# Conceptual Modeling of Context based Recommendation System

Arati R. Deshpande
PAHER University
Research Scholar, Faculty of Computer Engg.
Udaipur

Emmanuel M., PhD
Pune Institute of Computer Technology
Professor, Department of Information Technology
Pune

## ABSTRACT

Recommendation systems suggest the personalized list of items such as products, people and activities to reduce the search in a large amount of available information, by filtering the relevant information which the user will prefer to explore more. Currently, many web applications include the recommendation systems to enhance the user's experience and trust and thereby enable the service provider to retain the customers. Context based recommendation systems are aimed at providing relevant recommendations to users using the context as additional information in computation of recommendation. Context is the information concerning the situation of user interaction with the system along with the information of users and items. The acquisition, storage and representation of context are the requirements of the context based recommendation system. The context modeling deals with the representation of context in a form which can be suitable for storage and access to compute the recommendation. A conceptual model of context and recommendation system using the graphical object oriented model is proposed in this paper. It is converted into a relational database model for storage and access. This model can be used to implement the design of a context based recommendation system in many domains of applications.

## General Terms

Information Retrieval, Recommendation System, Context based system

## Keywords

Context aware recommendation, Context modeling, Context storage, Object model, Relational model

## 1. INTRODUCTION

Recommendation systems (RS) are part of many web applications like e-commerce, e-learning systems, social networking and also mobile applications. The users' history of interaction with the system in terms of explicit ratings or implicit actions and/or the contents of items are used to compute the list of recommended items to a user in these systems. The recommendation approaches are collaborative filtering, content based filtering and hybrid filtering [1]. The content based filtering uses the similarity of items to already rated items of the user and recommends the items that are most similar. The collaborative filtering uses the similarity of current user with the users, who have rated the same items as the current user and recommends the items that are rated by most similar users. The hybrid filtering uses both content and collaborative methods for the recommendation. The recommendation problem is defined as predicting the rating of the user for items not seen by the user yet and giving the top N predictions as the recommended list of items [2]. A matrix of ratings given by a set of users for a set of items forms the input. The rating matrix $R$ is defined as $U \times I$ with the dimensions as users and items. A function $r(u, i)$ defines the prediction of rating of user *u* on an item *i* which is not seen by the user [3].

Context based recommendation systems improve the quality of recommendation by incorporating the context information in addition to the information of users and items. Context based methods use the context as the third dimension in addition to user and item dimension of recommendation. The example of context can be the time of rating or location of the user or intent of purchase. The attributes of user like age or interest and attributes of items like price, discount can also be taken as context in some systems. The context can be incorporated in recommendation with pre filtering or post filtering or contextual modeling approaches. The pre filtering method used in [4] filters the user and item matrix according to the current context and uses it to reduce the problem to two dimensional problem of recommendation as depicted in Fig 1. The pre filtering approach has the advantage of using the existing recommendation algorithm. The post filtering method applied in [5] filters the recommended items generated by two dimension recommendation method, according to the current context of user as illustrated in Fig 2. The contextual modeling method in [6] uses the context in the recommendation process itself, which is shown in Fig 3. As the context is used as the information in all the three approaches, the context has to be accessed and stored for the recommendation.

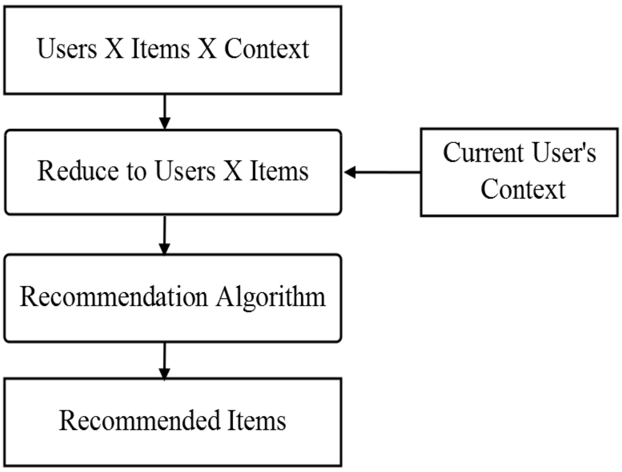

**Fig 1: Pre Filtering Approach**

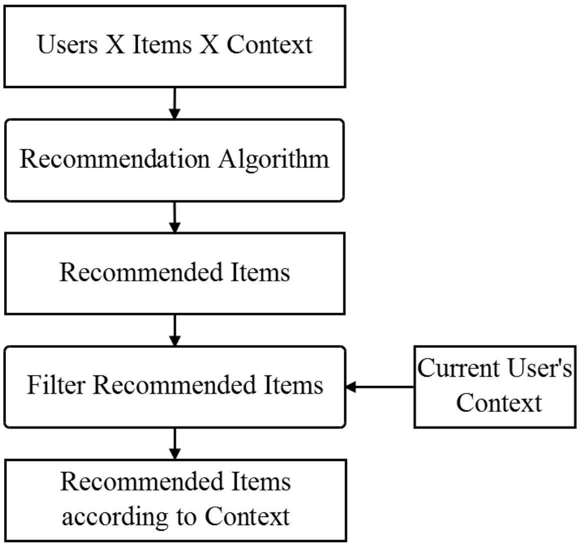

**Fig 2: Post Filtering Approach**

The context acquired from the interaction of users with the system may not be in the form required by the recommendation process. The information can be the time stamp which has to be converted into day, date and time or season for the recommendation. This information of context is represented as a hierarchical form in [7].

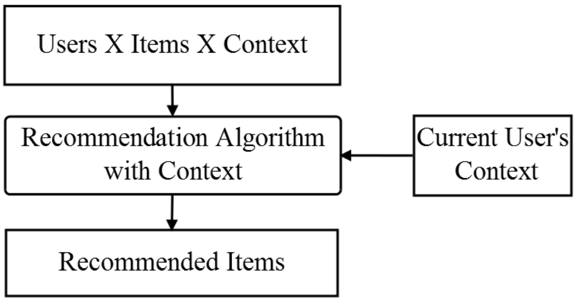

**Fig 3: Contextual Modeling Approach**

The modeling of context can be done with different methods which are taken from pervasive and ubiquitous computing. The proposed model uses the object oriented model, to model the context and extends it with the recommendation system and user interaction. This object model can be converted into a relational database model for storage and access of context.

## 2. CONTEXT IN COMPUTING

Context is defined in many ways by researchers. It plays an important role in the field of philosophy, human behaviour and psychology. This context is also used in a software system to adapt the software functionality according to the context. In context aware computing related to ubiquitous computing and pervasive computing, the context is defined by many researchers with the environment in which the computing takes place.

### 2.1 Definition of Context

The context is defined in Webster's dictionary as interrelated conditions in which something exists or occurs. In a computer system, it is the value of the device, use or user. In [8] context is defined as three important aspects: where you are, who you are with and what resources are nearby. Accordingly context encompasses more than just user location, like noise, network connectivity communication and social situation. Context aware systems adapt according to the context and also respond to the changes in context. Context is defined as the user's location, description of people and objects nearby and changes to those objects in [9].

Context is the circumstance or situation in which a computing task takes place [10]. The context information is characterized by different properties such as temporal, imperfect, alternative representation and high interrelation. As the context is temporal it can be static and dynamic. The static context information is the information that will not change which is invariant like date of birth. This can be obtained by users. The dynamic context is the context information that changes frequently like location of a user or time. This is obtained by sensors. Context is imperfect in the sense that the context information is not always correct and may be inconsistent or incomplete. The context information may be out of date or can contain faulty information. The same context has different representations in different levels of abstraction and has relationships between alternate representations. The context has a relationship or dependency between the properties, people, devices, communication and activities. Context data is defined as a piece of information that is composed of at least one piece of sensor data and one piece of meta information used to interpret the sensor information [11]. In this context data is modelled as a multistage model with raw sensor data at the lowest level and higher order context at upper level, which can be derived from another context or raw sensor data or meta information. Context class is defined as the type of context data with features and attributes which can have instances with values for attributes. Generic attributes are defined for the context which is validity, relevance, reliability and context history.

Many researchers define context as applicable to specific systems, which make use of information like location, temperature and user. A widely accepted definition was provided in [12] as "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between the user and an application, including the user and applications themselves". The context types are categorized as Activity, Identity, Location and Time.

### 2.2 Views of Context

The two views of context, namely a representational view and an interactional view is proposed in [13]. In interactional view context is considered as the part of the activity. Context is related to activity and activity also decides the context. For example, when a user searches for a book, the context of the user's age and intent of purchase influences the search activity for type of book and also searching a particular category of books can decide the context like age or purchasing for children. The context and its values are defined by the activity, dynamically instead of statically defining the context before the processing of information or activity.

In representational view the context is a form of information, which can be extracted before processing, which is stable and it is separated from the activity which is happening. The contexts are identified before the system design. The design and implementation of the system are done by considering the identified contexts. For example, location of user, age of a person or time can be the contexts identified in the tourist place recommendation. A system is context aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task [14]. For the design of intelligent systems, the designers will have to decide the relevant context before implementation which is an important task. This representational view of context is used in many recommendation systems.

## 2.3 Context Life Cycle

The management of context is an integral part of context aware applications. It manages the context, from acquisition to computing [15]. A context life cycle consists of four phases namely Context acquisition, Context modeling, Context reasoning and Context dissemination. Context acquisition deals with the extraction of context from the source of context information like sensors, system and user activity which may not be in processed form. The context acquired will be processed and stored for computation by context aware system. This requires the context to be represented in different forms of abstraction which is modelled using context modeling techniques. Context reasoning infers the information of context to decide the change to user and or context and also derives the high level contexts to model the real world context. Context dissemination gives the access to context by applications in an efficient way to process according to application requirements.

# 3. CONTEXT MODELING

The context is modelled using different techniques to be stored for processing [16], [17]. The Key-Value models store context as key value pairs like environment variables. Markup Scheme models use markup languages like XML. Graphical models apply the Unified Modeling Language (UML) or Object Role Modeling (ORM) and can be used to derive the entity relationship model for relational databases. Object oriented models use objects and classes for context representation. Logic based models use the rules to define the context. Ontology based models use ontologies for context representation and reasoning. The context modeling in recommendation system uses any of the above methods for effective retrieval of context.

## 3.1 Context Modeling in Recommendation System

Many context based recommendation systems model, context information as a representational view. The context information to be used is identified for the application at the design time. The context is stored as an attribute associated with the rating or the interaction of a user with an item. The context is considered as the dimension added to user and item dimensions in a recommendation system [18]. Each type of context like time, location and intent of purchase is considered as a dimension. Each dimension consists of a set of contextual elements which can take values from a particular domain. In a context dimension the contextual elements are related in a hierarchical way. This hierarchy can be stored as a predefined form for each context dimension. One context dimension can have different hierarchical structure as depicted in Fig 4. The selection of variable from the hierarchy for each context dimension and their combination depends on the application.

Ontology is a formal, explicit specification of concepts. The context is modelled as ontology in mobile catering recommendation for user, object and situation representation with restaurant and food as objects in [19]. The context is modelled as an entity or attribute in a conceptual data model depending on contextual dependency in [20]. The context is modelled as a class with types as user, item, system and other context in [21]. In this each context type has variables for each dimension and value pair. This model uses the object oriented concept for modeling the rating of one user on an item which is associated with zero or more of the four context types defined. This is a more generic model for recommendation using a context. This model has not considered all users and hierarchy of context dimension. The proposed model in this paper incorporates the system of recommendation with the actions and hierarchy of context dimension. It is converted into conceptual database model for storage and access of context and recommendation.

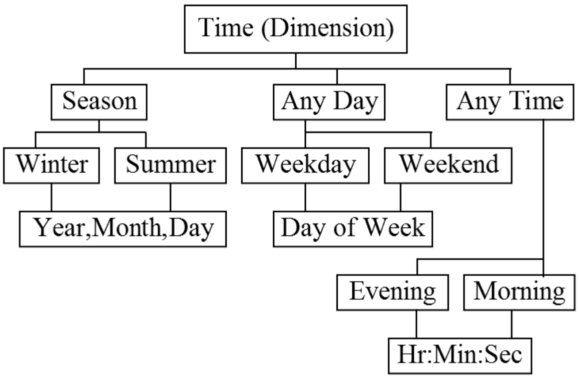

**Fig 4: Time Context Hierarchy**

# 4. CONCEPTUAL MODEL

A conceptual model of context based recommendation is proposed in terms of an object oriented model for recommendation with context and user action. The object oriented model properties which are persistent can be converted into a relational database model for storage and access. The problem of top N recommendation is defined as estimating the ranking or rating of an item by a user $u$ which he/she has not seen and extracting the top N items.

The recommendation system is defined as

$RS = \{U, I, A, C, RU\}$ where $U$ = set of users, $I$ = set of items, $A$ = set of actions, $C$ = set of context dimensions and $RU$ = set of rules.

$U = \{u_1, u_2 \dots u_m\}$, each $u_k$ Defines a user profile with a set of attributes which can be $u_k = \{id, name, dob\}$, for attributes id, name and date of birth respectively, where 1<= $k$ <= $m$ and $m$ = number of users.

Similarly $I = \{i_1, i_2 \dots i_n\}$, each $i_k$ Defines an item profile with a set of attributes which can be $i_k = \{iid, iname, icat, iprice\}$, for attributes item id, item name, item category and item price respectively, where 1<= $k$ <= $n$ and $n$ = number of items.

$A = \{a_1, a_2 \dots a_p\}$, each $a_k$ Defines the action profile with a set of attributes which can be $a_k = \{aid, name, value, u, i, c_1, c_2 \dots c_d\}$, for attributes action id, action name, action value, user, item and context dimensions $c_1$ to $c_d$ respectively, where $d$=number of context dimensions, 1<= $k$ <= $p$ and $p$ = number of action types.

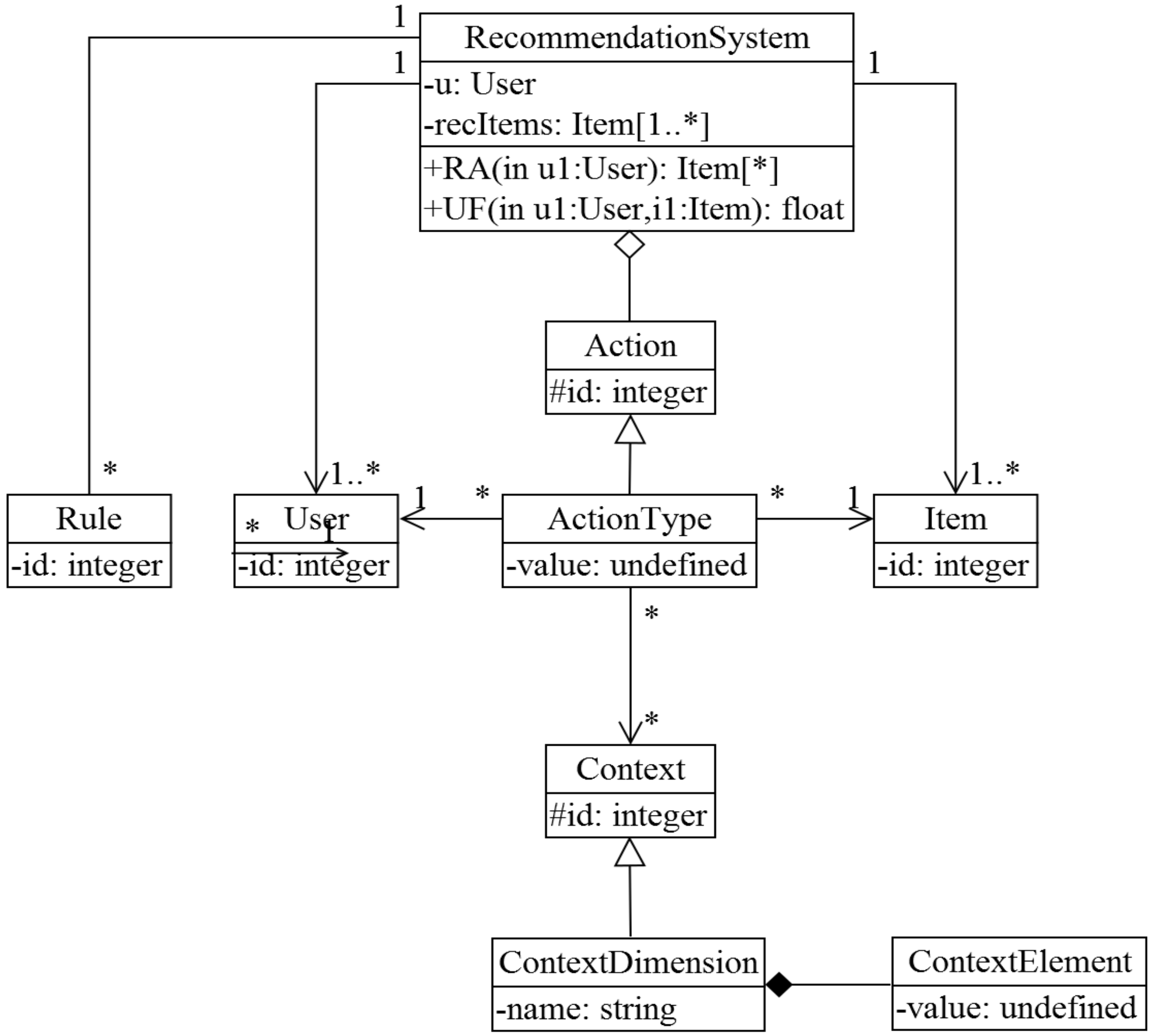

**Fig 5: UML Class Model of Context Based Recommendation System**

$C = \{c_1, c_2 \dots c_d\}$, each $c_k$ is $c_k = \{id, name, elements\}$, for attributes context dimension id, context dimension name and context dimension elements respectively. The elements of a context dimension $c_k$ can have a hierarchical structure.

$elements = \{c_{de1}, c_{de2}, \dots c_{dev}\}$, each $c_{dev}$ defines the element *v* for context dimension *d*. Each $c_{dev}$ takes a value from a predefined set of domain values.

$RU = \{ru_1, ru_2, \dots ru_l\}$, each $ru_k$ defines a rule such that $ru_k = \{c_{de1}, c_{de2}, \dots c_{dex}, a\}$ where $c_{dex}$ is a context element *ex* for a context dimension *d*, *a* is the action type which can be characterized by some value and 1<= *k* <= *l*, *l* = number of rules.

The recommendation problem is defined as predicting the utility *UF* of an item *i* for a user *u* ϵ*U* for each of the item *i* ϵ *I* in current context *C′* where *C′* is the context for current user and *i* is not seen by user *u*. This prediction is the function *P* of *U, I, C, A* and *RU* which are stored for the recommendation system *RS*.

$$UF(u, i, C') = P(U, I, C, A, RU) \quad (1)$$

A recommendation method or algorithm *RA* uses the prediction function and gives the top N items with highest utility function values which are not yet seen by the current user.

The conceptual model of the recommendation is depicted in Fig 5 as a UML class model. The object of class ‘RecommendationSystem’ generates context based recommendation with users, items, actions, contexts and rules. The user profile is stored in an object of class ‘User’ and item profile is stored in an object of class ‘Item’. Each ‘ActionType’ is a subclass of class ‘Action’ which can store information of each interaction of a user with each item corresponding to that action. For example, action types can be rating action, viewing action, buying action or tagging action. Each ‘ActionType’ object is associated with zero or more ‘ContextDimension’ objects which have a set of ‘ContextElement’ objects, having specific values. The Rule objects define the relation of context dimension elements with type of actions. These rules can be used for generating recommendation by the recommendation method *RA*.

## 5. CASE STUDY

The above conceptual recommendation model is applied to a movie recommendation system. In this users rate the movies and also tag the movies which are the two actions of users on movies as items. The context dimension in this case, is the time at which the actions are done by users which has the predefined hierarchical structure.

The persistent classes are modelled as relational tables according to the rules given in [22]. The User, Movie and Rules are the entities mapped to tables. Each action type is mapped to a table as 'ActionRating' and 'ActonTag'. Each type of context which is a context dimension is mapped to a table as 'ContextTime' and 'ContextPlace'. The context elements of a context dimension are mapped to the attributes of that context dimension table.

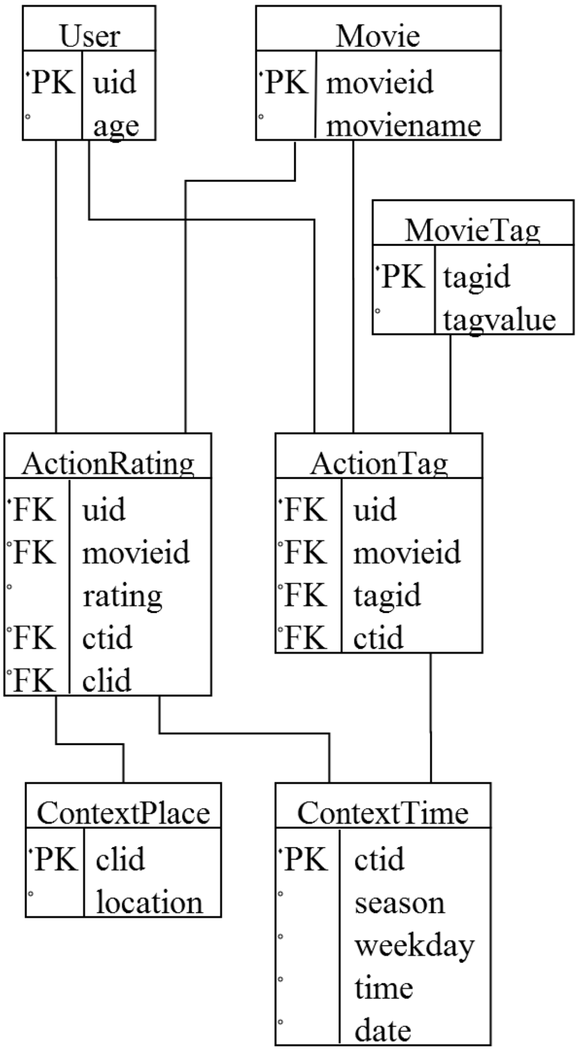

**Fig 6: Relational Tables for Movie RS**

The design of relational tables of the movie recommendation system obtained by mapping the conceptual model of recommendation system is given in Fig 6. The context is stored in tables with the attributes as context elements in the hierarchy. These are identified as required context elements of the recommendation system at design time [23], [24]. These tables store the information which is used by the context based recommendation algorithm to predict the movies for recommendation. The recommendation algorithm can be any existing collaborative algorithm if pre filtering method is used. The instances (tuples) of User, Movie, 'ActionRating' and 'ActionTag' tables are reduced according to context of the current user and rules, to predict the recommendation.

## 6. CONCLUSION

Context based recommendation systems generate the personalized recommendation using the current context of the user, item or interaction of users with the item. The conceptual model of the context based recommendation given in this paper is applied to the design of movie recommendation system. This is a generalized model of context based recommendation which can be extended for any other domain and platform of context based recommendation system like books, products, people or activities. The implementation of the proposed design of context based recommendation to generate more relevant recommendations will be part of future work. The rules can be generated from context and actions using data mining techniques. These rules can be used to improve the quality of the recommendation. The combining of rules and user actions with context to provide recommendation can be explored for combining with any of the three methods of context based recommendation like pre filtering approach.

## 7. REFERENCES

[1] Ricci, F., Rokach, L. and Shapira, B., 2011. Introduction to recommender systems handbook. In Recommender systems handbook (pp. 1-35). springer US.

[2] Bobadilla, J., Ortega, F., Hernando, A. and Gutiérrez, A., 2013. Recommender systems survey. Knowledge-based systems, 46, pp.109-132.

[3] Adomavicius, G. and Tuzhilin, A., 2005. 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), pp.734-749.

[4] Ramirez-Garcia, X. and Garcia-Valdez, M., 2015. A Pre-filtering Based Context-Aware Recommender System using Fuzzy Rules. In Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization (pp.497-505). Springer International Publishing.

[5] Cremonesi, P., Garza, P., Quintarelli, E. and Turrin, R., 2011, October. Top-n recommendations on unpopular items with contextual knowledge. In 2011 Workshop on Context-aware Recommender Systems. Chicago.

[6] Pandey, A.K., Kumar, A. and Rajendran, B., 2013. Contextual Model of Recommending Resources on an Academic Networking Portal. In Proceedings of 3rd International Conference on Computer Science & Information Technology, CCSIT (pp. 421-429).

[7] Adomavicius, G. and Tuzhilin, A., 2015. Context-aware recommender systems. In Recommender systems handbook(pp. 191-226). Springer US.

[8] Schilit, B., Adams, N. and Want, R., 1994, December. Context-aware computing applications. In Mobile Computing Systems and Applications, WMCSA 1994. First Workshop on (pp. 85-90). IEEE.

[9] Schilit, B.N. and Theimer, M.M., 1994. Disseminating active map information to mobile hosts. IEEE network, 8(5), pp.22-32.

[10] Henricksen, K., Indulska, J. and Rakotonirainy, A., 2002. Modeling context information in pervasive computing systems. Pervasive Computing, pp.79-117.

[11] Zimmer, T., 2004, March. Towards a better understanding of context attributes. In Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on (pp. 23-27). IEEE.

[12] Abowd, G., Dey, A., Brown, P., Davies, N., Smith, M. and Steggles, P., 1999. Towards a better understanding of context and context-awareness. In Handheld and ubiquitous computing (pp.304-307). Springer Berlin /Heidelberg.

[13] Dourish, P., 2004. What we talk about when we talk about context. Personal and ubiquitous computing, 8(1), pp.19-30.

[14] Bauer, C. and Dey, A.K., 2016. Considering context in the design of intelligent systems: Current practices and suggestions for improvement. Journal of Systems and Software, 112, pp.26-47.

[15] Mcheick, H., 2014. Modeling Context Aware Features for Pervasive Computing. Procedia Computer Science, 37, pp.135-142.

[16] Strang, T. and Linnhoff-Popien, C., 2004, September. A context modeling survey. In Workshop Proceedings.

[17] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A. and Riboni, D., 2010. A survey of context modeling and reasoning techniques. Pervasive and Mobile Computing, 6(2), pp.161-180.

[18] Adomavicius, G., Sankaranarayanan, R., Sen, S. and Tuzhilin, A., 2005. Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions on Information Systems (TOIS), 23(1), pp.103-145.

[19] Ai, D.X., Zuo, H. and Yang, J., 2013. Ontology-Based Context Modeling for Mobile Catering Recommendation. In *Advanced Materials Research* (Vol. 662, pp. 953-956). Trans Tech Publications.

[20] Choi, D., Kim, N. and Hung, D.T., 2012. Conceptual data modeling for realizing context-aware services. Expert Systems with Applications, 39(3), pp.3022-3030.

[21] Mettouris, C. and Papadopoulos, G.A., 2013. Contextual modeling in context-aware recommender systems: a generic approach. In Web Information Systems Engineering–WISE 2011 and 2012 Workshops (pp. 41-52). Springer, Berlin, Heidelberg.

[22] Ambler, S.W., 2000. Mapping objects to relational databases: What you need to know and why. Ronin International.

[23] Lombardi, S., Gorgoglione, M. and Panniello, U., 2013. The effect of context on misclassification costs in e-commerce applications. Expert Systems with Applications, 40(13), pp.5219-5227.

[24] Stefanidis, K., Pitoura, E. and Vassiliadis, P., 2005. On supporting context-aware preferences in relational database systems. In International Workshop on Managing Context Information in Mobile and Pervasive Environments.