Health, demographic change and wellbeing

Personalising health and care: Advancing active and healthy ageing

H2020-PHC-19-2014

Research and Innovation Action

ACANTO

A CyberphysicAl social NeTwOrk using robot friends

D4.8 Social activity recommendations: Definition of the Social Activities recommendation system.

Deliverable due date: 01.02.2017	**Actual submission date: 27.3.2017**
Start date of project: February 1, 2015	**Duration: 42 months**
Lead beneficiary for this deliverable: ATOS	**Revision: 1.0**
Authors: Ivo Ramos (ATOS), Ingo Brauckhoff (ATOS), Maurizio Marchese (UNITN)	
Internal reviewer: Paolo Bevilacqua	

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		
P	Public	**X**
CO	Confidential, only for members of the consortium (including the Commission Services)	

Contents

Executive Summary

This document defines the implementation of the social activities recommendation system. The design, implementation and test of the algorithms used for the recommendations of social activities in selected environments, contributing for the evolution of the user profiles will be described.
First, in section 1, we designate the architecture of the Social Activities Recommendation System. Then, in section 2, we present the User profile, Social Activity and Environment features. Finally, we demonstrate the design, implementation and test of the algorithms for the recommendation system. This information is captured, stored, analyzed and processed to provide the development of the recommendation algorithms.
We would like to add that this document has been submitted two months later than the original plan due mainly to: (i) the unavailability of real data; (ii) the consequent need to define and create an artificial dataset to test the approach; (iii) minor delays related with issues during the implementation and development due to the required training with the technical infrastructure selected for the ACANTO system in other work packages. Nevertheless, we would like to point out that this delay does not have a significant impact in the project since it does not affect the overall project timeline.

Introduction

In this deliverable, we present our contribution for the design, implementation and testing of the preliminary technological proposal towards the ACANTO Task 4.4: Social Activities recommendations. In the first phase of the work of Task 4.4 our effort has been focused on the applicability of the implementation of the specific set of models used in ACANTO to represents user's profiles, social activities, environment features.

As we mention in a previous task (Task 4.1), the creation of the user profiles and user communities as defined in WP2 will be based initially on the similarity of static features and later, on the dynamic observation coming from the ACANTO Cyber Physical Social network. These features are initially described in this deliverable.

The social activities recommendation implementation will thus be presented considering three aspects: a) Description of the selected technological architecture; b) Contextualization of the recommendation implementation by describing the user profile, social activities and environmental features; c) Description of the implementation of the recommendation algorithms.

The structure of the deliverable is as follow:

- In the first section, we first focus on the description of the technological environment, then we describe the data structures needed by the users, the activities and the environment, to understand the context over which we build the activity recommendation.

- In the second section, we describe the implementation of the recommendation system in ACANTO, implementing a hybrid system, exploiting the best parts of content based and collaborative filtering. Also, we present the tests done for evaluating the implementation.

2 Social Activities Recommendation System

2.1 Social Activities Recommendation System Architecture

In this section we will provide an overview of the architecture and technologies used for the Social Activities Recommendation System. The purpose of the Cyber Physical Social Network (CPSN) is to create a community of users, therapists and formal and informal caregivers (friends and relatives) who can enter prescriptions on the user and receive information on her/his state. The user's physical state will be obtained by electronic sensors, placed at the *FriWalker*, and data is collected during the execution of activities in order to extract 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 by the user initially. Afterwards, connections are established among user profiles with the creation of circles with their participation. Some activities will be proposed to different user *circles*, the execution of activities is supervised, and data will be collected to define the satisfaction level and to deliver recommendations for similar or different activities (depending on the case). The recommended activity can also be planned for a group of people / circle which the user belongs to. It can also come from the observation of other elderly adults with similar profiles or be associated with the micro-tasks execution relevant to the circles that the user belongs to. Figure 1: CPSN Architecture and Data Flow depicts the main components that comprise the CPSN and are described below.

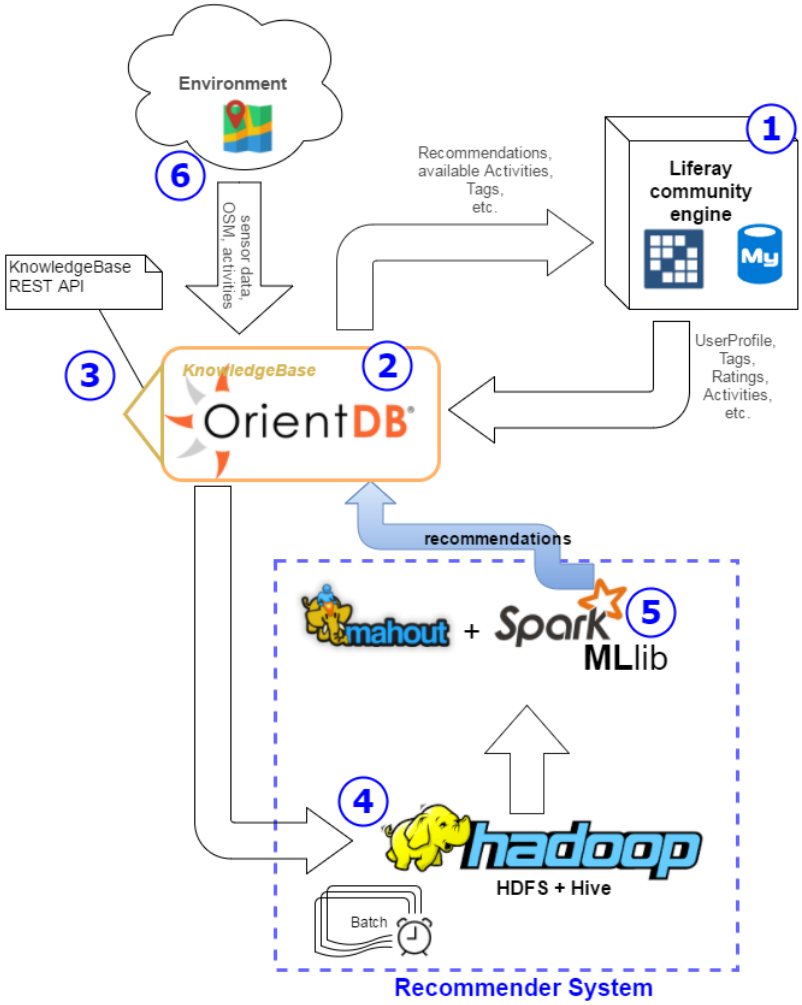

Figure 1: CPSN Architecture and Data Flow

(1) **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 also be stored here [1].

(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* [2].

(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 onto the data already in storage. A command line tool and a *Java Database Connectivity* (JDBC) driver are provided to connect to Hive.

(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 is composed by the "Life zones", which are the regions 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*.

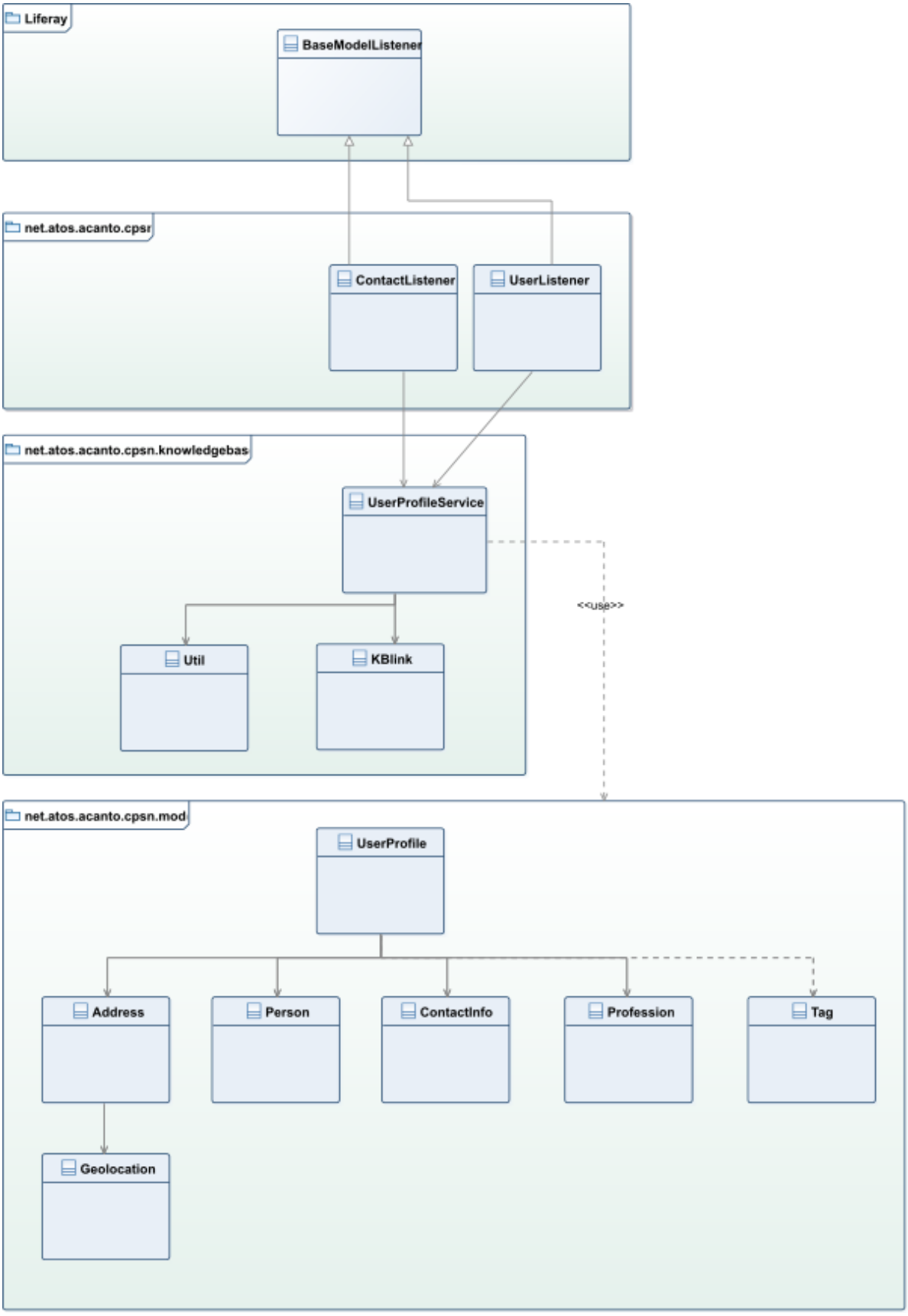

Figure 2: Activity Service

Figure 2: Activity Service and Figure 3: Liferay integration schema shows the Unified Modeling Language (UML) [3] class diagrams of the implementation used by the portlets

in the *Liferay* environment. The layers are (from bottom to top): model, service, controller and core. We prefer not go into the details of the implementation of the portlets.

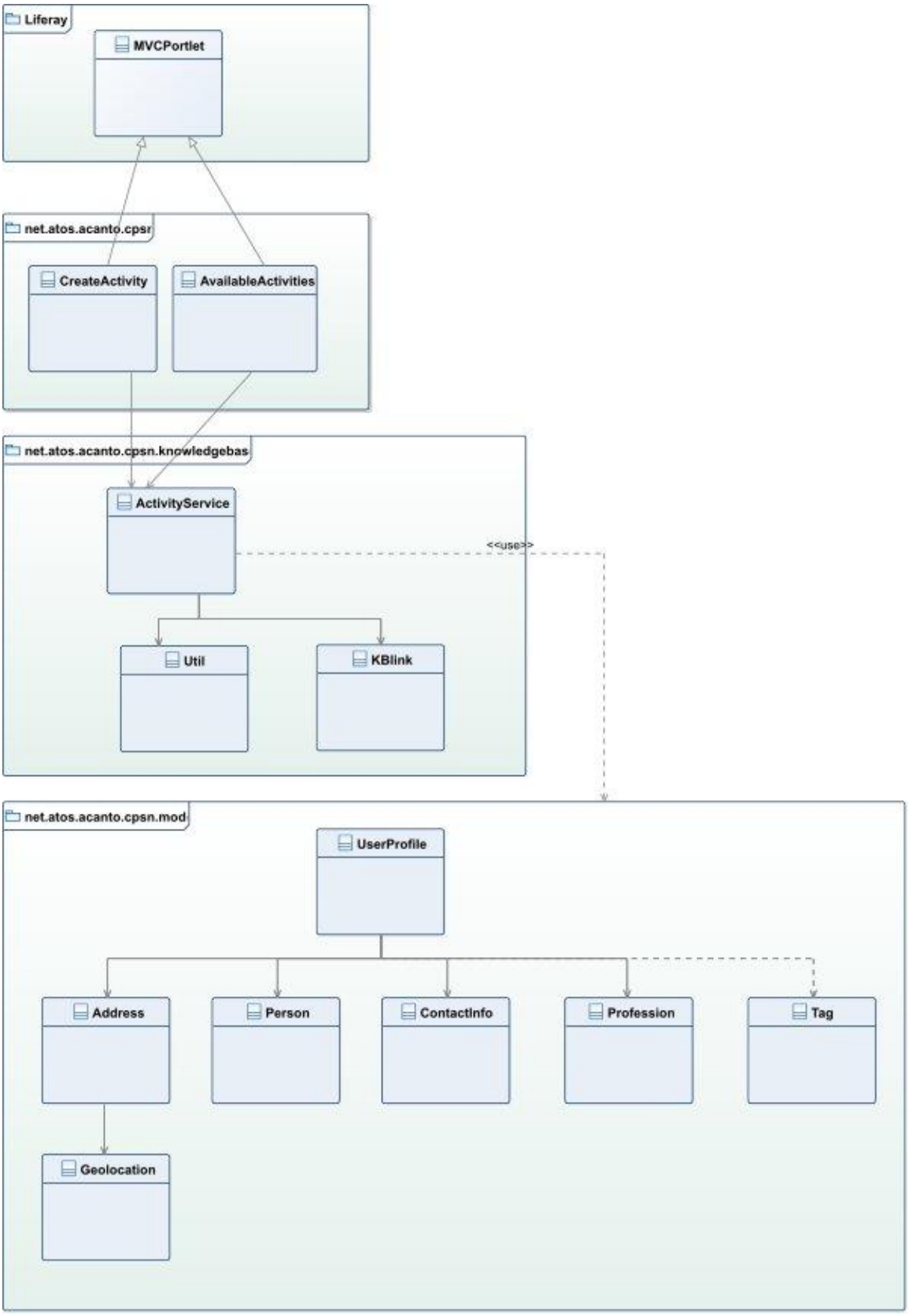

Figure 3: Liferay integration schema

Figure 4: Activity enrolment state diagram, describes the process of how the user interacts with a recommended Activity. The status of the enrollment is stored as property of edges of type ATTENDS, and can hold the following values:

- **enrolled** – when the user just signed up for an activity
- **passed** – when the activity has completed and the user participated
- **cancelled** – when the user made up her/his mind and chose not to go

The *rejected* status value is set on edges of type RECOMMENDED to record the fact that the recommended activity has not been accepted by the user.

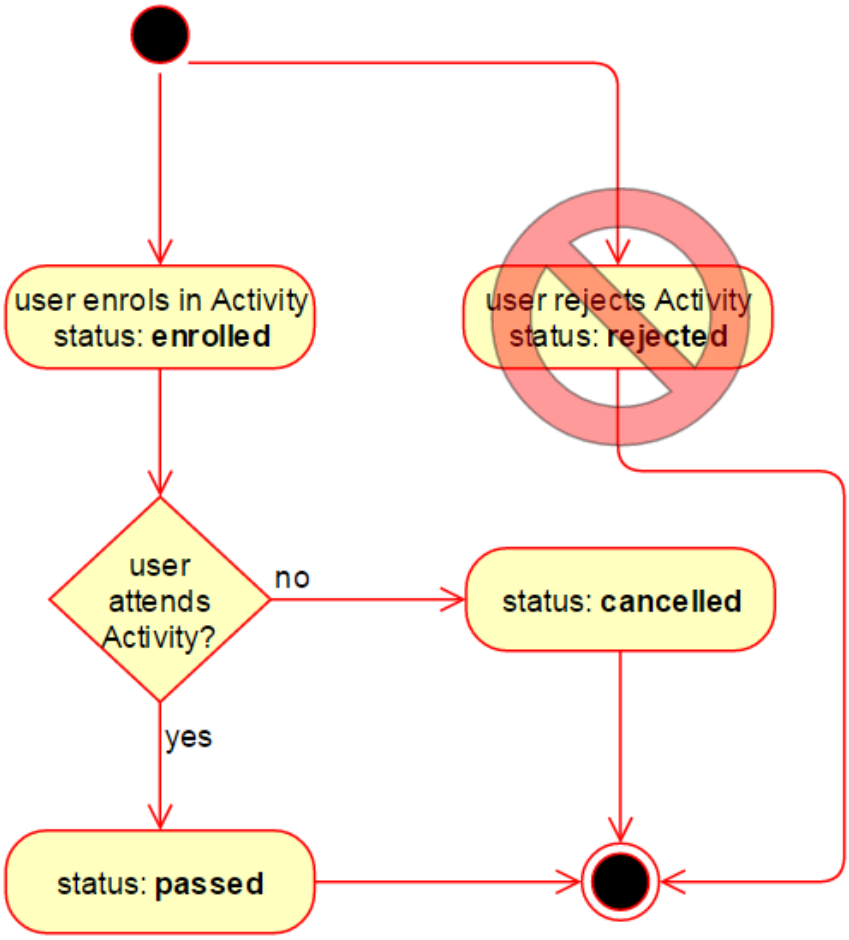

Figure 4: Activity enrolment state diagram

Figure 5: ACANTO Social Activities displays the actual aspect of the "creation of new social activities" dialogue. In the upper third there are the input fields for the basic information, like name, descriptions, starting time and date, as well as some ACANTO parameters, e.g. whether the Activity supports the use of a *FriWalker* and if devices will be provided on site.

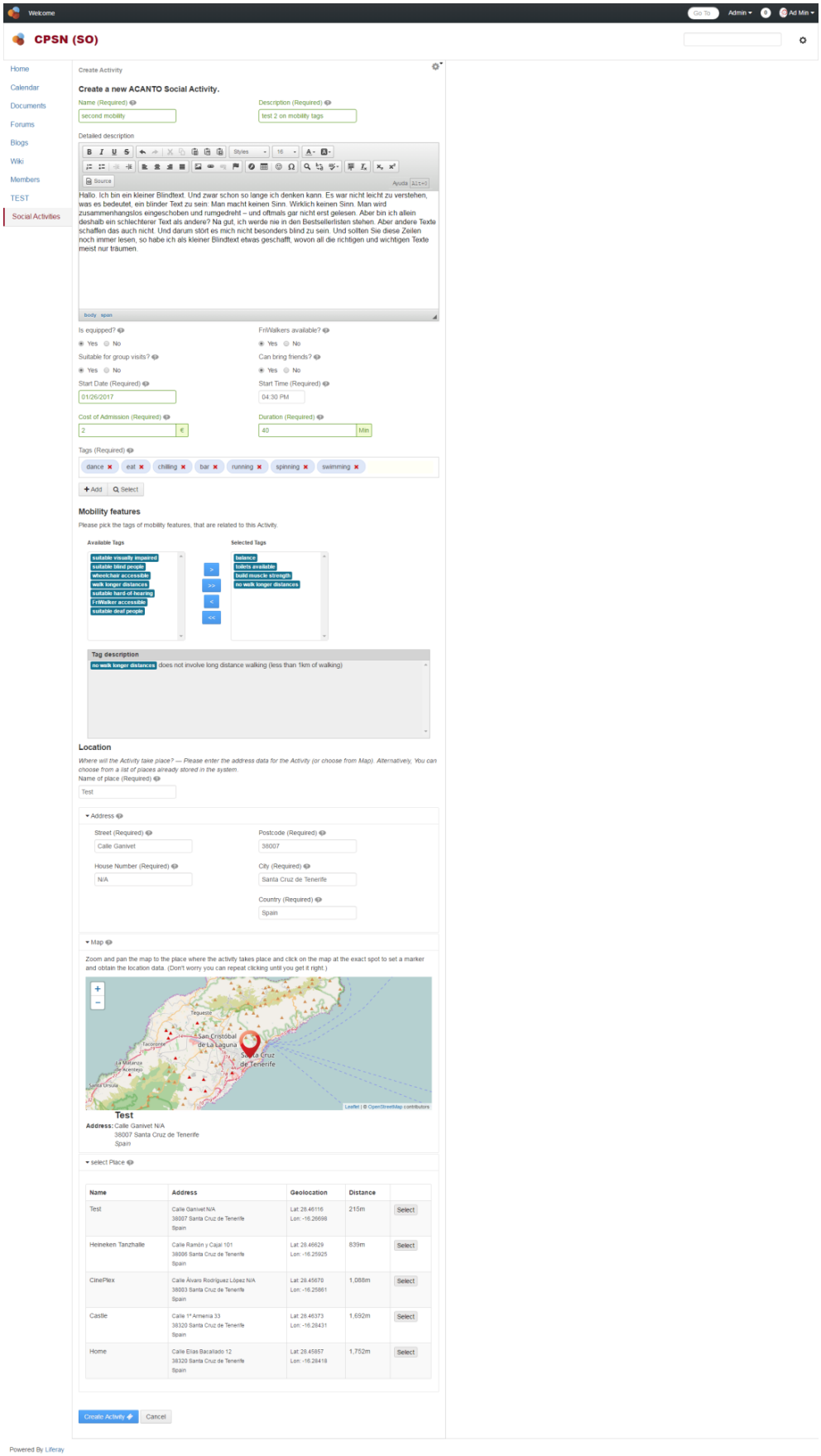

Figure 5: ACANTO Social Activities portlet

The second third is dedicated to the categorization of the activity: there is one facility to enter the tags – a list of already available tags is provided through the *Select* button – and

another one to pick the available *mobility tags* from a list. Currently available mobility tags are shown in Table 1 below. Different descriptions are provided forassociated to their different use cases, i.e. activity creation or prescription recommendation.
The third section is dedicated to the selection and modification of a *Place*: common input fields for the address (street name, house number, postcode, etc.), a map to select a location by panning, zooming and clicking on the spot, and a list of locations already stored as Places in the *KnowledgeBase*. In the background the geolocation data is stored as latitude/ longitude pair if possible, to enable tracking for external applications like the *FriTab*.

label	activity description	prescription description
balance	involves balance activities	needs more balance activities
suitable visually impaired	suitable for visually impaired	poor eyesight
toilets available	toilets available nearby	needs to have toilets nearby
build muscle strength	helps to build muscle strength	needs to build muscle strength
suitable blind people	suitable for blind people	blind
wheelchair accessible	wheelchair accessible	wheelchair user
walk longer distances	involves lots of walking (more than 1km)	needs to walk longer distances (more than 1km)
suitable hard-of-hearing	suitable for those with bad hearing	poor hearing
FriWalker accessible	FriWalker accessible	FriWalker user
no walk longer distances	does not involve long distance walking (less than 1km of walking)	cannot walk long distances (must walk less than 1km)
suitable deaf people	suitable for deaf people	deaf

Table 1 mobility tag examples

2.2 Description of the User profile, Social activity and Environment features

In our work on the implementation of the recommendation system we have used the latest models that present the User Profile, Activities and Environment as described and discussed in details in the document D2.4 – "*User, activity and environmental description*". These models have reached a good level of maturity and they are currently used by different components of the implementation of the ACANTO Infrastructure, including the specific recommendation system described in this deliverable. A few changes could be required during the validation and testing phases of the different

modules and therefore they are subject to possible updates until the end of the project. In brief, we have used and implemented the following models (for a more detailed description please check the full text in Deliverable 2.4):

- The **User Profile** model is the general description of the data representation and relationships for the users of the system. It intends to centralize data coming from different sources, mainly personal information, preferences, interests, mobility constrains, previous activities, etc.
- The **Activity** model shows the relationship of the activity with the user that executes it and the activity evaluation provided by the user.
- The **Social Activity** models are related with activities, like walking in the park with someone else, going shopping or to an art exhibition. It is crucial to understand the possible characteristics of social activities that could be relevant for a better recommendation and organization of activities.
- The **Circles** define the concept of "group", necessary to define activities with a social dimension. Essentially, they will be generated between people with compatible profiles and with common interests.
- The **Environment** models represent the definition of the "life zones", i.e., the regions inside the environment in which prominently the users carry out their daily activities.

We report here, for completeness, the schema of the models we have used in this deliverable, namely: **User profile**, **Social activity** and **Environment features**. For a more detailed description please consult the document D2.4.

The **user profile** aims to collect data coming from different sources, such as personal information, preferences, interests, mobility constraints and previous activities. 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 6: User profile schema illustrates the main blocks of information to present the user at the ACANTO system, together with the user circles and the locations.

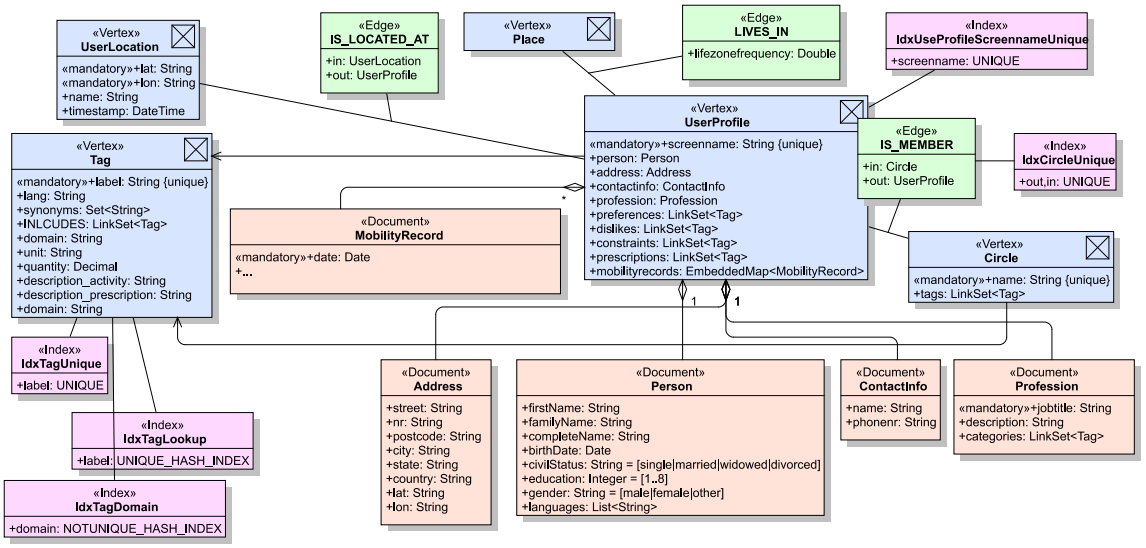

Figure 6: User Profile Schema

Figure 7: Social Activity schema depicts the structure of an Activity node in the

Knowledge Base and its relations. Every vertex of type Activity has a series of properties to describe the nature of a social event. The location of an Activity points to a Place node, which holds information about the address, the name of the place and optionally its geolocation (latitude and longitude).
The relationships between Activity and User Profiles are described using edges of type ATTENDS and RECOMMENDED, the latter of which is being created by the recommender system. Once a user interacts with a recommended Activity an ATTENDS edge is created, in case he enrolls to attend the activity, or the RECOMMENDED edge's property status is set to 'rejected' if the user chooses to reject the proposed activity.

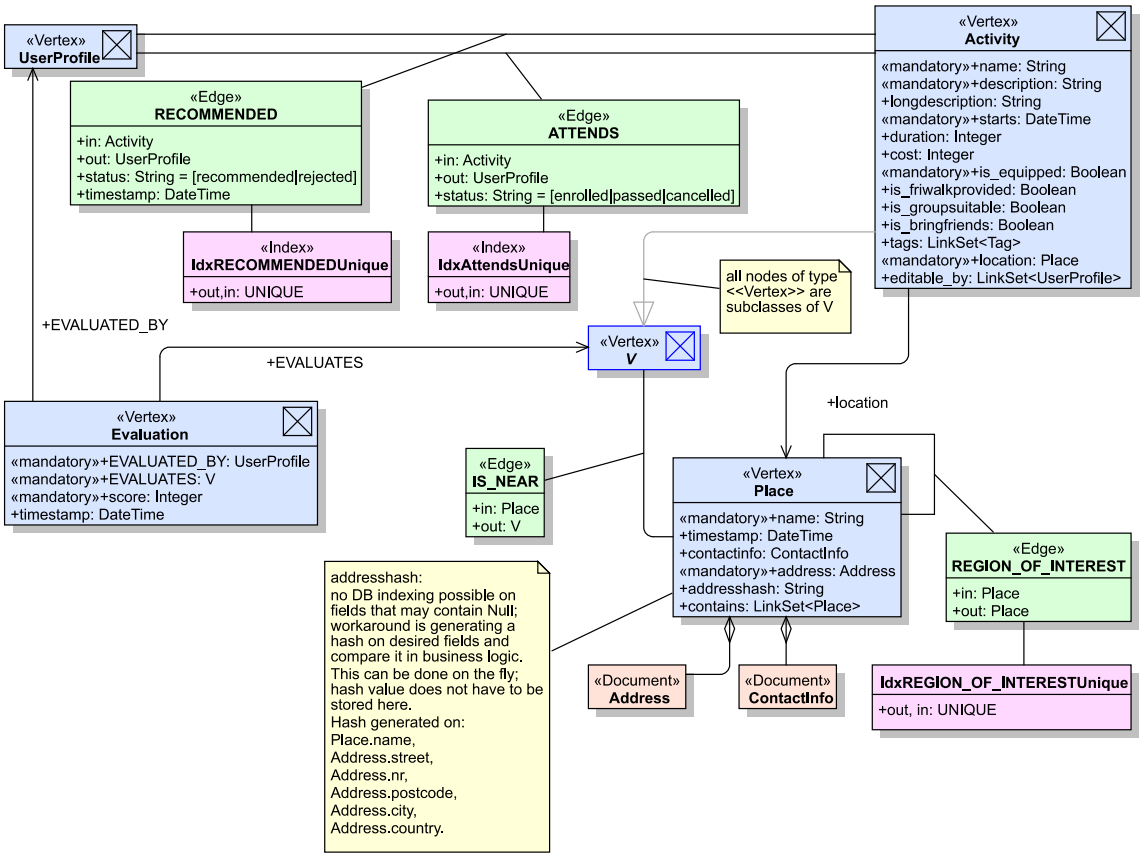

Figure 7: Social Activity schema

The **environment model** contains both static and dynamic information about the world. The static information refers to knowledge about the world that does not change or does

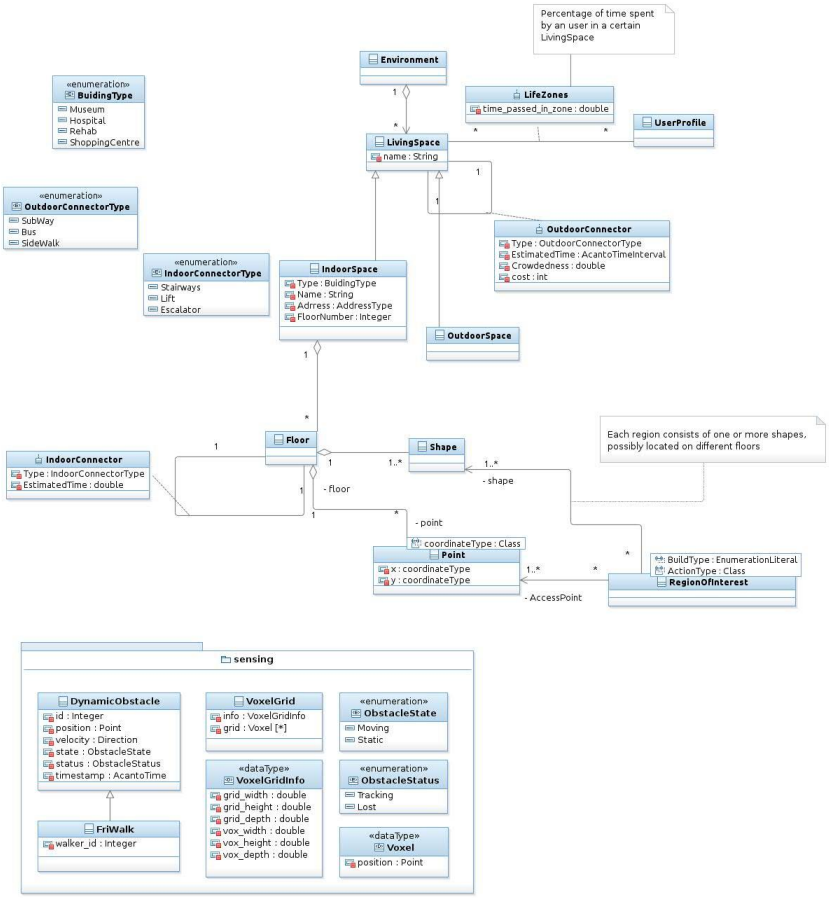

Figure 8: Class diagram offering an overview of the environment description

not change regularly. This includes both metric information (required to produce detailed plans) and information on the "semantics" of the different places. The dynamic information refers to knowledge about moving objects (i.e. obstacles, people, and - in the use case involving the FriWalk - other walkers) in the direct vicinity of the user as well as dynamic information collected by sensors (on the FriWalk or on smartphones, tablets or other wearable sensors) during the execution of the activities.

2.3 Design, implementation and test of the algorithms used for the recommendations

In this section we will describe the design, implementation and testing of the algorithms used for the recommender system.

2.3.1 Design

The recommender system is designed to use big data architecture, techniques, and Apache Mahout's [https://mahout.apache.org/] 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 the analysis of 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.

The algorithms used for our purpose are based on collaborative filtering and content based filtering. The basic assumption of collaborative filtering is that if users shared the same interests in the past, e.g. when they attended the same event, they will also have similar preferences in the future. So, if, for example, a user A and a user B have both visited the History Museum, and user A has recently attended another history related event, which user B does not know yet, the rationale is to propose this event to B.

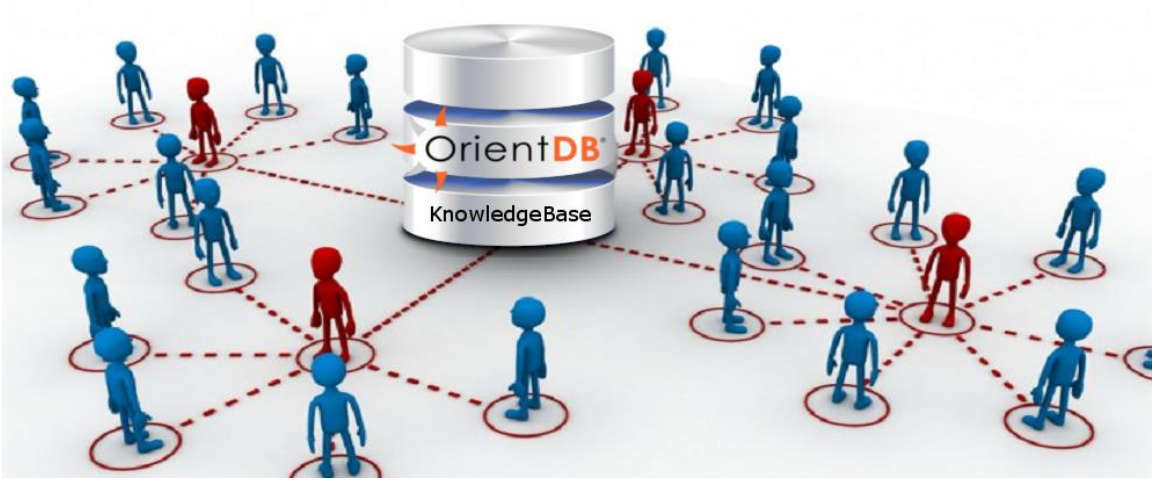

Figure 9: Representation of the collaborative recommendation

Content-based filtering is based on the availability of manually created or automatically extracted item descriptions together with a profile that assigns importance to these characteristics. For example, the characteristics of an event might include a theme, a specific topic, and the kind of the event. Similar to event descriptions, user profiles may also be automatically derived and "learned" either by analyzing user behaviors and feedback, or by asking the user explicitly about his/her interests and preferences. When compared to the content-agnostic approaches described above, content-based filtering has two advantages: It does not require a large user group to achieve reasonable recommendation accuracy, and a new event can be immediately recommended once the event's attributes become available.

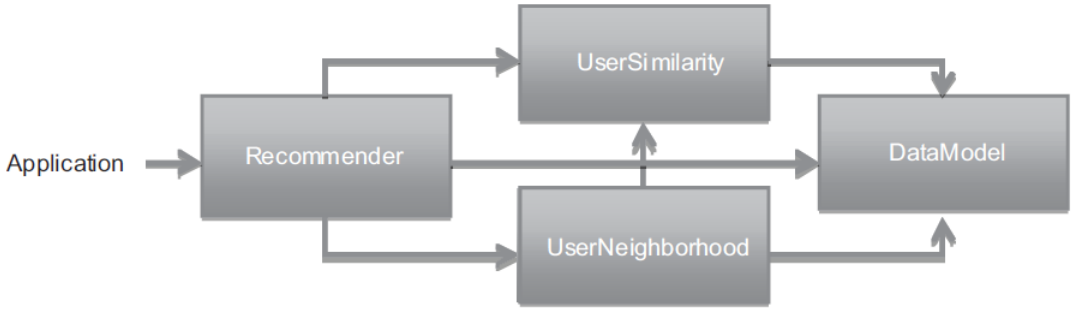

Figure 10: Diagram description of recommender prediction

These algorithms provide an easy way to find suitable new events for each user. Users have preferences for certain events, and these preferences must be distilled out of the data. The data itself is represented by a utility matrix, giving for each user-item pair a value that represents the degree of preference that a user has for that specific event. These values come from an ordered set, e.g., integers from one to five, and represent the number of 'stars' that the user gave as a rating for that event. We assume that the matrix

is sparse, meaning that most entries are *unknown*. An unknown rating implies that we have no explicit information about the user's preference for this event. The preliminary recommendation will then be filtered by other constraints of each user that may apply, such as location or mobility constraints.

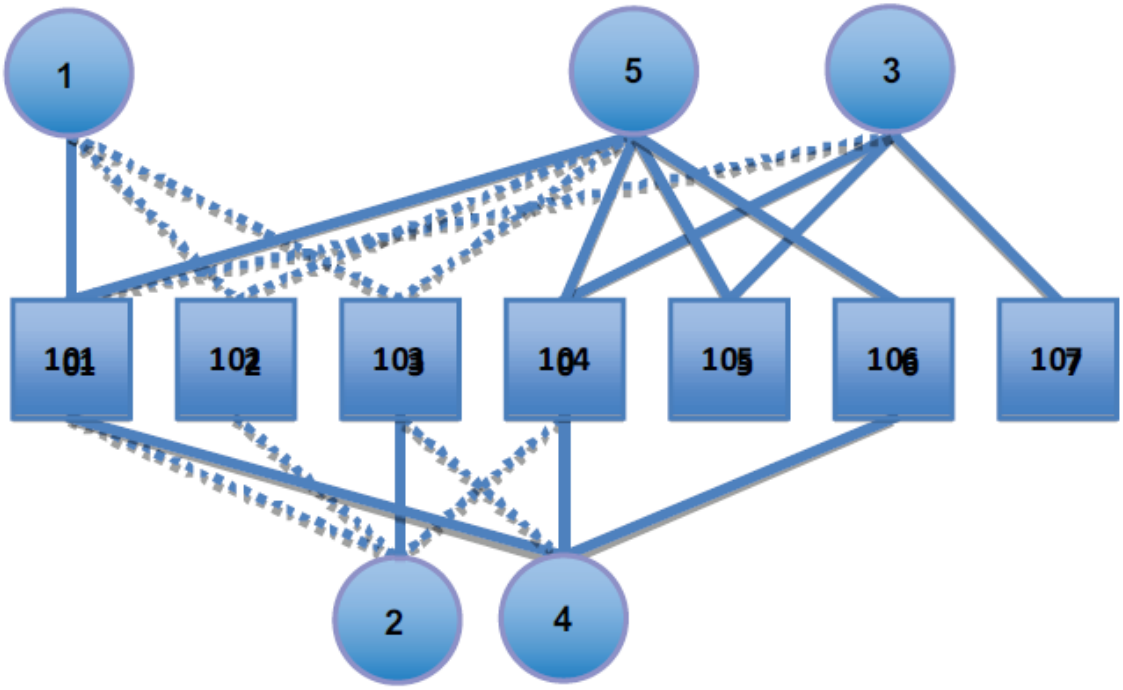

Figure 11: Relationship between users 1 to 5 and items events 101 to 107. Dashed lines represent associations that appear negative when the user does not like the item but expresses a relationship anyway.

2.3.2 Implementation

The recommender system works in two ways: collaborative recommendation and content-based recommendation. These implementations are based on Pearson correlation, log likelihood and Spearman correlation as well. For collaborative recommendations the system takes all available events and all user scores associated to them. If an activity is suitable for a user, it is validated by applying the user's constraints, and afterwards recommended if it passes the validation. Each activity recommendation takes also constraints into account like the geolocation area, viability and restrictions provided by the caregivers, relatives and medical doctors. If the activities have not yet been evaluated, the recommendation will be based on their *content*, i.e. by applying a content-based recommendation algorithm. The accuracy of the recommendations will improve as more data becomes available to be analyzed. The recommender system reads the data from the OrientDB KnowledgeBase and stores it in HDFS [3] as RDD [4] in order to make it available to the Hadoop cluster. A batch process is run periodically (say, every 24 hours) to include new activities, scores and user profile updates into the recommendations.

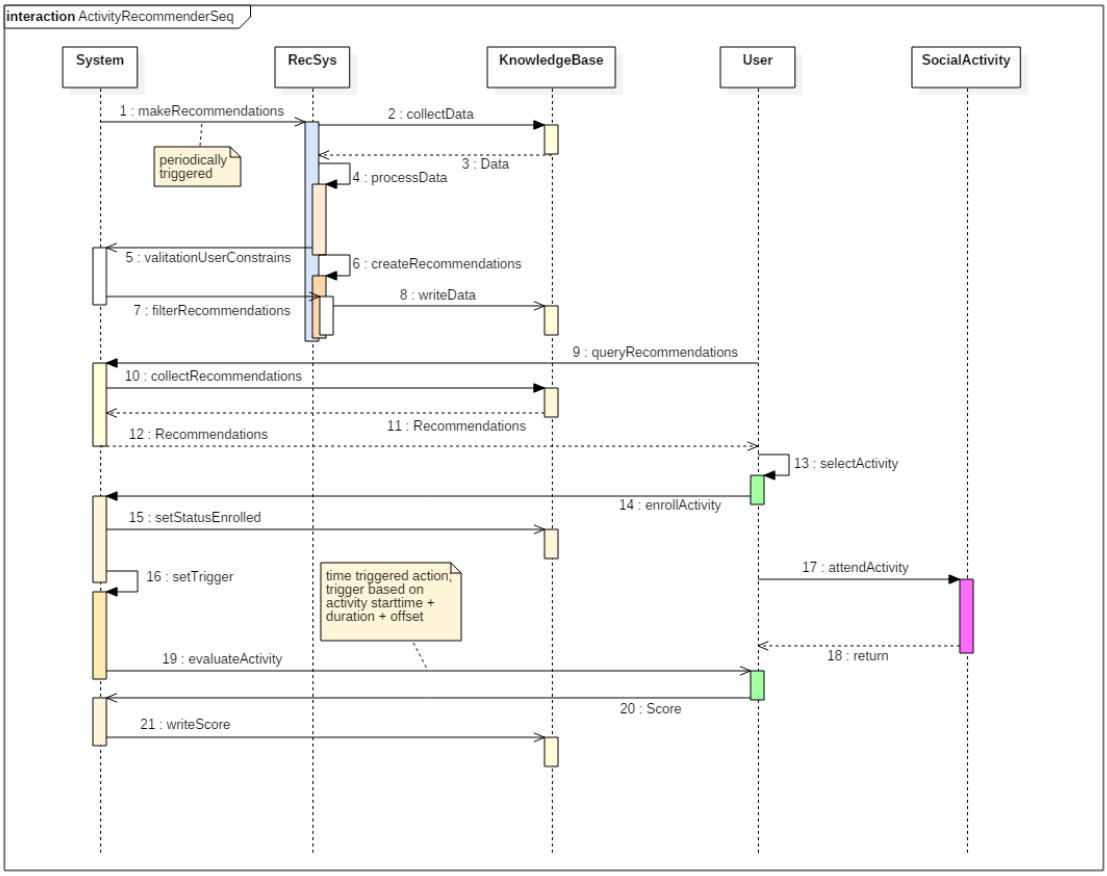

Figure 12: Recommender system process

The different recommendation algorithms are implemented in ACANTO as follows: The algorithms are provided as a Java (jar) library that can be directly linked into any Liferay application or portlet. The ACANTO library provides the following routines for accessing the recommendation engine through the *RecommenderActivities* class:

- **collaborativeFilter**
- **contentFilter**

The *RecommenderActivities* class provides recommendations for a user derived from its UserProfile. These methods have two implementations depending on the passed parameters.
The ***collaborativeFilter*** is computed by the Mahout *PearsonCorrelationSimilarity()* routine. This *PearsonCorrelationSimilarity()* method takes as input a *DataModel* – a list of user id, activity id and preference value for each item, and returns a *UserSimilarity* object that contain define a notion of similarity between two users. Implementation return values in the range -1.0 to 1.0, with 1.0 representing perfect similarity. Depending on the number of users, different recommendations will result. The *collaborativeFilter* method returns a list of *RecommendedItem* where each item has a *getItemId()* and *getValue()* method.
The *contentFilter* routine can be called independently and as many times as needed, e.g. once for every user to compute its personal recommendations.

The ***contentFilter*** provides recommendations in real time for a single user. It requires a user id, available activities and the maximum number of recommendations it should deliver.
The *contentFilter* provides a way to group users by matching their preferences or to find similar activities. This method allows recommending activities to user groups if the grouping is based only on some preferences of the *UserProfile* and is not as comprehensive as the circle recommender. The *contentFilter* finds activities that match with the tags referenced by the *UserProfile* preferences. This method can also be used to match users with similar preferences but leaving out the user's constraints. The parameters passed to this method are an activity id or user id and the output is a list of similar users or activities.

2.3.3 Testing

To test the system, we created an artificial dataset based on different datasets from the UCI machine learning repositories. Each UserProfile was created from "merge world cities", "born name" and "yelp academic" datasets. The main fields were extracted from the "Barry Becker" dataset from the 1994 Census database. The artificial dataset has about 30.000 different UserProfiles with their respective preferences, random likes and dislikes. Each profile was associated with activities in order to create useful data for the recommender system. The following table has a list of parameters that were used to create the dataset:

Work class:	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
Education:	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. education-num: continuous.
Marital-status:	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
Occupation:	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
Relationship:	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
Race:	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
Sex:	Female, Male.
Native-country:	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti,

	Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad & Tobago, Peru, Hong, Holand-Netherlands.
Preferences:	up to 20 sequential tags like "tag1", "tag2",…,"tag20". This provides an easy way to cross-validate the prediction results.

We created twenty tags, "tag1", "tag2", …, "tag20" to categorize the preferences and assigned three random tags to each UserProfile. Each tag is a representation of a personal preference like "art", "run", "read", etc. In a production environment, these tags should be defined using basic activities such as "run", "walk" and technical tags provided by the caregivers, relatives and medical doctors. Categories should not be created without normalization between user preferences and activity tags, because they are used to match one against the other. We also added three tags each to all activities. Furthermore, about 1.000 places around the USA were created and assigned to each user as its location. Finally, we created over 6000 activities distributed across the places, generated three recommended activities for each user using the *collaborativeFilter*, and created an evaluation by the user in a scale of one to five for all three activities. The score was created as weighted random and relies on the number of matches with the user preferences. For example, if the number of matches is three, then the resulting score is around five, because both user profiles and activities have three tags in common.

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, Activities (specifically Social Activities) 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. It is also related with the WP5 – "Execution Support of Social Activities", specifically the task 5.2 (Activity Planning), where the output of the Recommender system is processed and translated to an executable plan. 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 on the techniques developed in WP3, at task 3.2 – "Perception of the environment", as well as from the evolving social data coming directly from the users' social network).
Moreover, the components developed in this work have been designed and implemented in compliance with the overall cloud ACANTO infrastructure developed in WP7.
Finally, the work of this WP will be integrated and used in the experimental validation of WP8.

Conclusions

In this deliverable, we started by explaining the update made to the social activity model previously presented at the document D2.3 and updated in the D2.4. Following, we described the first version of the social activity 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.
Finally, we explained the process of design, implementation and test of the algorithms used for the social activities recommendations as well as the creation and implementation of the artificial dataset.

3 Bibliography

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

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

[3] Addison-Wesley, Unified Modeling Language User Guide, The (2 ed.). A, ISBN 0321267974, p. 496., 2005.

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

[5] "Kafka Confluent Platform," Apache, 2016. [Online]. Available: http://docs.confluent.io/2.0.0/platform.html. [Accessed 11 May 2016].

[6] Apache, "Apache Spark MLlib," 2016. [Online]. Available: http://spark.apache.org/mllib/. [Accessed 8 May 2016].

[7] OrientDB, "OrientDB RESTful HTTP," 2016. [Online]. Available: http://orientdb.com/docs/last/OrientDB-REST.html. [Accessed 2016].

[8] Addison-Wesley, Unified Modeling Language User Guide, The (2 ed.). A, ISBN 0321267974., 2005, p. 496.

[9] I. R. Luigi Palopoli, D2.3 User, activity and environmental description (preliminary), 2016.

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

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

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

[13] A. Consortium, ACANTO project, 2015.