X Modal
X Cultural
X Lingual
X Domain
X Site
Global OER Network

Grant Agreement Number: 761758
Project Acronym: X5GON
Project title: X5gon: Cross Modal, Cross Cultural, Cross Lingual, Cross Domain, and Cross Site Global OER Network
Project Date: 2017-09-01 to 2020-08-31
Project Duration: 36 months
Document Title: D4.2 – Final prototype of user modelling architecture
Author(s): Jasna Urbančič, Erik Novak
Contributing partners: **JSI**, UCL, NA
Date:
Approved by:
Type: P
Status: Final
Contact: Erik Novak (erik.novak@ijs.si)

Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Revision

Date	Lead Author(s)	Comments
21/06/2019	Jasna Urbančič	Initial draft
10/07/2019	Victor Connes, Walid Ben Romdhane, Colin de la Higuera, Philippe Leray, Fatma Miladi	Added contributions from Univerité de Nantes
19/07/2019	Sahan Bulathwela	Added contributions from University College London
25/08/2019	Colin de la Higuera	Draft review
30/08/2019	Jasna Urbančič, Erik Novak	Final version

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

No table of figures entries found.

ABBREVIATIONS

Abbreviation	Definition
OER	Open Educational Resource
NLP	Natural Language Processing
LMS	Learning Management System
Y1	Year 1
Y2	Year 2
Y3	Year 3
PRM	Probabilistic Relational Model

ABSTRACT

In this document we report on the work done in WP4 concerning the user modelling. We added the support for Moodle LMS to the X5GON Connect service, which is used to collect the user activity data. We developed several approaches to user modelling and used several of them in the recommendation engine to provide personalized recommendations.

1. INTRODUCTION

In this document we present the final user modelling architecture prototype. As there were no major deviations from the initial version in terms of the architecture, we focused on the most significant changes the design and the user modelling itself. The architecture considers two approaches:

1. The users login to the X5GON dashboard developed within WP2 and give information about their interests and learning patterns which is used for creating the user's model,
2. The recommender engine creates user models by extracting concepts from material a user has viewed on OER repositories and uses them for determining the user's interests.

As the X5GON dashboard is not fully developed yet, we focused our efforts into the second approach. The architecture also contains a mechanism for merging both approaches, as well as enabling real-time updating of the user models. This mechanism is based on the X5GON user activity tracker library (renamed to the X5GON Connect Service) described in D2.1 – Requirements & Architecture report and D2.2 – Final Server-Side Platform. The library is able to identify the user's movement across different OER repositories as well as the X5GON dashboard. In addition, it provides the user activity information is used for updating the user models.

The document is structured as follows. Section 2 describes the data used in creating the user's model. The data consists of both the OER material metadata and the user activity data. Next, in Section 3 we present the user modelling architecture, while in Section 4 we describe various approaches to user modelling. We present how we use user models in the recommendation engine in Section 5, and finally conclude the report with Section 6.

2. OER MATERIAL AND USER ACTIVITY DATA

We identified that there are two types of data that are useful for developing the user models – the OER material metadata and user activity data. While the material metadata consists of the data extracted and enriched through the ingesting and material processing pipeline, the user activity data was acquired through the user activity tracking library, renamed to the X5GON Connect Service.

Both data types were described in several previous deliverables – therefore, we suggest the reader to access the following deliverables for the following information:

- *Deliverable 2.2 – Final server-side platform* for the data acquisition process,
- *Deliverable 4.1 – Initial prototype of user modelling architecture* for the detailed description of the user activity data and how we use it, and
- *Deliverable 4.3 – Early prototype of recommendation engine* for the explanation on how we combine both types of data.

For the recent updates about the collected data, we point the reader to Section 2 of deliverable *D4.4 - Final prototype of recommendation engine*.

3. USER MODELLING ARCHITECTURE

The user modelling architecture was first designed in Y1 and is shown in **Figure 1**. Even though its full description is found in *Deliverable 4.1 – Initial prototype of user modelling architecture,* we present a brief overview of its structure.

The architecture is designed to handle three different approaches:

The OER provider approach. In this approach, the user models are dynamically updated using the information of the OER materials viewed by the user. The information of what materials were viewed by the user is provided by the Connect Service library integrated in the OER repositories.

The X5GON Dashboard approach. The user manually provides his or her topics of interests in the X5GON dashboard which is used to modify the user model.

The combined approach. A combination of the previous two approaches.

In Y2 we successfully implemented the OER provider approach and enabled the personalized recommendations for users accessing the network through partner repositories. As the X5GON Dashboard is still in development, the second approach and the combined approach have not been addressed yet.

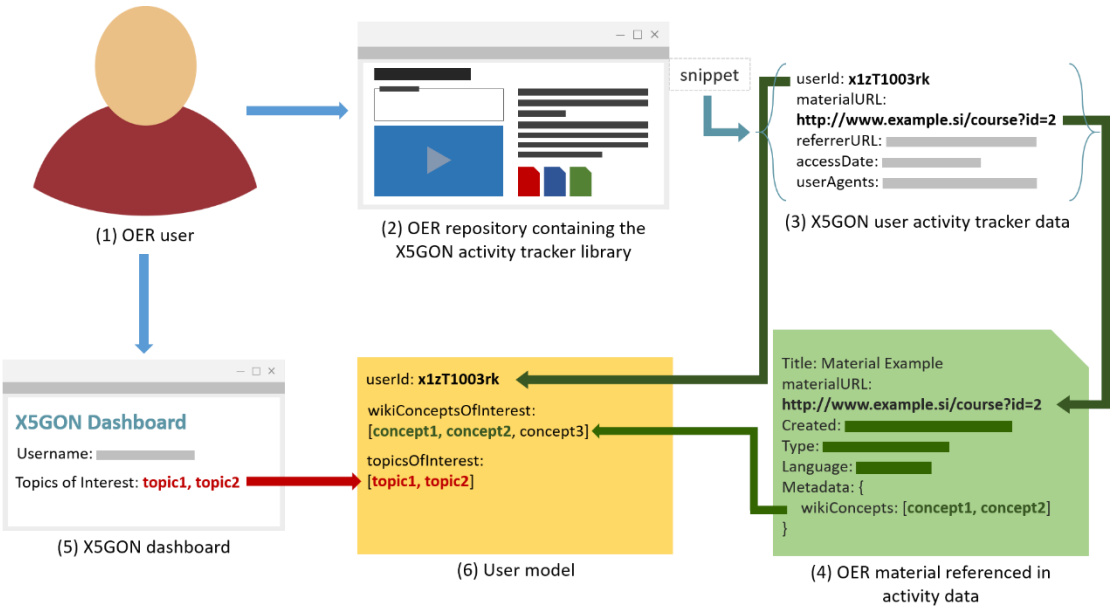

Figure 1: *User modelling architecture. It shows the schema of the OER provider and X5GON dashboard approaches, as well as how they how they can be combined.*

3.1. CONNECT SERVICE FOR USER ACTIVITY DATA ACQUISITION

The Connect Service is a crucial element in the user modelling architecture as it acquires the user activity data. In Y1 we developed a JavaScript library which was used to monitor the users' activity in the partnering OER repositories, e.g. Videolectures.NET and Universitat Politècnica de València. Afterwards, in Y2 the library received only minor corrections – enabling a more secure transition between the OER providers and the platform. In addition, the library was also integrated into the following repositories: edu-sharing, maintained by Universitat de Osnabrueck, and eUčbeniki, a Slovenian repository of text book materials for primary school and high school. The Connect Service's functionality is described in detail in *Deliverable 2.2 – Final server-side platform.*

However, we discovered that the JavaScript library was not compatible with all repository systems – among them being Moodle LMS, one of the most popular LMS systems around the globe. To this end, we have dedicated some effort in developing the Moodle plugin.

3.1.1. Connect service for Moodle

As Moodle is one of the most commonly used LMS one of the goals of Y2 was the development of fully functional easy-to-use version of the Connect Service for Moodle. We have experienced problems with material acquisition from Moodle-based OER repositories in Y1, therefore the current version of the Connect Service for Moodle also handles material acquisition.

The Connect Service is enabled with X5GON Moodle plugin. The Connect Service is activated for the selected Moodle pages and resources. The selection depends on the plugin parameters and the resources metadata as only open resources are connected to X5GON. The plugin parameters should be set by the Moodle administrator at the time of installation.

The Connect service supports the following functionalities:

1. **Connecting the selected resources.** The Connect Service is integrated on the administrator selected courses pages and their sub-pages. Once any of the open resources located on the selected pages were accessed by the user, the service notifies the X5GON platform and provides it the user activity data.
2. **Acquiring the user activity traces.** Additionally, the Moodle plugin is able to provide richer user activity information – called *user actions* – to the X5GON platform. These actions can be detected on any type of Moodle modules in the course pages. The user actions can be user interaction with the media players of external or internal resources – such as the play and pause commands – and action on the YouTube media players embedded in the selected Moodle course pages. As with the regular Connect Service, the user is required to provide consent to be monitored.
3. **Accessing the resource metadata via the Data API.** In addition to providing the user activity and action information, the Moodle plugin is also able to provide the metadata of the accessed open resources. This is done through the data API web-service, developed to respond to the external requests about OERs metadata. This enables the X5GON platform to directly send the Moodle resources towards the material processing pipeline – without having to go through the collector component in the ingesting and material processing pipeline.

Because of the specific needs of one of the pilots, the Moodle plugin collects more user activity information as the javascript version. However, most of the additional attributes are strictly associated with the Moodle LMS – consisting of the Moodle page information, the Moodle resource information, and the user actions on the Moodle course.

More information about the Moodle plugin and the data it collects is in the Moodle plugin documentation [1]. The latest version of the plugin for installation is accessible at [2].

4. VARIOUS APPROACHES TO USER MODELLING

Once the user activity data is being collected, the question of how to use it becomes relevant. In this section, we aim to describe four ideas of how to use that data to model users' interests.

4.1. EMBEDDING USERS INTO SEMANTIC SPACE OF MATERIALS

The most intuitive approach to model users' interests is by embedding users into the OER materials semantic space. In the pre-processing step we map materials into the semantic space using their Wikipedia concepts. After mapping, each material is represented with a sparse vector or a dictionary, where the keys are Wikipedia concepts. The value associated with a key in the dictionary indicates how strong is the presence of its associated Wikipedia concept in the material. To measure the strength of its presence, we used the *support* attribute of the Wikipedia concept, an attribute produced during the material enrichment process. It counts how many times the Wikipedia concepts is found in the material content. An example of the material embedding is displayed in Equation 1.

$$materialX = \begin{Bmatrix} conceptX_1 : valueX_1 \\ conceptX_2 : valueX_2 \\ \dots \\ conceptX_{N_X} : valueX_{N_X} \end{Bmatrix}, \qquad \sum_{i=1}^{N} valueX_i = 1$$

Equation 1: Material representation in semantic space of Wikipedia concepts.

We can embed the users into the material semantic space by using the material embeddings of materials the user has viewed. Here, we assume that the user is interested in the materials he or she has accesses. Therefore, the user's interests are the sum of concepts that are found in the viewed materials. An illustrative example of the user's interest computation is shown in Figure 2. When a user is first registered in the X5GON platform, his or her user model is empty. The user model is then initialized when the user accesses an OER material for the first time. For the initialization of the user model, we assign the material model - as defined in Equation 1 – to the user. Afterwards, when the user visits additional materials, the recommender engine updates its user model following the equation presented in Equation 2. The user models are thus represented as an average of the its viewed materials.

$$user\ model = \frac{(N-1) \cdot user\ model + materialX}{N}$$

Equation 2: Formula for user model updates.

Figure 2 is an illustrative representation of the user model update. Since the materials model is represented by a dictionary where the sum of its concept values is equal to 1, the sum of all values in the user model is also equal to 1. This way the user model has the same structure as the material models, thus can be placed in the materials semantic space.

In addition, once the user views enough materials, its user model will converge to a particular value – becoming semi-stationary since new materials will not contribute much to the user model, e.g. if the material is the kth material it viewed, then the materials contribution to the user model is $\frac{1}{k}$, which for large k it goes towards zero. To this end, it is better to only consider the viewers most recent viewed materials, including the temporal component to the user model.

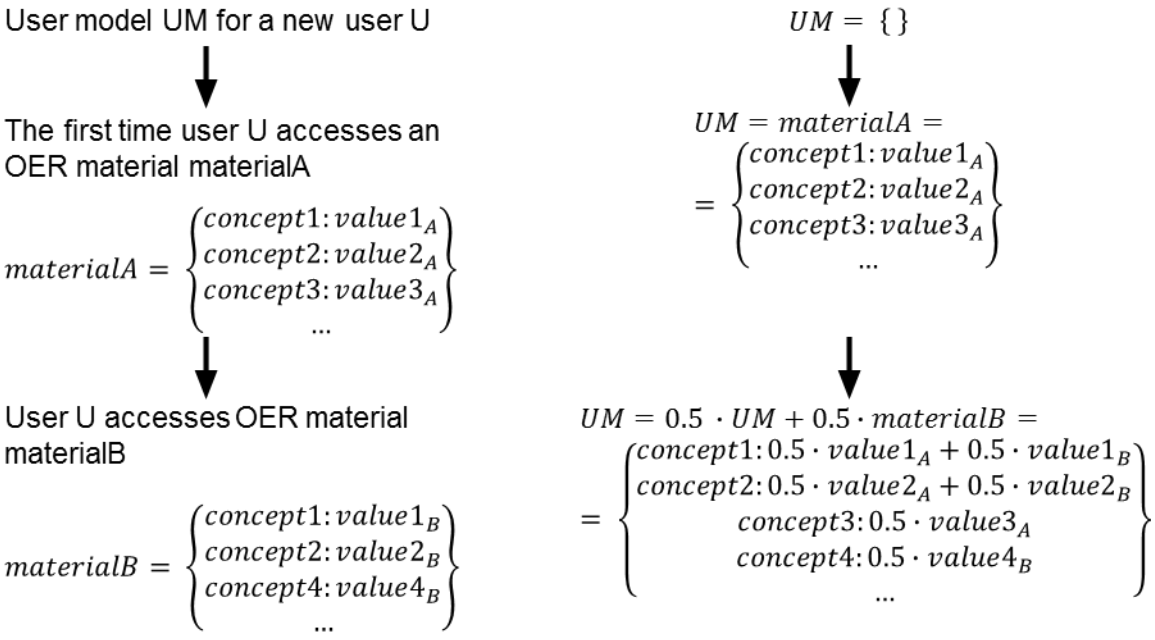

Figure 2: Illustrative example of user's interest computation.

Temporal aspect can be considered by rewriting Equation 2 with the parameter β, $0 < \beta \leq 1$, as shown in Equation 3. If β equals to 1, user model is made up of the last material the user accessed, otherwise the weight of previously accessed materials decreases every time the user accesses new material. In addition, it is worth considering to represent the user as a time window containing the materials the users viewed. Currently we use Equation 2 to update user models, however explorations with taking temporal aspect into account are possible in the future.

$$user\ model = (1 - \beta) \cdot user\ model + \beta \cdot materialX$$

Equation 3: User model updates with respect to the temporal aspect.

The user models are stored in the production PostgresQL database. This allows us to update the models in real-time when new records of users' activity are acquired by the platform.

4.2. TrueLearn: Bayesian Learner models for matching OERs to learners

Recently, with the emergence of online learning platforms [3], machine learning shows promise in providing high quality personalised teaching to anyone in the world in a cost-effective manner [4].

While excelling on the personalisation front, design of a futuristic recommendation system for education should be done with additional features in mind: (i) Cross-modality and (ii) cross-linguality are vital to identifying and recommending educational resources across different modalities and languages that are most likely to help the learner. (iii) Transparency empowers the learners by building trust between the learner and the system while supporting the learner's metacognition processes such as planning, monitoring and reflection (e.g. Open Learner Models [5]). (iv) Scalability ensures that a high-quality learning experience can be provided to large masses of learners over longer periods of time, essential in facilitating lifelong learning. (v) Data efficiency enables the system to work with less data, e.g. learning from implicit

engagement data [6]. We design TrueLearn, a recommendation system for OERs, considering all these desired features.

Drawing inspiration from two main paradigms used by the Learning Analytics research community, namely Item Response Theory [7] and Knowledge Tracing [8], we develop a set of models able to capture learner's knowledge and compatibility with OER materials.

In terms of content representation, we extract Knowledge Components (KCs), atomic units of knowledge that can be learned and mastered by a learner [8]. TrueLearn devices the same feature space described in Section 4.1 (specifically, the semantic space of Wikipedia Topics) to represent KCs. Wikifier [9] is used to infer the most relevant Wikipedia topics in OER materials and to estimate the depth in which these topics are covered. These content representations are consumed by TrueLearn to infer the learner's model.

TrueLearn, the final algorithm developed for learner modelling extends from TrueSkill [10], a Bayesian matchmaking algorithm developed to infer skills of online game players based on their performance in the games played. The main idea behind TrueLearn is to treat learner interactions with OERs as games played between learners and OERs to infer the skill learners demonstrate of different Knowledge Components.

TrueLearn model was evaluated on an OER dataset created with VideoLectures.Net data and obtained a recall of 0.821 (F1 score of 0.677) which is a 102% improvement of recall (and 69% improvement of F1 score) over the TrueSkill [10] baseline model. Furthermore, TrueLearn algorithm's accuracy (0.672) and precision (0.608) metrics also outperform TrueSkill by 51.4% and 16.5% respectively.

For a detailed report of the formulation of TrueLearn model, the experiments and their results, we direct you to *Deliverable D1.3 – Initial Content Representation*s.

4.3. CONTRIBUTION FROM NANTES

The data collected through the Connect Service is now starting to be able to track the user activity. Nevertheless, the data was weak in the following two senses:

- Only the data about the webpages was available; not about the actual final resources which were used/consumed by the user; and
- Date-stamps only allow to know when such a webpage was accessed.

The ambition is to build for each user a topic vector which can be compared with the topic vector of the actual resources. Through the connect service, a user will leave traces of its activities. A path (of activities) is composed of items (w_i, d_i), meaning that she was on page w_i at the date d_i.

Therefore, a user will have a path of traces represented as $p = (w_0, d_0), (w_1, d_1), \ldots, (w_n, d_n)$.

Let's say a webpage w_i contains the following resources $R_{i0}, R_{i1}, \ldots, R_{im}$. Then, each resource R_{ij} has an associated vector of concepts, as obtained through the Wikifier. We denote this vector as $c(R_{ij})$, where the value $c(R_{ij})[k]$ corresponds to the Wikipedia concept located at position k. Our goals are to find answers to the following questions.

1. If we are given a path of resources p (i.e. we know what resources have been visited by the user) for a user u and we assume an additional hypothesis that

the resources have been consumed, what does the vector $c(u)$ look like? In other words, what are the learner's topics?

2. If we consider a resource R and the previous webpage w_i containing resources $R_{i0}, R_{i1}, \ldots, R_{im}$, what is the probability $Pr(R_{ik}|R)$, e.g. the probability that the user's previous resource was R_{ik}? We assume that the sum of all conditional probabilities equal to $\sum_k Pr(R_{ik}|R) = 1$, i.e. the user was on webpage w_i and consumed one resource.
3. If we suppose that webpage w_i was visited before webpage w_j, and that in page w_i resource R was consumed, what is $Pr(R_{ik}|R)$, i.e. the probability of consuming resource R_{ik} (any of the resources accessible from page w_j) after R?
4. Given a sequence of webpages, what is the profile (the vector of topics) corresponding to the user? This would be $\sum_i \Pr(p_i)c(p_i)$.

Through dynamic programming we provide a positive answer to all 4 questions – but the tests have not been finalized. We will report on the final evaluation in one of the future deliverables.

4.4. PROBABILISTIC RELATIONAL MODEL

We propose to build a recommender system by using the Probabilistic Relational Model (PRM) formalism. A PRM is composed of two components: (1) a relational schema of the domain, and (2) a probabilistic model which describes the probabilistic dependencies in the domain.

Our relational schema has two entity classes called user and document, and two relationship classes, called consultation and Is-Similar-To.

We also propose one dependency structure where we define the fact that one first pertinence indicator (direct pertinence) related to one document depends on the number of times this document has been consulted. Besides, we define the indirect pertinence of one document as the weighted sum of the pertinences of its similar documents – where the weight is related to the degree of similarity between both documents.

This indirect pertinence will be used to predict the interesting documents to recommend when one user is reading one target document.

Right now, the relational schema and the database have been populated from the X5GON Database.

This PRM must now be implemented and tested with this database. Results will be compared with the one obtained with the recommender system actually used in X5gon project. We will also be able to learn the structure and/or the parameters of our PRM from the actual database. Our model will also be improved when new data will be available in order to consider more interesting features from user profile.

5. USER MODELLING AND RECOMMENDATION ENGINE

We extended the recommendation engine developed in Y1, which only supported content-based recommendations, to allow personalized recommendations in Y2.

We first implemented the User-Item similarity to find the relevant material based on the Wikipedia concepts that appear in the materials previously accessed by the user. The User-Item similarity is based on user embedding into the semantic space of the materials described in Section 4.1. We find relevant materials using the k-nearest neighbours strategy based on the cosine distances between the materials' and user embeddings.

We also use collaborative filtering, where we use user modelling in an implicit way. We model the user with a set of other users who accessed the same material as the user in question, and base the recommendations on all other materials that the set of similar users has accessed.

More details on about the two described strategies to provide personalized recommendations can be found in *D4.4 – Final prototype of the recommendation engine*.

We are aware that evaluating the user models is a very difficult task. We aim to evaluate it through the recommender engine, comparing the recommendations provided by personalized recommender engine with those of content based recommender. We discuss the early results of content based recommender engine in *D5.2 – Second report on piloting.*

6. CONCLUSION

In this document we reported on the changes in the user modelling architecture and the approaches to user modelling that have been made in Y2. The most significant change in the user modelling architecture is the development of the Moodle plugin for the X5GON Connect Service, which allows the platform to collect user activity data in a Moodle LMS. Compared to Y1 there has been a lot of focus on user modelling as we have been developing production-ready user models. One of the approaches to user modelling was integrated into the recommendation engine and has been deployed.

In the future we will consider exploring the development of user modelling which include the temporal component, as well as consider to represent the user model as a time window of viewed materials. In addition, we will continue our work in the TrueLearn described in Section 4.2, and finalize the tests on the user traces presented in Section 4.3

REFERENCES

[1] Universite de Nantes, “latest/documentation - master -x5gon / x5gonmoodleplugin-prod -GitLab,” [Online]. Available: https://gitlab.univ-nantes.fr/x5gon/x5gonmoodleplugin-prod/tree/master/latest/documentation. [Accessed 12 August 2019].

[2] Universite de Nantes, “latest/src - master - x5gon / x5gonmoodleplugin-prod - GitLab,” [Online]. Available: https://gitlab.univ-nantes.fr/x5gon/x5gonmoodleplugin-prod/tree/master/latest/src. [Accessed 12 August 2019].

[3] E. Allen and J. Seaman, “Online nation: Five years of growth in online learning,” *Technical report,* 2007.

[4] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas and J. Sohl-Dickstein, “Deep knowledge tracing.,” *Advances in Neural Information Processing Systems 28,* pp. 505-513, 2015.

[5] S. Bull and J. Kay, “Smili: a framework for interfaces to learning data in open learner models, learning analytics and related fields,” *International Journal of Artificial Intelligence in Education,* p. 26(1):293–331, 2016.

[6] Q. Zhao, F. M. Harper, G. Adomavicius and J. A. Konstan, “Explicit or implicit feedback? Engagement or satisfaction? A field experiment on machine-learning-based recommender systems,” *Proc. of the 33rd Annual ACM Symposium on Applied Computing,* p. 1331–1340, 2018.

[7] G. E. Rasch, “Probabilistic Models for Some Intelligence and Attainment Tests,” p. Volume 1, 1960.

[8] A. T. Corbett and J. R. Adnerson, “Knowledge tracing: Modeling the acquisition of procedural knowledge,” *User modeling and user-adapted interaction,* p. 4(4):253–278, 1994.

[9] J. Brank, G. Leban and M. Grobelnik, “Annotating documents with relevant Wikipedia concepts,” *Proc. of Slovenian KDD Conf. on Data Mining and Data Warehouses (SiKDD),* 2017.

[10] R. Herbrich, T. Minka and T. Graepel, “TrueSkill(tm): A Bayesian skill rating system,” *Advances in Neural Information Processing Systems 20,* 2007.

[11] “User modeling - Wikipedia,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/User_modeling. [Accessed 19 03 2018].