Computing and Informatics, Vol. 37, 2018, 1149–1183, doi: 10.4149/cai_2018_5_1149

GUMCARS: GENERAL USER MODEL FOR CONTEXT-AWARE RECOMMENDER SYSTEMS

Sergio Inzunza, Reyes Juárez-Ramírez
Samantha Jiménez, Guillermo Licea

Universidad Autónoma de Baja California
Calzada Universidad 14418, Tijuana BC, 22390 México
e-mail: {sinzunza, reyesjua, sjimenez, glicea}@uabc.edu.mx

Abstract. Context-Aware Recommender Systems (CARS) are extensions of traditional recommender systems that use information about the context of the user to improve the recommendation accuracy. Whatever the specific algorithm exploited by the CARS, it can provide high-quality recommendations only after having modeled the user and context aspects. Despite the importance of the data models in CARS, nowadays there is a lack of models and tools to support the modeling and management of the data when developing a new CARS, leaving designers, developers and researchers the work of creating their own models, which can be a hard and time-consuming labor, and often resulting in overspecialized or incomplete models. In this paper, we describe GUMCARS – a General User Model for Context-Aware Recommender Systems, where the main goal is to help designers and researchers when creating a CARS by providing an extensive set of User, Context and Item aspects that covers the information needed by different recommendation domains. To validate GUMCARS, two experiments are performed; first, the completeness and generality of the model are evaluated showing encouraging results as the proposal was able to support most of the information loaded from real-world datasets. Then the structural correctness of the model is assessed, the obtained results strongly suggest that the model is correctly constructed according to Object-Oriented design paradigm.

Keywords: User-model, context-aware, recommender-systems, cars, GUMCARS

Mathematics Subject Classification 2010: 68-N01, 68-N30, 68-T30

1 INTRODUCTION

A recent research trend in Recommender Systems (RS) is the inclusion of context information in the recommendation algorithms, as contextual information has been proved to help increasing the prediction accuracy of recommender systems [24, 86]. This type of RS are known as Context-Aware Recommender System (CARS). Contextual information plays an important role in CARS, as user behavior is affected by the user current context [16], e.g., time, location, mood, and weather. CARS are based on the idea that similar users, in similar contexts, like similar items, and that user's preferences change according to his/her contextual situation [39]. Therefore, in CARS, as in most personalization systems, a user model is an essential component used to store the information about the user, his/her context, and interactions with the systems, which can later be used to adapt and personalize the system in order to improve the experience of the user in future interactions [49, 37].

Despite the advances in context information management [73], the design and development of context-aware systems, such as CARS, remain significantly more challenging than traditional systems, especially without supporting tools that facilitate this process, as to add context-aware capabilities to a software system brings design and development overhead inherited from the complexity of managing (acquiring, aggregating, storing) the user and context information [80].

Nowadays there is a lack of tools that support and facilitate the development of CARS [44], especially infrastructures for user and context information management [6, 49], which leaves developers and researchers the work of designing and implementing their own context-aware models to manage the information needed by recommendation algorithms, based on their knowledge and with no model to use as a reference, often resulting in overspecialized inefficient or incomplete models [80].

Even when some proposals of user model (e.g. [40]) and context model (like [89]) exist in the literature, they are not designed specifically for recommender systems, which means that such models do not consider key information for recommender systems, like items' data or ratings history. Also, some proposals are too abstract, designed at ontological level (e.g. [94]); others present the information categories, but not the specific attributes of user or context (e.g. [89]), and getting them to implementation implies a lot of work.

Summarizing, a context-aware user model that can be used in a range of CARS domains, which could improve CARS designer and developer efficiency, providing a data structure to manage the information of the context, user and his/her interaction with the system has not yet been proposed in the literature.

In this paper, we propose a General User Model for Context-Aware Recommender Systems (GUMCARS), that addresses the above-described problem providing a generic model for CARS information that can be used by CARS designers and developers to support the information for their CARS implementation, or as reference model that can be easily adapted to specific project needs.

GUMCARS proposes a taxonomic categorization of the information used by CARS, and provides an extensive set of User, Context and Item aspects that covers the domain recommendation most commonly found in CARS literature, as well as the relation between these information elements resulted from the interaction of the user with the system. The result is a user model for context-aware recommender system that aims to achieve balance between *Completeness* so it can be used into a CARS systems with minimum modifications, and *Generality* so it can be extended to suit specific project needs.

The main contributions of this paper are:

1. From a *context-aware recommender systems* perspective, this paper presents a taxonomy that organizes the information needed by CARS systems to perform recommendations.
2. From a *user modeling* perspective, GUMCARS presents a general user model for context-aware recommender systems that can be adapted to suit specific needs, or used as a basis for future user model developments.

The remainder of this paper if structured as follows. Section 2 presents the related work. Section 3 presents the proposed taxonomy of CARS information. Section 4 describes a generic user model for CARS, Section 5 presents an experiment performed to assess GUMCARS completeness, and then Section 6 describes another experiment where the structural correctness of the proposal is evaluated. Finally, Section 7 presents the conclusions and future work.

2 RELATED WORK

In order to generate relevant recommendations, CARS needs to model the information of the user, the interaction of the user with the system, and the context where the interaction takes place [53].

A user model is defined as the knowledge about the user, explicitly or implicitly encoded, which is used by the system to improve user interaction [78].

The recent advance in technology that offers "anytime, anywhere, anyone" computing, has enabled software system to acquire more information about the user and his/her surroundings, which introduced the challenge of context-aware user modeling. A user model can be considered context-aware if it can express aspects of the user's contextual situation and helps software systems to adapt their functionality to the context of use [83].

A commonly used definition of context comes from Dey and Abowd [28], they define context as 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 a user and an application, including the user and application themselves. From an *infrastructural* perspective, context provides computing devices with information about their environment as provided by other system components. In order to provide such information, context needs to be classified in different 'types' or 'dimensions' of context, e.g. physical, computational, etc.

Dourish [30] introduces a taxonomy of context, according to which context can be classified into *representational* or *interactional* view. The representational view assumes that the contextual attributes are identifiable and known a priori, and hence, can be captured and used within the context-aware applications. In contrast, the interaction view assumes that the user behavior is induced by an underlying context, but that context itself is not necessarily observable. As the interactional view of context is an approach borrowed from psychology [3] that considers context non-observable, it cannot be used to explicitly store the context information into a computer system. Therefore, this work is based on the representational view of context, as this approach allows to identify context attributes and to model it.

The areas of user modeling and context representation are well-established research topics by themselves. Next we review some of the most related publications on modeling user or context information in software systems.

Heckman [40] introduced the General User Model Ontology (GUMO) for the uniform interpretation of distributed user models. GUMO is intended to represent the information of the user for adaptive systems, the proposal presents a long list of user aspects to be considered in such systems organized into 12 *Basic User Dimensions*. As GUMO is created at an ontological level, it does not present further organization of the user characteristics or any architecture or implementation design that could help CARS developers in their effort of designing a context-aware user model.

Kaklanis et al. [52] proposed another user model in their efforts of the Virtual User Modeling and Standardization 'VUMS'. Their proposal aims to model user preferences for graphical interfaces, as well as some cognitive and physical human abilities. As the proposal is aimed at modeling people with disabilities and elderly people, the user aspects are limited to such interests. Therefore, this model would be of little help when designing a context-aware user model for CARS.

Jawaheer et al. [49] classify the different types of user feedback as a source of information for user modeling in recommender systems. However, they do not describe what user or context information must be included in such a model. They conclude the work with a series of future research challenges, including the need for a unified user model for recommender systems.

With respect to context modeling, Zimmermann et al. [98] defined five fundamental categories for context information: Activity, Time, Relations, Individuality, and Location. They describe such categories as the design space of context models for context-aware application to build upon. Later on, Verbert [89], through an extensive survey on context-aware systems, increases the number of categories for context information up to 8, namely, Computing, Location, Time, Physical Conditions, Activity, Resource, Social Relations, and User. Unlike Zimmermann, Verbert presented a series of subcategories for each context category, e.g., Computing is categorized into *software*, *hardware*, and *network*. However, a low-level detail of what information about context designers and developers should consider when creating their own context-aware user model has not been described by either proposal.

As the need for tools that help in the design and development of CARS is known in the literature, some proposals have emerged proposing tools for that matter.

Mettouris and Papadopoulos [61] present a tool designed to facilitate the development of reusable user models for CARS. The researchers publish the proposal as a web application, where developers can describe the context aspect that will be included in their model design. Even when their works [61] help developers as a tool to enlist context aspects and the possible values of each aspect, the work of identifying the aspects that will be included in the model is left to designers and developers. This tool does not allow to specify the data type of each aspect or relations between aspects, neither the proposal uses a formal or semi-formal notation (such as UML or XML) to help to take their list of aspects closer to a model design. Mettouris proposals [61] differ from GUMCARS proposal as follows: their tool is intended as a place to register, test and share a list of context aspects that later can be used to create a model, and our proposal is closer to the design and implementation phases as it presents the conceptual design and the UML model of a context-aware user model.

3 CARS INFORMATION TAXONOMY

In a relatively short time of CARS existence, a lot of different proposals have been created [74], either to test new algorithms or to measure how including certain information can help the recommendation process to generate better results. In most cases, each proposal arbitrary selects which information to take into account.

In order to create a general user model for CARS, and before the identification of what attributes will be included in our general model, we needed to define a classification of the concepts involved, that can be used by the model as a basis to structure the information it contains.

In this section, we describe a taxonomy proposal of the information that CARS use to generate context-based predictions. This taxonomy represents one of the contributions of our work, as it organizes the high-level information of the user, context and items that CARS need to work with, and, to the best of our knowledge, is the first taxonomy for CARS information.

The taxonomy comprises four categories of information:

1. User information,
2. Context information,
3. Activity information, and
4. Item information.

Next, each category and its corresponding sub-categories are described.

3.1 User Information

This top level category represents all the information of the user that describes him/her as a person, as such information is heterogeneous, and based on [40], we further organize the user category into eight subcategories: *Physiology*, *Contact information*, *Personality*, *Emotion*, *Role*, *Interest and Preference*, *Demographic*, *Mental*.

Compared to Heckman's [40] list of 12 dimensions of user information (contact information, demographics, ability and proficiency, personality, characteristics, emotional state, physiological state, mental state, motion, role, nutrition, and facial expression); we do not consider a category for user's characteristics as we consider (supported by Heckman's [40] itself), that there is no clear separation between personality traits and characteristics; we propose to support both information in the *Personality* subcategory. Heckman uses the ability and proficiency dimension to support information about abilities and disabilities of the user, such information is supported in the *Physiological* subcategory of our taxonomy. Heckman uses the motion user dimension to represent whether the user is *walking*, *sitting*, *goingUpStairs*, etc. We consider such information to be part of the activity of the user, and is considered in the *Activity* information category (described in Section 3.3). The last dimensions we do not include in our taxonomy are *Nutrition* and *Facial expressions*, as we considered them very specific, this can increase the computational complexity of the model and most important, will make the model more cumbersome and harder to understand.

We also include a category called *Interest and Preference*, as CARS literature expresses that end users will consider useful a recommendation only if it fits in his/her interests and preferences [74].

Next, the proposed organization of the user information category is depicted in Figure 1, then each subcategory is briefly described.

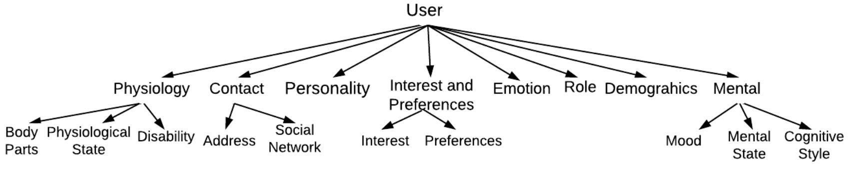

Figure 1. Proposed taxonomy for CARS user information

Physiology. Physiological aspects represent the information about user's body and its functionality. This information is further subdivided into:

Body Parts: This represents information about parts of the human body. The body parts subcategory is subdivided into *Head*, *Hand*, *Foot*. And could the list of body parts of humans as described in [20].

Physiological State: Generic class to store information about physiological states, whose attributes are *name*, *level* and *isNormal*, these attributes can be used to represent states like respiration, blood pressure, heart rate, etc.

Disability: Superclass for specific disabilities classes. As designing a class for all the existing disabilities is outside the scope of this work, the model currently included only *ColorBlindness* and *Musculoskeletal* disability classes as a probe of concept, specific disabilities can be added as needed by specific projects.

Contact Information. For CARS, it is important to know whom is the user interacting with in the system, in order to provide tailored recommendations [74]. Contact subcategory refers to user aspects that identify a person (such as *name*, *last name*, and *email*), and the user's address. This information is available in most commercial systems since the user has to create an account and provide his data. Contact information is further subdivided into:

Address: used to represent the data related to the user's address, which could include information such as *house number*, *street name*, and *country*.

Social Network: represents the contact information of the user in social network sites that can be used by recommender systems to infer user preferences from their social activities as done by [7] and [81].

Personality. Describes permanent or very slow changing patterns that are associated with a person [41]. CARS can use the personality information to decide on what items will fit better to the user, for example, [14] uses the user personality as an important factor in their travel CARS. Certainly, the personality subcategory can be subdivided with the list of personality traits, but as there are several personality models in psychology like Big Five [50] and Cattell's traits of personality [56], we opt to include the basic elements in the taxonomy to support all the personality models, and not to stick to any particular model (as described in Section 4.2.1).

Emotion. Emotions are subjective human experiences [37]. Even when emotion information may look similar to a personality, there are different terms of duration of its contained information. Emotions tend to be closely associated with a specific event (context), and have a short duration of minutes up to an hour [40], while personality reflects long-term user characteristics.

Role. This user subcategory represents information about the role the user is playing at a certain moment. According to [43], the user can take several roles and frequently change between them, for example, a user can visit a town as tourist or businessperson and the CARS should be able to recommend different places to visit depending on such a role.

Demographic. This subcategory represent user's demographical information such as *age* or *family status*. This information can be used by CARS to improve the recommendations, for example [27] uses demographical information, such as *gender* and *socio-economic* information in a context-aware music recommendation system. Demographical information along with interest and preference can be used by CARS to improve the interaction of the user with the system, using such information to adapt the user interface accordingly as presented in [56].

Mental. Used to describe the user's state of mind, this subcategory is subdivided into *mood*, *mental state* and *cognitive style* subcategories that can be used by CARS to provide more tailored recommendations to users. For example [7] uses the user *mental stress*, while [22] uses the user mood in their CARS.

Interest and Preference. This user information category allows to explicitly store user preferences for, or interest in some type of items. In a CARS presented in [95], the interest of the user is used to recommend places and events to visit accordingly, a user with music interest will be recommended to attend concerts, while a user with shopping interest will be recommended to visit near shopping malls. Similarly the CARS of [95] also implicitly collects the preferences of the users, which help the system to decide what specific type of place to recommend, in the music-lover user example, the system could use the preferences of the user to decide whether recommend country, rock or classical events. This user information category is further subdivided into:

Interest: Represents the interest of the user in certain topics or items.

Preference: Represents the preference of the user for certain items in relation with a certain context.

3.2 Contextual Information

This category represents information about the environment that surrounds the user, as well as other information about the items or the recommendation system itself that can be used to characterize the situation of the elements considered in the CARS. We organize the context information into the following six subcategories: *Computing*, *Location*, *Time*, *Physical condition*, *Resource* and *Social relation*.

Compared to the organization of contextual information proposed in [89], that describes a list of 5 types of context like *Computing*, *Location*, *Time*, *Physical Condition*, *Activity*, *Resource*, *User*, and *Social Relations*; we do not consider a subcategory for the activity information, as we consider such information in a category at a higher level. We do not consider the user information to be part of the context category, as we have a more detailed categorization of the user information, and it is supported in a top-level category of the proposed taxonomy. The similarities and differences between the matching types of context from [89] and the proposed taxonomy are discussed along with the description of each subcategory.

Our taxonomy supports other proposals of context information organization, like the one presented in [11] that argues that is important to include interactions between the environment (supported in *Computing* and *Physical Conditions* subcategories), the user (supported in *User* category), their tasks *Activity*), and other users (*Social Relation*) as part of the contextual information.

Next, the proposed taxonomy for context information category is depicted in Figure 2, and then each subcategory is briefly described.

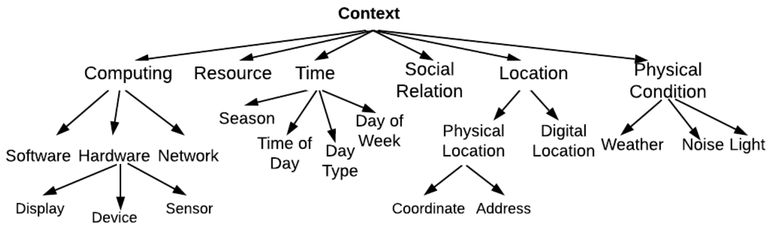

Figure 2. Proposed taxonomy for contextual information user by CARS

Computing. This subcategory represents the information that describes computational elements. Following [89], this subcategory is further divided into:

Software: This sub-subcategory describes the characteristics of software systems that may be available to the user under a certain context, or software systems that the user is interacting with at certain moment. Let us consider a CARS that is recommending persons to a friend on different social networks, for such a system, the information about what applications the user have installed on their smart-phone may be very useful.

Hardware: Comprises information about the hardware as a physical item, such as *Vendor* and *Model*. We follow the Composite Capabilities/Preference Profile (CC/PP) [93] to further organize the hardware sub-subcategory into *Sensors*, *Display*, and *Device* that represent all the computing devices (like tablet, laptop and smart-phone) and include information of the minimum components of the computer device like *RAM*, *Processor* and *Storage*.

Network: Information related to properties of the networks that are being used by computing devices. As for uses of *Network* information, CARS can take into account if the user's mobile device is currently on WiFi connectivity or via the cellular network in order to recommend High Definition or Standard Definition videos.

Resource. The resource information models relevant characteristics that describe elements and the services that items or places provide, for example, if a restaurant has wheelchair ramps, or if a hotel has transportation to and from the airport. The resource sub-subcategory of context can also be used to describe digital resources that are relevant to users, for example, a learning material can be used by a CARS to recommend learning topics to students.

Time. Time subcategory represents information about the time of the day, time of the week or time of the year (season), this information allows CARS to record the moment that some actions took place. The use of *time* information in CARS is very common, as its values are easy to gather and have a deep impact on the user decisions [38].

Social Relation. This subcategory of contextual information refers to social associations or connection between the user of the system and other persons that may or may not be part of the CARS. The user social relations can help CARS to tailor items specific to a current user companion, or to infer user preferences based on the knowledge the system has about other users related to him/her [79].

Location. In the proposed taxonomy, the location subcategory refers to information that relates a user or an item with a physical or a digital location. Location information is used by CARS to recommend the user items like restaurants or movie theaters near him/her. Location is so often used in CARS, that there is a whole research topic called location-aware recommender systems [32].

Physical Location: Comprises information used to identify a specific point in the world, using the coordinates (latitude and longitude) or an address.

Digital Location: Refers to information of where a digital resource (like a web page, video or song) can be located on the Internet. A common example of a digital locator is a URL (Uniform Resource Locator) that specifies the location of a computer in a network.

Physical Condition. Describes the environmental conditions where user or items are situated at a certain point in time. Physical condition describes physical conditions external to user or items like the level of crowdedness and the level of traffic. This subcategory is further subdivided in:

Weather: Represents weather information that CARS can use to better tailor suggestions, for example, the weather is important when recommending places to visit as recommending an outside place with a rainy weather may not be well perceived by users [26].

Noise: Refers to information about the level of noise at a certain place, at a given time.

Light: Used to represent the light level as well as the light source of the physical environment.

3.3 Activity Information

This information category represents the activity performed by the users of the system and relates such an activity to the user and the context.

The activity of the user can help CARS perform more accurate recommendations, for example, Spotify, a music service and recommender system, identifies when the user is running or biking and plays inspirational music to keep them going.

Unlike other proposals like [40, 89] that consider the activity information as part of the contextual information, we create a category for such information at the same level of *User* or *Context* information.

3.4 Item Information

This category represents the information about the items of the CARS catalog like *Video*, *Book* and *Audio*. The importance of items information for CARS is paramount as items along with the transaction log are the base information for the CARS algorithms [74, 4].

4 THE GENERAL USER MODEL FOR CONTEXT-AWARE RECOMMENDER SYSTEMS

This section introduces GUMCARS, a General User Model for Context-Aware Recommender Systems that along with the information taxonomy, represents the main contributions of this paper.

The goal of GUMCARS is to structure the user, context, and items aspects that can be used by CARS systems either as the input for recommendation algorithms or by the system itself to improve interaction with the user. GUMCARS can be used by researchers as a basis for future model developments, as it contains a holistic view of the information used by the research community to generate context-based recommendations, by designers of CARS as the model can serve as a guide when selecting the information aspects to create their own model, and by CARS developers as the model is expressed not only from a conceptual perspective (the taxonomy), but also is presented from a software architectural point of view, and is ready to support the information needed by the most common recommendation domains.

Based on Dourish [30] representational view of context, GUMCARS is formed by a finite set of user, context, and item attributes. The methodology used to gather the specific aspects that form GUMCARS is described in the following subsection. Then, the proposed architecture for GUMCARS is presented.

4.1 Gathering of CARS Information Aspects

As GUMCARS intends to structure a finite set of user, context and item aspects that can be used by CARS as input information for the recommendation algorithm. In the previous section the proposed taxonomy for the information organization was described, as the taxonomy is an organization of concepts and does not contain the specific attributes that will support the CARS information, the next step towards the creation of GUMCARS structure was the identification of such specific information aspects, for which a systematic literature review of CARS was performed and presented in [46].

For the Systematic Literature Review (SLR), the Kitchenham [54] methodology was followed, the SLR reviews the user modeling, context-awareness, and CARS literature to respond the following questions:

1. Which user aspects have been used in CARS literature as input information for the recommendation algorithms?

2. Which context aspects have been used in CARS?
3. What items are being recommended by CARS?
4. What aspects about such items are being used by the recommendation algorithms?

The execution of the SLR consisted in defining the search queries, executing such queries in the selected literature sources, gathering all the matching publications and the grooming of the results based on a series of predefined filters that goes from reading the title to reading the entire publications. The results obtained from SLR was an extensive list of aspects (like address, mood, companion, season, battery level, show size, etc.) and aspects values (like partner, family, and friends for the companion aspect) that publications have used in their CARS proposals. For a more detailed description of the followed steps or the obtained results please refer to [46].

4.2 The GUMCARS Model Architecture

This section describes the architectural view of the model, compared with the conceptual view (taxonomy), the architectural view goes deeper in detailing what specific aspects are considered for each information category. The architecture of GUMCARS is designed following the object-oriented (OO) paradigm [12], and the semi-formal language Unified Modeling Language (UML) [13].

GUMCARS architecture uses the same four top-level categories proposed in the taxonomy, which are called (packages) in the UML notation, the sub-categories and sub-subcategories are *classes*, and the aspects are *class attributes*.

The rest of this section describes the proposed architecture for each category.

4.2.1 User Aspects Considered in GUMCARS

The UML class diagram of user information is presented in Figure 3, and the counting of classes and attributes considered for this package is presented in Table 1. In addition, representative examples of what user aspects were found through the SLR, and how they were organized in the proposed model are given. For a detailed documentation of each class and attribute please refer to `http://bit.ly/GUMCARS_user`, where the GUMCARS model is used as the core component of the user modeling framework for CARS.

For the model architecture, the *body part* sub-subcategory presented in the taxonomy is specialized into four types of body parts: *Foot*, *Arm*, *Head* and *Hand*. Aspects of this classes are being used in CARS, for example *Shoe Size* which is an attribute of the *Foot* class, is considered by [44], while *Ring Size* and *Bracelet Size* attributes mapped to *Hand* and the *Arm* classes, respectively, can be used by a CARS recommending fashion accessories like the proposed in [57].

Contact information like *Address* and *email address* are used by [84]; the physical *address* and *phone number* are also attributes of the *Contact* class as used by [71].

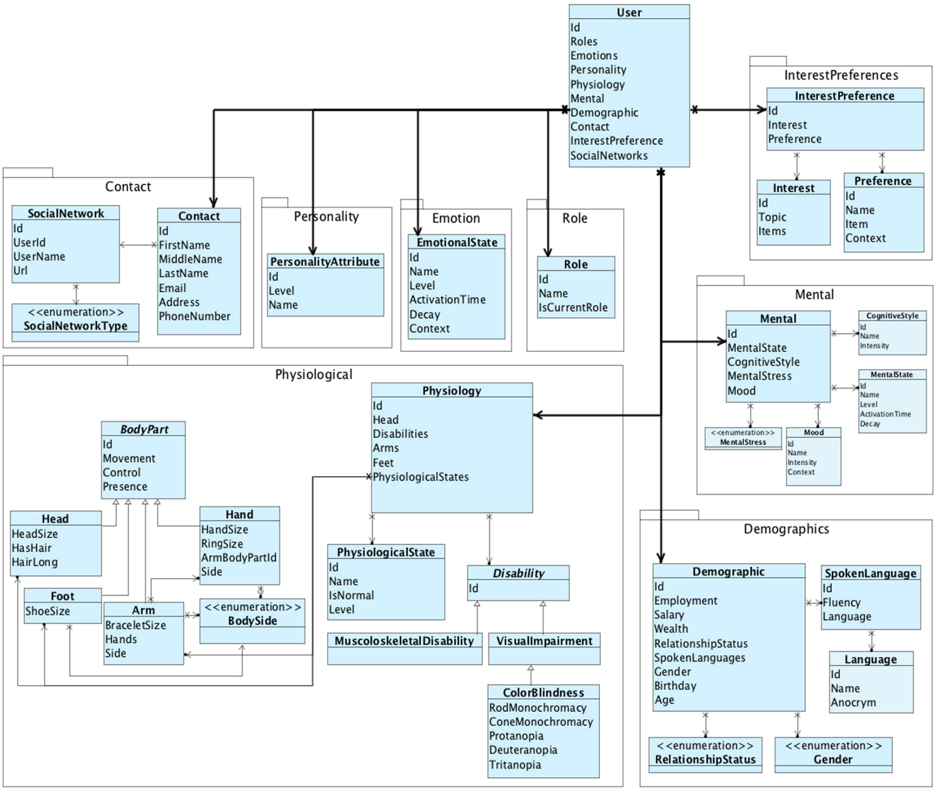

Figure 3. User aspects considered in GUMCARS

The personality information is used for example by [51] as a way to improve the music preference predictions in their CARS. GUMCARS support personality information containing base attributes such as *Level* and *Name* that can support different personality models or can be specialized to support a specific personality model.

The demographics information was commonly used in the reviewed literature, for example [92, 17] use *age*, *gender*, and *employment* among others. In this package, *Spoken Language* class was added to support cases like [7] where the *language* that the user speak is considered by their CARS.

4.2.2 Context Aspects Considered in GUMCARS

In this section, Figure 4 presents UML class diagram for the context information, and Table 2 presents the number of classes and attributes considered in GUMCARS; then, representative examples of what context aspect were found through the SLR,

Subcategory	Classes	Attributes
Physiology	12	33
Contact	3	23
Personality	1	4
Interest and Preferences	3	10
Emotion	1	6
Role	1	3
Demographics	5	28
Mental	5	20
Total	31	127

Table 1. Number of classes and attributes of GUMCARS for User information

and how they were organized in the proposed model are given. A detailed documentation of each class can be found at `http://bit.ly/GUMCARS_context`.

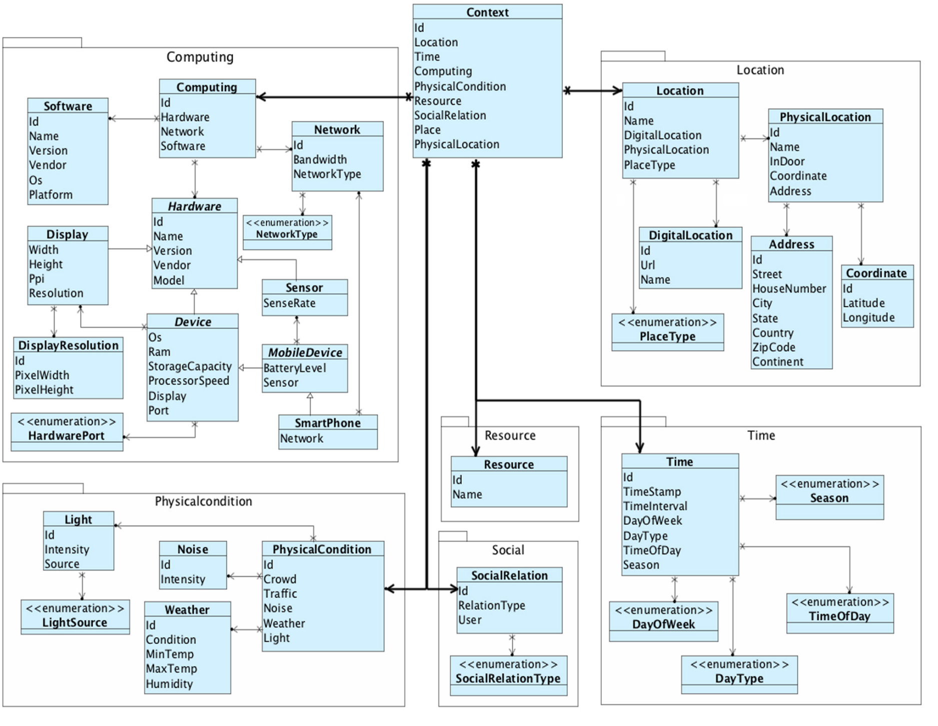

Figure 4. Context aspects considered in GUMCARS

Subcategory	**Classes**	**Attributes**
Computing	12	67
Resource	1	2
Time	5	33
Social Relation	2	11
Location	6	40
Physical Condition	5	19
Total	31	172

Table 2. Number of classes and attributes of GUMCARS for context information

The information of computing devices like smart phones is used by [26]. Also [77] considers that storing the battery level of mobile computing devices is important for CARS. To support this information, the *device* class of the *Computing* package, is further specialized into *mobile device* class, also *sensor* class is created as an specialization of *hardware* class.

Time is one of the most used contextual information in CARS, as found in the SLR. For example, [26] uses the *time of the day* to tailor news, [92] uses *weekday* and *weekend* categorization, and [71] uses the type of day (e.g. holiday or weekend) as valuable information in their CARS. GUMCARS further organizes the *Time* package adding *Time of Day*, *Day of Week*, *Season* and *Day Type* classes.

The aspects referring to other packages like the *Social Relation*, *Location* and *Physical conditions* were also found in the SLR, for example [16] considers the user companion when recommending movies, [8] uses the location of the user (expressed in latitude and longitude) when recommending places to visit, and [71] uses the weather condition as valuable information in their CARS.

4.2.3 Activity Information Considered in GUMCARS

Recommender systems use the information of what item the user consumed and in what context such activity took place, as base information for the prediction generation [74]. A good example of how all this information is used is presented in [71], their CARS consider the activity of the user (running, driving, standing, etc.), along with some contextual information (like location, time of day, day of week) and information about what song the user is currently listening, to decide what to recommend next.

With the *Activity* package, GUMCARS supports the past and current activity of the user, the item consumed in such a transaction, the contextual information, and the rating given by the user. GUMCARS contains a total of 5 classes and 18 attributes to represent the activity information as depicted by the UML diagram in Figure 5. For a detailed description of each class and attribute please refer to `http://bit.ly/GUMCARS_act`.

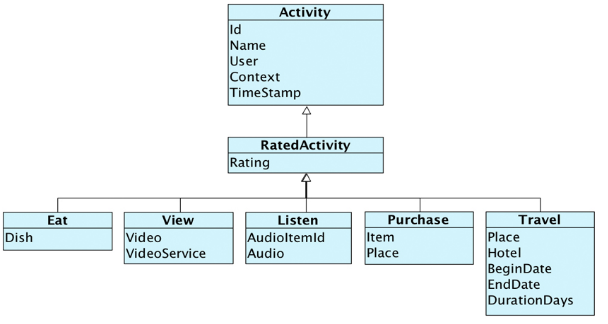

Figure 5. Act aspects considered in GUMCARS

4.2.4 Items Information Considered in GUMCARS

In addition to User, Context and Activity information, GUMCARS also includes the base aspects related to the *Items* that a CARS could recommend. These aspects are included in an *Item* package, which contains classes to represent the elements the reviewed publications are recommending as shown in Figure 6. The item package contains a total of 8 classes and 41 attributes to represent the information about the items. For a UML diagram of the classes and a detailed description of each class and attribute, please refer to `http://bit.ly/GUMCARS_items`.

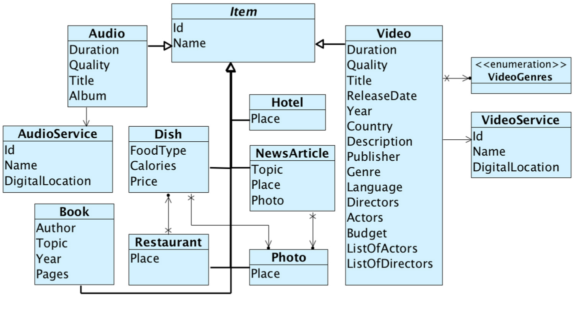

Figure 6. Item aspects considered in GUMCARS

Next sections present two evaluations performed to the GUMCARS model, the first one is dedicated to a mathematical evaluation using complexity metrics intended to evaluate the *Correctness* of the model. The second describes a practical evaluation that evaluates the *Completeness* and *Generality* of GUMCARS by measuring its ability to hold data from real-world datasets.

5 EVALUATING GUMCARS COMPLETENESS AND GENERALITY

According to [58], the quality of a conceptual model can be categorized in

Syntactic: the model contains only statements that are valid in the language;

Semantic: how well the model represents the important elements of the domain; and

Pragmatic: how the intended audience interprets the models.

In this experiment, we perform a *Semantic* quality evaluation of GUMCARS. The *Semantic* quality of the model (M) with respect to the target domain (D), can be defined by two quality attributes: *Validity* and *Completeness* [58].

Validity ensures that the elements included in the model are relevant to the modeled domain. As the GUMCARS model is based on the SLR [46] performed explicitly in the target (CARS) domain, all the included elements in GUMCARS were stated as important for the domain by their respective publications.

Therefore, in this iteration, we focus on evaluating the *Completeness* of GUMCARS, which refers to how many of the relevant elements of the domain are included in the model. In [64], *Completeness* is also defined as *whether the model supports all the information required by target systems*. This experiment also allowed us to validate the *Generality* of the model that according to Moody [65] describes how wide the application scope of the model is. In this work, we define *Generality* as the ability of the model to support data from different domains.

5.1 Materials

We used a total of 8 datasets gathered from the literature. We selected only datasets that were not considered during the creation of the model, that were created for recommendation purposes, and, that contain data about the 3 main elements (*Users*, *Items* and *Context*). The datasets correspond to different item domains, with different quantity and type of information. Table 3 shows the details of each dataset.

In addition, to perform this evaluation, an implementation of GUMCARS is used, such implementation is part of a wider project where the proposed model is used as part of a CARS modeling framework (`http://um4rs.com`). Such implementation was created by exporting the UML representation of the model to a C# code library. We consider the code to be just another representation of the model, as there is no change between the architectural view of the model and the implemented code. In this experiment, as well as in the one described in Section 6, we opted to use the implemented model as a way to automatize the evaluation and avoid biases.

Dataset	Description	Type	Attributes				Instances
			User	Context	Item	Total	
Japan	Restaurant dataset by [70]. Describes the restaurant, time, environment, and user's social information.	Food	3	14	17	34	800
Trip Advisor	Hotel booking dataset by [96]. Contains information about the hotel, the context, and the user.	Hotel/ Travel	3	2	4	9	14 176
Expedia	Travel booking dataset from expedia.com. Contains 150 aspects about the travel destination, aspects of the user and the context.	Hotel/ Travel	4	15	154	173	62 100
STS	A travel dataset by [31]. Is rich in user and context information including Big Five personality traits.	Hotel/ Travel	8	15	4	27	2 535
DePaul Movie	Movie dataset by [97]. Contains information about the time and companion of the user.	Movie	1	4	1	6	50 043
LDOS CoMoDa	A movie rating dataset by [55]. This dataset contains emotional information about the user while rating/watching the movie, as well as movie and context information.	Movie	9	8	14	31	2 296
Concert Tweets	A music dataset by [1]. This dataset contains information about concert events, the user, and the context of the user when attending the events.	Music	1	7	4	12	249 470
InCar Music	A music dataset by [9]. This dataset contains information about the song, the mood and driving style of the user while listening to the song, as well as other contextual information.	Music	2	8	8	18	4 012

Table 3. Information of datasets loaded into GUMCARS

5.2 Method

First, we mapped all the possible aspects found in each dataset to their corresponding attribute in GUMCARS classes. This enables us to identify what dataset aspects are supported, and what aspects are not supported by GUMCARS. An aspect is considered to be supported if GUMCARS contains an equal or similar attribute in their classes. For example, *Social* aspect of LDOS CoMoDa dataset refers to *the companion of the user to watch the movie*, and was mapped to *Context.SocialRelation.RelationType* enumerator in the model, that represents the same information.

Then, a simple script was developed for each dataset, where the dataset file was read and its contained features are mapped into GUMCARS implemented representation, following the mapping created in the first step. During this step, no changes were made to GUMCARS code.

The *Completeness* of GUMCARS was calculated as the percentage of the total numbers of aspects contained in the dataset (D), that were successfully mapped into the model (M), i.e. $\mathit{Completeness} = M/D$. Each dataset aspect scores a 1 into the D variable if a corresponding attribute exists in GUMCARS, and all of its values were fully supported by the model. In cases where a corresponding attribute exists in GUMCARS, but some of the values present in the dataset were not supported by the model aspect, such attributes scored a 0.5 into D and considered as *Partially Supported*.

5.3 Results

A total of 31 user, 73 context, 206 items aspects, and 323 332 instances were contained in the 8 datasets, and were mapped (separately) into GUMCARS. Along the datasets, some aspects are repeated, being the most common *userId*, *itemId*, *rating* and *weather*. As a result, in this experiment, 22 unique *User* aspects were used, 64 *Context* aspects, and 194 *Item* aspects of 6 different types of items, namely: *food*, *hotels*, *travels*, *movies*, *concerts* and *songs*, were used.

For *Japan* dataset, 6 of the 32 attributes were not supported, and 1 attribute (RelationType) was considered partially supported, as 2 (*Boss* and *Subordinate*) of the possible 6 values could not be mapped to GUMCARS's *RelationType* enumerator. This result in a *Completeness* value of 0.81, which means that the 81 % of the attributes were supported. We got *Completeness* of 89 % for *TripAdvisor*, 96 % for *Expedia*, and 70 % for *STS*. For *DePaul Movie* we obtained a *Completeness* of 100 %, and 90 % for *LDOS CoMoDa*. In the *ConcertTweets* dataset, we were able to map all the contained attributes to GUMCARS, thus a *Completeness* value of 100 %. Finally, for *InCar Music* dataset we obtained 83 % of *Completeness* as 3 of its 18 attributes were not supported.

In general, GUMCARS supported the 93.55 % of the user aspects, 75.34 % of contextual aspects, and 96.12 % of item aspects contained in the eight datasets used in this experiment. The obtained *Completeness* for each dataset as well as the non supported attributes are presented in Table 4.

Dataset	Total Attributes	Not Supported		Partially		Compl-etness
Japan	34	6	Budget Num. of Male (Partners) Num. of Female (Partners) Lowest Age (Partners) Highest Age (Partners) Partner Status	1	Relation type	0.81
Trip-Advisor	9	1	User time zone	0		0.89
Expedia	173	7	Orig_destination_distance Is_package (is a travel package) Srch_adults (num of adults) Srch_children (num of children) Srch_rm (num of rooms) Cnt (number of similar events) Posa_continent (point of sale)	0		0.96
STS	27	8	distance knowledgeOfSurroundings budget travel goal means of transport POI's category1 POI's category2 POI's category3	0		0.70
DePaul Movie	6	0		0		1.00
LDOS CoMoDa	31	3	Physical (health condition) Decision (to watch the movie) Interaction (n-th interaction)	0		0.90
Concert-Tweets	12	0		0		1.00
InCar Music	18	3	DrivingStyle RoadType Sleepiness	0		0.83

Table 4. Results of mapping and loading dataset into GUMCARS model

5.4 Discussion

The average value of *Completeness* for the 310 aspects tested was a 0.88 of 1, with a standard deviation of 0.09, which strongly suggests that GUMCARS is capable of supporting most of the aspects considered by the datasets.

The lowest *Completeness* value for a dataset corresponds to *STS*, which scored a 0.70, as 8 out of the 27 aspects considered in the dataset were not supported by

GUMCARS. This dataset contains some very specific traveling aspects like *Travel Goal* and *Traveled Distance* that would have been difficult to foresee.

We consider that GUMCARS performed well in supporting the datasets, especially in user and item information with 94 % and 96 % of *Completeness*, respectively. The obtained 75 % of *Completeness* for the context information, even when this is a good value, is lower than the obtained value for others datasets. We attribute this to the fact that recommendation system field just started to consider contextual information [2], and there is not an established standard of what context information works best for CARS. Also, the amount of information of the environment (context) that surrounds users is enormous, and trying to create a contextual model that encompasses everything is practically impossible. Nevertheless, we strongly believe that GUMCARS has the needed base to allow model designers and researchers to tailor or extend the model to their specific contextual needs.

In general, this experiment allowed us to evaluate to some degree the *Completeness* of the model by mapping into it different datasets with different characteristic obtaining encouraging results. Also, the fact that the dataset contains information from different domains, allows us to prove GUMCARS *Generality*, as it was able to support most of the aspects of the tested domains. As for the non supported elements, they were included after finishing the evaluations, and the increased version of GUMCARS will be published as open-source.

6 EVALUATING GUMCARS STRUCTURAL CORRECTNESS

Assessing the quality of a complex model is generally a difficult task [44], since GUMCARS is, to the best of our knowledge, the first user model designed specially to build Context-Aware Recommender Systems, a direct comparison with other proposals is difficult or even misleading.

As the goal of GUMCARS is to be used in CARS implementations to support their data, we considered important to assess some quality attributes of the *architectural* and *implemented* representation of the model. Therefore, in this experiment, we focus on evaluating the *Correctness* of GUMCARS, which is a very important quality attribute of conceptual models [58].

The *Correctness* of a model is defined as whether the model conforms to the rules of data modeling techniques [64]. Since GUMCARS was created following the Object-Oriented (OO) design technique, we used software quality metrics to assess the level of *Correctness* of the proposed model, which (according to [36]) allow software designers to evaluate the quality characteristics of OO models, objectively.

6.1 Materials Used

For this evaluation, the *Depth of Inheritance Tree*, *Number of Children* and *Coupling Between Objects* metrics from the (CK) metric suite [18] were selected, as these metrics that can be applied either to class diagram or to code [36]. We also considered important to assess the Maintainability of GUMCARS, as it expresses the easiness with which the model could be adapted to suit specific needs [25], therefore we also include the *Maintainability Index* metric in this evaluation. Next, the used metrics are briefly described.

- *Depth of Inheritance Tree (DIT)*: The depth of a class within the inheritance is defined as the maximum length from the class node to the root of the class hierarchy tree and is measured by the number of ancestor classes [18]. A low value of DIT implies less complexity in the model organization, a value between $1 \leq DIT \leq 6$ is desired [69].
- *Number Of Children (NOC)* Number of immediate sub-classes subordinated to a class in the class hierarchy [18]. The value of NOC represents the number of classes that inherit from a specific class. An optimal value for NOC is between 0 and 11, values from 12 to 28 are regular and a value greater than 29 is considered bad NOC [35].
- *Coupling Between Objects (CBO)* Coupling Between Objects, also known as Class Coupling measures the relationships between entities [19]. CBO is a measure of how many classes does a single class use. The CBO value threshold proposed by Microsoft [67] is a 0–9 range for optimal (green) CBO value, a 10–80 range for acceptable (yellow) value, and CBO greater than 80 is considered critical (red) value.
- *Maintainability Index (MI)* ISO/IEC 9126 [47] defines maintainability as *"the capability of the software product to be modified. Modification may include corrections, improvements or adaptation of the software to changes in the environment"*. To measure the *maintainability* of the GUMCARS we used the *Maintainability Index* (MI) metric, which measures maintainability by taking the size, complexity, and self-descriptiveness of the classes into account, resulting in a MI range between 0 and 100, where 100 represents the best possible value of maintainability of the system [68].

6.2 Method

The selected metrics could be applied either to UML class diagram performing the calculation by hand or to classes in code using tools to perform the calculations.

To avoid any bias in the metric applications, and to get the most objective results, we opted to apply the metrics to the code representation of GUMCARS (described in Section 5), using automated tools to perform the calculations.

The GUMCARS's values for *DIT* and *NOC* metric were gathered using NDepend. For the *CBO* and *MI* metrics, Visual Studio IDE was used. Both tools gave a numeric value of each class for the evaluating metric. The results of each metric are loaded into SPSS for the data analysis.

6.3 Results

According to [18], a lower DIT value is preferred, with a threshold between 1 and 6 for an optimal value. This experiment revealed a maximum DIT value of 4, a mean value 1.41 with a Standard Deviation (SD) of 0.69, as can be seen in Table 5, placing GUMCARS inside the optimal threshold for this metric. The deepest leaf found in GUMCARS inheritance tree is *Hardware* > *Device* > *MobileDevice* > *SmartPhone* in the context package.

The NOC mean value of GUMCARS is 0.41 with an SD of 1.40 and a maximum number of 8 children for a single class, that correspond to *Item* class, as all the types of items considered in GUMCARS are dependent of *Item* class.

In this experiment, we got a CBO value of 1.42, with SD of 1.79 and a maximum value of 10. The maximum CBO value corresponds to *User* class, as it is dependent of 10 other classes, even when the 10 is outside the optimal value (as shown in Figure 7), it just occurred once, and the rest of the 85 classes fall between 1 and 7.

The last metric applied in this experiment is MI, the mean value obtained is 94.8, with SD of 2.89 and the lowest value of 90 with only 1 occurrence, and corresponds to *Item* class.

Next, Figure 7 shows the frequency distribution for the DIT, NOC and CBO metrics, and Figure 8 shows the distribution for MI values. Table 5 summarizes the results obtained from the application the quality metrics to GUMCARS. Then, Table 6 shows some representative examples of the raw data obtained from the metrics, showing some best and worst cases.

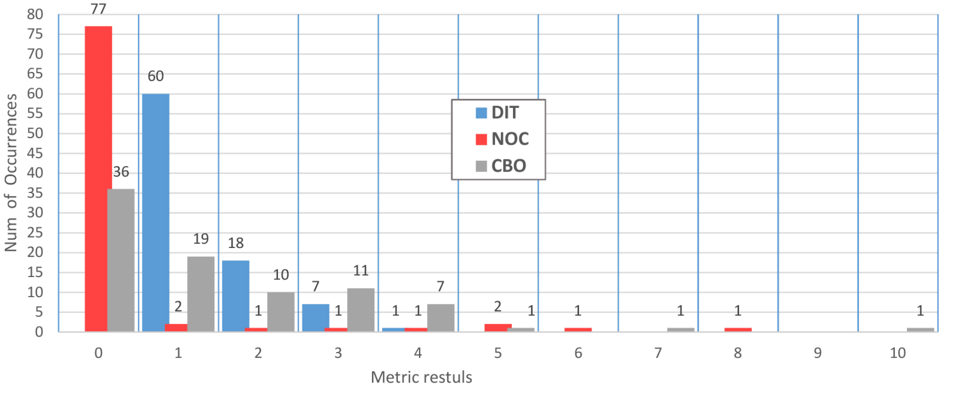

Figure 7. Frequency distribution of DIT, NOC and CBO

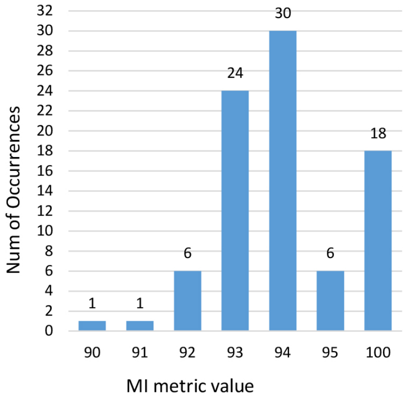

Figure 8. Frequency distribution of MI values

Metric	Min	Max	Mean	SD	Threshold	Better when	Tool
DIT	1	4	1.41	0.69	$1 \leq DIT \leq 6$	Lower	ND
NOC	0	8	0.41	1.40	$0 \leq NOC \leq 29$	Lower	ND
CBO	1	10	1.42	1.79	$0 \leq CBO \leq 80$	Lower	VS
MI	90	100	94.83	2.89	$0 \leq MI \leq 100$	Greater	VS

Table 5. Results of applying quality metrics to GUMCARS

6.4 Discussion

The low value obtained for DIT (1.41) reflects that the model organization is not complex, which will help GUMCARS to be understood and used by its target users (CARS designers, researchers, and developers). This DIT value supports *Correctness* with respect to the organization of the model classes.

For the NOC metric, we obtained a 0.41 with a maximum of 8, this places GUMCARS in the optimal values part of the threshold. The result of NOC strongly

Class	**Location in Model**	**DIT**	**NOC**	**CBO**	**MI**
User	*(top level)*	1	0	10	90
Context	*(top level)*	1	0	7	92
Item	*(top level)*	2	8	0	91
Activity	*(top level)*	1	6	2	93
RelationType	Context.Social	1	0	0	100
SmarthPhone	Context.Computing	4	0	4	95
MentralStress	User.Mental	1	0	0	100

Table 6. Representative examples of raw data obtained from the application of the metrics

suggests that the proposed model uses a correct level of abstraction, as there is not a single class with too many dependent children.

Coupling Between Objects (CBO) metric yielded a mean 1.42 value which reflects that GUMCARS has an optimal level of coupling between classes. From the 86 total classes of the model, only one lays outside the optimal value, namely the *User* class, that got a CBO of 10, placing it in the *good* part of the threshold. Certainly, the *User* CBO could be improved sub-dividing the class, but we need to consider the increase of complexity by adding another level on the inheritance tree, thus increasing the overall DIT value. Overall, the result of this metric supports the general *Correctness* of the proposal, showing that the model represents the intended domain using a correct abstraction level.

This experiment yielded a mean value of 94.83 for the MI metric, meaning that the proposal's correct organization of classes and attributes resulted in an optimal level of MI. Figure 8 shows that most of the classes have an MI of 93 and 94. The lowest value of MI found in the whole model was 90 that corresponds to *Item* class, as all the items that the model considers are dependent on this class. This means that any change made to the *Item* class will be propagated through other classes, and that designers need to consider this before making any change to *Item* class. Nevertheless, an MI value of 90 falls very close to an optimal value, and we consider it a good value considering that *Item* class is some of the cornerstones of the GUMCARS.

Overall, in this experiment we evaluated the *computational* representation of the proposal, obtaining very good results for all the quality metrics, which strongly suggest that GUMCARS contains a high level of structural *Correctness*, and can be used by CARS designers and developers to support the data needed for their systems, and in cases where the model does not consider some specific aspect(s), it will support such a modification with minimum possibilities of affecting the overall structure of the model.

7 CONCLUSIONS AND FUTURE WORK

In this paper, a Generic User Model for Context-Aware Recommender Systems was described in terms of: the taxonomy to organize the information, the model structure, an extensive set of user, context, and item aspects needed by CARS to generate recommendations. The goal of GUMCARS is to help CARS designers, developers and researchers presenting them a base data model that can use directly or as a reference when developing their own CARS.

The proposed model was evaluated using some real world datasets from CARS domain, with the intention to assess the model completeness and generality. The obtained results from this experiment strongly suggest that GUMCARS has a great degree of completeness as 88 % of the dataset aspects were supported. This experiment also showed that the model has a great level of generality, being able to support the data from 6 different domains of CARS. This implies that the model has the needed attributes and classes to support the information that CARS needs.

Another evaluation performed to GUMCARS allowed us to assess the *correctness* respect to the structural quality of the model. The application of 4 quality metrics for object-oriented models design showed that GUMCARS contains an optimal level of *Depth of Inheritance*, *Number of Children* and *Maintainability*, and an acceptable level of *Class Coupling*. Overall, the results from the second experiment, strongly suggest that GUMCARS has a high level of structural *correctness*, which implies that the attributes are organized in the precise classes and that proper relationships between classes are present in the model.

The main limitation of this work is that the proposed model is based on CARS literature, therefore the model contains only the elements that CARS researchers considered in their publications. Based on the results obtained from the evaluations here presented, we are sure that the model will work as expected; nevertheless, we acknowledge that a dynamic validation of the model is needed to support the correct functionality of the model.

Future work on the GUMCARS proposal is aimed to test the model in a real implementation of CARS systems. Currently, there is ongoing work to create an entire modeling framework for CARS using GUMCARS as a core to organize the information. Another interesting research path is to put GUMCARS outside CARS domains, to see what modifications the model needs to support the information needed by an adaptive system in a Human-Computer Interaction area, or by an Intelligent System in a Medical field.

GUMCARS proposal contributes to user modeling area, presenting the first user and context model designed specifically for context-aware recommender systems. The model has an impact in the real world as it is ready to be implemented in CARS, and enables to avoid the work of structuring and organizing the systems information, saving developers and designer such work. We firmly believe that GUMCARS also contributes to the research community as it can be used as a reference for researchers to create more detailed or more specific user models. For example, a research seeking to create a computational model to represent human disabilities can take advantage of the *Physiological* package of GUMCARS.

REFERENCES

[1] Adamopoulos, P.—Tuzhilin, A.: ConcertTweets: A Multi-Dimensional Data Set for Recommender Systems Research. 8$^{\text{th}}$ ACM Conference on Recommender Systems (RecSys 2014), Foster City, Silicon Valley, USA, 2014.

[2] 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, Vol. 17, 2005, No. 6, pp. 734–749, doi: 10.1109/TKDE.2005.99.

[3] Adomavicius, G.—Tuzhilin, A.: Context-Aware Recommender Systems. Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys '08), 2008, p. 335–336, doi: 10.1145/1454008.1454068.

[4] Aggarwal, C. C.: An Introduction to Recommender Systems. In: Aggarval, C. C.: Recommender Systems: The Textbook. Springer, 2016, pp. 1–29.

[5] Albert, W.—Dixon, E.: Is This What You Expected? The Use of Expectation Measures in Usability Testing. Proceedings of Usability Professionals Association Conference, Scottsdale, AZ, 2003.

[6] Alegre, U.—Augusto, J. C.—Clark, T.: Engineering Context-Aware Systems and Applications: A Survey. Journal of Systems and Software, Vol. 117, 2016, pp. 55–83, doi: 10.1016/j.jss.2016.02.010.

[7] Alhamid, M. F.—Rawashdeh, M.—Al Osman, H.—Hossain, M. S.—El Saddik, A.: Towards Context-Sensitive Collaborative Media Recommender System. Multimedia Tools and Applications, Vol. 74, 2015, No. 24, pp. 11399–11428.

[8] Bagci, H.—Karagoz, P.: Context-Aware Location Recommendation by Using a Random Walk-Based Approach. Knowledge and Information Systems, Vol. 47, 2016, No. 2, pp. 241–260, doi: 10.1007/s10115-015-0857-0.

[9] Baltrunas, L.—Kaminskas, M.—Ludwig, B.—Moling, O.—Ricci, F.—Aydin, A.—Lüke, K.-H.—Schwaiger, R.: InCarMusic: Context-Aware Music Recommendations in a Car. In: Huemer, C., Setzer, T. (Eds.): E-Commerce and Web Technologies (EC-Web 2011). Springer, Berlin, Heidelberg, Lecture Notes in Business Information Processing, Vol. 85, 2011, pp. 89–100.

[10] Bangor, A.—Kortum, P.—Miller, J.: Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. Journal of Usability Studies, Vol. 4, 2009, No. 3, pp. 114–123.

[11] Beale, R.—Lonsdale, P.: Mobile Context Aware Systems: The Intelligence to Support Tasks and Effectively Utilise Resources. In: Brewster, S., Dunlop, M. (Eds.): Mobile Human-Computer Interaction – MobileHCI 2004. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3160, 2004, pp. 240–251.

[12] Booch, G.: Object-Oriented Analysis and Design with Applications. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1994.

[13] Booch, G.—Rumbaugh, J.—Jacobson, I.: The Unified Modeling Language for Object-Oriented Development. Unix Review, Vol. 14, 1996, No. 13, `http://www.ccs.neu.edu/research/demeter/course/f96/readings/uml9.pdf`.

[14] Braunhofer, M.—Ricci, F.: Contextual Information Elicitation in Travel Recommender Systems. In: Inversini, A., Schegg, R. (Eds.): Information and Communication Technologies in Tourism 2016. Springer, Cham, 2016, pp. 579–592.

[15] Brooke, J.: SUS – A Quick and Dirty Usability Scale. Usability Evaluation in Industry, Vol. 189, 1996, No. 194, pp. 4–7.

[16] Chen, B.—Yu, P.—Cao, C.—Xu, F.—Lu, J.: ConRec: A Software Framework for Context-Aware Recommendation Based on Dynamic and Personalized Context. 2015 IEEE 39th Annual Computer Software and Applications Conference (COMPSAC), 2015, Vol. 2, pp. 816–821.

[17] Chen, G.—Chen, L.: Augmenting Service Recommender Systems by Incorporating Contextual Opinions from User Reviews. User Modeling and User-Adapted Interaction, Vol. 25, 2015, No. 3, pp. 295–329, doi: 10.1007/s11257-015-9157-3.

[18] Chidamber, S. R.—Kemerer, C. F.: A Metrics Suite for Object Oriented Design. IEEE Transactions on Software Engineering, Vol. 20, 1994, No. 6, pp. 476–493, doi: 10.1109/32.295895.

[19] Chowdhury, I.—Zulkernine, M.: Using Complexity, Coupling, and Cohesion Metrics as Early Indicators of Vulnerabilities. Journal of Systems Architecture, Vol. 57, 2011, No. 3, pp. 294–313, doi: 10.1016/j.sysarc.2010.06.003.

[20] Churchill, E.—Laubach, L. L.—McConville, J. T.—Tebbetts, I.: Anthropometric Source Book. Volume 1: Anthropometry for Designers. NASA, Yellow Springs, Ohio, 1978.

[21] Clarke, S.: Measuring API Usability. Dr. Dobb's Journal, Vol. 29, 2004, No. 5, pp. S1–S5.

[22] Codina, V.—Ricci, F.—Ceccaroni, L.: Distributional Semantic Pre-Filtering in Context-Aware Recommender Systems. User Modeling and User-Adapted Interaction, Vol. 26, 2015, No. 1, pp. 1–32, doi: 10.1007/s11257-015-9158-2.

[23] Coleman, D.—Ash, D.—Lowther, B.—Oman, P.: Using Metrics to Evaluate Software System Maintainability. Computer, Vol. 27, 1994, No. 8, pp. 44–49, doi: 10.1109/2.303623.

[24] Colombo-Mendoza, L. O.—Valencia-García, R.—Rodríguez-González, A.—Alor-Hernández, G.—Samper-Zapater, J. J.: RecomMetz: A Context-Aware Knowledge-Based Mobile Recommender System for Movie Showtimes. Expert Systems with Applications, Vol. 42, 2015, No. 3, pp. 1202–1222, doi: 10.1016/j.eswa.2014.09.016.

[25] Counsell, S.—Liu, X.—Eldh, S.—Tonelli, R.—Marchesi, M.—Concas, G.—Murgia, A.: Re-Visiting the 'Maintainability Index' Metric from an Object-Oriented Perspective. 2015 41st Euromicro Conference on Software Engineering and Advanced Applications, 2015, pp. 84–87.

[26] De Pessemier, T.—Vanhecke, K.—Martens, L.: A Personalized and Context-Aware News Offer for Mobile Devices. In: Monfort, V., Krempels, K. H., Majchrzak, T. A., Turk, Ž. (Eds.): Web Information Systems and Technologies (WEBIST 2015). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 246, 2016, pp. 147–168.

[27] Deng, J. J.—Leung, C. H. C.—Milani, A.—Chen, L.: Emotional States Associated with Music: Classification, Prediction of Changes and Consideration in Recommendation. ACM Transactions on Interactive Intelligent Systems, Vol. 5, 2015, No. 1, Art. No. 4.

[28] Dey, A. K.—Abowd, G. D.: Towards a Better Understanding of Context and Context-Awareness. Computing Systems, Vol. 40, 1999, pp. 304–307.

[29] Djoudi, B.—Bouanaka, C.—Zeghib, N.: A Formal Framework for Context-Aware Systems Specification and Verification. Journal of Systems and Software, Vol. 122, 2015, pp. 445–462.

[30] Dourish, P.: What We Talk about When We Talk about Context. Personal and Ubiquitous Computing, Vol. 8, 2004, No. 1, pp. 19–30, doi: 10.1007/s00779-003-0253-8.

[31] Elahi, M.—Braunhofer, M.—Ricci, F.—Tkalcic, M.: Personality-Based Active Learning for Collaborative Filtering Recommender Systems. In: Baldoni, M., Baroglio, C., Boella, G., Micalizio, R. (Eds.): International Conference of the Italian Association for Artificial Intelligence (AI*IA 2013). Springer International Publishing, Cham, Lecture Notes in Computer Science, Vol. 8249, 2013, pp. 360–371.

[32] Fan, X.—Hu, Y.—Li, J.—Wang, C.: Context-Aware Ubiquitous Web Services Recommendation Based on User Location Update. 2015 International Conference on Cloud Computing and Big Data (CCBD), 2015, pp. 111–118, doi: 10.1109/CCBD.2015.20.

[33] Fang, Q.—Xu, C.—Hossain, M. S.—Muhammad, G.: STCAPLRS: A Spatial-Temporal Context-Aware Personalized. ACM Transactions on Intelligent Systems and Technology, Vol. 7, 2016, No. 4, Art. No. 59.

[34] Fenton, N.—Bieman, J.: Software Metrics: A Rigorous and Practical Approach. Third Edition. CRC Press, 2014.

[35] Filó, T. G. S.—Bigonha, M. A. S.—Ferreira, K. A. M.: A Catalogue of Thresholds for Object-Oriented Software Metrics. The First International Conference on Advances and Trends in Software Engineering (SOFTENG 2015), 2015, pp. 48–55.

[36] Genero, M.—Piattini, M.—Calero, C.: A Survey of Metrics for UML Class Diagrams. Journal of Object Technology, Vol. 4, 2005, No. 9, pp. 59–92, doi: 10.5381/jot.2005.4.9.a1.

[37] Germanakos, P.—Belk, M.: Human-Centred Web Adaptation and Personalization from Theory to Practice. Springer, Human–Computer Interaction Series, 2016.

[38] Hamed, A. A.—Roose, R.—Branicki, M.—Rubin, A.: T-Recs: Time-Aware Twitter-Based Drug Recommender System. 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2012, pp. 1027–1031, doi: 10.1109/ASONAM.2012.178.

[39] Hawalah, A.—Fasli, M.: Utilizing Contextual Ontological User Profiles for Personalized Recommendations. Expert Systems with Applications, Vol. 41, 2014, No. 10, pp. 4777–4797, doi: 10.1016/j.eswa.2014.01.039.

[40] Heckmann, D.: Ubiquitous User Modeling. Ph.D. thesis, Saarland University, Germany, 2005.

[41] Heckmann, D.—Schwarzkopf, E.—Mori, J.—Dengler, D.—Kröner, A.: The User Model and Context Ontology GUMO Revisited for Future Web 2.0 Extensions. Proceedings of the International Workshop on Contexts and Ontologies: Representation and Reasoning (C & O:RR), 2007, pp. 37–46.

[42] Heitlager, I.—Kuipers, T.—Visser, J.: A Practical Model for Measuring Maintainability. 6th International Conference on the Quality of Information and Communications Technology (QUATIC 2007), 2007, pp. 30–39, doi: 10.1109/QUATIC.2007.8.

[43] Hsieh, C. K.—Yang, L.—Wei, H.—Naaman, M.—Estrin, D.: Immersive Recommendation: News and Event Recommendations Using Personal Digital Traces.

Proceedings of the 25th International Conference on World Wide Web (WWW '16), 2016, pp. 51–62, doi: 10.1145/2872427.2883006.

[44] Hussein, T.—Linder, T.—Gaulke, W.—Ziegler, J.: Hybreed: A Software Framework for Developing Context-Aware Hybrid Recommender Systems. User Modeling and User-Adapted Interaction, Vol. 24, 2014, No. 1-2, pp. 121–174, doi: 10.1007/s11257-012-9134-z.

[45] Inzunza, S.—Juárez-Ramírez, R.—Jiménez, S.: User Modeling Framework for Context-Aware Recommender Systems. In: Rocha, Á., Correia, A. M., Adeli, H., Reis, L. P., Costanzo, S. (Eds.): Recent Advances in Information Systems and Technologies (WorldCIST 2017), Vol. 1. Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 569, 2017, pp. 899–908.

[46] Inzunza, S.—Juárez-Ramírez, R.—Ramírez-Noriega, A.: User and Context Information in Context-Aware Recommender Systems: A Systematic Literature Review. In: Rocha, Á., Correia, M. A., Adeli, H., Reis, P. L., Mendonça Teixeira, M. (Eds.): New Advances in Information Systems and Technologies. Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 444, 2016, pp. 649–658.

[47] ISO/IEC TR 9126-2:2003 – Software Engineering – Product Quality – Part 2: External Metrics. International Organization for Standardization, Geneva, Switzerland, 2003.

[48] ISO 9241-11:1998 – Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs). The International Organization for Standardization, 1998.

[49] Jawaheer, G.—Weller, P.—Kostkova, P.: Modeling User Preferences in Recommender Systems: A Classification Framework for Explicit and Implicit User Feedback. ACM Transactions on Interactive Intelligent Systems, Vol. 4, 2014, No. 2, Art. No. 8, doi: 10.1145/2512208.

[50] John, O. P.—Donahue, E.—Kentle, R.: The Big Five: Factor Taxonomy: Dimensions of Personality in the Natural Language and in Questionnaire. In: Pervin, L. A. (Ed.): Handbook of Personality: Theory and Research. Guilford Press, New York, 1990, pp. 66–100.

[51] Kaminskas, M.—Ricci, F.: Contextual Music Information Retrieval and Recommendation: State of the Art and Challenges. Computer Science Review, Vol. 6, 2012, No. 2-3, pp. 89–119, doi: 10.1016/j.cosrev.2012.04.002.

[52] Kaklanis, N.—Biswas, P.—Mohamad, Y.—Gonzalez, M. F.—Peissner, M.—Langdon, P.—Tzovaras, D.—Jung, C.: Towards Standardisation of User Models for Simulation and Adaptation Purposes. Universal Access in the Information Society, Vol. 15, 2016, No. 1, pp. 21–48.

[53] Kim, H. N.—Ha, I.—Lee, K. S.—Jo, G. S.—El-Saddik, A.: Collaborative User Modeling for Enhanced Content Filtering in Recommender Systems. Decision Support Systems, Vol. 51, 2011, No. 4, pp. 772–781.

[54] Kitchenham, B.—Brereton, O. P.—Budgen, D.—Turner, M.—Bailey, J.—Linkman, S.: Systematic Literature Reviews in Software Engineering – A Systematic Literature Review. Information and Software Technology, Vol. 51, 2009, No. 1, pp. 7–15, doi: 10.1016/j.infsof.2008.09.009.

[55] Košir, A.—Odić, A.—Kunaver, M.—Tkalčič, M.—Tasič, J. F.: Database for Contextual Personalization. Elektrotehniški Vestnik, Vol. 78, 2011, No. 5, pp. 270–274.

[56] Kuflik, T.—Kay, J.—Kummerfeld, B.: Challenges and Solutions of Ubiquitous User Modeling. In: Krüger, A., Kuflik, T. (Eds.): Ubiquitous Display Environments. Springer, Berlin, Heidelberg, Cognitive Technologies, 2012, pp. 7–30.

[57] Landia, N.: Building Recommender Systems for Fashion. Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17), 2017, p. 343, `http://dl.acm.org/citation.cfm?doid=3109859.3109929`.

[58] Lindland, O. I.—Sindre, G.—Solvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Software, Vol. 11, 1994, No. 2, pp. 42–49, doi: 10.1109/52.268955.

[59] Luo, J.—Feng, H.: A Web-Based Framework for Lightweight Context-Aware Mobile Applications. International Journal of Database Theory and Application, Vol. 9, 2016, No. 4, pp. 119–134.

[60] McCabe, T. J.: A Complexity Measure. IEEE Transactions on Software Engineering, Vol. SE-2, 1976, No. 4, pp. 308–320, doi: 10.1109/TSE.1976.233837.

[61] Mettouris, C.—Papadopoulos, G. A.: Designing Context Models for CARS Incorporating Partially Observable Context. In: Christiansen, H., Stojanovic, I., Papadopoulos, G. (Eds.): Modeling and Using Context (CONTEXT 2015). Springer, Cham, Lecture Notes in Computer Science, Vol. 9405, 2015, pp. 118–131, doi: 10.1007/978-3-319-25591-0_9.

[62] Mettouris, C.—Papadopoulos, G. A.: Using Appropriate Context Models for CARS Context Modelling. In: Kunifuji, S., Papadopoulos, A. G., Skulimowski, M. J. A., Janusz, K. (Eds.): Knowledge, Information and Creativity Support Systems (KICSS '2014). Springer International Publishing, Cham, Advances in Intelligent Systems and Computing, Vol. 416, 2016, pp. 65–79.

[63] Microsoft: Code Metrics Values. 2015, `https://msdn.microsoft.com/en-us/library/bb385914.aspx`.

[64] Moody, D. L.: Measuring the Quality of Data Models: An Empirical Evaluation of the Use of Quality Metrics in Practice. Proceedings of the European Conference on Information Systems (ECIS 2003), 2003, pp. 1337–1352.

[65] Moody, D. L.: Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions. Data and Knowledge Engineering, Vol. 55, 2005, No. 3, pp. 243–276, doi: 10.1016/j.datak.2004.12.005.

[66] Morris, W. N.: More on the Mood-Emotion Distinction. Psycoloquy, Vol. 3, 1992, No. 7, Art. No. 6.

[67] Naboulsi, Z.: Code Metrics – Class Coupling. 2011, `https://blogs.msdn.microsoft.com/zainnab/2011/05/25/code-metrics-class-coupling/`.

[68] Naboulsi, Z.: Code Metrics – Maintainability Index. 2011, `https://blogs.msdn.microsoft.com/zainnab/2011/05/26/code-metrics-maintainability-index/`.

[69] Naboulsi, Z.: Code Metrics – Depth of Inheritance (DIT). 2011, `https://blogs.msdn.microsoft.com/zainnab/2011/05/19/code-metrics-depth-of-inheritance-dit/`.

[70] Oku, K.—Nakajima, S.—Miyazaki, J.—Uemura, S.: Context-Aware SVM for Context-Dependent Information Recommendation. Proceedings of the 7th International Conference on Mobile Data Management (MDM '06), IEEE, 2006, doi: 10.1109/MDM.2006.56.

[71] Otebolaku, A. M.—Andrade, M. T.: Context-Aware Media Recommendations for Smart Devices. Journal of Ambient Intelligence and Humanized Computing, Vol. 6, 2015, No. 1, pp. 13–36.

[72] Perepletchikov, M.—Ryan, C.—Frampton, K.: Cohesion Metrics for Predicting Maintainability of Service-Oriented Software. Proceedings of the Seventh International Conference on Quality Software (QSIC 2007), 2007, pp. 328–335, doi: 10.1109/QSIC.2007.4385516.

[73] Perera, C.—Zaslavsky, A.—Christen, P.—Georgakopoulos, D.: Context Aware Computing for the Internet of Things: A Survey. IEEE Communications Surveys and Tutorials, Vol. 16, 2014, No. 1, pp. 414–454.

[74] Ricci, F.—Rokach, L.—Shapira, B.—Kantor, P. B.: Recommender Systems Handbook. Springer US, Boston, MA, 2011.

[75] Samoladas, I.—Stamelos, I.—Angelis, L.—Oikonomou, A.: Open Source Software Development Should Strive for Even Greater Code Maintainability. Communications of the ACM, Vol. 47, 2004, No. 10, pp. 83–87, doi: 10.1145/1022594.1022598.

[76] Sauro, J.: Measuring Usability with the System Usability Scale (SUS). 2011.

[77] Schedl, M.: Ameliorating Music Recommendation: Integrating Music Content, Music Context, and User Context for Improved Music Retrieval and Recommendation. Proceedings of International Conference on Advances in Mobile Computing and Multimedia (MoMM '13), Vol. 3, 2013, pp. 3–9, doi: 10.1145/2536853.2536856.

[78] Schreck, J.: Security and Privacy in User Modeling. Sringer Netherlands, Human-Computer Interaction Series, Vol. 2, 2003, doi: 10.1007/978-94-017-0377-2.

[79] Seth, A.—Zhang, J.: A Social Network Based Approach to Personalized Recommendation of Participatory Media Content. Proceedings of the 2nd International Conference on Weblogs and Social Media (ICWSM '08), 2008.

[80] Shani, G.—Gunawardana, A.: Evaluating Recommendation Systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. (Eds.): Recommender Systems Handbook. Springer, Boston, MA, 2011, pp. 257–297.

[81] Shapira, B.—Rokach, L.—Freilikhman, S.: Facebook Single and Cross Domain Data for Recommendation Systems. User Modeling and User-Adapted Interaction, Vol. 23, 2013, No. 2-3, pp. 211–247, doi: 10.1007/s11257-012-9128-x.

[82] Shatnawi, R.—Li, W.—Swain, J.—Newman, T.: Finding Software Metrics Threshold Values Using ROC Curves. Journal of Software Maintenance and Evolution: Research and Practice, Vol. 22, 2010, No. 1, pp. 1–16, doi: 10.1002/smr.404.

[83] Siolas, G.—Caridakis, G.—Mylonas, P.—Kollias, S.—Stafylopatis, A.: Context-Aware User Modeling and Semantic Interoperability in Smart Home Environments. 2013 8th International Workshop on Semantic and Social Media Adaptation and Personalization, 2013, pp. 27–32, doi: 10.1109/SMAP.2013.19.

[84] Skillen, K.-L.—Chen, L.—Nugent, C. D.—Donnelly, M. P.—Burns, W.—Solheim, I.: Ontological User Profile Modeling for Context-Aware Application Per-

sonalization. In: Bravo, J., López-de-Ipiña, D., Moya, F. (Eds.): Ubiquitous Computing and Ambient Intelligence (UCAmI 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7656, 2012, pp. 261–268.

[85] Smacchia, P.: NDepend Code Metrics Definitions. 2016.

[86] Stefanidis, K.—Ntoutsi, E.—Petropoulos, M.—Nørvåg, K.—Kriegel, H.-P.: A Framework for Modeling, Computing and Presenting Time-Aware Recommendations. In: Hameurlain, A., Küng, J., Wagner, R., Liddle, S. W., Schewe, K. D., Zhou, X. (Eds.): Transactions on Large-Scale Data- and Knowledge-Centered Systems X: Special Issue on Database- and Expert-Systems Applications. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 8220, 2013, pp. 146–172.

[87] Storey, V. C.—Trujillo, J. C.—Liddle, S. W.: Research on Conceptual Modeling: Themes, Topics, and Introduction to the Special Issue. Data and Knowledge Engineering, Vol. 98, 2015, pp. 1–7, doi: 10.1016/j.datak.2015.07.002.

[88] Troelsen, A.—Japikse, P.: ADO.NET Part III: Entity Framework. In: Troelsen, A., Japikse, P.: C# 6.0 and the .NET 4.6 Framework. Apress, Berkeley, CA, 2015, pp. 929–999.

[89] Verbert, K.—Manouselis, N.—Ochoa, X. et al.: Context-Aware Recommender Systems for Learning: A Survey and Future Challenges. IEEE Transactions on Learning Technologies, Vol. 5, 2012, No. 4, pp. 318–335, doi: 10.1109/TLT.2012.11.

[90] Walter, T.—Parreiras, F. S.—Staab, S.: An Ontology-Based Framework for Domain-Specific Modeling. Software and Systems Modeling, Vol. 13, 2014, No. 1, pp. 83–108.

[91] Wu, D.—Chen, L.—Zhou, Y.—Xu, B.: A Metrics-Based Comparative Study on Object-Oriented Programming Languages. Proceedings of 27$^{\text{th}}$ International Conference on Software Engineering and Knowledge Engineering, 2015, pp. 272–277, doi: 10.18293/SEKE2015-064.

[92] Wu, S.—Liu, Q.—Wang, L.—Tan, T.: Contextual Operation for Recommender Systems. IEEE Transactions on Knowledge and Data Engineering, Vol. 28, 2016, No. 8, pp. 2000–2012.

[93] W.W.W. Consortium: Composite Capabilities/Preference Profiles. `http://www.w3.org/Mobile/CCPP`, 2004.

[94] Yaman, Ç.—Çiçekli, N.: A Content Recommendation Framework Using Ontological User Profile. In: Gelenbe, E., Lent, R. (Eds.): Computer and Information Sciences III. Springer, London, 2013, pp. 381–389.

[95] Yin, H.—Cui, B.: Spatial Context-Aware Recommendation. Spatio-Temporal Recommendation in Social Media. Springer Singapore, SpringerBriefs in Computer Science, 2016, pp. 41–63.

[96] Zheng, Y.—Mobasher, B.—Burke, R.: Context Recommendation Using Multi-Label Classification. Proceedings of the 13$^{\text{th}}$ IEEE/WIC/ACM International Conference on Web Intelligence (WI 2014), Vol. 2, 2014, pp. 288–295, doi: 10.1109/WI-IAT.2014.110.

[97] Zheng, Y.—Mobasher, B.—Burke, R.: CARSKit: A Java-Based Context-Aware Recommendation Engine. Proceedings of the 2015 IEEE International

Conference on Data Mining Workshop (ICDMW '15), 2015, pp. 1668–1671, doi: 10.1109/ICDMW.2015.222.

[98] Zimmermann, A.—Lorenz, A.—Oppermann, R.: An Operational Definition of Context. In: Kokinov, B., Richardson, D. C., Roth-Berghofer, T. R., Vieu, L. (Eds.): Modeling and Using Context (CONTEXT 2007). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4635, 2007, pp. 558–571.

Sergio Inzunza received his Ph.D. degree in engineering from the Autonomous University of Baja California i México in 2018 focused on building a user and context model for context-aware recommender system. He received his Master's degree in computer science from the same university in 2014. He is passionate about software engineering and creating tool for developers.

Reyes Juárez-Ramírez has a Bachelor's degree in computer engineering, a Master's degree in computer science and Ph.D. in computer science (2008). Since 2002 he is Professor-Researcher at Universidad Autónoma de Baja California, Campus Tijuana, México. His research areas are software engineering, human-computer interaction, knowledge engineering, and engineering education and training. He is the Technical Coordinator of the Mexican Network of Software Engineering since 2014 and IEEE Senior Member since 2014.

Samantha Jiménez holds her Ph.D. from the Universidad Autónoma de Baja California. She received her Bachelor's degree in 2011 in computer systems and Master's in engineering in 2013 from University of Colima (Colima, México). Currently she is Lecturer at Universidad Autónoma de Baja California Tijuana, México. Her research interests are in the areas of human computer-interaction, dialogue systems, affective computing, educational systems.

Guillermo Licea is Full Time Professor at Universidad Autónoma de Baja California. He has Ph.D. in computer science from CICESE Research Center. He is Member of the research group "Software Technology and Interactive Systems" and his research interests include software engineering and the improvement of engineering education using new education strategies and software tools.