (90.0..., 180.0...)
weather (W)	0 (tornado), ..., 47 (isolated thundershowers)
movement (M)	0 (still), ..., 6 (in vehicle)

situation-aware recommender system to the most relevant ones. Alternatively, they can be used as weights for the contextual dimensions when determining the similarity between a pair of situations.

4.2 Step 2: Determination of Similarity Between Situations

Every user situation is connected to a contextual situation, which consists of k contextual conditions (one for each of the k considered contextual dimensions). Therefore, the similarity of two situations x and y is determined according to their underlying contextual conditions.

In the course of the previously described user study, it has been sufficient to use an exemplary set of rather coarse-grained contextual conditions for each contextual dimension. The more fine-grained contextual conditions we consider for the envisaged recommender system are listed in Table 2.

4.2.1 Normalization of Distance Values

Since the contextual conditions have completely different ranges of values, we normalize the respective distance values to the interval $[0;1]$.

Season, day of the week, time of day: The contextual conditions of these contextual dimensions feature fixed extreme values with a minimum value of 0 and are equally distributed with discrete steps of 1. Therefore, for normalization, the contextual conditions are first divided by their cardinality. Moreover, these contextual dimensions are characterized by a smooth transition from their maximum value to their minimum value. We consider this through the following equation, where x_i and y_i constitute the respective contextual condition of contextual dimension i (already divided by the cardinality of the set of contextual conditions) of situation x and y. $(x_i - y_i)$ is the normalized distance value. The multiplication by 2 is needed to get a result in the interval $[0;1]$.

$$(x_i - y_i) = \begin{cases} 2 * (1 - |x_i - y_i|) & \text{if } |x_i - y_i| > 0.5 \\ 2 * (|x_i - y_i|) & \text{else} \end{cases}$$

Temperature, heart rate: The contextual conditions of these contextual dimensions do not feature fixed extreme values and are not equally distributed.

We argue that the same absolute difference between two contextual conditions has greater impact for contextual conditions that appear more often and less impact for contextual conditions that appear less often. Therefore, the distance between two contextual conditions is calculated by means of their cumulative distribution function. To obtain this function for temperature, we use the frequency distribution of the average day temperature in Regensburg, Germany from 1947 to 2016 (with a minimum value of -20 °C and a maximum value of 29 °C)[2]. For heart rate, there are no appropriate data we could rely on. Hence, we assume the corresponding contextual conditions to be equally distributed with a minimum value of 40 and a maximum value of 200[3].

Place: Here, each contextual condition actually consists of two values, i.e. latitude and longitude. Therefore, the distance between two geographic coordinates has to be calculated. For short distances (e.g. within a country), it would be sufficiently precise to use the Euclidean distance [17]. For the sake of general applicability, however, we rely on the great-circle (orthodromic) distance, which takes into account the surface of the Earth [29]. Finally, this distance value is divided by the maximum possible distance between two geographic coordinates in order to get a result in the interval $[0; 1]$.

Weather, movement: The contextual conditions of these contextual dimensions constitute Yahoo weather codes[4] and Google activity codes[5] on an ordinal scale, respectively. We compare two contextual conditions by calculating the difference between the respective internal codes and normalizing the result by dividing it by the maximum value. Please note that because of the ordinal scale of weather and movement, distance calculations are actually not meaningful here. Nevertheless, the respective contextual conditions are at least appropriately ordered, which enables meaningful uses of the calculated distances.

4.2.2 Overall Distance Measure

The normalization results in a normalized distance value for each of the k contextual dimensions. In order to integrate them to an overall distance measure, we rely on the k-dimensional Euclidean distance [17]:

$$d(x, y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2 * \omega_i}$$

where $(x_i - y_i)$ can be replaced by the single normalized distance values. Through ω_i, the distance value for a particular contextual dimension is weighted according to the relevance of this contextual dimension as determined in Sect. 4.1.3, with $\sum \omega_i = 1$. The final similarity value is given as:

[2] `https://www.dwd.de/DE/leistungen/klimadatendeutschland/klarchivtagmonat.html`

[3] `https://healthyforgood.heart.org/move-more/articles/target-heart-rates`

[4] `https://developer.yahoo.com/weather/documentation.html/#codes`

[5] `https://developers.google.com/android/reference/com/google/android/gms/location/DetectedActivity` (original order of activity codes changed)

$$sim(x, y) = 1 - d(x, y)$$

In order to derive the users' current situation, the system has to determine the similarity values of the corresponding contextual situation to each situation stored in the system. The one showing the highest similarity value is used as the basis for the computation of recommendations.

4.3 Step 3: Computation of Recommendations and Acquisition of Feedback

Since this paper does not aim at developing a new technique for recommendation computation, an existing algorithm matching the basic design decisions outlined in Sect. 3.2 has to be identified. This is true for Spotify's "Get Recommendations based on Seeds" as part of the Spotify Web API[6].

The basic design decision of pre-filtering in the form of user splitting is realized through the determination of the users' situation as outlined in Sect. 4.2. Spotify's seed recommender then allows specifying up to five seed items, which the users have previously defined for the determined situation, as input. Seed items can be seed genres, seed artists, and seed tracks. Because of the content-based nature of Spotify's seed recommender, it is also possible to specify additional criteria that should be considered for recommendation generation. In particular, these are minimum values, maximum values, and target values for tuneable song attributes. Based on these input data, Spotify's seed recommender returns a playlist of songs (both known and unknown ones). From this set of items, our situation-aware recommender system selects the first song for playback without any user interaction. Nevertheless, returning an entire playlist of songs features the advantage of enabling the users to immediately request the next recommendation. After all, the users' situation is unlikely to change within only one song.

Regarding the acquisition of feedback, users have the possibility to add a recommended song to the set of seed items of their current situation. This constitutes an explicit form of positive feedback. Negative feedback cannot be directly used as input for Spotify's seed recommender.

5 Implementation

In order to show how the proposed concept of situation-aware recommender systems could be put into practice and thus demonstrate its feasibility, we implement the system design discussed in the previous section in a prototypical software application. We realize this in the form of an Android app (4.4 Kitkat, API level 19) because smartphones offer convenient access to a large variety of contextual dimensions, especially the users' movement and heart rate.

[6] `https://developer.spotify.com/web-api/get-recommendations/`

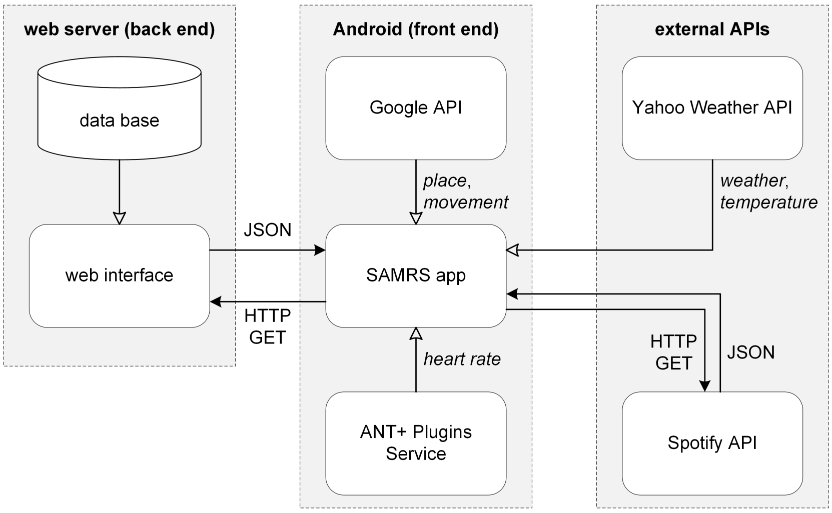


Fig. 5 Architecture of the situation-aware music recommender system

5.1 Architecture

Fig. 5 depicts the architecture of the situation-aware music recommender system (SAMRS). It is organized in three blocks: the web server in the back end, the Android operating system in the front end, and external APIs.

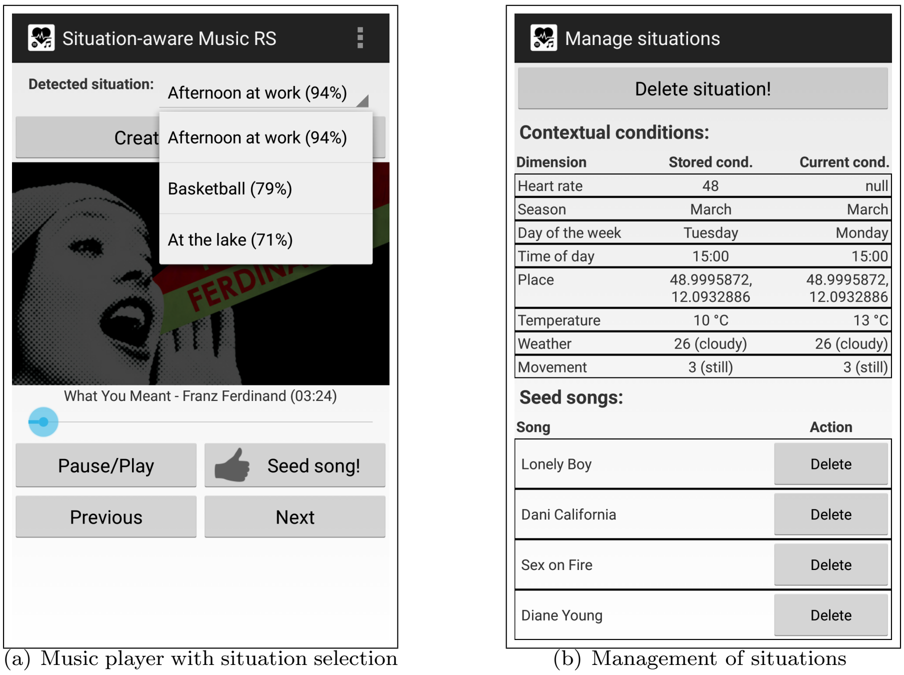
The main component of the front end is the SAMRS app. Via the native java class `Calendar`, it offers access to *time of day*, *day of the week*, and *season*. The Google API provides the app with the contextual dimensions *place* (via GPS) and *movement* (via Google Play Services). ANT+ Plugins Service[7] is needed to send *heart rate* data from a fitness tracker to the app. Since Google does not offer any weather-related interfaces, we rely on the Yahoo Weather API[8] to get data on the contextual dimensions *weather* and *temperature*.

In the back end, we rely on the LAMP (Linux, Apache, MySQL, PHP) solution stack. The web interface handles the communication between the SAMRS app and the database. In particular, it processes all database operations including the corresponding calculations. For example, the current contextual conditions, which are determined through the sensors and APIs described before, are sent from the SAMRS app to the web interface. There, the contextual situation is compared to all situations stored in the database. Finally, the most similar situations including the corresponding seed songs are returned to the SAMRS app.

The seed songs are eventually needed by the Spotify API. Its main task is the generation of song recommendations based on seed items. The Spotify track IDs contained in the returned JSON documents can directly be used to stream the recommended songs within the SAMRS app.

[7] `https://www.thisisant.com/directory/ant-plugins`

[8] `https://developer.yahoo.com/weather/`



Detected situation: Afternoon at work (94%)

What You Meant - Franz Ferdinand (03:24)

Contextual conditions:

Dimension	Stored cond.	Current cond.
Heart rate	48	null
Season	March	March
Day of the week	Tuesday	Monday
Time of day	15:00	15:00
Place	48.9995872, 12.0932886	48.9995872, 12.0932886
Temperature	10 °C	13 °C
Weather	26 (cloudy)	26 (cloudy)
Movement	3 (still)	3 (still)

Seed songs:

Song	Action
Lonely Boy	Delete
Dani California	Delete
Sex on Fire	Delete
Diane Young	Delete

(a) Music player with situation selection

(b) Management of situations

Fig. 6 Screenshots of the prototypical software application

5.2 Practical Usage

In the following, we describe a practical usage example from the perspective of the fictitious user Alice to illustrate the mechanics of the developed prototype.

When Alice uses the prototype for the first time, there are no situation definitions stored in the database yet. Therefore, she creates a new situation based on the current contextual conditions. Instead of defining specific seed songs, Alice is supposed to select a seed genre for initialization because this is considered faster and easier for her. In addition, she can optionally specify target values for the tuneable song attributes available through the Spotify API (cf. Sect. 4.3). The new situation is then stored in the database along with five seed songs fitting the specified seed genre and song attributes.

If there already are situations stored in the database, the one that is most similar to Alice's current environment is determined. By default, all contextual dimensions mentioned in the preceding subsection are considered for similarity calculations weighted according to their relevance resulting from the empirical study discussed in Sect. 4.1.3. However, Alice is able to adjust this to her preferences. Since the contextual conditions defining Alice's current situation may change, situation detection is repeatedly performed in fixed intervals. This interval is five minutes by default but can be adjusted by Alice. The situation most similar to Alice's current environment is then automatically selected and the corresponding seed songs are used for recommendation generation. The SAMRS app creates a temporary private Spotify playlist[9] according to the resulting set of recommendations and immediately streams the first song of this playlist. Nevertheless, Alice is able to display the next most similar situations and select another one as the most appropriate one (cf. Fig. 6(a)). If she is not satisfied with any of her existing situations, she can also create a new one.

Fig. 6(a) depicts the main activity of the implemented Android app, which follows the typical design of a music player. Moreover, the activity shows the detected situation (along with the option to select or create another one) and offers a feedback button to add the current song to the set of seed songs. The set of seed songs can be managed through the activity depicted in Fig. 6(b). The situation management activity also lists all contextual dimensions with the contextual conditions that are stored for the selected situation compared to the contextual conditions of Alice's current environment.

6 Evaluation

In addition to demonstrating the mere feasibility of the proposed concept of situation-aware recommender systems (cf. Sect. 5), we also evaluate its meaningfulness. The empirical study conducted in this regard is discussed in the following.

6.1 Evaluation Scope

It is important to note that the focus of this evaluation is on the meaningfulness of the general idea of taking the users' situation into consideration for recommendation generation. Consequently, we exclude all other connected aspects such as the detection of the users' current situation by the system. If we evaluated the general idea *including* the autonomous detection of situations, for example, we would not be able to tell which share of an increase (or decrease) in recommendation quality can be ascribed to the general idea and which share stems from a high (or low) accuracy of the situation detection concept.

Hence, the systems compared in the evaluation are supposed to differ only in the particular aspect of considering the users' situation for recommendation generation or not. This results in the following two candidate systems, which both rely on Spotify's seed recommender as their recommendation technique.

Baseline approach: Recommendation generation is based on all seed songs taken together, i.e. regardless of the situation in which they have been

[9] Creating a temporary private playlist is necessary because the Spotify API does not support queuing songs.

supplied to the system. Thus, the input to Spotify's seed recommender is a sample of the users' general music preferences.

Proposed approach: Our situation-aware recommender system distinguishes between situations and only uses those seed songs as input that have been supplied in the given situation. Thus, the input to Spotify's seed recommender resembles the users' music preferences in this particular situation.

In order to measure the quality of recommendations, we rely on the users' satisfaction with the recommendations in particular situations. Satisfaction is expressed on a 5-star rating scale for each recommendation.

6.2 Study Design

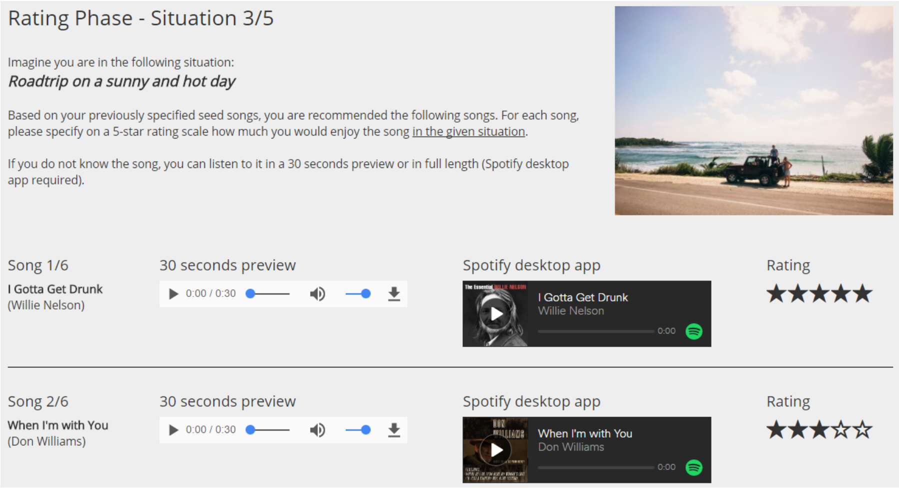
Traditionally, recommender systems are ideally evaluated on the basis of popular datasets that are publicly available. Since we propose a completely new variation of the commonly accepted recommendation categories, however, it is impossible for us to rely on such a reference dataset – which is a general problem in the area of context-aware recommender systems as well [28]. This makes it necessary to base our evaluation on real users' judgments obtained in an empirical study.

Similarly to our first user study described in Sect. 4.1.3, this study is realized by means of a web survey. Since it is considerably more time-consuming, we organize it in the following three phases.

Initialization phase: As the study requires the participants to rate music recommendations in the light of particular situations, it is crucial that the participants are able to imagine listening to music in these situations as easily as possible. Therefore, we provide a set of 25 exemplary situations from which the participants are supposed to select those five in which they can most easily picture themselves listening to music. In order to ensure a sufficient diversity of the participants' selection, the 25 exemplary situations are organized in five categories and the participants are only allowed to select one situation per category.

Preparation phase: For each of the five situations selected in the initialization phase, the participants are required to specify at least two and up to five seed songs that are representative of their music preferences in the particular situation. In order to find appropriate seed songs, the participants are able to perform track searches, artist searches, and genre searches.

Rating phase: For each of the five situations selected in the initialization phase, the participants finally get six song recommendations based on the seed songs specified in the preparation phase. Similarly to our first user study described in Sect. 4.1.3, this adds up to 30 ratings per participant. Each song is supposed to be rated on a 5-star rating scale according to how much the participants would enjoy the song in the particular situation. The interface of the rating phase is depicted in Fig. 7.



Rating Phase - Situation 3/5

Imagine you are in the following situation:
Roadtrip on a sunny and hot day

Based on your previously specified seed songs, you are recommended the following songs. For each song, please specify on a 5-star rating scale how much you would enjoy the song in the given situation.

If you do not know the song, you can listen to it in a 30 seconds preview or in full length (Spotify desktop app required).

Song 1/6	30 seconds preview	Spotify desktop app	Rating
I Gotta Get Drunk (Willie Nelson)	0:00 / 0:30	I Gotta Get Drunk Willie Nelson 0:00	★★★★★

Song 2/6	30 seconds preview	Spotify desktop app	Rating
When I'm with You (Don Williams)	0:00 / 0:30	When I'm with You Don Williams 0:00	★★★☆☆

Fig. 7 Rating phase of the empirical study

6.3 Results

The web survey was completed by N = 67 participants (41 male, 24 female)[10], who provided 2,010 ratings in total. 35 participants were randomly assigned to the situation-aware group receiving recommendations from the proposed situation-aware approach and 32 participants to the control group receiving recommendations from the non-situation-aware baseline approach. The median of the participants' age is 24 (SD = 3.428), which is due to the large number of students (44). The majority of the participants stated that they were interested (25) or very interested (25) in music.

Since rating as the dependent variable cannot automatically be assumed to be an interval variable, we rely on the Mann-Whitney-U test for statistical analysis because it requires the dependent variable to be only on an ordinal scale. The test indicates that users receiving situation-aware seed song recommendations (Median = 3) are significantly more satisfied (U = 346; z = -2.771; $p \leq 0.01$) than users receiving common seed song recommendations (Median = 2). In addition to this, we calculate an effect size estimate based on Pearson's product-moment correlation coefficient $r = \frac{|z|}{\sqrt{N}}$ [25]. According to the classification of Cohen [18], the effect size r = 0.33853 we get for our sample data indicates a medium effect for taking the users' situation into consideration for recommendation generation.

If we assume rating to be an interval variable, we are able to perform the independent t-test as the parametric equivalent to the Mann-Whitney-U test. In addition to the interval scale, the t-test features two further requirements: normal distribution and homogeneity of variance. Normal distribution can be assumed because the corresponding null hypothesis is not rejected ac-

[10] Providing personal data was optional.

cording to the Shapiro-Wilk test ($p > 0.05$ for both groups). Homogeneity of variance can be assumed as well because the corresponding null hypothesis is not rejected according to the Levene test ($p > 0.05$). In accordance with the Mann-Whitney-U test, the t-test indicates that the mean rating values of the situation-aware group (Mean = 3.057) are significantly higher ($p \leq 0.01$; increase of 16 %) than those of the control group (Mean = 2.635). In conclusion, we regard the results of the t-test as a confirmation of the results of its non-parametric equivalent.

6.4 Discussion and Limitations

Overall, the evaluation results clearly show the potential of situation-aware recommendations. Although we expected the mean rating value of both groups to be higher, we regard the increase of 16 % in our sample data as a notable improvement in terms of user satisfaction. From follow-up discussions with selected participants, we derived two main reasons for the lower than expected mean rating values. First, the limit of five seed songs set by Spotify impedes a detailed specification of one's music preferences. Many participants stated that they wanted to indicate more seed songs. This is confirmed by the fact that they indicated 19.12 seed songs on average (SD = 4.977), which is an unexpectedly large figure considering the maximum amount of 25 seed songs per participant. And second, participants often selected seed songs that they indeed enjoyed listening to in a particular situation but that are actually not representative of their overall taste of music in that situation.

Furthermore, there are some limitations to discuss. First, the situations that the web survey is based on are predefined by us. However, we constructed the set of exemplary situations based on informal interviews with potential study participants that describe themselves as very interested in music. Second, ensuring that the situations selected by the participants in the initialization phase feature a sufficient diversity may be interpreted as giving the situation-aware group an advantage. However, comparing the two groups would be meaningless if the situations (and consequently, the participants' music preferences in these situations) did not differ to a certain extent. Third, the evaluation is not based on the prototype described in Sect. 5. However, this would have resulted in a considerably lower number of participants due to smartphone requirements (e.g. Android, ANT+) and, more importantly, due to the time and effort needed. Finally, it could be argued that the evaluation should also include comparable context-aware recommender systems in addition to the baseline approach not considering any contextual side information. However, there was no directly comparable (i.e. only differing in the particular aspect of considering the users' contextual situation or not) context-aware recommender system relying on seed recommendations which we could have used.

7 Conclusion

Despite their notable success in numerous areas, one major shortcoming of traditional recommender systems is their inability to adjust to users' short-term preferences resulting from varying situation-specific factors. In this work, we introduced the notion of situation awareness for recommender systems, which goes one step further compared to the existing solution approach of context-aware recommender systems. In particular, situation-aware recommender systems blend different kinds of contextual side information together to actually comprehend the users' current situation instead of only being aware of the contextual conditions of the situation. Our evaluation clearly showed the potential of the idea of taking the users' situation into consideration for recommendation generation.

Even though we ensured the comprehensiveness of our proposal by covering all phases of the design science research methodology, the concept of situation-aware recommender systems is still in its infancy. Thus, there are several opportunities for future work. The major ideas we see for extending the fundamental concepts presented in this paper are the following:

Optimization of contextual dimensions: In the course of designing our music recommender system, we asked potential users to evaluate the influence of different contextual dimensions on their music preferences. As opposed to this "relevance assessment", the alternative "relevance detection" conducts statistical analyses on existing rating data to determine which contextual dimensions actually have considerable influence on users' preferences [34]. Since relevance detection requires a large rating data base [34], we had to rely on relevance assessment for initial development but may be able to optimize the contextual dimensions as soon as we have obtained a sufficient amount of usage data.

Adjustment of situations: Once users of our music recommender system have defined a particular situation, they can change the associated seed songs but the underlying contextual conditions are fixed. Similarly to the consideration of contextual dimensions, it may be meaningful to automatically optimize the situations as soon as there are enough usage data. The adjustment of situations could be done with the help of well-established classification or clustering methods, for example. If these techniques proved themselves valuable, they might even entirely replace the currently used distance measure.

Consideration of input data quality: So far, we assumed perfect certainty for all situation-specific input data. While the conditions of some contextual dimensions can safely be assumed to be correct (e.g. time of day), the conditions of others are more likely to be incorrect (e.g. weather). For movement, it is already possible to request a confidence value for the determined contextual condition from the Google API. Similar values would be needed for the other contextual dimensions as well.

Extensions regarding empirical studies: Even though the evaluation yielded both promising and reliable results, it may be worth conducting follow-up studies which consider several lessons learned in the meantime. For example, more objective metrics for determining the participants' satisfaction with the provided recommendations could be considered. Moreover, more advanced participant attributes (e.g. mainstreaminess [39]) in addition to the participants' interest in music may lead to further valuable insights.

Generalizability to other application domains: Choosing music recommendation for the more detailed discussion of the otherwise generic design steps for the design of situation-aware recommender systems was perfectly suitable and yielded promising results. Nevertheless, applying the generic steps to other application domains would help confirm the generalizability of our findings. Moreover, this could bring some interesting differences between application domains to light, such as diverging practical relevances of specific contextual dimensions.

Acknowledgements The research leading to these results was supported by the "Bavarian State Ministry of Education, Science and the Arts" as part of the FORSEC research association. The authors would like to thank Kilian Müller for his support regarding the implementation of the software prototype as well as the web surveys used in this work. Moreover, the authors would like to thank the anonymous reviewers for their helpful comments.

Conflict of Interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

Data Availability

The datasets during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

1. Adams, M.J., Tenney, Y.J., Pew, R.W.: Situation Awareness and the Cognitive Management of Complex Systems. Human Factors: The Journal of the Human Factors and Ergonomics Society **37**(1) (1995)
2. Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-aware Recommender Systems. AI Magazine **32**(3), 67–80 (2011)
3. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering **17**(6), 734–749 (2005)
4. Aggarwal, C.C.: Context-Sensitive Recommender Systems. In: C.C. Aggarwal (ed.) Recommender Systems: The Textbook, pp. 255–281. Springer International Publishing, Cham (2016)
5. Asoh, H., Motomura, Y., Ono, C.: An Analysis of Differences Between Preferences in Real and Supposed Contexts. In: Proc. of the 2nd Workshop on Context-aware Recommender Systems (CARS) (2010)

6. Balabanović, M., Shoham, Y.: Fab: Content-based, Collaborative Recommendation. Communications of the ACM **40**(3), 66–72 (1997)
7. Baltrunas, L.: Context-aware Collaborative Filtering Recommender Systems. Phd thesis, Free University of Bozen-Bolzano, Bozen, Italy (2011)
8. Baltrunas, L., Amatriain, X.: Towards Time-Dependant Recommendation Based on Implicit Feedback. In: Proc. of the 1st Workshop on Context-aware Recommender Systems (CARS) (2009)
9. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context Relevance Assessment and Exploitation in Mobile Recommender Systems. Personal and Ubiquitous Computing **16**(5), 507–526 (2012)
10. Bauer, C., Novotny, A.: A Consolidated View of Context for Intelligent Systems. Journal of Ambient Intelligence and Smart Environments **9**(4), 377–393 (2017)
11. Bazire, M., Brézillon, P.: Understanding Context Before Using It. In: Proc. of the International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT), pp. 29–40 (2005)
12. Burke, R.: Knowledge-based Recommender Systems. In: A. Kent (ed.) Encyclopedia of Library and Information Science: Volume 69 - Supplement 32, pp. 180–200. Marcel Dekker, Inc., New York, NY, USA and Basel, Schweiz (2000)
13. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction **12**(4), 331–370 (2002)
14. Burke, R., Felfernig, A., Göker, M.H.: Recommender Systems: An Overview. AI Magazine **32**(3), 13–18 (2011)
15. Celma, O., Lamere, P.: If You Like Radiohead, You Might Like This Article. AI Magazine **32**(3), 57–66 (2011)
16. Chen, C.M., Tsai, M.F., Lin, Y.C., Yang, Y.H.: Query-based Music Recommendations via Preference Embedding. In: Proc. of the 10th ACM Conference on Recommender Systems (RecSys), pp. 79–82 (2016)
17. Clapham, C., Nicholson, J.: The Concise Oxford Dictionary of Mathematics, 5. edn. Oxford Univ. Press, Oxford (2014)
18. Cohen, J.: A Power Primer. Psychological Bulletin **112**(1), 155–159 (1992)
19. Dey, A.K.: Understanding and Using Context. Personal and Ubiquitous Computing **5**(1), 4–7 (2001)
20. Dourish, P.: What We Talk About When We Talk About Context. Personal and Ubiquitous Computing **8**(1), 19–30 (2004)
21. Endsley, M.R.: Situation Awareness Global Assessment Technique (SAGAT). In: Proc. of the IEEE 1988 National Aerospace and Electronics Conference (NAECON), pp. 789–795 (1988)
22. Endsley, M.R.: A Survey of Situation Awareness Requirements in Air-to-Air Combat Fighters. The International Journal of Aviation Psychology **3**(2), 157–168 (1993)
23. Endsley, M.R.: Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors: The Journal of the Human Factors and Ergonomics Society **37**(1), 32–64 (1995)
24. Feng, Y.H., Teng, T.H., Tan, A.H.: Modelling Situation Awareness for Context-aware Decision Support. Expert Systems with Applications **36**(1), 455–463 (2009)
25. Fritz, C.O., Morris, P.E., Richler, J.J.: Effect Size Estimates: Current Use, Calculations, and Interpretation. Journal of Experimental Psychology: General **141**(1), 2–18 (2012)
26. Herbig, P.A., Kramer, H.: The Effect of Information Overload on the Innovation Choice Process. Journal of Consumer Marketing **11**(2), 45–54 (1994)
27. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly **28**(1), 75–105 (2004)
28. Kaminskas, M., Ricci, F.: Contextual Music Information Retrieval and Recommendation: State of the Art and Challenges. Computer Science Review **6**(2-3), 89–119 (2012)
29. Kells, L.M., Kern, W.F.: Plane and Spherical Trigonometry (1935)
30. Levy, Y., Ellis, T.J.: A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research. Informing Science: The International Journal of an Emerging Transdiscipline **9**, 181–212 (2006)
31. Liu, N.H.: Comparison of Content-based Music Recommendation Using Different Distance Estimation Methods. Applied Intelligence **38**(2), 160–174 (2013)

32. Melville, P., Mooney, R., Nagarajan, R.: Content-Boosted Collaborative Filtering for Improved Recommendations. In: Proc. of the 18th National Conference on Artificial Intelligence (AAAI), pp. 187–192 (2002)
33. Niu, L., Lu, J., Zhang, G.: Managerial Cognition. In: Kacprzyk et al. (ed.) Cognition-Driven Decision Support for Business Intelligence, *Studies in Computational Intelligence*, vol. 238, pp. 31–37. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
34. Odic, A., Tkalcic, M., Tasic, J.F., Kosir, A.: Relevant Context in a Movie Recommender System: Users' Opinion vs. Statistical Detection. In: Proc. of the 4th ACM RecSys Workshop on Context-aware Recommender Systems (CARS) (2012)
35. Panniello, U., Tuzhilin, A., Gorgoglione, M.: Comparing Context-aware Recommender Systems in Terms of Accuracy and Diversity. User Modeling and User-Adapted Interaction **24**(1-2), 35–65 (2014)
36. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems **24**(3), 45–77 (2007)
37. Salas, E., Prince, C., Baker, D.P., Shrestha, L.: Situation Awareness in Team Performance: Implications for Measurement and Training. Human Factors: The Journal of the Human Factors and Ergonomics Society **37**(1) (1995)
38. Schedl, M.: Ameliorating Music Recommendation: Integrating Music Content, Music Context, and User Context for Improved Music Retrieval and Recommendation. In: Proc. of the International Conference on Advances in Mobile Computing & Multimedia (MoMM), pp. 3–9 (2013)
39. Schedl, M., Bauer, C.: An Analysis of Global and Regional Mainstreaminess for Personalized Music Recommender Systems. Journal of Mobile Multimedia **14**(1), 95–112 (2018)
40. Stanners, M., French, H.T.: An Empirical Study of the Relationship Between Situation Awareness and Decision Making. Tech. Rep. DSTO-TR-1687, Defence Science and Technology Organisation, Department of Defence, Australian Government (2005)
41. Stanton, N.A., Stewart, R., Harris, D., Houghton, R.J., Baber, C., McMaster, R., Salmon, P., Hoyle, G., Walker, G., Young, M.S., Linsell, M., Dymott, R., Green, D.: Distributed Situation Awareness in Dynamic Systems: Theoretical Development and Application of an Ergonomics Methodology. Ergonomics **49**(12-13), 1288–1311 (2006)
42. Tzanetakis, G., Cook, P.: Musical Genre Classification of Audio Signal. IEEE Transactions on Speech and Audio Processing **10**(5), 293–302 (2002)
43. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly **26**(2), xiii–xxiii (2002)
44. Zheng, Y., Burke, R., Mobasher, B.: Splitting Approaches for Context-aware Recommendation. In: Proc. of the 29th Annual ACM Symposium on Applied Computing (SAC), pp. 274–279 (2014)