A web-based environment to support on-line and collaborative group recommendation scenarios

David CONTRERAS [a], Maria SALAMÓ [b] and Jordi PASCUAL [b]

[a] *Facultad de Ingeniería y Arquitectura, Universidad Arturo Prat, Avenida Arturo Prat, 2120, Iquique, Chile*

[b] *Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via de les Corts Catalanes, 585-08007, Barcelona, Spain*

Abstract. Nowadays there has appeared many recommendation scenarios that involve groups of inter-related users intending to participate in a group activity. Though there have been attempts to establish group recommendation, most of them focus on off-line environments. In this paper we present a novel web-based environment that supports on-line group recommendation scenarios. Specifically, we propose a Collaborative Advisory CHannel for group recommendation called gCOACH. First of all, we present the conceptual architecture of gCOACH, which is domain independent and it is based on a web-based environment for enabling the interaction of the users from anywhere. Secondly, we introduce the multiple interaction modalities of this environment developed to communicate, coordinate and persuade the group in a case-based group recommender. We demonstrate its usability through a live-user case-study.

Keywords. Group Recommendation, Conversational Case-Based Recommendation, Critiquing, Interaction

1. Introduction

Recommender Systems (RSs) help users search large amounts of digital contents and services by allowing them to identify the items that are likely to be more attractive and useful [21]. RSs are playing an increasingly important role in many on-line scenarios [29], for example, e-commerce services are one example of these scenarios [20]. In particular, recommender systems has the ability to provide even the most tentative shoppers with compelling and timely product suggestions. Moreover, there are many recommendation scenarios that involve groups of inter-related users such as movies, trips, restaurants, and museum exhibits. But to date, much of the research in the area of recommender systems has focused mainly on recommending products to individuals rather than groups of people intending to participate in a group activity [17].

These types of scenario have motivated recent interest in group recommendation and to date a variety of early-stage systems have been developed in domains such as group web-page recommendation [18], recommending vacations or tours to groups of tourists [13,7], recommending music tracks and playlists to large groups of listeners [3, 11], and, recommending movies and TV programs to friends and family [30].

The role of the group recommender system (GRS) is to make suggestions that reflect the preferences of the group as a whole, while offering reasonable and acceptable options to individual group members. In this regard, a key aspect of group recommendation concerns the way in which individual preferences are captured and stored into the user profiles. These user profiles are combined to recommend a product (or a set of products) for the group at the completion of the group session. Though there have been attempts to establish group recommendation, most of them focus on user profiles extracted from off-line environments. Taking into account the growth of activities and environments that offer group interactions (e.g., virtual spaces or ubiquitous personalization services), we consider that group recommendations in on-line environments are becoming nowadays commonplace. When the recommendations are made for a group of on-line users, new challenges and issues arise to provide compelling item suggestions as well as new interaction mechanisms to keep aware the members of the group and to facilitate communication among them.

In this paper we consider a web-based environment that supports on-line group recommendation scenarios. The proposal is called gCOACH (COllaborative Advisory CHannel for group recommendation). In particular, gCOACH aims at being an on-line framework that facilitates group interaction and communication among members. For this reason, gCOACH uses a conversational case-based recommender [27], which is a form of recommendation that is well suited to many product recommendations. It uses the description of the product base in terms of a complete set of features instead of using the preferences (mainly provided off-line) from other users to make a recommendation, as occurs in collaborative filtering recommenders [25]. Conversational recommenders have proven to be especially helpful for users with ill-defined needs and preferences. Notice that recommender systems can be distinguished by the type of feedback that they support; examples include *value elicitation*, *ratings-based feedback* and *preference-based feedback* [28]. Specifically, we use a form of user feedback called *critiquing* [16], where a user indicates a directional feature preference in relation to the current recommendation. Feature critiques typically take the form of a directional or replacement. Directional critiques effect an increase or decrease over numerical feature values. For instance, a cheaper camera when the price of the current recommendation is 300$ implies [price $\leq$ 300$]. On the other hand, replacement critiques allow for the substitution of any value for a non-numeric feature. For example, different manufacturer if the current recommendation is a Sony camera means [manufacturer $\neq$ Sony]. Critiquing is a form of feedback that strikes a useful balance between the information content of the feedback and the level of user effort or domain expertise that is required.

In summary, the main contributions of this paper are two-fold. First of all, gCOACH not only elicits users' preferences in the form of critiques during the session, in an environment that is domain-independent, but also it provides multiple interaction modalities and interface components to keep awareness of member's decisions and to provide suggestions among them. Second, we evaluate the usability of the proposal based on a live-user study.

The rest of this paper is organized as follows: Section 2 reviews related work on group recommender systems; Section 3 describes our proposed gCOACH framework;

Next, Section 4 discusses the usability of the proposal; Finally, Section 5 gives conclusions and future work.

2. Related Work

Most of the research on group recommendation investigated the core algorithms used for *recommendation generation*. Two different strategies have been mostly used for generating group recommendations: aggregating individual predictions into group predictions (*aggregated predictions*) [1,9] or aggregating individual models into group models (*aggregated models*) [4,10,13,26]. Differences among these strategies differ in the timing of data aggregation step. Apart from the recommendation generation, group recommender systems can be distinguished based upon the different features used in the design of the GRSs [7,10]. In this paper we are interested in group recommenders that work in an on-line environment. For this reason, we pay attention to those features that heavily influence *preference elicitation*.

Preference elicitation refers to the manner in which information is acquired from user. For example, preferences may be acquired by asking users directly *explicit preference elicitation* or by inferring their preferences from their actions and feedback (*implicit preference elicitation*). In the case of the former, systems such as the Travel Decision Advisor [6] and PolyLens [17] both acquire preferences by asking users to specify them explicitly; either in the form of preferred features or item ratings. In contrast, group recommender systems such as FlyTrap [3] and Let's Browse [8] acquire the preferences by monitoring a user's interactions. Specifically, we consider that preference elicitation is also related to: (1) the way user interacts with the GRS; (2) the type of domains that the GRS can work with; (3) the outcome of the GRS; and, (4) the group size..

First of all, considering the *user interaction*, most of the approaches use an off-line interaction with the users. That is, users provide some feedback and the GRS provides a recommendation without further user interaction with the system (e.g., [17,5]). On the other hand, in an on-line interaction, users are active and they are engaged in a collaborative session. For example, *Travel Decision Forum* [6] and CATS [13,15] allow this kind of interaction. Our proposal, gCOACH, follows this vein by supporting on-line group recommendation scenarios.

Secondly, according to the *domains* that GRS can work with, nearly all GRSs are defined for an specific domain. For example, a recipe recommender [1], CATS [13] and *Travel Decision* [6] deal with tourist domain, and [30] propose to recommend movies and TV programs to friends and family. There are few proposals that are domain-independent [5] but it uses an off-line environment for group interaction. It is worth noting that gCOACH is domain-independent too and, as said above, it allows on-line interaction with the users.

There are habitually two forms for presenting the outcome of the GRS: a single recommendation that must be a successful selection for the group (e.g., [13]) or a list of items ordered according to the preferences of the group members (e.g., [17]). There are some GRSs that generate a different outcome, for example in [6] instead of returning recommendations, it returns a list containing the group preferences. Our proposal uses a conversational case-based recommender which aims at guiding the user over the search space by using critiquing as feedback. For this reason, it is more convenient to use a single recommendation instead of a list of products ordered. However, it is remarkable that gCOACH defines two areas for keeping a list of recommendations. First one is devoted to store the preferred products of the group, called *stack area*. Second area receives and stores *suggestions* in a proactive way. This suggestions may come from the GRS itself or from other group members.

Finally, not all the GRS are capable of working with any group size. Some GRSs are limited to small groups of two, three or four users [8,13] and some others have no limitation on the group size [30]. In gCOACH, we can define any group size although it is true that small groups will benefit the most of the environment than large groups.

3. gCOACH: group-based COllaborative Advisory CHannel

This section presents gCOACH framework, which supports on-line group recommendation scenarios. This framework allows several users to participate in a group activity that involves searching a product for the whole group. First of all, Section 3.1 presents the conceptual architecture and it details in depth each layer. Next, the interaction modalities defined in gCOACH are shown in detail in Section 3.2.

3.1. Conceptual Architecture of gCOACH

The conceptual architecture of gCOACH is based on a web-based environment developed on a client-server model for enabling the interaction of the users from anywhere. Figure 1 shows the conceptual architecture of our proposal divided into three main layers: a Space Client, a Space Server, and a Space Recommender.

Specifically, the Space Client is mainly devoted to be the view layer. It is in charge of capturing the interaction modalities and the communication among members of the group. The Space Server acts as a controller and it is responsible for connecting the users with the recommender. Note that Figure 1 depicts some events among the Space Client to the Space Server and the actions among Space Server to the Space Recommender. Finally, last layer contains the model of the whole architecture, which includes the Group Conversational Recommender algorithm (GCR) –which contains the individual user model (IM) of each member of the group, and the set of products available to recommend (i.e., known as case base (CB) in conversational recommenders)– and the Group Management Module (GMM). Next sections describe in depth each one of these layers.

3.1.1. Space Client Layer

This layer is a space that offers interaction, collaboration, and awareness among users. This space allows concurrent connection of one or more users (i.e., note that Figure 1 only displays 2 users for helping in the reading comprehension). Users can interact with a recommendation object, with a stack object, with a suggestion box object, or with an awareness object. As Figure 1 details, the stack object is shared among users while the remaining objects are independent for each user.

A *Recommendation Object* (RO) is an object that represents a product in the interface. It contains product information and the description of all features of the current recommendation, interactive elements to perform critiques, and collaborative elements

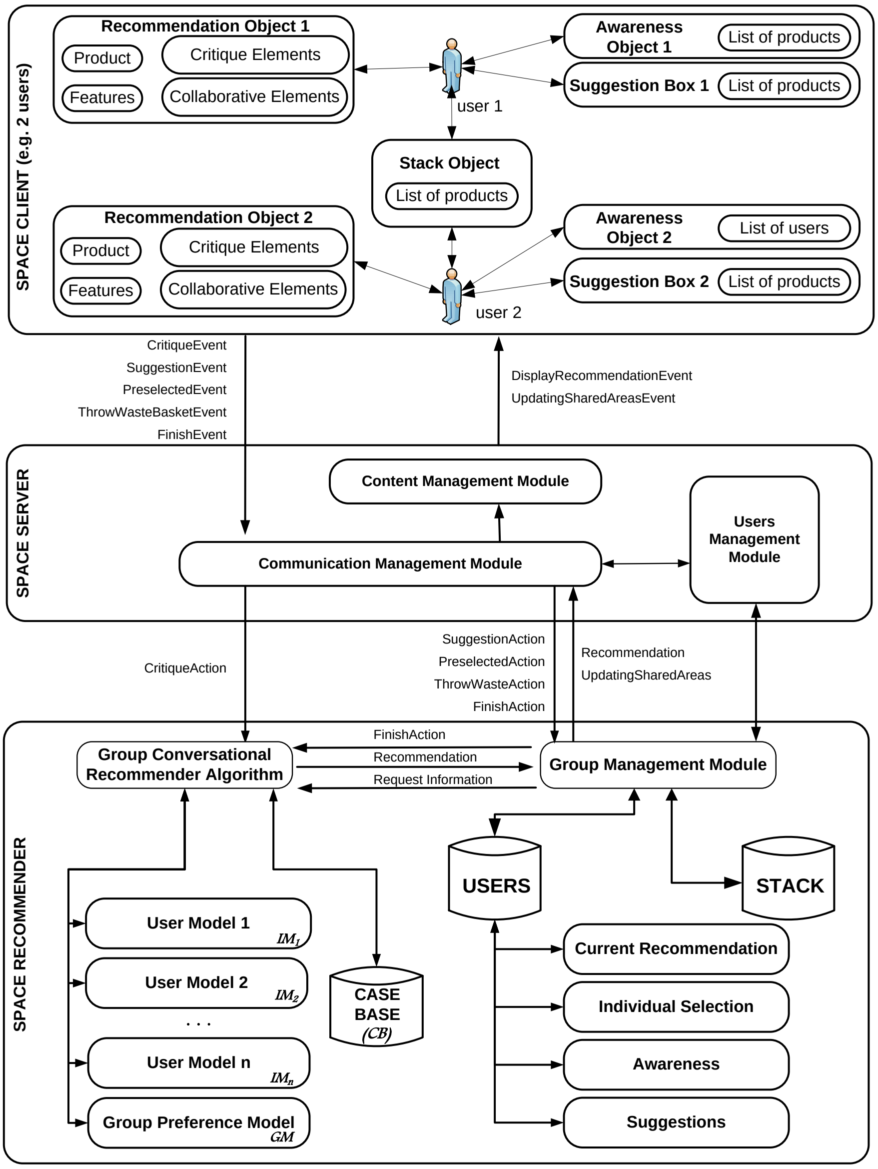


Figure 1. Conceptual architecture of gCOACH framework

to perform collaboration and communication among users (e.g., sending a suggestion to another user). For example, in a SKI domain the product information includes one or more images of the resort and the accommodation, each feature is described as a pair description-value (e.g., price-1000$), and the interactive critique elements available for each one of the features (greater than, lower than for numerical features such as price or different if the feature is nominal such as country). Note that the recommendation object is domain independent as its elements are general enough to a wide range of domains.

A *Stack Object* (SO) is an object that contains all the products that are of interest for the group members. It is a common object for the group and it is updated each time a user operates over it. Users add or remove their preferred products and the whole group will be aware of the main preferences of each group member. Apart from this, this object (SO) also stores the level of satisfaction of the whole group for every particular product. This information is for helping users in reaching a consensus.

The *Suggestion Box* (SB) object contains a list of products that have been suggested to the user by anyone of the members or by the recommender itself. This object also stores the level of satisfaction of the user to the suggestions received.

The *Awareness Object* (AO) includes a list of products with the information of the current product view of each one of the members of the group. This object keeps the group aware of the current navigation of each group member and the level of satisfaction of that particular user to the product he is currently viewing.

The user's interactions in the Space Client layer trigger five types of events that are sent to the Space Server. These events are: the *CritiqueEvent*, the *SuggestionEvent*, the *PreselectedEvent*, the *ThrowWasteBasketEvent*, and the *FinishEvent*. The *CritiqueEvent* occurs when the user performs a critique by touching the critique elements on the RO. On the other hand, the *SuggestionsEvent* is activated when the user wants to collaborate with another user by performing a drag and drop of a product from RO to SB. With this event, the user adds a product to the list of suggested products of a particular member of the group. The intention of the sender is to provide one recommendation that satisfies their requirements and the sender wish to draw the attention to it of another group member. By the contrary, instead of drawing attention to one member as is the case of the SuggestionEvent, the *PreselectedEvent* is aimed at focusing the attention to the whole group. It occurs when the user performs a drag and drop from RO to the list of products in SO for denoting that this product is of the user interest and she wants to store it as one of her main preferences. During the recommendation process users change their mind and maybe one of the previously preferred or suggested products that were of her interest are no longer preferred. To enable an update of the user's preferences, the *ThrowWasteBasketEvent* let the user to perform a drag and drop from SB or SO to the waste basket area. Finally, there is an event for finishing the recommendation process. This event is called *FinishEvent* and it occurs when the user wants to do the final selection of the desired product from the list of products in SO.

The Space Client layer also receives two events from the Space Server layer: the *DisplayRecommendationEvent* which is in charge of displaying a new product recommendation on the RO, and the *UpdatingSharedAreasEvent* which updates the shared areas (SO, SB, and AO) with the collaborative interactions from other users.

3.1.2. Space Server Layer

This layer is responsible for the communication between the Space Client and the Space Recommender. Basically, it has three components. The first component is the *Users Management Module* which stores and manages users' information such as user identification and states (critiquing or sending suggestions), and the RO the user is interacting with. Second component is the *Content Management Module* which is responsible for updating the RO visualization (i.e., product image and features values) in each recommendation cycle. Third component is the *Communication Management Module* which maps user's events from the Space Client to recommendation's actions in the Group Conversational Recommender module and, in reverse, recommendation actions to user events.

In the following, we present how the user's events are mapped to the actions into the Space Recommender. Note that Section 3.1.3 details in depth the action in the Space Recommender.

1. The *CritiqueEvent* is converted to the *CritiqueAction*, as described in Signature 1. It contains the user that performed the critique, *userId*, the current recommended product, *productId*, the feature that receives the critique, *featureId* (e.g., *price*), the type of critique, *typeCritique* (i.e., (↑) greater than , (↓) lower than , or (<>) different), and the critique value, *critiqueValue* (i.e., the current value of the feature that have received the critique).

$$CritiqueAction(userId, productId, featureId, typeCritique, critiqueValue) \tag{1}$$

   For example, $critiqueAction(Red,\ HotelGarniStGeorg,\ price,\ \uparrow,\ 419)$ describes that *Red* user sends a critique about *Hotel Garni St Georg* for obtaining a recommendation product with a price higher than 419$. As described in Section 3.1.3, this critique is later stored in the individual user model (IM_j).
2. The *SuggestionEvent* is transformed to the *SuggestionAction* (see Signature 2), which contains the user that performed the suggestion, *userFromId*, the user which the product is suggested, *userToId*, and the suggested product, *productId*.

$$SuggestionAction(userFromId, userToId, productId) \tag{2}$$

   For example, $SuggestionAction(Red, Blue,\ HotelGarniStGeorg)$ is showing that *Red* user sends product *Hotel Garni St Georg* to the *blue* user.
3. The *PreselectedEvent* is mapped to the *PreselectedAction*, as described in Signature 3. The PreselectedAction contains as parameters the user that performed the pre-selection action, *userId* and the product added to the list of products in SO, $preSelectedProductId$.

$$PreselectedAction(userId, preSelectedProductId) \tag{3}$$

   For instance, $PreselectedAction(Blue,\ HotelGarniStGeorg)$ indicates that *Blue* user wants to add *Hotel Garni St Georg* to the stack.
4. The *ThrowWasteBasketEvent* is converted to the *ThrowWasteAction* (see Signature 4) whose parameters are the user that performed the throw waste basket action, *userId*, the product that is sent to waste basket, *productDeletedId*, and the area –SB or SO– that contains the product, *areaId*.

$$ThrowWasteAction(userId, productDeletedId, areaId) \tag{4}$$

   For example, $ThrowWasteAction(Blue,\ HotelGarniStGeorg, SO)$ removes *Hotel Garni St Georg* from the stack entered by user *Blue*. A product can be by several members of the group. This is necessary because users select one of their preferences in the stack as the final product chosen in the recommender before finishing the recommendation session.

5. The *FinishEvent* maps to the *FinishAction* described in Signature 5, which contains the user that performed the finish action, *userId* and the individual product selected, $productSelectedId$.

$$FinishAction(userId, productSelectedId) \tag{5}$$

For example, $FinishAction(Red,\ HotelGarniStGeorg)$ means that *Red* user has chosen *Hotel Garni St Georg* as their preferred product. Thus finishing the recommendation process to her.

Now we depict the actions received from the Space Recommender layer in the Space Server layer and how it sends to the Space Client layer.

1. The *Recommendation* action is transformed into the *Communication Management Module* to the *DisplayRecommendationEvent* previously introduced in Section 3.1.1. As shown in Signature 6, it contains the user who performed the critique, $userId$, the new recommended product, $productId$, and a list of features (i.e., the value for each product features), $featureValues$. Later, from the *Communication Management Module* the mapped event is sent to the *Content Management Module* which is responsible for updating the RO visualization through the *UpdatingSharedAreasEvent* (i.e., product image and features values) in each recommendation cycle.

$$Recommendation(userId, productId, featureValues) \tag{6}$$

2. The *UpdatingSharedAreas* action is connected to the *UpdatingSharedAreasEvent*. The definition of this action is described in Signature 7, which shows that the updated area (i.e., SO, SB or AO), $areaId$, the pre-selection, suggestion or awareness product identifier, $productId$, the entity that generates the update (i.e., the update action may be triggered by a user in case of the SO and SB, or by the system in case of SB and AO), $entityId$, and the compatibility of the current product with the individual user model (IM_j), $compatibility$.

$$UpdatingSharedAreas(areaId, productId, entityId, compatibility) \tag{7}$$

3.1.3. Space Recommender Layer

The Space Recommender layer is composed by two different modules: the *Group Conversational Recommender* (GCR) Algorithm and the *Group Management Module* (GMM). Next we describe each module in detail.

First, the *Group Conversational Recommender* Algorithm contains a case base (CB) of products, a set of individual user models (IM) (i.e., one for each user in the environment), and a group preference model (GM). Specifically, the case base $CB = \{p_1, ..., p_n\}$ contains a set of products or items for recommendation where p_i is the ith product. Each product is described with a set of a features ($F = \{f_1, ..., f_m\}$) (e.g., price, duration, or location). Each user is associated with an individual preference model, IM, that is made up of the critiques that they apply throughout the course of the recommendation session. We denote the individual preference model of a particular user j as IM_j, where each element in IM_j is a single unit critique. As new critiques are made by the user, their preference model is updated as described in [12,14]. This involves the addition of new critiques but may also involve the removal of past critiques if they conflict with, or are subsumed by the most recent critique. In addition, a group preference model (*GM*) is also maintained by combining the individual user models and associating critiques with the users who contributed them. GCR algorithm is a module that performs the recommendation process, controls data access (massive products with diversity of features), and updates group and individual user models during the recommendation process.

Briefly, the recommendation process starts when the user is inside of a *Space Client* and she performs one critique about a product feature through a Critique Element displayed on a Recommendation Object, this critique is sent to the *Group Conversational Recommender (GCR)* through the *Communication Management* module placed on the *Space Server*. Once received the *CritiqueAction* in *GCR* algorithm, it updates the individual user model (IM_j) and the group preference model (*GM*) with the new preference and it selects the next recommendation from the full set of products in the case base, *CB*. The recommendation is generated based on a quality score that considers both the similarity with the previous recommendation and the compatibility to the individual and the group models, as described in [14,15]. In this paper we have used the compatibility score defined in [19] but it is possible to use a different score in the framework such as those based on Reinforcement Learning [22,24]. Next, the *Group Conversational Recommender module* sends a new recommendation to the *Group Management* module. This module, using both the *Communication Management* and the *Content Management* modules, starts a *Recommendation* action to be displayed on the Recommendation Object of the user that performed the critique.

Second, the *Group Management Module* (GMM) coordinates the users' interactions performed using the collaborative elements in the Space Client. In particular, GMM contains a set of users U with all the information related to each one and the stack that stores the preselected products. Moreover, for each member of the group, GMM stores the personal information of the user, the current recommendation (i.e., the last product recommended by GCR to the user that is displayed currently in the recommendation object), the individual selection of that user when she finishes the recommendation process, the awareness that contains the current product view of each user and the compatibility to this user, and the suggestions that is a list of products received either from others or proactively from the recommender. GMM receives several actions from the *Communication Management Module*: (1) the *SuggestionAction* stores the product suggested to the list of suggestions of the receiving user; (2) the *PreselectedAction* adds a new product in the stack; (3) the *ThrowWasteAction* removes the selected product from the stack or the list of suggestions; and (4) the *FinishAction* finishes the user session. Note that GMM coordinates users' interactions. The particular implementation of the interaction modalities for communicating and allowing suggestion among users are deeply described in next section.

3.2. Interaction modalities in gCOACH

Figure 2 depicts the interface screen shown to each user in their browser. The example interface shown is focused on a skiing package domain. It is divided in several areas, each one with a specific interaction modality (i.e., individual or social) and functionality. Note first that each one of the users is represented by a color (look at the top of the figure). In the example the user is the red one. This color is used in the interface to denote to who belongs each one of the objects in the interface (i.e., RO, SO, SB, and AO).

Figure 2. The main gCOACH interface with a skiing package view

First area denoted with number one in Figure 2 represents the RO in the conceptual architecture shown in Figure 1. This area is devoted to *individual interaction*, which has three different views: category view, subcategory view, and product view. In gCOACH, the user interaction starts at a category view, which is able to focus the user in a specific category of the products (in our case, we use the location in the domain used in the experiments, see Figure 3a). Once selected a category, they are moved to a subcategory that in the domain analyzed corresponds to a resort view (see Figure 3b) where they can see all the available hotels. If the user selects one hotel, they arrive to an specific skiing package, as depicted in Figure 2.

Here, in the product view, a product is described in terms of their features and the particular value for each one of them. Additionally, each one of the features contains one or two buttons for performing critiques (i.e., these are the critique elements in Figure 1).

(a) gCOACH category view

(b) gCOACH subcategory view

Figure 3. Due to space limitations, we only show the individual interaction of the two views

The user is able to make a critique[1], which affords the user an opportunity to provide informative feedback. The user feedback represents the *CritiqueAction* in the Space Server layer (see section 3.1.2). This feedback is introduced to the individual model (*IM*) and the group preference model (*GM*). Next, GRC algorithm uses this informative feedback about the user taste and it answers this action by replacing the product displayed with a new recommendation that better matches with the preference expressed (i.e., the *RecommendationAction* which maps to the *DisplayRecommendationEvent* in Space Server layer described in section 3.1.2). Currently, GRC algorithm implementation is the one proposed in [14,15]. Furthermore, when individual users arrive to a particular product recommendation that satisfy their requirements and wish to draw it to the attention of the other group members, they can do this by performing a drag and drop and adding it to the suggestion box of another user or to a stack area, which is a social interaction modality.

Second area in gCOACH interface is for *keeping awareness* among members of the group –see AO in the conceptual architecture and look at number 2 in Figure 2. This area is refreshed by the system through the *UpdatingSharedAreas* action which maps to *UpdatingSharedAreasEvent* formalized in section 3.1.2. In addition, this area contains a list of color boxes, each one representing a group member. Each color box show which product is currently looking up each user. Users can browse this product by performing a click on it. Besides, the color box contains a 0-5 hearts ranking that represents how much compatible is the product to the user that currently is looking at it. The ultimate goal of this heart ranking is to know in which products are the users currently interested in and, if considered interesting by the user, suggest it to the group members. With this goal in mind, this area also allows the user to make suggestions of the current recommendation on area 1 to a specific user by doing a drag and drop of the product into the target user box. Then, the suggested product will appear in the suggestions box of the target user. These collaborative actions correspond to collaborative elements in the conceptual architecture (see Figure 1).

Third area is the *suggestion box*, depicted in Figure 2, number 4, which is represented as SB in the conceptual architecture as shown in Figure 1. These suggestions may come from anyone of the group members (i.e., the *SuggestionsAction* described in 3.1.2) or as a result of a *proactive* suggestion of the recommender algorithm (i.e., the *UpdatingSharedAreas* action, previously described in section 3.1.2). GRC algorithm suggests a product to the whole group when one or more cases exceeds a certain critical *compatibility threshold* with respect to the group preference model. As the previous area, each product is identified by the border color and shows few features including a compatibility ranking with the current user. Furthermore, it is also available the option of clicking on the product to take a look on it in the area of individual interaction.

Another social interaction modality is the *STACK area* (shown in Figure 2, number 5), which represents the SO in the conceptual architecture. It serves as repository of particular holiday recommendations the user is interested in and it is also useful to draw the attention of the other group members over a particular product (i.e., the *PreselectedAction* described in section 3.1.2). The stack stores summaries of the user's recommendations, as well as displaying compatibility information relating to group compatibility. Each product recommendation appears boxed with the color of the user that added it to the list and shows a summary of its features (in the skiing domain, *hotel name*, *resort*

[1] With a critique the user express a preference over a specific feature in line with their personal requirements (e.g., *cheaper* or *higher star rating for hotel*, etc.).

name, *kind of resort*, *number of stars*, and *price*). In addition of this content, a measure of compatibility between the current user and the item appears through a visual ranking of 0-5 hearts (0 hearts means a non suitable product for the current user and 5 heart means a perfect item to choose). In this area, the display is done through the *UpdatingSharedAreas* action shown in section 3.1.2. At any time, when users detect an attractive product in this area, they can open it in the individual interaction area to examine it by performing a simple click on it.

We consider an area for removing products that the user is not interested in anymore, this area was called the *waste basket* area (see Figure 2, number 3). It is used to the disposal of products from the *suggestions box area* or the STACK area . This functionality is activated when the user perform a drag and drop from one of these areas to the waste basket area (i.e., the *ThrowWasteAction* described in section 3.1.2).

Finally, Figure 2 shows at the top of the interface a *Finish session* button, which can be used by the user to finish her session. At this point, the user receives a new web page with the information of the products that she has added to the stack. The user has to choose one of them as her final decision product. Once all the users have selected their final decision product, an automatic consensus [23] is performed and a product is sent to each member as the final chosen product.

4. Evaluating the Usability in gCOACH

In this section we evaluate the usability with real users. Given the high fidelity of our prototype, we used the Summative usability evaluation method, which focuses on gathering both qualitative and quantitative data [2].

4.1. Setup and methodology

We recruited 44 participants, diverse in features such as age, gender, computer skills and experience in web-based environments. These participants were joined in groups of four heterogeneous participants for each test. The test was performed using a SKI domain that contains 153 European skiing holidays described by 41 features related to the resort (e.g., country or transfer time) and the accommodation (e.g., rating, price, or restaurant facilities). The test was conducted by a moderator and an observer. Users were requested to join in a group using an initial web page of the interface and they perform a search task of their favorite ski vacation. The test protocol consisted of four stages:

1. **Pre-test interview:** In this stage the moderator welcomed the user, briefly explained test objectives and asked about their previous experience with ski vacations and recommender systems.
2. **Training**: During this stage users were freely navigating on the web interface of gCOACH. They were asked to locate a predefined product using individual or social interaction modalities. The training stage finished when users discovered this product.
3. **Test:** In this phase users performed, without guidance, a test task that consisted of selecting a product that best satisfies the group preferences for going skiing together. To this end, users were asked to navigate, communicate, and provide suggestions with the aim of finding a consensus in the group to purchase a prod-

uct. However, users were free to finish the search process once they have found a product that best satisfies their preferences. Among the products in the stack, the user selects the preferred one. When all users finished the searching process, among the group preferred products (one for each user), GCR recommends the one that best satisfies the group. This product is shown as the final product for the group. During the task, a computer recorded the test session and the observer made annotations.

4. **Post-test questionnaire:** Users were asked to fill out a Web form that contains a satisfaction questionnaire consisting of 10 questions (see Table 1) that users answered using a five-point likert scale, where 1 correspond to “strongly disagree" and 5 to “strongly agree".

After the test, we analyzed the post-questionnaire to extract relevant data concerning usability. Next section describes this evaluation analysis.

4.2. Analysis of usability

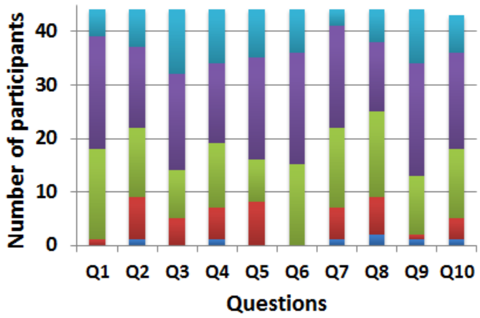
To collect user satisfaction measures we designed a post-test questionnaire, depicted in Table 1. Figure 4 presents collected data using a bar chart and a pie chart.

Table 1. Post-test questionnaire

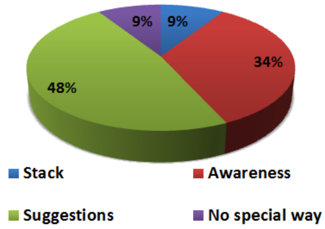
Question Number	Statement
Q1 (Effectiveness)	The items recommended to me matched my interests.
Q2 (Learnability)	The interface provides an adequate way for me to express my preferences.
Q3 (Learnability)	The recommender’s interface provides sufficient information.
Q4 (Learnability)	I became familiar with the recommender system very quickly.
Q5 (Usefulness)	It is easy for me to inform the system if I dislike/like the recommended item.
Q6 (Usefulness)	The recommended items effectively helped me find the ideal product.
Q7 (Satisfaction)	The recommender made me more confident about my selection/decision.
Q8 (Satisfaction)	The interface let me know easily where are the rest of my teammates at all times.
Q9 (Satisfaction)	Overall, I am satisfied with the recommender.
Q10 (Intention to use in the future)	I will use this system for buying products in the future.

Figure 4a depicts the results obtained from the post-questionnaire. Note that these results are related to the subjective perception of users but are quantitative data which gives us valuable information about users’ perception of usefulness and usability of our gCOACH framework. Overall, the quantitative results obtained from the questionnaire are very satisfactory. It is worth noting that 88.4% of the responses were ranked with 3 or more points, 2.3% of responses correspond to a minimal score (1 value), 9.3% were replied to a question with value 2.

Considering the learnability of the gCOACH (i.e., questions Q2 to Q4), a 84.1% of participants’ responses show that the users found the system easy to learn and evaluated this aspect with 3 or more points. The majority of the participants (35 of 44) have positively evaluated the Q2 question (i.e., 80.0% of participants ranked over 3 points this question). Moreover, the obtained result in Q3 is very satisfying too as 39 participants ranked over 3 points (i.e., 88.6% of the participants) and none of participants replied



(a) Grouping rating for each question in the post-questionnaire



(b) Area of most attention in the interface

Figure 4. Usability analysis of gCOACH interface

this question with a minimal score. In addition, the answers to question Q4 show that a 84.1% of participants evaluate positively this aspect (i.e., 37 of 44 ranked over 3 points this question).

With regard to the satisfaction of users, results of Q7 to Q9 show that a 86.4% of the participants positively evaluated this aspect with 3 or more points. In this way, the result in Q7 shows that a 85.0% of the participants perceive that the recommender makes more confidence about their selection (i.e., 35 participants that evaluate over 3 points this question). Additionally, the results in Q8 are satisfactory too where a 79.5% of the participants ranked with 3 or more points and the result in Q9 indicates that the majority of users was satisfied with the system (i.e., a 95.5% ranked this question with 3 or more points).

In addition, user's opinion reference to whether or not gCOACH is useful to them (i.e., usefulness aspect), responses to Q5 and Q6 answers depict that a 90.9% of the participants it evaluated with 3 or more points. Specifically, a 84.1% of the participants ranked with 3 or more points the Q5 question (i.e., 37 of 44) and it's worth noting that in the case of Q6 all of participants ranked this question with 3 or more points. Moreover, when the users answer about their intention to use this system in the future (Q10) only 1 participant ranked with a minimal score which means that users have a good perception about usefulness of the gCOACH system.

Finally, regards to perceived effectiveness of users during the recommendation process, results of Q1 show that a 97.7% of the participants evaluated positively this aspect (i.e., 43 of 44 participants ranked this question with 3 or more points).

We also asked users about the area on the interface that they paid more attention to the recommendations received: the stack area, the suggestion box area, the awareness area, or if there is not a preferred area. We report the results on the user's perception of most useful area in Figure 4b. A 48% of users prefer the suggestions box area as its main source of information about member's activities and for choosing a product. Note that suggestions received in this area come from teammates or proposed in a proactive way by the recommender algorithm. Next, the most preferred area with a 34% value is the awareness area. This means that group interaction is highly influenced by observation as users prefer to observe which products consult their teammates and then select one of them. A 9% of users prefer the stack area as its main source of recommendations. Finally, a 9% of users has not a clear preference over an area because they have been looking equally at all of them.

5. Conclusions

In this paper, we argued that there has been many recommendation scenarios that involve groups of inter-related users intending to participate in a group activity. But to date, most of the state-of-the-art approaches has focused on particular domains or on off-line environments. We have described in depth our proposal called gCOACH (COllaborative Advisory CHannel for group recommendation), which is an independent domain web-based environment that supports on-line group recommendation scenarios. It has been described the conceptual architecture of the environment. Moreover, it has been explained the developed user-interaction modalities available for the users to enhance communication, coordination and persuasion in the group participants. The usability of this novel interface has been evaluated with live-users. The results show that 88.4% of participants responded positively to the various social interaction modalities. The results also depict that mostly users prefer to be aware about the products that are suggested by other group members and it is their main influence for choosing products. Considering all the interaction modalities, as future work, we plan to add the behavior implicitly detected in gCOACH interface to define roles (i.e., leader, collaborator, or follower) among the teammates to influence in the group recommendation algorithm.

Acknowledgments

D. Contreras is supported by a doctoral fellowship "Becas Chile". This work has been supported by project ”DIANA: DIscourse ANAlysis for knowledge understanding” (TIN2012-38603-C02-02) from the Spanish Ministry of Science and Innovation.

References

[1] S. Berkovsky and J. Freyne. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proc. of the fourth ACM conference on Recommender systems*, pages 111–118. ACM, 2010.

[2] D. Bowman, J. Gabbard, and D. Hix. A Survey of Usability Evaluation in Virtual Environments : Classification and Comparison of Methods 1 Introduction and motivation 2 Distinctive characteristics of VE evaluation. *Presence: Teleoperators and Virtual Environments*, 11(4):404–424, 2002.

[3] A. Crossen, J. Budzik, and K. J. Hammond. Flytrap: Intelligent group music recommendation. In *Proc. of the Int. Conf. on Intelligent User Interfaces*, pages 184–185, USA, 2002. ACM.

[4] H. De Silva, K. Moessener, and F. Carrez. Group knowledge management for context-aware group applications and services. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 2851 –2855, 2009.

[5] I. Garcia, S. Pajares, L. Sebastia, and E. Onaindia. Preference elicitation techniques for group recommender systems. *Inf. Sci.*, 189:155–175, Apr. 2012.

[6] A. Jameson. More than the sum of its members: Challenges for group recommender systems. In *Proc. of the International Working Conference on Advanced Visual Interfaces*, pages 48–54, Italy, 2004.

[7] A. Jameson and B. Smyth. Recommendation to groups. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The adaptive web*, pages 596–627. Springer-Verlag, Berlin, Heidelberg, 2007.

[8] H. Lieberman, N. V. Dyke, and A. Vivacqua. Let's browse: A collaborative web browsing agent. In *Proc. of the International Conference on Intelligent User Interfaces*, pages 65–68, New York, 1999. ACM.

[9] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Model. User-Adapt. Interact.*, 14(1):37–85, 2004.

[10] J. Masthoff. Group recommender systems: Combining individual models. In *Recommender Systems Handbook*, pages 677–702. Springer US, 2011.

[11] J. McCarthy and T. Anagnost. Musicfx: An arbiter of group preferences for computer aupported collaborative workouts. In *Proceedings of Conference on Computer Supported Cooperative Work*, pages 363–372, 1998.

[12] K. McCarthy, L. McGinty, B. Smyth, and M.Salamó. Social interaction in the cats group recommender. In P. Brusilovsky, editor, *Proceedings of the Workshop on the Social Navigation and Community-Based Adaptation Technologies at Adaptive Hypermedia (AH-06)*, Dublin, Ireland, 2006.

[13] K. McCarthy, L. McGinty, B. Smyth, and M. Salamó. The Needs of the Many: A Case-Based Group Recommender System. In *Proc. of the European Conf. on Case Based Reasoning*, pages 196–210. Springer Verlag, 2006.

[14] K. McCarthy, M. Salamó, L. Coyle, L. McGinty, and B. S. . P. Nixon. CATS: A Synchronous Approach to Collaborative Group Recommendation. In *Proceedings of the FLAIRS 2006 Conference*, pages 1–16. Springer Verlag, 2006. Florida, USA.

[15] K. McCarthy, M. Salamó, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. Group recommender systems: A critiquing-based approach. In *Proc. of the Int. Conf. on Intelligent User Interfaces*, pages 267–269, Australia, 2006.

[16] L. McGinty and J. Reilly. On the evolution of critiquing recommenders. In *Recommender Systems Handbook*, pages 419–453. Springer, 2011.

[17] M. O'Connor, D. Cosley, J. Konstan, and J. Riedl. PolyLens: A Recommender System for Groups of Users. In *Proc. of European Conf. on Computer-Supported Cooperative Work*, pages 199–218, 2001.

[18] S. Pizzutilo, B. D. Carolis, G. Cozzolongo, and F. Ambruoso. Group modeling in a public space: Methods, techniques and experiences. In *Proceedings of WSEAS AIC 05*, Malta, 2005. ACM.

[19] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Incremental Critiquing. In M. Bramer, F. Coenen, and T. Allen, editors, *Research and Development in Intelligent Systems XXI. Proceedings of AI-2004*, pages 101–114. Springer, 2004. UK.

[20] F. Resatsch, S. Karpischek, U. Sandner, and S. Hamacher. Mobile sales assistant: Nfc for retailers. In *Proc. of Int. Conf. on HCI with mobile devices and services*, pages 313–316. ACM, 2007.

[21] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[22] M. Salamó and S. Escalera. Increasing retrieval quality in conversational recommenders. *IEEE Transactions on Knowledge and Data Engineering*, 24(10):1–14, 2012.

[23] M. Salamó, K. Mccarthy, and B. Smyth. Generating recommendations for consensus negotiation in group personalization services. *Personal Ubiquitous Comput.*, 16(5):597–610, June 2012.

[24] M. Salamó, J. Reilly, L. McGinty, and B. Smyth. Improving Incremental Critiquing. In *16th Artificial Intelligence and Cognitive Science*, pages 379–388, 2005.

[25] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, pages 291–324. Springer-Verlag, 2007.

[26] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier. Analysis of strategies for building group profiles. In *User Modeling, Adaptation, and Personalization*, Lecture Notes in Computer Science, pages 40–51. Springer Berlin / Heidelberg, 2010.

[27] B. Smyth. Case-based recommendation. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, pages 342–376. Springer-Verlag, 2007.

[28] B. Smyth and L. McGinty. An Analysis of Feedback Strategies in Conversational Recommender Systems. In P. Cunningham, editor, *Proceedings of the 14th National Conference on Artificial Intelligence and Cognitive Science*, 2003. Dublin, Ireland.

[29] C.-Y. Wang, Y.-H. Wu, and S.-C. Chou. Toward a ubiquitous personalized daily-life activity recommendation service with contextual information: a services science perspective. *Information Systems and E-Business Management*, 8:13–32, 2010.

[30] Z. Yu, X. Zhou, Y. Hao, and J. Gu. Tv program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction*, 16:63–82, 2006.