# Recommendation System for Tourist Attraction Information Service

Alexander Smirnov, Alexey Kashevnik, Andrew Ponomarev, Nikolay Shilov, Maksim Schekotov, Nikolay Teslya
St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
St.Petersburg, Russia
{smir, alexey, ponomarev, nick, shekotov, teslya}@iias.spb.su

*Abstract*—**The paper proposes a description of information decision support system in the tourism domain and a set of methods and algorithms for generating recommendations for a user that allow significant increase of the system usability. The system generates for the user recommendations which attractions at the moment are better to attend based on the user preferences and the current situation in the location area. The system also allows showing the user information about interesting attraction in more detail, which is based on analyzing information evaluations made by other users.**

***Keywords—recommendations, ratings, ontology, context management, tourism, mobile devices.***

## I. Introduction

Recently, the tourist business has become more and more popular. More and more tourists prefer to use Internet services to book hotels, buy flights, and search attractions to see instead of booking complete tours. In this regard, information retrieval systems, which allow finding information about the tourist trip and provide the tourist interested information during the trip, are becoming more and more popular. The most valuable systems support online information searching in different Internet sources instead of using local information databases. However, such systems have to personify and implement the context-based filtering of information before providing it to the tourist. These systems have to recommend the tourist, which attractions are better to attend, and identify, which information about attraction is better to be shown to the tourist.

The presented recommendation system for tourist attraction information service allows recommending attractions, which are better to attend, based on the tourist preferences and the context information of the location area. The system allows the tourist to see detailed description of the interesting attraction acquired from Internet sources. Recommendation system chooses an Internet source that provides description of the interested attraction based on other tourists ratings.

The system is service-based and uses the smart space technology, which allows providing for information sharing between different services of the system.

Section 2 contains the overall system description and architecture including role of the presented recommendation system for the tourist attraction information service. In Section 3 the attraction recommendation problem is discussed and the proposed context-aware approach to attractions recommendation is presented. Section 4 presents the image and text blocks recommendation models. Section 5 describes recommendation system services interaction for propose the tourist attractions and their description information. The main results are summarized in conclusion.

## II. Recommendation System Description

### A. System Overview

The recommendation system described in the paper is developed as a distributed service-oriented application that allows tourists to get useful information using mobile client (

Fig. 1).

The system accumulates tourist's interests and context-related information and searches for accessible at the moment and interesting for the tourist attractions based on this information [1].

Implementation of the recommendation system is based on the Android OS that is one of the most popular operating systems for mobile devices today. For interoperability support between mobile clients and different services the smart space technology is used, which allows providing for information sharing between different devices. For this purpose the Smart-M3 information sharing platform [2] is used, which provides implementation of the smart space technology. The key idea of this platform is that the smart space is device, domain, and vendor independent. Smart-M3 assumes devices and software entities to be able to publish their embedded information for other devices and software entities through simple, shared information brokers. The platform has a decentralized architecture and allows seamless integration with other systems, services, and program modules.

Modern tendencies of information and communication technologies require development of stable and reliable infrastructures to extract and keep different kinds of information and knowledge from various members of the smart environment. The smart space assumes more than one device that uses common resources and services.

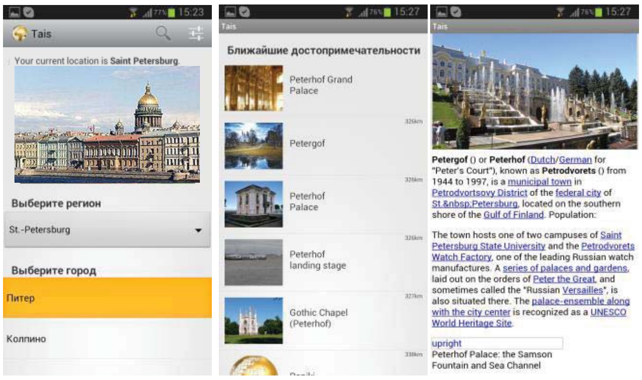


Fig. 1. Tourist attraction information service: screenshots

## B. *System Architecture*

In accordance with the developed architecture, the recommendation system consists of knowledge processors and system information broker. Knowledge processors include a client module and different services (see Fig. 2). The client module is implemented in the user mobile device and provides possibilities for interaction with the tourist. The client module acquires the tourist context and keeps his/her profile, that includes the tourist preferences for the system personification and usability.

Available services include: the recommendation service, the context service, the attraction information service, the public transport service, the ridesharing service, and the taxi service. The recommendation service implements information filtering that is provided for the tourist by other services. The context service acquires and provides information about current situation in the area (e.g. location, traffic situation, weather). The attraction information service searches for information around the tourist's location in different Internet resources (like Wikipedia, Wikivoyage, Wikitravel, Panoramio, Flickr). The public transport service finds ways to reach an attraction by public transport. The ridesharing service tries to find drivers, who move in the same direction and to determine possibilities of joint trips. The taxi service provides information about accessibility and prices for taxi for tourist trip to an attraction.

## III. ATTRACTIONS RECOMMENDATION

To improve the user experience of the tourist attraction information service the list of attractions presented to the user should be ordered with respect to a predicted degree of interestingness for the specified user as well as reachability (taking into account the current situation in the area).

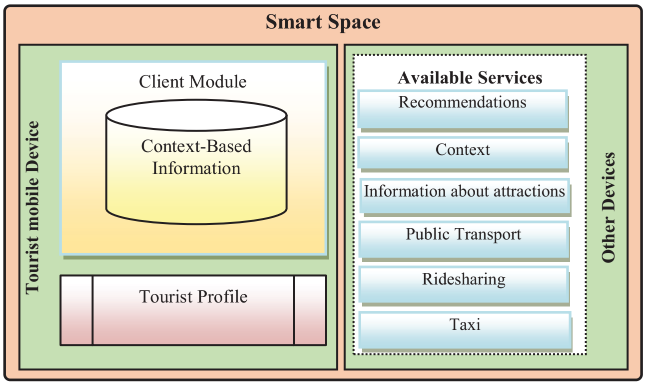


Fig. 2. Architecture of the recommendation system for the tourist attraction information service

The attraction's degree of interestingness is estimated by the recommendation service. This service takes user ratings associated with each attraction by all users as an input. According to the conventional classification (e.g. [3]), it performs user-based collaborative filtering. One of the promising directions to improve the predictive quality of recommendation systems in general (and collaborative filtering systems among them) is context-awareness [4]. The context describes conditions in which the user rates an object or asks for recommendations.

In the proposed tourist attraction information service the following context attributes are distinguished:

a) time;
b) company in which the user visited the attraction (alone, with a friend or with the family);
c) weather (sunny, rainy, etc).

Values are assigned to these attributes in mostly automated fashion. For example, the user opens the attraction evaluation screen being near to that particular attraction (according to the mobile device's GPS sensor). In this case the time attribute is filled in with the current time and current weather is queried from the context service. However, there is also a possibility to set the values of context attributes manually in the evaluation screen of the mobile application. It is convenient, for example, if a user wants to rate the attractions seen during the day upon returning to the hotel in the evening. To facilitate deferred evaluation the proposed system tracks attractions the user visits and shows unrated visited attractions in a special screen. The user does not have to assign values to each context attribute. If a context attribute is not given a value, it is assumed to have value "any".

There are three general approaches to take context into account in recommendation systems [4]: (a) contextual pre-filtering; (b) contextual post-filtering; (c) contextual modelling.

The advantage of the contextual pre-filtering and post-filtering approaches is that they are compatible with classical (not context-aware) recommendation algorithms. The context awareness in these approaches comes true by transformation of either input or output of the classical recommendation algorithm. In the contextual pre-filtering approach, the rating data that is not related to the context is filtered out before applying the recommendation algorithm. On the other hand, in the contextual post-filtering approach the resulting list of recommendations is ordered or filtered taking into account context values.

In the contextual pre-filtering approach all the ratings that are irrelevant to the context discarded from the rating matrix before the recommendation algorithm is applied. For example, if in some attraction recommendation service the context includes weather conditions, then making recommendations in a rainy day should not use ratings assigned in sunny days. This approach aggravates the important problem inherent to collaborative filtering systems – rating matrix sparsity. The main goal pursued by contextual pre-filtering methods is to take into account the context, but not let rating matrix to become too sparse.

In the proposed system the context generalization method [5] (one of the contextual pre-filtering methods) is used for taking context into account. In this method, the rating matrix is filtered not only by exact values of context attributes, but also by its possible generalizations. To use this method the context model has to support context generalization. In most general form, it means that at least one context attribute must be defined on a set with a strict partial order relation of generalization ($\rightarrow$). Let $A$ be a set of attribute values and $a_i$, $a_j \in A$. Then notation $a_i \rightarrow a_j$ means that value $a_j$ is a generalization of $a_i$. A context is usually represented by $m$ attributes. Let $c = (c_1, \ldots, c_m)$ and $c' = (c'_1, \ldots, c'_m)$ are two contexts. We define $c'$ as a generalization of $c$ ($c \rightarrow c'$) iff there exists at least one $i \in \{1,..,m\}$, such that $c_i \rightarrow c'_i$. We call context $c$ *incompatible* with $c'$ iff neither $c \rightarrow c'$ nor $c = c'$. In most cases, the generalization relation forms some kind of a hierarchy (or multiple hierarchies).

In the proposed system the context generalization is enabled by following:

a) The set of Time attribute values includes not only exact date and time values but also “any” value and aggregate values for each season, day type (working day or weekend) and time of day (morning, afternoon, evening). The generalization relation is defined naturally.

b) The set of Company attribute values includes values “alone”, “with friends”, “with family” and “any”. “Any” value is defined to be a generalization of any other value.

c) The set of Weather attribute values includes values “sunny”, “rainy”, “cloudy”, “snowy” and “any”. “Any” value like in (b) is defined to be a generalization of any other value.

For example, the exact context could be (Time: “July 31, 2013 17:30”; Company: “with family”; Weather: “sunny”). This context can be generalized to (Time: “summer”; Company: “with family”; Weather: “sunny”) or even to (Time: “summer”; Company: “any”; Weather: “any”).

It is obvious that a context can be generalized in several ways and directions. In systems with many attributes and many levels of granularity of attributes, enumerating all possible context generalizations is a problem and various heuristics are used for picking appropriate generalizations [5]. In the proposed system, there are not so many possible generalizations, so all of them are enumerated through implicit directed graph traversal procedure. The nodes of this graph are attribute values and the arcs are generalization relations.

A user rates attractions on a five-point scale (1 – bad, 5 – excellent). The rating obtained from the user (*raw* rating) is normalized to reduce individual bias in assessment: some users tend to put relatively high ratings to all attractions, others in contrary tend to put relatively low ratings. Normalized rating $\tilde{r}_{uj}$ given by user $u$ to attraction $j$ is defined by formula:

$$\tilde{r}_{uj} = r_{uj} - \frac{1}{|K_u|+1}\left(3 + \sum_{k \in K_u} r_{uk}\right),$$

here, $r_{uj}$ is raw rating of the attraction $j$ given by user $u$, and $K_u$ is a set of all attractions rated by user $u$. The idea of normalization is to shift from user-oriented five-point scale to calculations-oriented zero-centered scale. The sign of the normalized rating corresponds to general attitude of the user (whether it is positive or negative) and the absolute value of the rating corresponds to the strength of that attitude. The straightforward way to normalize ratings is to subtract scale average (i.e. “3”) from each rating. It would work nice if users normally used all the range of five-point scale. However, most users in fact rate items using some subset of the scale, e.g., only “3”, “4” and “5”. In this case subtracting scale average would result in non-negative normalized ratings missing the fact that the user definitely likes items he/her rated “5” and probably doesn’t like items rated “3”. Hence, the normalization procedure should capture not only the scale characteristics but also the observed usage of this scale. Therefore, a popular method of normalization is subtracting average user rating from all his/her ratings. This method works well in most cases but have some subtle drawback which turns out when there are only a few ratings. For example, when the user rated only two items – both with “5” – then normalization over the average user rating would turn these ratings into zeroes. I.e. *a priori* notion of five-point scale with “5” as the best mark is lost in favour of adaptation to the observed usage of this scale. To alleviate this drawback in the proposed system we use slightly modified version of the normalization over the average user rating. During the normalization we add one fake rating of “3” (scale average) to the set of user ratings having a purpose to stick other ratings to the original notion of the scale. This modification is significant when there are a few ratings (in the example above two “5” ratings become positive) but its contribution to the normalized ratings vanishes as the number of users’ ratings grows.

Attraction rating estimation for a given user is performed in two steps:

1) a group of users with ratings similar to the given user’s is determined;

2) rating of attraction is estimated based on ratings of this attraction assigned by users of the group.

While building the list of recommendations, several possible generalizations of the context is used. For each context generalization ratings received in contexts incompatible with this generalization are not taken in to account.

User group is determined by k-Nearest Neighbours method (kNN). The similarity between users $u$ and $v$ is calculated as a cosine measure between normalized ratings vectors of users according to the following formula:

$$s_{uv} = \frac{\sum_{o \in O} \tilde{r}_{uo} \tilde{r}_{vo}}{\sqrt{\sum_{o \in O} \tilde{r}_{uo}^{\,2}} \sqrt{\sum_{o \in O} \tilde{r}_{vo}^{\,2}}} \,.$$

Here $O$ is a set of attractions rated by both users $u$ and $v$.

Attraction rating estimation for the user is based on ratings of that attraction assigned by other users of the group with respect to their similarity to the user. It is calculated as a weighted average of normalized ratings among group members:

$$r^{*}_{uj} = \frac{\sum_{v \in G} \tilde{r}_{vj} s_{uv}}{\sum_{v \in G} \left| s_{uv} \right|} \,,$$

here $G$ is the group of the user.

The resulting list of attractions $L$ presented to the user $u$ is sorted in descending order of:

$$s_j = k r^{*}_{uj} + (1-k)\left(1 - \frac{d^{w}_{j}}{\max_{i \in L} d^{w}_{i}}\right),$$

here $k \in [0,1]$ is a model parameter correlated to the importance of the attraction rating estimation in favor of its reachability; $d^{w}_{j}$ is the estimation of time to reach the attraction $j$.

## IV. Intelligent Information Search for Recommendation

The tourist attraction information service processes two types of information describing attractions, namely images and text blocks. The information is retrieved from external sources. That means that the providers of the proposed service do not directly control quality, completeness and even relevance of the information presented to the user. One of the exploited methods to ensure information quality is automated information filtering and ranking. Thereby information retrieved from external sources before being presented to a user is processed by several filtering and ranking algorithms. The purpose of these algorithms is to enhance the user experience through providing the user with more reliable and potentially useful information about attractions.

### *A. Images Ranking*

Presented system allows the user to view images, connected to the attraction. The images are retrieved from public sources (e.g. Wikipedia, Panoramio or Flickr). It is natural that some images are better than others and it is desirable to show to the user mostly good images, but image services do not usually annotate images with quality tags. Moreover, the image quality itself is no doubt rather subjective concept.

For automated filtering and ranking of objects, two general approaches can be used: content analysis and user evaluation. Content analysis assumes development of a formal model of the object and linking the parameters of this model to some quantitative measure of quality. The user evaluation approach moves the complexity of quality estimation to the users.

As a development of the formal model for assessing image quality is a complex task (which is not solved yet) and image quality itself is quite controversial concept, it was decided to build a filtering and ranking technique based on user ratings. It means that the user can rate images presented to him/her, the ratings are collected, saved to the recommendation services’ information storage and further used to select and rank images presented to users.

The following requirements should be taken into account by the image selection algorithm:

a) A user should mostly see images positively rated by other users.

b) To make a list of images presented to a user more diverse it should include new images that have no ratings yet. This is also important to collect more ratings.

c) Negatively rated images should also be shown, because negative rating may have been assigned to them by mistake. Certainly, there should not be many of them in one particular list of images.

The requirements enumerated have different (in some cases mutually opposite) influence on the resulting list. The list of images is supposed to be built by an algorithm taking into account weights assigned to different factors.

Users can assign images binary ratings: "Like" or "Dislike". Scales with more elements are used mostly in specialized systems where it is important to measure the user's attitude to an object more precisely. It has to be noted that the cardinality of a scale is correlated with the user effort of rating objects according to that scale. It is easy to decide if an object is "good" or "bad", but not so easy to decide how many points to assign to it in, say, ten- or even hundred-points scale. So, the cardinality of the rating scale should be chosen as a trade-off between precision and user friendliness. For example, in the proposed system five-points scale is used to capture the users' attitude about attractions, which is a part of the core functionality. On the other hand, the image filtering and ranking is a secondary feature that should be unobtrusive to the user. "Dislike" is needed to reduce the possibility of presenting to users bad or irrelevant images with minimal administrative effort.

Overall image rating is calculated as a sum: $v = v^+Q^+ - v^-Q^-$, where $v^+$ and $v^-$ are weights of "Like" and "Dislike" ratings respectively (in the current implementation both of them have value 1), and $Q^+$ и $Q^-$ are numbers of "Like" and "Dislike" ratings respectively.

The algorithm of image list populating takes the following input parameters:

- $N_{max}$ - maximal size of the requested images list.

- $L_v$ – set of images with known ratings; this set can be seen as a union of $L_v^+$ - the set of images with positive overall rating and $L_v^-$ - the set of images with negative or zero overall rating.

- $L_n$ - set of new images.

- $k_1, k_2, k_3$ ($k_1, k_2, k_3 \geq 0$, $k_1 + k_2 + k_3 = 1$) - weights corresponding to relative importance of each factor: $k_1$ - presence of positively rated images in the resulting list; $k_2$ - presence of new images in the resulting list, and $k_3$ - presence of negatively or zero rated images.

- $B$ ($B < 0$) - minimal overall rating for an image to be shown to users.

The resulting list $L$ is formed according to the following algorithm:

1) The list $L$ is assumed to be empty.

2) $\lceil k_1 N_{\max} \rceil$ randomly selected elements of $L_v^+$ are appended to $L$. If there are less than $\lceil k_1 N_{\max} \rceil$ elements in $L_v^+$, then all elements from $L_v^+$ are appended.

3) The size of $L$ is increased up to $\lceil k_1 N_{\max} \rceil + \lceil k_2 N_{\max} \rceil$ elements. For that purpose it is appended randomly drawn elements from $L_n$ and if there are not enough of them, random elements of $L_v^+$ are used. If it turns out that the size of $L$ can not be increased up to specified value using these two sets, then the resulting list will be shorter than $N_{max}$.

4) min($N_{max}$ - $|L|$, $\lceil k_3 N_{\max} \rceil$) randomly selected elements of $L_v^-$ with overall rating $B$ or higher are appended to $L$. Here $|L|$ is length of the list after completing three previous steps of the algorithm. If there are not enough elements in $L_v^-$, then elements are drawn randomly from $L_n$, and if there are not enough even there, from $L_v^+$.

5) The resulting list is sorted in descending order of overall image rating.

On each step of the algorithm drawn images are selected in random uniformly, i.e. each image of the source set has equal probability of being chosen.

Values of parameters $B$, $k_1$, $k_2$, $k_3$ in practical implementation depend on the user role and other information from his/her profile. For example, a user with the role "expert" (familiar with the selected location) can be more interested in new, unrated images – for this kind of users the values of $k_2$ and $k_3$ will be greater than $k_1$. On the other hand, a user with the role "traveler" (e.g., planning his/her first trip to the location area) will see better images, i.e. value of $k_1$ will be greater than $k_2$ and $k_3$. Moreover, users of this kind will see less images than expert users ($N_{max}$ is less). The specific parameter values are refined during the tourist attraction information service operation.

## B. *Descriptions Ranking*

Beside images, users of the tourist attraction information service see textual information describing the interesting attraction. For each attraction, several text blocks are retrieved from different external sources, but all of these blocks cannot be shown simultaneously as it would bloat the user interface. So, the text blocks are shown to the user in turn. Therefore, one of the blocks should be selected to be shown first and the rest blocks should be ordered to form such a sequence that would give the user a complete notion of the interesting attraction as soon as possible.

It was decided to employ the combination of content analysis methods and user evaluation methods to build the sequence of text blocks to be shown to the user. As in the case of images, the users can rate text blocks – this information is saved to the recommendation service's storage and is further used in the ranking formula. Beside user ratings, text characteristics are used to rank blocks in several languages (e.g., Russian and English).

Hence, the list of factors taken into account when choosing the next block to be shown is the following:

a) user rating;
b) text characteristics: length, lexical diversity;
c) similarity of block contents with the last shown block.

Quantifications of these factors are merged into overall text block score ($v_i$) which is used for choosing the next text block to show to the user.

Let us see these factors in some greater detail. A user can mark a text block as "good". Overall user rating of block ($u_i$) is calculated as the number of users marked block as "good".

The main purpose of text characteristics analysis is finding out: a) the extent to which the text block is appropriate for getting summary about the tourist attraction; b) the lexical diversity of the text block. It is assumed that information provider services are reliable enough and contain quality texts, so spam protection is not an important task for the presented service. Currently two text characteristics are used: text block length in bytes ($l_i$) and the number of nouns and named entities divided by block length ($q_i$) as a rough measure of lexical diversity and quality of the text. It should be noted that not all block lengths are convenient for getting information on a mobile device. So block length is evaluated by function that has small values for very short blocks, high value for blocks of length about 10KB (which is the empirical length found during experiments on Wikipedia articles) and decreases with further growth of block length. For the evaluation of block length, the following formula has been proposed:

$$f(x) = \frac{k_1 x^3}{e^{k_2 x} - 1}.$$

Here $k_1$ and $k_2$ are normalization coefficients which values are tuned in such a way that $f(l_i)$ has maximum value of 1 corresponding to the recommended block length (10KB). The function itself does not have special interpretation, it was chosen solely because it reflects the desired way of block length scoring.

Block similarity is defined as a cosine measure between two block contents represented in form of a vector space model [7, 8]. Smaller values of the cosine measure correspond to higher chances of block to be shown next. The rationale here is to show user diverse blocks, trying to form the most complete notion of object (an attraction).

The vector space model representation of a block is built according to the following procedure:

1) text is normalized on case, split into words, named entities are recognized and each word is marked with its part of speech and transformed to canonical form;

2) words of block that are neither named entities nor nouns are discarded;

3) vector $b_i$ is composed where dimensions correspond to words (nouns in canonical form and named entities) found in any text blocks, and values are defined as $c_j/C$ where $c_j$ is number of times word occurs in text block, $C$ - number of words in block after step 2.

The employed procedure of building vector space representation is well-known [7], the main particularity of the version of this procedure used in the proposed system is that only named entities and nouns matter in the resulting vector.

The overall text block score is calculated by the following formula:

$$v_i = k_1 \frac{u_i}{\max\limits_{j \in \{1,\ldots,N\}} u_j} + k_2 f(l_i) + k_3 q_i + k_4 \frac{1 - \cos^{names}(b_i, prev)}{2}$$

Parameters $k_1$, $k_2$, $k_3$, $k_4$ correspond to the influence of different factors on overall score and in current implementation have equal values. $N$ – the number of candidate text blocks available. $cos^{names}(b_i, prev)$ – cosine measure between text block $i$ and the last block shown to user.

The first addend corresponds to the user evaluation of the attraction. If there are no rated blocks (therefore maximal user rating in denominator is zero) then this addend is assumed to have "zero" value.

The second addend of the formula accounts for block length, the third – text diversity metric. Finally, the fourth addend accounts for text block similarity with the last block shown – the value of cosine measure is transformed from the range [-1;1] to [0;1] with cosine value -1 mapping to the highest text block score. When choosing the first block to show, this addend is assumed to have "zero" value.

The block with maximal overall score is shown to the user.

## V. Recommendation System Services Interaction

Service interaction scenario of recommendation system is presented in Fig. 3. The client module publishes information about the tourist context and preferences in the smart space. The context service reads this information and publishes information about the current situation in the area of tourist location. The attraction information service publishes the list of attractions that meets the tourist context and preferences and reaching possibility meets the current situation in the location area. The recommendation service reads information about attractions, tourist preferences, and context from the smart space, gets information from the rating service, and generates a list of most appropriate for the tourist at the moment attractions related to the his/her interests. Based on information from the transport service, the recommendation service analyzes which attractions and in which order can be proposed to the tourist for visiting. Then, the recommendation service publishes this information in the smart space and it becomes accessible for the client module, which presents it to the tourist.

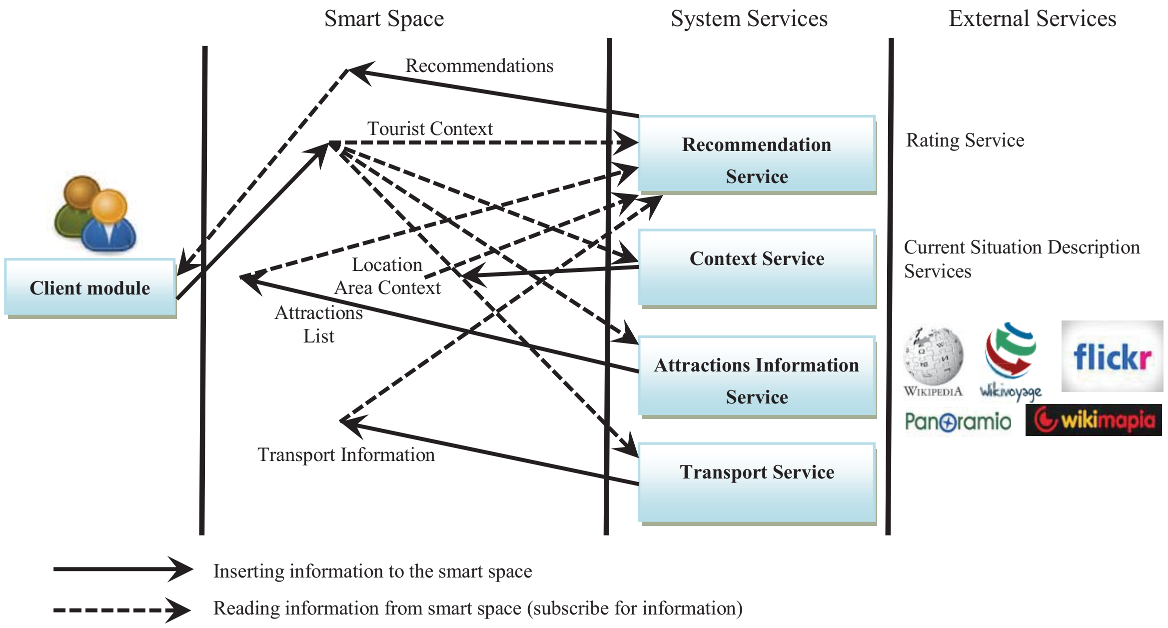


Fig. 3. Tourist attraction recommendation system services interaction based on smart space technology

The recommendation service reads information about attractions, tourist preferences, and context from the smart space, gets information from the rating service, and generates a list of most appropriate for the tourist at the moment attractions related to the his/her interests. Based on information from the transport service, the recommendation service analyzes which attractions and in which order can be proposed to the tourist for visiting. Then, the recommendation service publishes this information in the smart space and it becomes accessible for the client module, which presents it to the tourist.

The recommendation system allows the tourist to browse the attractions' descriptions for making a decision about visiting them. In this case, the client module publishes the information that the tourist is interested in an attraction. The attraction service gets this information, searches for descriptions of this attraction, and publishes links to information sources with these descriptions to the smart space. The recommendation service analyzes these information sources, user preferences, ratings made by other tourists for these sources and shares description (or descriptions) that are better for the tourist at the moment with the client module.

## VII. Conclusion

The paper proposes a system for making recommendations for the tourist information service. The system proposes a list of attractions, which is better satisfy tourist's interests and the current situation in the location area. Also, the system provides the tourist with descriptions of interesting attraction extracted from different Internet sources ranked based on other tourists' ratings. The proposed architecture, system implementation, recommendation methods, and algorithms allow implementing intelligent attraction information processing that significantly increases the system usability.

The proposed recommendation system uses binary ratings ("Like" or "Dislike") for attraction images and descriptions. For the future work, the authors plan to study possibility to use triple ratings ("Like", "Dislike", or "Irrelevant"), which would allow recommendation system to ignore images and descriptions that are irrelevant to the attraction.

## ACKNOWLEDGMENT

This research is a part of grant KA322 «Development of cross-border e-tourism framework for the programme region (Smart e-Tourism)» of Karelia ENPI programme, which is co-funded by the European Union, the Russian Federation and the Republic of Finland. The presented results are also a part of the research carried out within the project funded by grants # 13-01-00286 and 13-07-12095 of the Russian Foundation for Basic Research.

## REFERENCES

[1] A. Smirnov, A. Kashevnik, S. Balandin, S. Laizane, "Intelligent Mobile Tourist Guide: Context-Based Approach and Implementation" in Proc. Internet of Things, Smart Spaces, and Next Generation Networking, *13th International Conference, NEW2AN 2013, and 6th Conference (ruSMART 2013)*, Aug. 2013, Springer, LNCS 8121, pp. 94-106.

[2] J. Honkola, H. Laine, R. Brown, O. Tyrkko, "Smart-M3 Information Sharing Platform", *in Proc. IEEE Symp. Computers and Communications (ISCC'10). IEEE Comp. Soc.*, June 2010, pp. 1041-1046.

[3] M. Balabanovic, Y. Shoham "Fab: Content-based, Collaborative Recommendation" *Communications of the ACM*, vol. 40, no. 3, 1997, pp. 66-72.

[4] G. Adomavicius, B. Mobasher, F. Ricci, A. Tuzhilin, "Context-aware recommender systems", *AI Magazine*, vol. 32(3), 2011, pp. 67–80.

[5] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach", *ACM Trans. Inf. Syst. vol. 23(1)*, Jan. 2005, pp. 103-145.

[6] L. Boratto, S. Carta, A. Chessa, M. Agelli, "Group Recommendation with Automatic Identification of Users Communities", *in Proc. of IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT '09)*, vol. 3, 2009, pp. 547-550.

[7] C. Manning, P. Raghavan, H. Schutze, "Introduction to Information Retrieval", *New York: Cambridge University Press*, 2008, p. 482.

[8] P. Turney, P. Pantel, "From Frequency to Meaning: Vector Space Models of Semantics", *Artificial Intelligence Research*, 37, 2010, pp. 141-188.