Health, demographic change and wellbeing

Personalising health and care: Advancing active and healthy ageing

H2020-PHC-19-2014

Research and Innovation Action

D4.6 User communities' creations based on user's profile matching (dynamic and adaptive profile).

Deliverable due date: 01.02.2017	**Actual submission date: 24.03.2017**
Start date of project: February 1, 2015	**Duration: 42 months**
Lead beneficiary for this deliverable: UNITN	**Revision: FINAL**
Authors: Maurizio Marchese (UNITN), Marcelo Dario Rodas Britez (UNITN), Ivo Ramos(ATOS), Ingo Brauckhoff (ATOS)	
Internal reviewer: Luigi Palopoli	

The research leading to these results has received funding from the European Union's H2020 Research and Innovation Programme - Societal Challenge 1 (DG CONNECT/H) under grant agreement n°643644

Dissemination Level		
R	Restricted	
CO	Confidential, only for members of the consortium (including the Commission Services)	
PU	Public	**X**

Contents

Executive Summary 4
Introduction 5
1. User Profiles 5
1.1. Static profiles 5
1.2. Dynamic and adaptive profiles 9
2. Circles recommendations system 11
2.1. Circle Recommender System Design 11
2.2. ACANTO Overall Recommendation System Architecture 12
2.3. Circle Recommendation System Implementation and testing 14
2.3.1. Implementation details 14
2.3.2. Similarity Vector and Parameters 16
2.3.3. Testing 17
3. Relation with other work packages 27
Conclusion 28
Bibliography 29

Executive Summary

This document defines the implementation of the user communities' creation using a recommendation system for ACANTO definition of users' groups, i.e. "circles". In the document, we will describe the design, implementation and test of the algorithms used for the recommendations of circles and how they contribute to the evolution of the user profiles. This deliverable is connected also to the deliverable 4.8 as part of a broader architecture for the recommendation systems used in ACANTO.

In section 1, we first describe the current status of the model of the User Profile and then we recall the overall architecture of the ACANTO cloud infrastructure. This model and the architecture were based on the requirements from deliverable 2.4 and the infrastructure from WP7. In section 2, we present the design, context, implementation of the circle recommendation system and we provide the details of the first testing on an artificial dataset of ca. 30.000 user profiles.

Finally, we describe the testing environment and evaluate preliminary results of the algorithms for the implemented recommendation system. This information is captured, stored, analyzed and processed to support further development of the recommendation algorithms as well as its experimentation on real and dynamic datasets. At present the current implementation uses the artificial dataset of ca. 30.000 static user profiles specifically created for our testing purposes. The results of our circles recommendations are input to the Social Network for further interactions with the users and to the social activities recommendations systems from Task 4.4 (deliverable 4.8).

We want to note here that this deliverable has been released two months later than the original plan due mainly to: (i) the unavailability of real data; (ii) the subsequent need to define and create an artificial dataset to test the approach; (iii) some minor implementation issues due to the required training with the technical infrastructure selected for the ACANTO system in other work packages. However, we also want to point out that this delay does not have an important impact in the project since it does not map into substantial delays for the overall project timeline.

Introduction

The purpose of this document is to describe the design and the initial implementations in the overall ACANTO systems of the support to the initial creation of user communities'.

In this respect, our approach envisions first a persons' profiles matching and small group creation based on the affinity/similarity among user's profile using unsupervised or semi supervised recommendation models based on both static features and dynamic observation flowing from the CPSN. The output of these first recommendations will be used to bootstrap the creation of "circles", i.e. micro-communities of older adults with similar interests (and eventually constraints) to whom propose appropriate activities with a substantial social dimension and support to their execution These will be a first step to the support of incremental strategies to form the creation of sustainable communities.

In section 1, we first describe the current status of the model of the User Profile and then we recall the overall architecture of the ACANTO cloud infrastructure. In section 2, we present the design, context, implementation of the circle recommendation system and we provide the details of the first testing on an artificial dataset of ca. 30.000 user profiles. Finally, we describe the testing environment and evaluate preliminary results of the algorithms for the implemented recommendation system. This information is captured, stored, analyzed and processed to support further development of the recommendation algorithms as well as its experimentation on real and dynamic datasets.

1. User Profiles

1.1. Static profiles

The static user profile aims to collect data coming from different sources such as personal information, preferences, interests, mobility constrains and previous activities. The user profile model is the general description of the data representation and relationship for the users of the system. The main parts of the user profiles are the personal information (first name, family name, complete name, birthday, civil status, the social profile (likes, dislikes and preferences), and the mobility record (constraints and prescriptions). The Figure 1 - User profile schema, illustrates the main blocks of information used to describe a user in the ACANTO system. It represents the output of the modeling activity in the related deliverable 2.4 "*User, activity and environmental description*". For more details please see the specific deliverable 2.4.

The diagram uses a style similar to the UML class diagram, but actually reflects the graph database schema. The similarity is useful since *OrientDB* [1] describes its schema with classes as well. The color-key to the diagram is: light blue = vertices, green = edges, light red = embedded documents, purple = indices.

The initial user profile is generated by answering to a questionnaire and is enlarged and developed subsequently with information provided by formal caregivers, relatives and medical doctors.

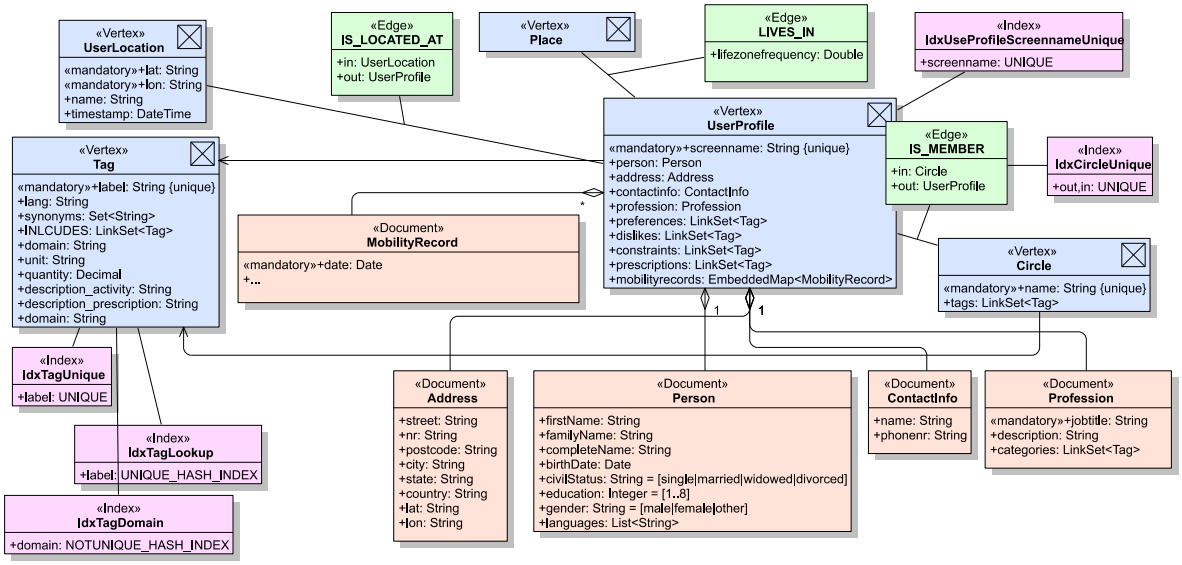


Figure 1. Graph Representation of the User Profile.

The graph representation is one mode to present information in a database, which is used in graph-databases such as *OrientDB* [1]. We use the graph representation because a social network is a natural use case for these systems. The representation is based on the graph theory, which characterizes 3 main elements: vertices, edges, and properties. The **vertices** are the main entities, the **edges** are the relationships within nodes, and the **properties** are characteristics that describe these two element types.

The user profile decomposes in the following way: a Vertex of type (or class) ***UserProfile*** with properties resembling embedded documents that are individually specified by their respective classes: *profession*, *person*, *address*, *contactinfo*, etc. *OrientDB* allows not only to store properties with a primitive type, like integer values or strings, but also whole embedded documents, links, sets, etc. [1].

It is worth mentioning the ***MobilityRecord*** structure, which is embedded into the *UserProfile* as a Map (*mobilityrecords* property) whose keys state the date of the day the record is related to in '**`yyyymmdd`**'-format. The linked document of class *MobilityRecord* has a property *date* which matches the key of the Map. The *MobilityRecord* will hold the data from the sensors and will be used to derive some statistics from these.

Circles are used to group *UserProfile*s and state only the name of the group as a property. The grouping is realized by connecting one or more *UserProfile* nodes to the *Circle* node through ***IS_MEMBER*** edges. A *UserProfile* can be part of a *Circle* only once – this is guaranteed by the unique index ***IdxCircleUnique***, which is generated on the *in* and *out* properties of the *IS_MEMBER* edges.

Associated to the *UserProfile* via the *IS_LOCATED_AT* edge is the *UserLocation*. It holds geolocation information *lat* (latitude) and *lon* (longitude) as well as a timestamp. This type of nodes will be useful to track the current location of a user.

Figure 2 – User Profile relations shows the relationships, marked by edges in the graph, of the *UserProfile* nodes between instances of the same type. At present, we defined the following types of edges:

- ***FOLLOWER_OF*** and ***CONNECTED_TO*** describe the network of followers and connections of the *UserProfile*. While *FOLLOWER_OF* is a unilateral relation, *CONNECTED_TO* resembles a bilateral or 'friendship'/'family-member' relation.

- ***CAREGIVER_OF*** will be used to give special rights to view and/or alter the contents of this connected *UserProfiles*. *CAREGIVER_OF* edges originate from *UserProfiles* which take the role of a doctor, therapist or nurse, and the amount of data that can be revealed this way will be adjusted accordingly. A doctor for example will have the amplest view on the collected data, while a simple caregiver might just be allowed to query the current location of the patient.

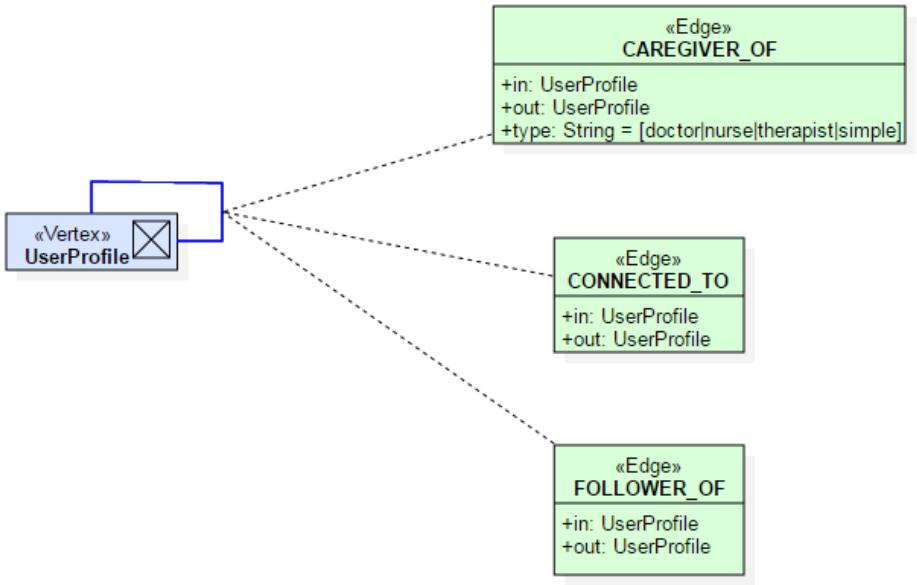


Figure 2: User profile relations

***Tag* nodes:** the tags are used to classify or categorize the nodes they are linked to, and therefore tags provide a semantic layer. They are used by the recommender systems to match comparable *UserProfiles*, to group them into *Circles* and to find appropriate *Activities* for a particular user. The important feature about them is their concise way to describe an activity or a person with just one or two words, easily to understand and interpret and most importantly, to limit the amount of different descriptions used to a smaller and controllable vocabulary.

Table 1. Tags attributes.

label	`String`	the name of the tag, semantic purpose
lang	`String`	the language in which the label is given; optional
synonyms	`Set<String>`	collection of labels with the same meaning
domain	`String`	a domain name to separate different applications of tags, defaults to "" (empty) as free domain and can take

		value "mobility" for use with the mobility profiling system.
description_activity	`String`	*mobility* domain only: an explanatory description for the tag, in contrast to the concise way of the label. This description will be used when tagging *Activity* nodes.
description_prescription	`String`	*mobility* domain only: an explanatory description for the tag, in contrast to the concise way of the label. This description will be used when tagging *UserProfile* nodes.

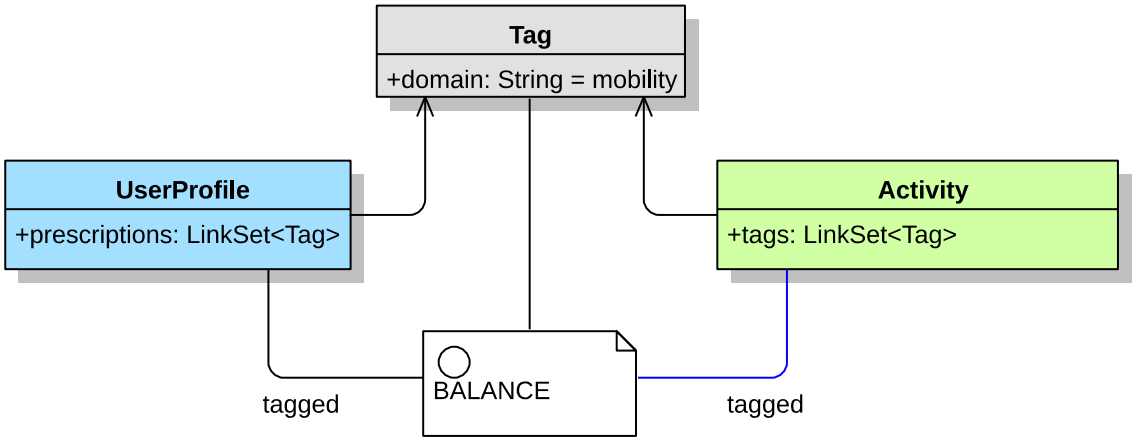


Figure 3. User Profiles, Tags and Activity relational representation.

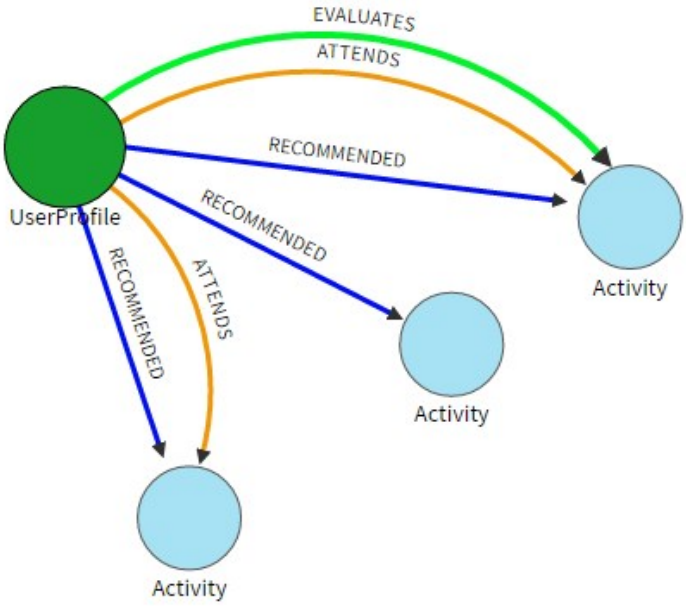


Figure 4. User Profile relations with activities.

Figure 3 illustrates the main architectural relationships between Tags and User Profiles and Activities, while Figure 4 provides the foreseen main relations between User Profiles and Activities, namely:

- RECOMMENDED: when a specific activity is recommended to a UserProfile (User) based on its preferences, *MobilityRecord, position etc.*
- ATTEND: when a User attends a recommended activity
- EVALUATE: when a User provides some feedbacks and evaluation of the attended activity

Knowledge Base

The Knowledge Base is the infrastructure that aggregates data from user profiles, circles, tags, evaluations, environments and activities, as shown in the Figure 5 - General Knowledge base representation. This package is used to differentiate the classes used for the recommendation system. The Knowledge Base will be accessible to the other components of the ACANTO information system through a Representational State Transfer Application Programming Interface (REST API) [2] or native APIs for a variety of programming environments [3], allowing easy re-use, scalability and support to a number of evolving services.

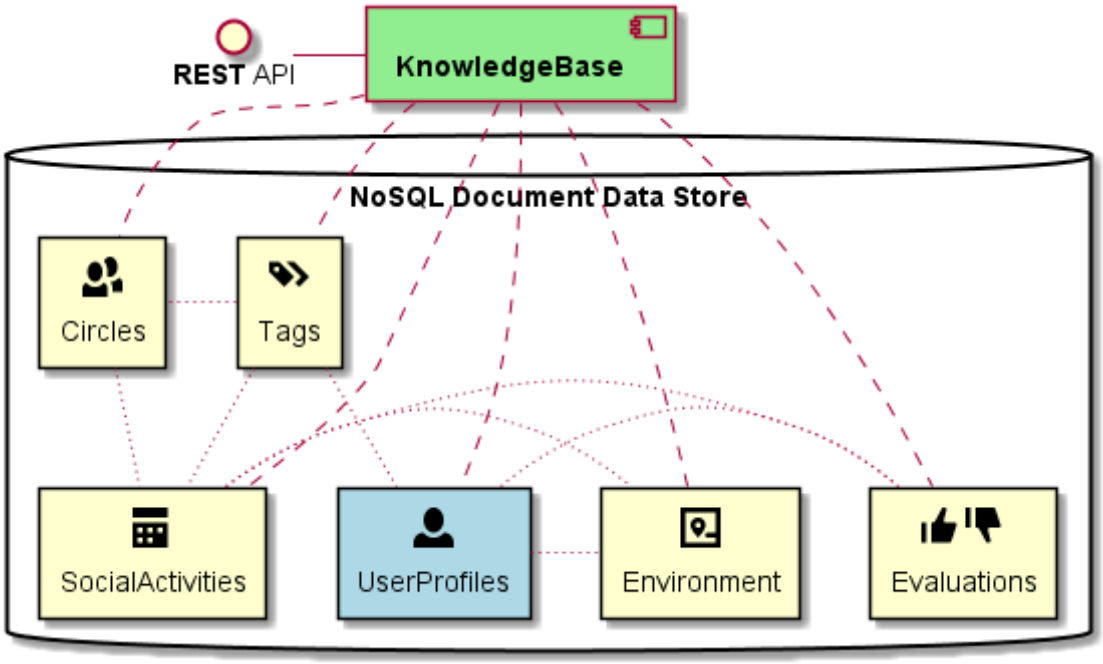


Figure 5. General Knowledge base representation.

1.2. Dynamic and adaptive profiles

An important element of the final circle recommendation system is the relationships between the UserProfiles model and the use of the dynamic data collected from other elements of the ACANTO CPSN in order to evolve the recommendations and adapt them through observations and data collected from the CPSN.

This can be envisioned not only in the use cased involving the FriWalk (equipped with a number of specific monitoring sensors) but also in a more open vision that adopts the same cyber-physical paradigm and create a CPSN-enabled ecosystem of devices (e.g., smart glasses, smart watches, smart headphones for visually impaired users) that share the same technology of the FriWalk. As long as these devices have some degree of

sensing and perceptual abilities, they can in principle be used to harvest information that can be used for the definition of the user profile.

As an example, dynamic location data can be collected using smart phones – provided that the users install a specific app and agrees on the sharing of their position data - and used to defined "life zones", i.e. regions inside the environment in which prominently the users carry out their daily social activities.

This information is capture in the Environment model that represent the definition of the "life zones", is related to a particular User Profile and is stored in the Knowledge base as indicated in Figure 6. For more details on the Environment and definition of LifeZones, please refer to deliverable 2.4 "User, activity and environmental description".

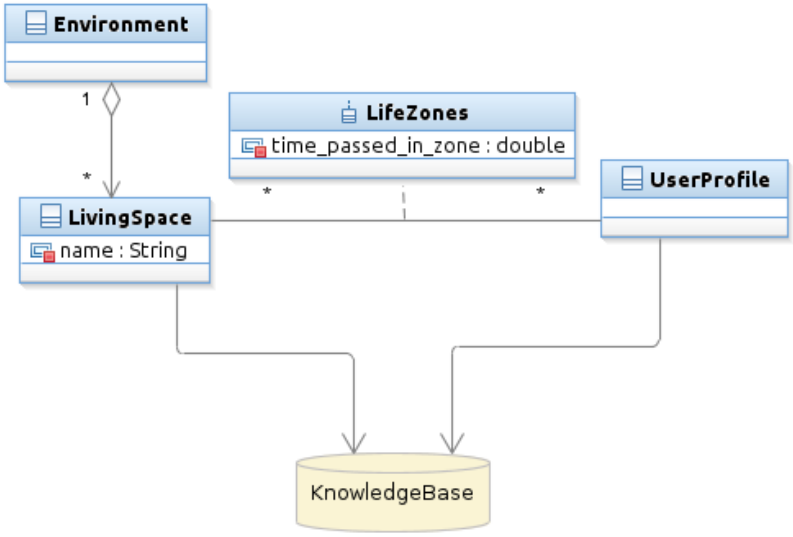


Figure 6: Life zones, users profile and their connection with the knowledge base

Other dynamic information will also be collected by monitoring the actual participation to the circles activities as well as through the collection of feedbacks to both the proposed circles and social activities.

When this information will become available during the experimentation phase, our circle recommendation system will be able to use it in order to adapt and evolve its recommendations dynamically.

For instance, if our two users - let's say Isabel and Martha, to maintain some persona from use cases presented in other work packages - spend most of their time in the same life zone, as long as they have compatible profiles, it would make a lot of sense to proposed them to participate to same circle, although their preferences overlap is not optimal. On the other hand, if the two users - Isabel and Michael - have compatible interests but their life zones are far away, their participation to the same circle is probably pointless, unless other considerations suggest otherwise (e.g., the users' willingness to be part of the same circle).

We plan to investigate the implementation of these dynamic information mainly through

post-processing (e.g. filtering techniques) of the circles' recommendations.

2. Circles recommendations system

In this section, we describe the design, implementation and testing of the algorithms used for the circles recommendation sub-system, which is part of the overall ACANTO recommendation system.

The circle is ACANTO specific concept of "user's group", used in ACANTO to target activities with a social dimension to similar user's group. Circles are first generated automatically between people with compatible profiles and with common interests. Then they evolve through the interaction of the involved users and the data collected during and after the activities (feedbacks, evaluations, etc.)

2.1. Circle Recommender System Design

The circle recommender system is designed to use the data architecture, techniques, and the Apache Mahout's [https://mahout.apache.org/] framework to build scalable and performant machine learning applications. In particular, we have used in our implementation work the Apache Mahout's pre-implemented recommender algorithms.

Mahout provides several types of recommender engines like content-based or collaborative filtering. The big data cluster provided by WP7 presently counts on four virtual machines for analysis of the data in parallel. The cluster supports the Hadoop stack, that can scale well up to thousands of nodes, working in parallel, to enable recommendations even in near real time.

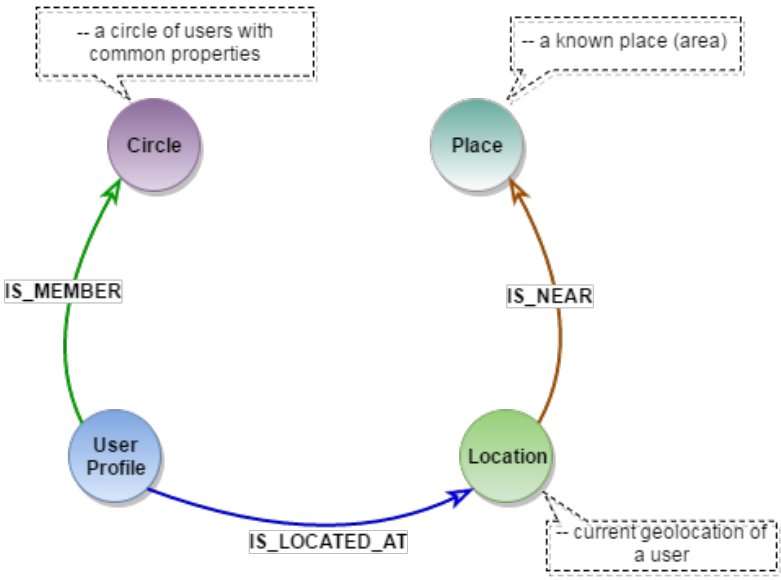


Figure 7. Vertex relations of the User profile, Circle, Location and Place.

In regard to the information structure, the figure 7 shows a proprietary diagram which shows the vertex relationships between *UserProfile*, *Circle*, *Location* and *Place*. Circles are nodes specifically used to group *UserProfiles*.

A *UserProfile* can be in principle a member of one or more circles (declared by IS_MEMBER typed edge) and is associated with a Location node that holds information

about the current geolocation of a user. The *synonyms* property collects *Tags* with the same semantic meaning, for example in another language. A *language* property states the language the tag is labelled in. The activity recommender system developed in Task 4.4 (see Deliverable 4.8) uses this information of *UserProfiles* and *Circles* to generate its social activity recommendations.

2.2. ACANTO Overall Recommendation System Architecture

The circle recommendation system is a component of the overall ACANTO Recommender System depicted in the Figure 8. In fact, one of the purpose of the proposed concept in ACANTO of a Cyber Physical Social Network (CPSN) is to support the creation and evolution of a community of users. The user's information is collected within the CPSN to extract information and patterns that may indicate behaviors, preferences, constraints, satisfaction level and opportunities related with the environment. The physical observation results will be combined with information provided initially by the user in order to bootstrap the recommendation system.

Using the available information, initial connections are proposed among the users with the creation of the concept of circles as described in the previous section. Then, some activities will be proposed to the different user *circles*, the execution of activities will be supervised (automatically by sensors where available as well as by users' feedbacks and evaluations) and data will be collected to define the satisfaction level and to deliver improved recommendations for similar or different activities (depending on the collected data).

Figure 8 shows the main components of the developed ACANTO CPSN Infrastructure Architecture where the circle recommendation system is included. It includes:

(1) **Liferay**

Liferay is a portal platform used as frontend for the CPSN to provide access to the social network and its features, such as forum, private messaging, chat, etc. – easily integrated via available portlets from the *Liferay* marketplace. *Liferay* comes with an internal database (in this case: *MySQL*) for its metadata; basic user information and login data will be stored here also [4].

(2) **KnowledgeBase: OrientDB.**

It is a flexible open source NoSQL document database, where we will store the user profiles, circles, activities, evaluations, environments, etc. The OrientDB is very fast (120k writes/s) and provides a full set of features, most notably SQL support and REST API [5].

(3) **Recommender System: Hadoop HDFS and Hive**

The *Hadoop Distributed File System* (HDFS) offers a way to store large files across multiple machines. The *Apache Hive* data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. A schema can be projected on to the data already in storage. A command line tool and a *Java Database Connectivity* (JDBC) driver are provided to connect to Hive.

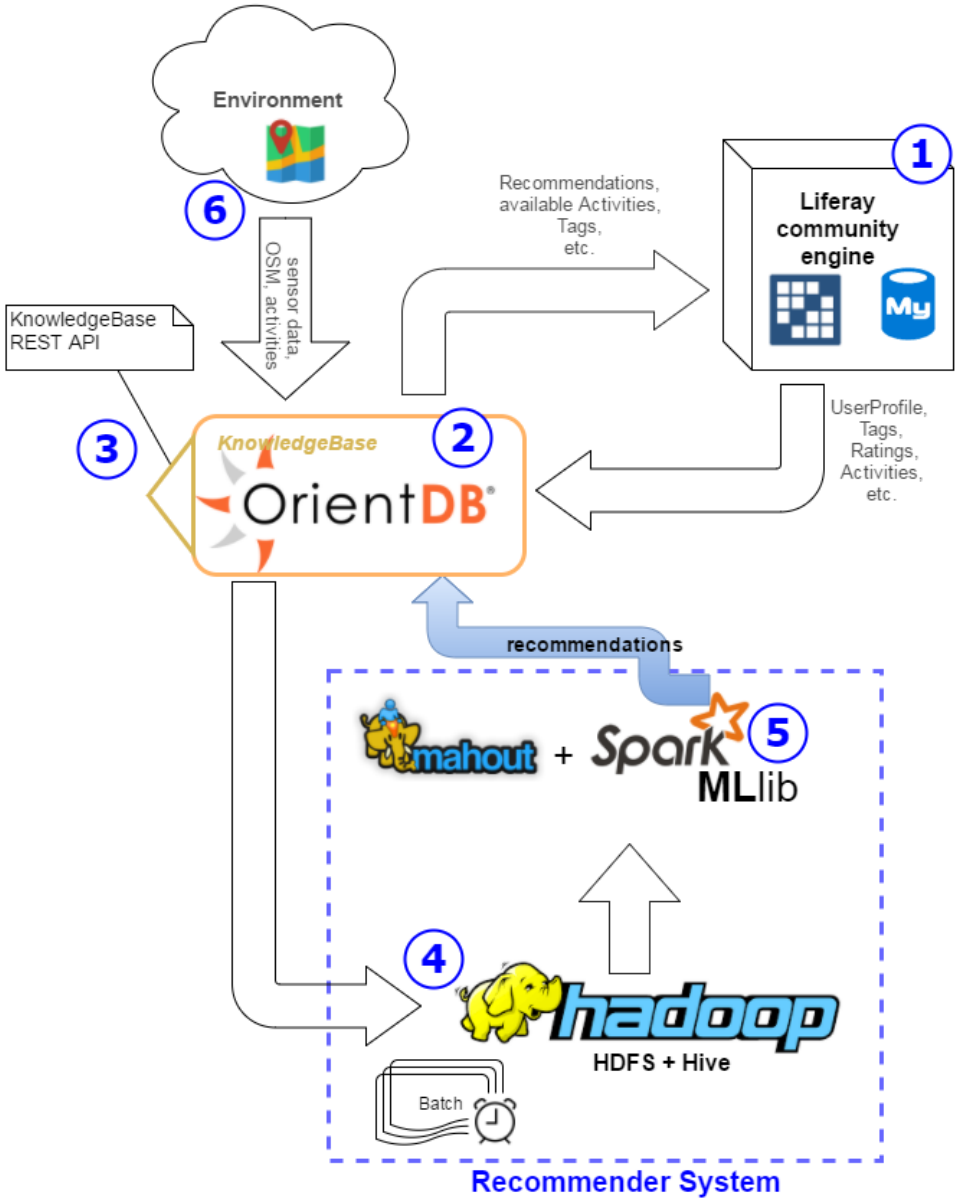


Figure 8. Architecture of the CPSN, and main Data Flow streams.

(4) **Recommender System: Recommendations, Mahout + Apache Spark MLlib** will be used to crunch the data and make the recommendations (circles, activities) The Apache Spark MLlib is Spark's machine learning (ML) library. Its goal is to make practical machine learning scalable and easy. It consists of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as lower-level optimization primitives and higher-level pipeline application programming interfaces (API). Apache Mahout is an additional framework for scalable machine learning algorithms. The output of the recommender system will be stored in the *KnowledgeBase* and can be displayed, e.g., in a portlet.

(5) **Environment**
Data ingestion from diverse sources like Activity Harvester, Open Street Map, sensors, etc. [2] The external data is conceived as a continuous event stream that

can be ingested by the CPSN. The events can be location updates, metrics from FriWalk or new social activities. The environment is modelled as a metric map with semantic information connecting the places where the activities can be executed. It can be composed by the "Life zones" which is the region inside the environment where the user carries out the daily activities. The life zone is defined when the user profile is created and the environment is set for a particular user, storing the information into the *KnowledgeBase*.

2.3. Circle Recommendation System Implementation and testing

The circle recommendation system works by executing first a clustering algorithms over the users of the system. The initial clustering algorithm we have used for a first test of the system is the well-known k-means algorithm and we have initially used in our tests the "preferences" tags of the users to build the similarity vector between users.

2.3.1. Implementation details

The general architecture of the circle recommendation system, basically follows the main three steps of the majority of clustering algorithms:

1. Pre-processing of the information: where the main features of the clusters are selected and pre-processed and transformed (if needed);
2. Execution of the clustering algorithm.
3. Post-processing of the results.

The pre-processing step is a fundamental step since normally the original data structure for managing the information is not in a proper format for being used as a similarity vector as needed for the majority of clustering algorithm, including k-means.

In our work, we have initially chosen to use the preferences tags selected by the users as the main evaluation criteria to study and demonstrate the feasibility of the approach. The selected clustering algorithm is the k-means algorithm [6], which is a simple but well-known algorithm for clustering. In brief, all objects need to be represented as a set of numerical features. In addition, the user has to specify the number of groups (referred as k) he wishes to identify.

In order to better present the k-means algorithm, in Table 2 we present examples of the similarity vectors of 4 users and used them in the following as a running example for describing the working of the clustering algorithm.

The basic steps for the k-means algorithm are the following:

1. Select initial k-centers which are the initial centers of the circles;
2. Compute the distances from all vectors to all k-centers;
3. Assign each vector to the nearest k-center;
4. Compute the average of all points assigned to all specific k-centers.
5. Replace the k-centers with the new averages;
6. Repeat the above steps till the end of a specific iteration value or up to attainment of a defined convergence criteria.

Table 2. Example of preferences representation (similarities vectors) for the clustering algorithm for four users's profile

	Preferences								
User	Tag01	Tag02	Tag03	Tag04	Tag05	Tag06	Tag07	Tag08	Tag09
U1	1	1	1	0	0	1	0	0	0
U2	1	1	1	0	0	1	0	0	0
U3	1	0	0	1	1	0	1	0	1
U4	1	0	0	1	1	0	1	1	0

The initial centers are selected randomly by the algorithm in the first step. Another very important value to calculate is the value k, that is the number of searched cluster. Its estimation depends on the specific semantic of the clustering problem. In our case, the meaning of the number of clusters we are seeking, is linked both to the total number of user profiles in our used data set and by the "ideal" average circle size (i.e. the average number of users present in a given circle). For our initial implementation and testing, our definition of the number of clusters (k) has been simply the division of the total number of users' profiles by a defined ideal circle size.

To follow the k-means algorithm execution in the simple running example we use here the data collected in Table 2. First, we need to define a distance measure and a k, for example squared Euclidean distance and k=2 respectively. With this definition, we follow the basic steps for the k-means algorithm:

- First, we select randomly U1 (circle1) and U3 (circle2) as the 2 starting centers of our clustering, i.e. circle1 {1,1,1,0,0,1,0,0,0} and circle2 {1,0,0,1,1,0,1,0,1}
- We then compute the Euclidean distances of U1 with all others (U2, U3, U4) and get the following results: 0, 7, 7.
- We also compute the Euclidean distances of U3 with the others (U1, U2, U4) as follows: 7, 7, 1.
- Then we group by the nearest distances, creating circle1 as (U1, U2) and circle2 as (U3, U4).
- The new center for circle1 is: {1,1,1,0,0,1,0,0,0}
- The new center for circle2 is: {1,0,0,1,1,0,0.5,0,0.5}
- Since the centers did not change significantly we may decide to stop the algorithm with the two-created circle (more on this convergence issue in the next section).

It is important to notice that we could add other parameters to the preferences vector shown in Table 2 like the user's address (latitude and longitude), or even the centroid of dynamically computed life zones as describe in the Section 1.3. The condition for adding more similarity variables is that the values of these new variables, between the users, should be measurable by a distance function.

The post-processing step is necessary to apply the inverse processing of the information done in the pro-processing in order to write back the created circles to the database with the initial user profiles features. In this way, the structure of the data models used in the overall system is maintained and is available to be displayed to the different users in the Social Network.

This post processing phase includes also an important process of the recommendation process, i.e. the construction of a comprehensible “name” of the cluster. K-means assign numbers as identifiers to the generated clusters, so we need to implement a semantic for building a more relevant and understandable name. We have defined the cluster name by calculating the percentage frequency of the preference in a cluster, and selecting the two more frequent preferences that we concatenate adding at the end the string “circle”. This implementation could be adapted and extend (e.g. use of more tags etc.) to provide additional information regarding the computed circles. In the testing section, we provide some examples of this naming process.

The accuracy of the recommendations will improve the more data is available to be analyzed. The recommender system reads the data from OrientDB KnowledgeBase and stores it in HDFS [6] as RDD [7] in order to be available to the Hadoop cluster. A batch process is run periodically (say, every 24 hours) to include new information for evolving the recommendations.

2.3.2. Similarity Vector and Parameters

The similarity vector is the data structure that represent the characteristics we want to compare within users to define the proximity between them. Each user can be thought of as being represented by a feature vector in an n dimensional space, n being the number of all features used to describe the objects to cluster. As a starting point, the features we have taken into account are the tags “preferences” of the users present in the User Profile model (see Figure 1). In future development and tests, other tags could be considered as well as other featured present in the User Profile model, such as Place, UserLocation and/or MobilityRecord. Moreover, as mentioned in Section 1.3 also information that is related to the UserProfile and collected in other elements of the ACANTO CPNS (such as information of the Life Zones of the users or monitoring the actual participation to the circles activities as well as through the collection of feedbacks to both the proposed circles and social activities) could be use in the definitions of the relevant features of the user’s profiles. This would allow a fine tuning of the circles recommendations when and if needed.

To implement the current feature vector, we create for each user profile a vector of the size of the full list of preferences tags in the database, where we save the presence or not of each specific preference in the specific user profile.

To execute the k-means clustering algorithm Mahout implementation, we need to define three types of parameters: parameters related to OrientDB, parameters related to input/output files for Hadoop, and parameters related to the k-means algorithm Mahout implementation. The first two types of parameters are infrastructure dependent, are related to technical aspects of data processing and do not affect the clustering process, while the k-means algorithm Mahout implementation have an impact on the clustering results. We collect the last type of parameters in Table 3.

Table 3. Parameters for the k-means algorithm Mahout implementation.

Parameter	Examples of values
Distance Measure	Squared Euclidean
K (number of clusters)	310; 620; 1240;
Convergence Delta	1.0; 3.0; 5.0;
Iteration Number	500;
Classification Threshold	0; 0.0001; 0.001

A brief description of the k-means algorithm Mahout implementations parameters is provided in the following:

- *Distance Measure*: the measure used to compute the distance between two vector points in our n-dimensional space;
- *Number of clusters* (K): indicates the number of cluster to be created;
- *Convergence Delta*: it indicates the minimum value of variation of the distance from the clusters' centers from one iteration to the next. It thus defines the convergence parameter for the specific clustering execution. When reached the algorithm stops;
- *Iteration Number*: it is the limit value of the k-mean algorithm iterations. When reached the algorithm stops;
- *Classification threshold*: it indicates the strictness / outlier removal parameter (value between 0 and 1). The more this parameter is set close to 1, the stricter will be the rule for considering a user to a cluster.

2.3.3. Testing

To test the recommendation system, we use the same artificial dataset presented in Deliverable 4.8 – Social Activity Recommendations - and based on different datasets from UCI machine learning repositories. More specifically, a number of UserProfiles were created from "merge world cities", "born name" and "yelp academic" datasets. The main fields were extracted from Adult Database [9]. The artificial dataset contains about 31.000 different User Profiles with their respective preferences, random likes and dislikes.

For the creation of the artificial dataset, each user profile was associated with activities tags in order to create useful data also for the social activity recommender system.

To have a better understanding of the used artificial dataset, we present some statistics of this dataset in the Table 4 and Table 5. Table 4 shows the general size of the dataset (users and preferences). Table 5 shows specific statistics on the use of Preferences Tags by users.

Table 4. Statistical Variables in the Dataset.

Variable	Value
# of Users	31781
# of Preferences Tags	19
Minimum # of Preferences Tags per User	3

Average # of Preferences per Tags User	4,5
Maximum # of Preferences per Tags User	5

Table 5. Statistical values of the Preferences Tags in the Dataset.

RID	Label	# of Users with this preference	% of Use
#22:10	Fishing	16565	52,12%
#24:27	Sewing	11964	37,65%
#24:21	Painting	6755	21,25%
#23:27	Card and board games	6690	21,05%
#22:20	Scrapbooking	6689	21,05%
#24:28	Dances	6678	21,01%
#21:29	Educational programs	6632	20,87%
#21:10	Exercise, yoga or tai chi classes	6626	20,85%
#22:2	Picnics	6606	20,79%
#22:28	Photography	6591	20,74%
#22:30	Trips	6590	20,74%
#22:4	Treasure hunts	6588	20,73%
#21:1	Lectures	6578	20,70%
#23:22	Drawing	6574	20,69%
#22:12	Arts and crafts	6540	20,58%
#22:11	Gardening	6519	20,51%
#23:19	Crocheting	6506	20,47%
#24:18	Knitting	6506	20,47%
#22:1	Support services and resources for seniors	6503	20,46%

Table 6 shows a generic exemplar - for description purpose - of a recommended circle of size 5 (in order to be easily readable), using the full population output of a given clustering execution.

Table 6. Example of a generated Circle.

Circle Information (Circle number 34):

- Name = Exercise, yoga or tai chi classes-Scrapbooking circle
- Total number of users = 5
- Total number of used preferences = 12
- Total number of not used preferences = 34

Preferences Information of the Circle:

- Preferences and total frequency in the Circle (size=12):
  1. #20:23_Scrapbooking=5;
  2. #19:17_Exercise, yoga or tai chi classes=5;
  3. #17:18_Educational programs=4;
  4. #19:10_Treasure hunts=3;
  5. #18:5_Painting=1;
  6. #17:6_Card and board games=1;

7. #19:5_Trips=1; 8. #20:17_Photography=1; 9. #17:24_Knitting=1; 10. #17:23_Lectures=1; 11. #17:17_Support services and resources for seniors=1; 12. #17:11_Sewing=1;
Preferences Information of one element (User #37:38) of the Circle (size=5): • Preferences and percentage of users with this preference: 1. Scrapbooking=100%; 2. Exercise, yoga or tai chi classes=100%; 3. Educational programs=80%; 4. Treasure hunts=60%; 5. Card and board games=20%;
Preferences Information of one element (User #37:85) of the Circle (size=5): • Preferences and percentage of users with this preference: 1. Scrapbooking=100%; 2. Exercise, yoga or tai chi classes=100%; 3. Treasure hunts=60%; 4. Photography=20%; 5. Support services and resources for seniors=20%;
Preferences Information of one element (User #37:471) of the Circle (size=5): • Preferences and percentage of users with this preference: 1. Scrapbooking=100%; 2. Exercise, yoga or tai chi classes=100%; 3. Educational programs=80%; 4. Knitting=20%; 5. Sewing=20%;
Preferences Information of one element (User #37:553) of the Circle (size=5): • Preferences and percentage of users with this preference: 1. Scrapbooking=100%; 2. Exercise, yoga or tai chi classes=100%; 3. Educational programs=80%; 4. Trips=20%; 5. Lectures=20%;
Preferences Information of one element (User #37:697) of the Circle (size=5): • Preferences and percentage of users with this preference: 1. Scrapbooking=100%; 2. Exercise, yoga or tai chi classes=100%; 3. Educational programs=80%; 4. Treasure hunts=60%; 5. Painting=20%;

A number of preliminary tests have been focused to understand how the variation on the variables convergence delta and classification threshold are affecting the results.

Another important consideration about the implemented algorithm, is that for processing the testing dataset with the parameters shown in table 3, the running time was about a few minutes (2-5 minutes).

We have started to explore the effect of the classification threshold using the values reported in Table 3. This study informed us on the stability of the algorithm by providing

the same solutions in the range 0.0 - 0.001. If we increase classification threshold to 0.01 (or greater) the algorithm creates just a single circle.

A second investigation was related to the variation of the number of clusters (k) by considering an ideal circle size equal to 25 (k=1240), 50 (k=620), 100 (k=310). For a visual analysis of the created clusters we include:

- in Table 7 a specific example of a recommended circle with k=1240
- in Table 8 shows a specific example of a recommended circle with k=620

Table 7. Recommended Circle with k=1240.

Circle Information (Circle number 65): • Name = Sewing-Exercise, yoga or tai chi classes circle • Total number of users = 25 • Total number of used preferences = 6 • Total number of not used preferences = 13
Preference frequency in cluster: {#22:30_Trips=25; #21:10_Exercise, yoga or tai chi classes=25; #24:27_Sewing=25; #22:10_Fishing=19; #22:2_Picnics=5; #21:29_Educational programs=2;}
User= #49:2544 Preferences Info [size=3 (100%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; }]
User= #49:3125 Preferences Info [size=3 (100%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; }]
User= #49:4202 Preferences Info [size=3 (100%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; }]
User= #49:4604 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #49:4618 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #49:5113 Preferences Info [size=3 (100%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; }]
User= #49:5986 Preferences Info [size=5 (79.2%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; Picnics=20%; }]
User= #50:853 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #50:1874 Preferences Info [size=3 (100%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; }]
User= #50:3459 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #50:6575 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #50:7167 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #50:7516 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #51:879 Preferences Info [size=5 (79.2%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; Picnics=20%; }]
User= #51:3438 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]

User= #51:5859 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #51:7744 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #52:1142 Preferences Info [size=5 (79.2%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; Picnics=20%; }]
User= #52:1418 Preferences Info [size=5 (76.8%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; Educational programs=8%; }]
User= #52:1442 Preferences Info [size=5 (79.2%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; Picnics=20%; }]
User= #52:3573 Preferences Info [size=5 (79.2%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; Picnics=20%; }]
User= #52:5950 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #52:6760 Preferences Info [size=4 (94%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; Fishing=76%; }]
User= #52:7048 Preferences Info [size=3 (100%); %-InCluster = {Trips=100%; Exercise, yoga or tai chi classes=100%; Sewing=100%; }]

Table 8. Recommended Circle with k=620.

Circle Information (Circle number 109): • Name = Arts and crafts-Gardening circle • Total number of users = 50 • Total number of used preferences = 16 • Total number of not used preferences = 3
Preferences frequency in cluster: {#22:11_Gardening=50; #22:12_Arts and crafts=50; #22:10_Fishing=47; #21:10_Exercise, yoga or tai chi classes=40; #22:28_Photography=10; #24:28_Dances=5; #23:27_Card and board games=4; #22:20_Scrapbooking=3; #22:4_Treasure hunts=3; #22:30_Trips=2; #24:21_Painting=2; #23:19_Crocheting=2; #21:1_Lectures=2; #22:1_Support services and resources for seniors=2; #24:18_Knitting=1; #21:29_Educational programs=1; }
User= #49:111 Preferences Info [size #=4 (93.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; }]
User= #49:1126 Preferences Info [size #=4 (93.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; }]
User= #49:1423 Preferences Info [size #=4 (78.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Photography=20%; }]
User= #49:2626 Preferences Info [size #=5 (75.6%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Crocheting=4%; }]
User= #49:3262 Preferences Info [size #=5 (75.6%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Lectures=4%; }]
User= #49:3308 Preferences Info [size #=5 (75.6%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Trips=4%; }]
User= #49:3473 Preferences Info [size #=4 (75%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Exercise, yoga or tai chi classes=80%; Photography=20%; }]
User= #49:5453 Preferences Info [size #=5 (75.6%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Support services and

resources for seniors=4%; }]
User= #49:5744 Preferences Info [size #=5 (76.4%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Card and board games=8%; }]
User= #49:6123 Preferences Info [size #=3 (98%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; }]
User= #49:6979 Preferences Info [size #=5 (75.6%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Crocheting=4%; }]
User= #49:7135 Preferences Info [size #=5 (75.6%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Support services and resources for seniors=4%; }]
User= #49:7841 Preferences Info [size #=4 (78.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Photography=20%; }]
User= #50:427 Preferences Info [size #=5 (75.6%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Painting=4%; }]
User= #50:1236 Preferences Info [size #=4 (78.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Photography=20%; }]
User= #50:1491 Preferences Info [size #=3 (98%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; }]
User= #50:2627 Preferences Info [size #=5 (76.4%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Card and board games=8%; }]
User= #50:3020 Preferences Info [size #=5 (76%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Scrapbooking=6%; }]
User= #50:3487 Preferences Info [size #=4 (93.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; }]
User= #50:3565 Preferences Info [size #=3 (98%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; }]
User= #50:3575 Preferences Info [size #=5 (76.4%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Card and board games=8%; }]
User= #50:3950 Preferences Info [size #=5 (76.8%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Dances=10%; }]
User= #50:5515 Preferences Info [size #=5 (76.4%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Card and board games=8%; }]
User= #50:7510 Preferences Info [size #=5 (76%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Treasure hunts=6%; }]
User= #50:7538 Preferences Info [size #=4 (93.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; }]
User= #50:7714 Preferences Info [size #=5 (76.8%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Dances=10%; }]
User= #51:678 Preferences Info [size #=5 (76%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Treasure hunts=6%; }]
User= #51:3113 Preferences Info [size #=5 (78.8%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Photography=20%; }]
User= #51:3131 Preferences Info [size #=5 (76.8%), %-InCluster = {Gardening=100%; Arts and

crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Dances=10%; }]
User= #51:3169 Preferences Info [size #=3 (98%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; }]
User= #51:4628 Preferences Info [size #=5 (75.2%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Educational programs=2%; }]
User= #51:5788 Preferences Info [size #=4 (75%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Exercise, yoga or tai chi classes=80%; Photography=20%; }]
User= #51:5937 Preferences Info [size #=5 (76.8%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Dances=10%; }]
User= #51:5967 Preferences Info [size #=4 (93.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; }]
User= #51:6339 Preferences Info [size #=5 (75.6%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Painting=4%; }]
User= #52:120 Preferences Info [size #=5 (78.8%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Photography=20%; }]
User= #52:129 Preferences Info [size #=4 (78.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Photography=20%; }]
User= #52:169 Preferences Info [size #=4 (93.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; }]
User= #52:730 Preferences Info [size #=4 (93.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; }]
User= #52:1156 Preferences Info [size #=4 (78.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Photography=20%; }]
User= #52:2130 Preferences Info [size #=5 (76.8%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Dances=10%; }]
User= #52:2555 Preferences Info [size #=5 (75.6%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Trips=4%; }]
User= #52:4672 Preferences Info [size #=5 (76%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Scrapbooking=6%; }]
User= #52:5234 Preferences Info [size #=5 (75.6%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Lectures=4%; }]
User= #52:6263 Preferences Info [size #=5 (76%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Treasure hunts=6%; }]
User= #52:6275 Preferences Info [size #=5 (75.2%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; Knitting=2%; }]
User= #52:7025 Preferences Info [size #=3 (98%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; }]
User= #52:7361 Preferences Info [size #=4 (75%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Exercise, yoga or tai chi classes=80%; Photography=20%; }]
User= #52:7526 Preferences Info [size #=4 (93.5%), %-InCluster = {Gardening=100%; Arts and crafts=100%; Fishing=94%; Exercise, yoga or tai chi classes=80%; }]

Finally, to compare in more analytic form the results obtained with the three-different k, we define a simple quality parameter "Quality of a Cluster (***QC***)" as follows:

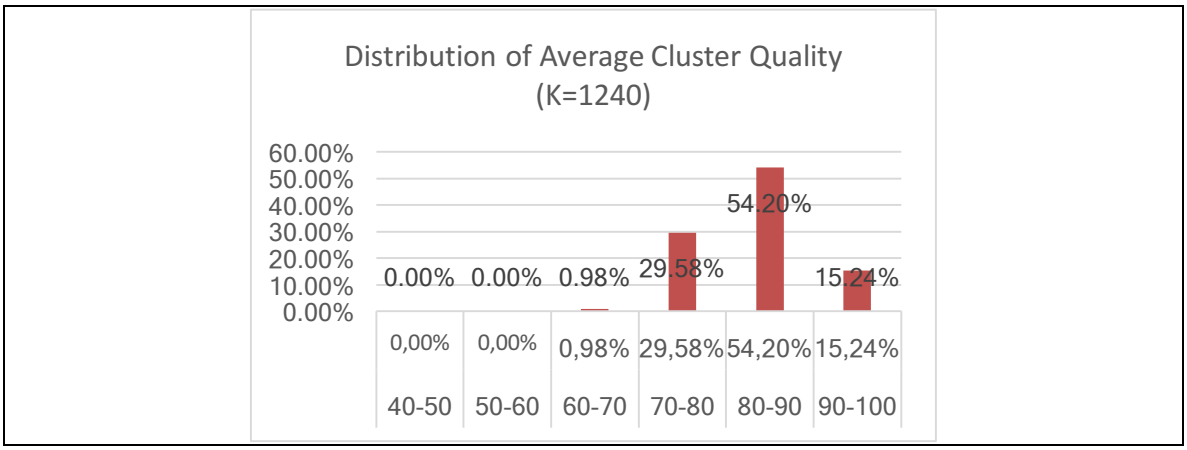
- we first compute the average percentages of occurrences of the preferences of a user profile in a circle and define this average as belonging percentage of a User (***q)***
- we then compute the average of the belonging percentages of every user profile of the circle and call it "Quality percentage of a Circle (***QC***)"

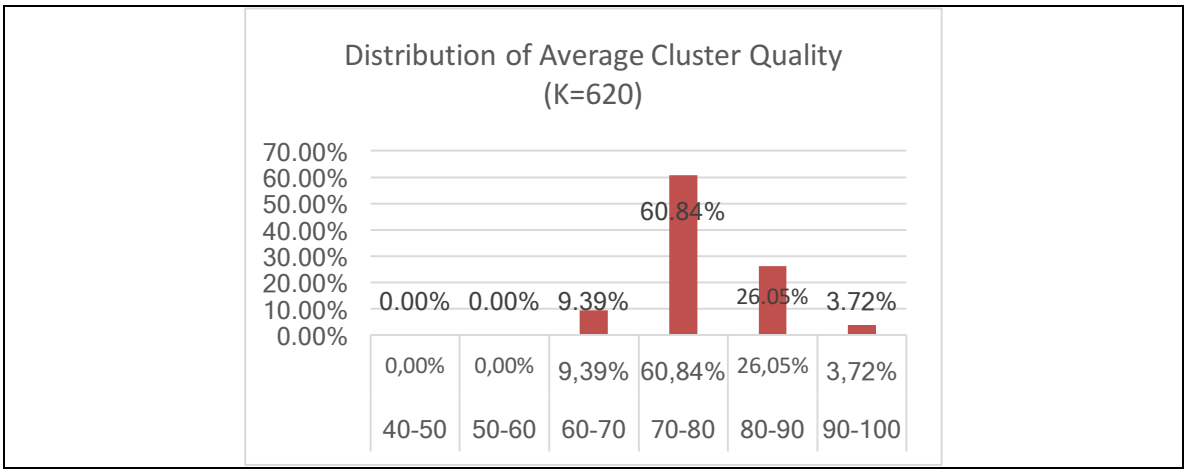
The variable ***q*** could be seen in the two tables examples above next to the preferences size as "(***q***%)", and is calculated for every user of a Cluster. The average number of these ***q*** correspond to our first simple estimation of the overall quality of a given cluster ***QC***.

The following diagrams - collected in Figure 9- are describing the QC distribution for the circles generated with the algorithm as a function of k. It is notable that ranges below 50% average quality are 0.0%. Moreover, with k = 1240 we obtain for a specific execution QC more than 70% overall quality precision. With k = 620 we obtain QC between 70% and 90%. Finally, with k = 310 we obtain QC between 60% and 80%.

What we find is thus that a larger number of k – which corresponds to smaller number of users per cluster – provides a better overall overlap between users' preferences (i.e. a small circle with a large number of common preferences). While a small number of k – i.e. larger number of users per cluster – provide more diversity among the users, with less overlap between users' preferences. This feature can be used to fine tune the recommendation of circles to specific needs of the Social Network. One may think to start with a circle compose of very similar users, but then in time – in order to support and provide diversity – can propose more numerous circles with a larger variety of interests.

In future work, we want to run the clustering algorithm on our test dataset a number of times in order to collect more statistics on the effect of the different clustering parameters.





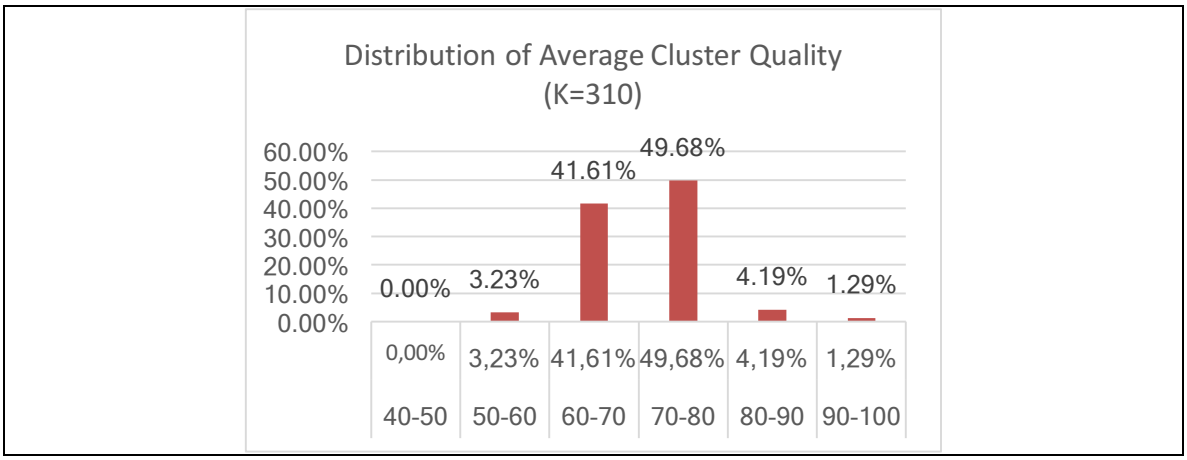


Figure 9: Average QC distribution for different k (number of clusters) parameter values

3. Relation with other work packages

The work presented in this document is based on the continuous model development carried out in WP2. Specifically, we have implemented and used the latest models that present the User Profile and Environment as described and discussed in details in Deliverable 2.4. We expect that these models will continue to evolve during the validation and testing phases of the different modules. The current models are in some sense "alive" and they are subject to possible updates until the end of the project.

The current implementation of the User Profile, Social Activities and Environment will also be the base for its evolution towards the inclusion of user's information coming from dynamic sources (i.e. sensing data coming from the cyber physical network based in the techniques developed in WP3, at task 3.2 – "Perception of the environment" as well as from the evolving social data coming from directly from the users' social network).

Moreover, the components developed in this work have been designed and implemented in compliance of the overall cloud ACANTO infrastructure developed WP7.

Finally, the work of this WP will be integrated and used in the experimental validation of WP8.

Conclusion

In this deliverable, we started by considering the updates made to the user profile model previously presented in the documents D4.1 and presented in more details in D2.4.

We, then described the first version of the user profile repository. In section two, we provided the architecture overview, the data flow and the description of the selected technologies being used during the development phase.

We detailed the process of implementation and test of the algorithms used for the recommendations of circles as well as the creation and implementation of the artificial dataset. The tests were analysed to understand in some details the obtained results, and to have a procedure for future testing.

In summary, we have implemented and shown the main properties of a first prototype for bootstrapping the creation of circles user communities. The current prototype is fast (2-5 minutes for a dataset of ca. 30.000 user profiles) and obtains consistent results, with high overall quality.

Bibliography

[1] OrientDB, "OrientDB," 2017. [Online]. Available: http://orientdb.com.

[2] OrientDB, "OrientDB REST," 2017. [Online]. Available: http://orientdb.com/docs/orientdb-rest.html.

[3] OrientDB, "OrientDB programming language bindings," 2017. [Online]. Available: http://orientdb.com/docs/last/programming-language-bindings.html.

[4] Liferay, "Liferay Digital Experience Platform," 2016. [Online]. Available: http://www.liferay.com. [Accessed 2 June 2016].

[5] OrientDB , "OrientDB Community," 2016. [Online]. Available: http://orientdb.com/orientdb/. [Accessed 21 March 2016].

[6] S. Owen, R. Anil and T. Dunning, Mahout in Action, New York: Manning, 2012, p. 387.

[7] G. Booch, J. and Rumbaugh and I. and Jacobson, The Unified Modeling Language User Guide. (2 edition), Addison-Wesley, 2005, p. 496.

[8] Apache, "Apache Kafka - A high-throughput distributed messaging system," Apache, 2016. [Online]. Available: http://kafka.apache.org. [Accessed 11 May 2016].

[9] R. Kohavi and B. Becker, "UCI Machine Learning Repository," Data Mining and Visualization, 01 May 1996. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Adult. [Accessed 11 May 2016].