Engineering Letters, 13:3, EL_13_3_22 (Advance online publication: 4 November 2006)

Designing a Web-based Testing Tool for Multi-Criteria Recommender Systems

Nikos G. Manouselis, and Constantina I. Costopoulou

***Abstract*—A plethora of real-life applications of recommender systems in the Web exists. These systems help users to deal with information overload, by providing personalized recommendations regarding online content and services. Due to the dynamic and changing parameters of the various application contexts, careful testing and parameterization has to be carried out before a recommender system is finally deployed in a real setting. This paper proposes a Web-based tool that allows for simulated testing of a special class of multi-criteria recommender systems, namely multi-attribute collaborative filtering systems. More specifically, it introduces a number of collaborative filtering algorithms that are based on Multi-Attribute Utility Theory (MAUT), and presents the design and implementation of a Web-based tool termed as the Collaborative Filtering Simulator (CollaFiS), which may be used for their simulated testing. The specification of CollaFiS is described using the Unified Modelling Language (UML). In addition, a characteristic scenario of the CollaFiS usage is illustrated.**

***Index Terms*—Collaborative filtering, Multi-Criteria Decision Making, Recommender systems, Simulation.**

I. Introduction

The area of Web-based recommender systems attracts high research interest due to its challenging open issues [1]. Internet users are often becoming overwhelmed by the flow of online information, and are in need of adequate tools to help them manage the situation [11]. Recommender systems are a family of such tools. The term "recommender system" generally describes any system that produces individualized recommendations as output, or has the effect of guiding the user in a personalized way to interesting or useful items, in a large space of possible options [7]. There is an abundance of real-life applications of recommender systems in the Web, which may help Internet users to deal with information overload by providing personalized recommendations regarding online content and services [23]. The application domains range from recommendation of commercial products such as books, CDs and movies, to recommendation of more complex items such as quality methods and instruments [20].

Manuscript received April 10, 2006.

N. G. Manouselis is with the Informatics Laboratory of the Agricultural University of Athens, 75 Iera Odos Str., 11855, Athens, Greece (phone: +30-6945-166400; fax: +30-210-5294199; e-mail: nikosm@aua.gr).

C. I. Costopoulou is with the Informatics Laboratory of the Agricultural University of Athens, 75 Iera Odos Str., 11855, Athens, Greece (e-mail: tina@aua.gr).

In a recommender system, the items of interest and the user preferences are represented in various forms, which may involve one or more variables. Particularly, in systems where recommendations are based on the opinion of others, it is crucial to incorporate the multiple criteria that affect the users' opinions into the recommendation problem. Several recommender systems have already been engaging multiple criteria for the production of recommendations. Such systems, referred to as multi-criteria recommenders, early demonstrated the potential of applying multi-criteria decision making (MCDM) methods to facilitate recommendation in numerous application domains. These include movie recommendation [25], [26], [29], [30], restaurant recommendation [38], product recommendation [2], [8], [10], [14], [17]-[19], [27], [34], [36], [37], and others [31].

Most evaluation studies of recommender systems [6], [9], [12], [28] indicate that careful testing and parameterization has to be carried out, before a recommender system is finally deployed in a real setting. On the other hand, most current multi-criteria recommenders usually remain at a design or prototyping stage of development. Testing methods and tools that may support their systematic implementation and evaluation in the context of real-life applications are limited. Experimental testing for multi-criteria recommenders could be greatly facilitated by a testing tool that would allow the simulated execution of multi-criteria recommendation algorithms under controlled experimental conditions, similar to tools proposed for single-criterion recommenders, e.g., [24]. In addition, single-criterion systems use, for their experimental testing, publicly available data sets from real operation, e.g., the MovieLens, EachMovie and Jester data sets [13], [22]. Such data sets may be freely used by recommender systems' researchers, in order to experimentally test new features or systems, e.g., proposed recommendation algorithms, in a simulated environment, before their actual deployment. In a similar manner, the systematic evaluation of multi-criteria recommenders would require their experimental investigation using data sets with multi-criteria evaluations [13]. Unfortunately, multi-criteria evaluation data sets from real-life applications are not publicly available until today, therefore only experimental data sets that have been collected through pilot user studies or synthetic (simulated) data sets can be used for this purpose.

To this direction, this paper proposes the design and development of a Web-based testing tool that supports the simulated study of a special class of multi-criteria recommenders, namely multi-criteria (or multi-attribute) collaborative filtering systems. The proposed testing tool allows for the creation of a synthetic data set with multi-criteria evaluations, while manually defining several of its features. It also provides the option of selecting and executing one or more of multi-attribute collaborative filtering algorithms upon the created data set. In this way, the performance of the algorithms may be studied under experimental conditions that mimic the ones expected during actual operation. The remainder of the paper is structured as follows. Section 2 introduces how collaborative filtering is modeled as a MCDM problem, and describes the multi-attribute algorithms that are considered for the tool. In section 3, the specification of the Collaborative Filtering Simulator (CollaFiS) is described. Section 4 describes a usage scenario of CollaFiS. Finally, in section 5 the conclusions of this study and directions for future research are outlined.

II. Multi-Attribute Collaborative Filtering Algorithms

In related research, the problem of recommendation has been identified as the way to help individuals in a community to find the information or products that are most likely to be interesting to them or to be relevant to their needs [16]. It has been further refined to the problem of (i) predicting whether a particular user will like a particular item (prediction problem), or (ii) identifying a set of *N* items that will be of interest to a certain user (top-*N* recommendation problem) [9]. A particular class of recommender systems is collaborative filtering ones. This section aims to introduce multi-attribute collaborative filtering and the algorithms that will be integrated in CollaFiS.

A. Collaborative Filtering Problem Modeling

Collaborative filtering systems help people make choices based on the opinions of other people, and actually automates the process of "word-of-mouth" recommendations: items are recommended to a user based upon evaluations assigned to them by other people with similar taste [32], [35]. In collaborative filtering, the recommendation problem can be formulated as follows [1]: let *C* be the set of all users and *S* the set of all possible items that can be recommended. We define $U^c(s)$ as a utility function $U^c(s): C \times S \rightarrow \Re^+$ that measures the appropriateness of recommending an item *s* to user *c*. It is assumed that this function is not known for the whole *C* x *S* space but only on some subset of it. Therefore, in the context of recommendation, we want to use the evaluations that users in *C* have provided about items in *S*, in order to be able for each user $c \in C$:

(i) to estimate (or approach) the utility function $U^c(s)$ for an item *s* of the space *S* for which $U^c(s)$ is not yet known; or,

(ii) to choose a set of *N* items $s \in S$ that will maximize $U^c(s)$:

$$\forall c \in C, s = \max_{s \in S} U^c(s) \qquad (1)$$

In most recommender systems, the utility function $U^c(s)$ usually considers one attribute of an item, e.g., its overall evaluation or *rating*. Nevertheless, utility may also involve more than one attributes of an item. The recommendation problem therefore becomes a multi-attribute or multi-criteria one.

Engaging Multi-Attribute Utility Theory (MAUT) [15], the recommendation problem in collaborative filtering systems may be defined as a decision problem with multiple variables, called *multi-attribute collaborative filtering*, in the following manner. The multiple attributes describing an item *s* are defined as a set of criteria upon which a user evaluates the item. The utility function $U^c(s)$ is then referred to as the *total utility* of an item *s*, which is calculated by synthesizing the *partial utilities* of the item *s* on each one of the criteria. The criteria are non-decreasing real valued functions, defined on *S* as follows:

$$g_i : S \rightarrow \Re / s \rightarrow g_i(s) \in \Re \qquad (2)$$

where $g_i(s)$ is the evaluation of the item *s* on the i^{th} criterion (i=1,…,n). Thus, the multi-criteria evaluation of a item $s \in S$ is given as a vector $\underline{g}(s) = [g_1(s), g_2(s), \ldots, g_n(s)]$.

The approach adopted for preference modeling belongs to the family of Value Focused ones [15]. It formulates the global preference model as an additive value function, where an importance weight is associated with each evaluation criterion. Assuming that there is no uncertainty during the decision making, the total utility of a item $s \in S$ for a user $c \in C$ can be expressed as:

$$U^c(s) = \sum_{i=1}^{n} u_i^c(s) = \sum_{i=1}^{n} w_i^c \cdot g_i^c(s) \qquad (3)$$

where $u_i^c(s)$ is the partial utility function of the item *s* on criterion g_i for the user *c*, $g_i^c(s)$ is the evaluation that user *c* has given to the item *s* on criterion g_i, and w^c_i is the weight indicating the importance of criterion g_i for the particular user *c*, with:

$$\sum_{i=1}^{n} w_i^c = 1 \qquad (4)$$

The linear function of (3) is the simplest and most popular form of an additive value function. Other forms that could be used include an ideal point model, dependencies and correlations, as well as diminishing utility forms [31].

B. Proposed Algorithms

Let us assume that a set of users $M \subseteq C$ has evaluated a subset of items *E* from the whole population of available items *S* (that is $E \subseteq S$). As mentioned before, the goal of recommendation is to provide to a particular user $c \in C$ (who we will refer to as the *active user*), with an estimation of the total utility of the items that this user has not evaluated yet. This corresponds to the prediction of the total utility $U^c(s')$, for each item $s' \in S$ that this user has not evaluated. For the

CollaFiS tool, three different multi-attribute collaborative filtering algorithms are considered: the *Similarity Per Priority (PW)* algorithm, the *Similarity Per Evaluation (PG)* algorithm, and the *Similarity Per Partial Utility (PU)* algorithm. These algorithms are described elsewhere [21] and belong to the family of "neighborhood-based" ones [6], [12]. They create a "neighborhood" of $D \subseteq M$ users that have similar preferences with the active user and who have previously evaluated s', and calculate the prediction of $U^c(s')$ according to how the users in the neighborhood *D* have evaluated s'. That is, if we assume that $z \in \Re^+$ is the number of users in a neighborhood *D*, the algorithms aim to predict $U^c(s')$ according to the *z* utilities $U^d(s')$ of this item for each neighbor $d \in D$. Differences exist in the way each algorithm calculates the similarity between the preferences of the active user to the preferences of other users. In addition, several design options are considered for the further parameterization of the algorithms, according to related literature for single-criterion recommendation algorithms [6], [12]. This leads to a number of different variations for each one of the three algorithms [21].

III. CollaFiS Specification and Deployment

CollaFiS is a Web-based testing tool that aims to support people interested in parameterizing and evaluating the introduced collaborative filtering algorithms under various experimental conditions. More specifically, the general requirements for CollaFiS have been the following:

- To allow the creation a synthesized data set of multi-criteria evaluations, while manually defining some of its features. For example, it should allow the manipulation of the total number of items that may be recommended; the number and scales of the criteria upon which items may be evaluated; the total number of users that provide an evaluation of items; and the total number of multi-criteria evaluations that users have already provided over items.
- To facilitate the preparation of a data set with multi-criteria evaluations for the execution of an algorithm on it. That is, to allow for splitting of a data set into a training and a testing component. The training component will have to be used as input for the tested algorithm, in order to provide recommendations about the items of the testing component. Several options for splitting the data set should be provided, e.g., "50%-50%", "80%-20%", "90%-10%", and "all-but-one" splittings.
- To allow for the selection of one from the considered multi-attribute collaborative filtering algorithms, allowing for their parameterization (according to the variety of available design options).
- To execute the parameterized algorithm and measure its performance according to different metrics. More specifically, to predict the evaluation of each item in the testing component, using the training component as input. Then, to measure the mean absolute error (MAE) of all predictions compared to the actual evaluations, the number of items in the testing component for which it was possible to produce a prediction (coverage), as well as the total execution time needed for calculating a single prediction.

In the following paragraphs, a software development process is followed in order to model these requirements into system specifications and implement the initial version of CollaFiS.

A. Software Development Process

Software development involves a number of steps to be carried out, so that a software system is properly modelled, analysed, designed, specified and implemented. For CollaFiS, we use the Unified Modelling Language (UML) and follow the Rational Unified Process (RUP) [4]. UML is the de-facto software industry standard modelling language for visualizing, specifying, constructing and documenting the elements of systems in general, and software systems in particular [5]. It provides a rich set of graphical artifacts to help in the elicitation and top-down refinement of software systems from requirements capture to the deployment of software components [33]. In UML, a system is described using different levels of abstraction and considering various views (i.e. Business view, Use Case view, Design and Process view, Implementation view). Each view is realized using different UML modelling tools (diagrams). UML is largely process-independent, meaning that it can be used with a number of software development processes. RUP is an iterative software development process that is especially well suited to UML. The RUP development starts with four process workflows (business modelling, requirements or system use case modelling, analysis and design, and implementation) that adopt the various UML views, and continues with five more process workflows (test, deployment, configuration management, project management and environment). The four UML–based process workflows are:

- *Business modelling* that describes the structure and dynamics of the business activities around the system. It results in UML Business Use Case Diagrams, Activity Diagrams and analysis-level Class Diagrams with business entities.
- *Requirements* or *System Use Case Modelling* that describes user requirements using UML Use Case Diagrams. It results in identifying actors, use cases, and Use Case Diagrams.
- *Analysis and design* that describes the structural and architectural aspects of the system. Analysis results in describing the flow of events in use cases by developing analysis-level UML Sequence or Collaboration Diagrams. Design results in developing design-level UML Sequence or Collaboration Diagrams, Class Diagrams, Statechart Diagrams, Component Diagrams and a Deployment Diagram.

- *Implementation* that takes into account the above UML Diagrams in order to develop software components and integrate them in an initial version of the software system.

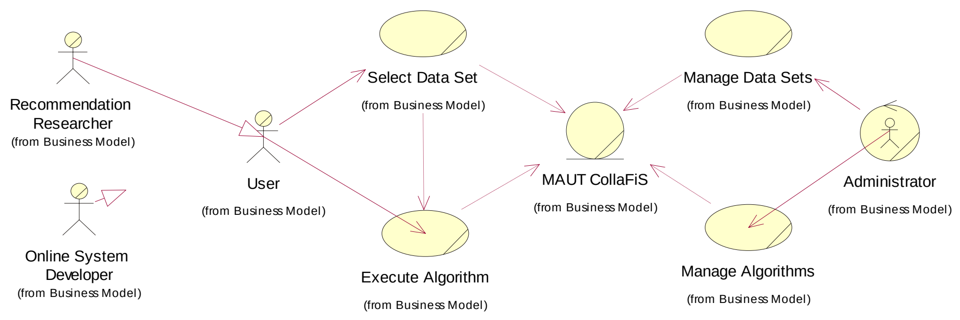


Fig. 1: The CollaFiS Business Use Case Diagram.

B. RUP Application for CollaFiS

In this section, the four UML-based process workflows for CollaFiS are described in detail.

Business Modeling

Business modelling concentrates on the business activities that will be generally supported by the testing tool (referred to here as the CollaFiS business), while the rest of process workflows focus on the software system that will be built. It concerns the identification of business actors (anyone or anything that is external to the CollaFiS business but interacts with it), business workers (roles within the CollaFiS business), and business use cases (a group of related workflows within the CollaFiS business that provide value to the business actors). A Business Use Case Diagram illustrates business use cases, business actors, and business workers for business activities, as well as the interactions between them. Activity Diagrams can also be used to model the workflow through a particular business use case.

The CollaFiS business refers to the usage of a Web-based testing tool that will facilitate its users with the execution of multi-attribute collaborative filtering algorithms on synthetic and real data sets. Thus, the business actors that will take advantage of the CollaFiS business are the following:

- *Recommendation Researchers*, who are working on multi-criteria recommenders and are interested into exploring how the proposed algorithms perform under various conditions, by controlling the algorithm and data set parameters. For example, they use CollaFiS to test how the proposed multi-attribute algorithms perform on various forms of data sets, e.g., very large and sparse ones.
- *Online System Developers*, who are interested to implement a multi-attribute collaborative filtering system in a particular environment. These developers are expected to use CollaFiS in order to test which algorithm performs better under conditions similar to the ones of a particular application context, e.g., an e-market.

Both business actors can be considered as a general business actor called *User* of the CollaFiS business. Furthermore, the business worker of CollaFiS is the following:

- *Administrator*, who is an entity internally concerned with the administration of the CollaFiS tool and its proper operation, managing user accounts, or uploading external data sets.

The above business actors and worker are involved in a number of business use cases, which are outlined in the CollaFiS Business Use Case Diagram (Fig. 1).

Requirements

Use cases and actors define the scope of the system built [4]. Use cases include anything that is inside the system. Actors include anything that is external, interacting with the system. Reconsidering the CollaFiS business actors, we identified these:

- *User*, a researcher or developer that uses CollaFiS in order to create some synthetic data set(s), and test how some algorithm variation(s) perform on a selected data set.
- *Administrator*, responsible for the proper functioning of CollaFiS, update with new data sets, and management of user accounts.

The above actors are engaged in a number of CollaFiS system use cases, which are:

- *Create User Account*, concerns submitting a registration request to the CollaFiS tool so that a user profile is created.
- *Approve User Account*, concerns checking and approving (or not) a user registration request.
- *Delete User Account*, concerns removing a user account from the tool.
- *Login*, concerns logging into the CollaFiS tool. It aims to allow only registered users to perform certain operations.
- *Create Data Set*, concerns using the tool to create a synthetic data set with the desired properties, defining parameters such as number of users, number of items, number and scales of criteria, as well as number of evaluations. After the data set is created, the user views its properties, e.g., mean evaluations per user, mean evaluations per item, percentage of items evaluated.
- *Prepare Data Set*, concerns processing a data set that has been created or imported, in order to prepare it for the algorithm execution. The initial data set can be divided into two or more component sub-sets. Each sub-set is then split into a training component and a testing one, of a desired size, e.g., the splitting may be “50%-50%” or “80%-20%”. After the preparation, the user views the properties of the

produced sub-sets and components, e.g., how many evaluations each component includes.

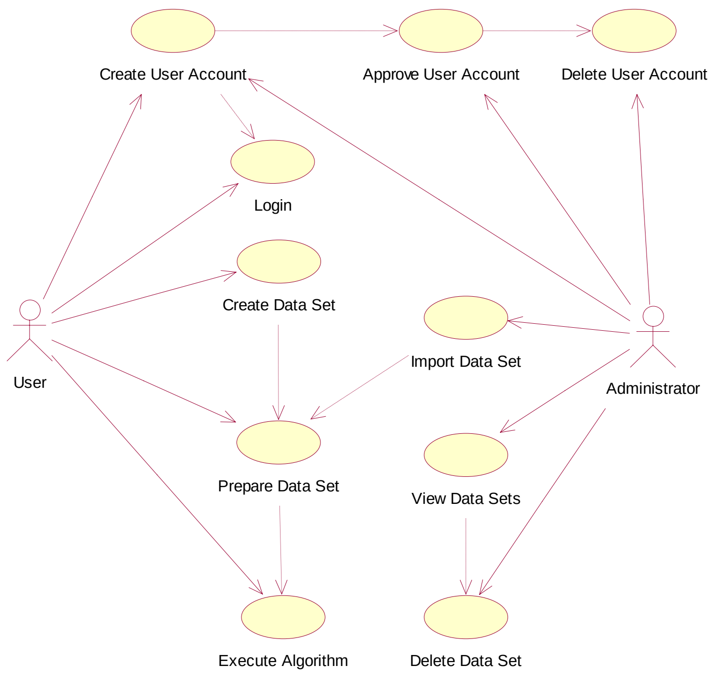


Fig. 2: The CollaFiS Use Case Diagram.

- *Execute Algorithm*, concerns selecting which algorithm will be used for the simulation and the targeted data set, defining the algorithm parameters to be considered, e.g., number of neighbors to be used for producing a recommendation, and executing the prameterized algorithm on the selected data set. After the simulation is completed, the results are presented to the user.
- *View Data Sets*, concerns providing an overview of the data sets currently available to the tool, and of their properties.
- *Import Data Set*, concerns uploading to the CollaFiS tool an existing data set (usually with real and not synthetic data), following the appropriate format so that the simulation tool can process it.
- *Delete Data Set*, concerns removing a data set from the CollaFiS tool.

Fig. 2 illustrates the CollaFiS Use Case Diagram that provides an overview of the identified actors and use cases, as well as, the interactions between them.

Analysis and Design

In the CollaFiS tool, use cases are supported by a set of corresponding sub-systems. The following CollaFiS sub-systems have been identified:

- *Interface sub-system*, responsible for the interaction with the users, passing information to and from users to CollaFiS.
- *Algorithms sub-system*, responsible for the execution of the algorithms, the production of recommendations, and the measurement of performance, based on the information in the data sets.
- *Data Set Processing sub-system*, responsible for processing an initial data set and preparing it for being used in a simulation.
- *Data Set Database sub-system*, responsible for storing the initial and the processed data sets.
- *User Account Management sub-system*, responsible for the creation, update and deletion of user accounts.
- *Users Database sub-system*, responsible for storing the information of the user accounts.

During the CollaFiS analysis, the interactions between the involved actors and the CollaFiS sub-systems are illustrated using Sequence Diagrams. Fig. 3 presents the Sequence Diagram for the Prepare Data Set use case. Similar Sequence Diagrams have been produced for the rest of the CollaFiS use cases.

During the CollaFiS design, a number of Class Diagrams have been developed, representing the information that CollaFiS sub-systems hold and exchange. Class Diagrams are used to display the classes in a system. Fig. 4 illustrates the Class Diagram for the Prepare Data Set use case. A Component Diagram displays the components in the system and the dependencies between them, where components are the physical modules of software code. Fig. 5 presents the Component Diagram for the case of the Prepare Data Set use case. Similar Component Diagrams have been produced for the rest of the CollaFiS use cases.

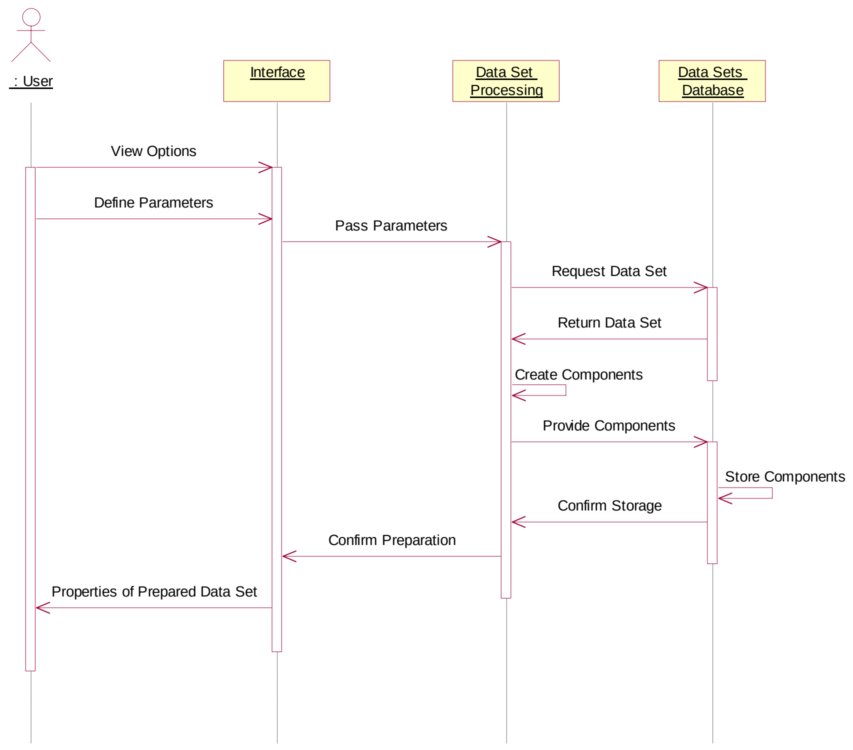


Fig. 3: The Sequence Diagram for Prepare Data Set use case.

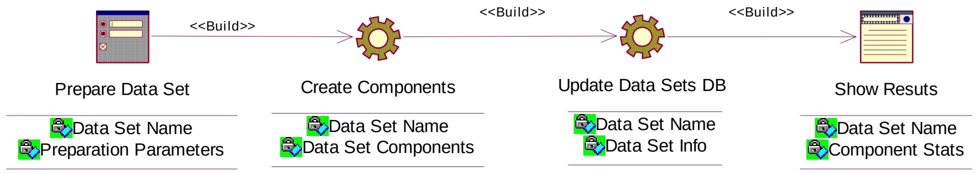


Fig. 4: The Class Diagram for Prepare Data Set use case.

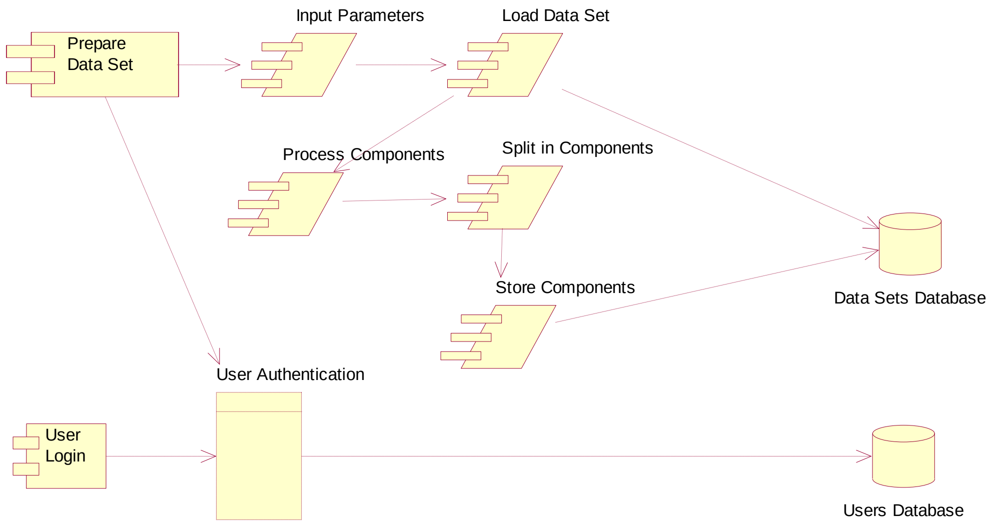


Fig. 5: The Component Diagram for Prepare Data Set use case.

The final result of the CollaFiS design is the Deployment Diagram. A Deployment Diagram is concerned with the physical deployment of the system, including issues such as the network layout and the location of components in the network. It illustrates all nodes of the system network, the connections between them, and the processes that will run on each one. Fig. 6 presents the CollaFiS Deployment Diagram.

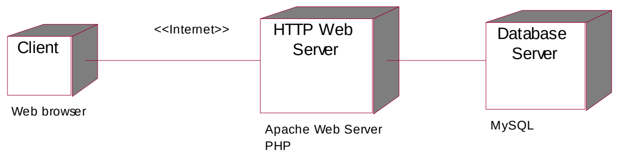


Fig. 6: The CollaFiS tool Deployment Diagram.

Implementation

The implementation process workflow has led to the initial version of the system. More specifically, following the results of the previous workflows, a first version of CollaFiS has been implemented. Fig. 7 and Fig. 8 present characteristic screenshots of the system. More specifically, Fig. 7 presents a screenshot of how the properties of a synthetic data set are provided prior to its creation. Fig. 8 presents a screenshot of characteristic simulation results.

The CollaFiS prototype has been developed using MySQL and PHP technologies in order to be easily accessed and used online. This version provides all functionalities related to the creation of synthesized data set, the parameterization of multi-attribute collaborative filtering algorithms, and their execution. A number of additional functionalities are considered for implementation in the next version of the system, before it is made public. More specifically, the users of CollaFiS will be allowed to upload their own data sets in the system using a text file with an appropriately specified SQL query, and to automatically create a series of synthetic data sets with varying properties upon which a single specific algorithm will be executed and tested.

IV. Scenario Of Use

In this section, we present a characteristic scenario of use for the implemented CollaFiS tool. More specifically, we present how several variations of a selected algorithm are compared in the context of supporting wine recommendation in an electronic market (e-market) with agricultural products. The selected case study is a Greek e-market that provides access to a product catalog of Greek wines (http://www.greekproducts.com/). The wines on offer come from various producers and areas, and often have totally different quality characteristics. The e-market under study is an established online market with a number of functionalities, which do not include recommendation so far.

Since there is no available data set of multi-attribute evaluations of wines from real users, we used CollaFiS to create a data set with multi-criteria evaluations of wines, similar to the one expected during actual operation. Then, a particular algorithm has been selected and executed. More specifically, the following steps have been performed:

Step 1: Definition of data set properties. In this step, the properties of the desired data set have been defined. More specifically, eight criteria affecting consumers' choice of wine have been adopted from related literature [3] as appropriate for the evaluation of wines, upon a scale of "1 to 7". In addition, 50 items have been considered, since the wine catalog of the e-market under study includes about 50 wines. Based on feedback from the e-market operator, it has been estimated that about 1,000 different users have visited the e-market in the last six months. Assuming that a 10% of them (100 users) will be actively providing evaluations for at least 3 wines each, it has been estimated that about 150 evaluations will be available in the system for producing wine recommendations.

Step 2: Creation of data set. According to the definition of the properties, the synthetic data set has been created using CollaFiS. It has been decided to include 50 items, 100 users, and 300 multi-criteria evaluations of items upon eight criteria.

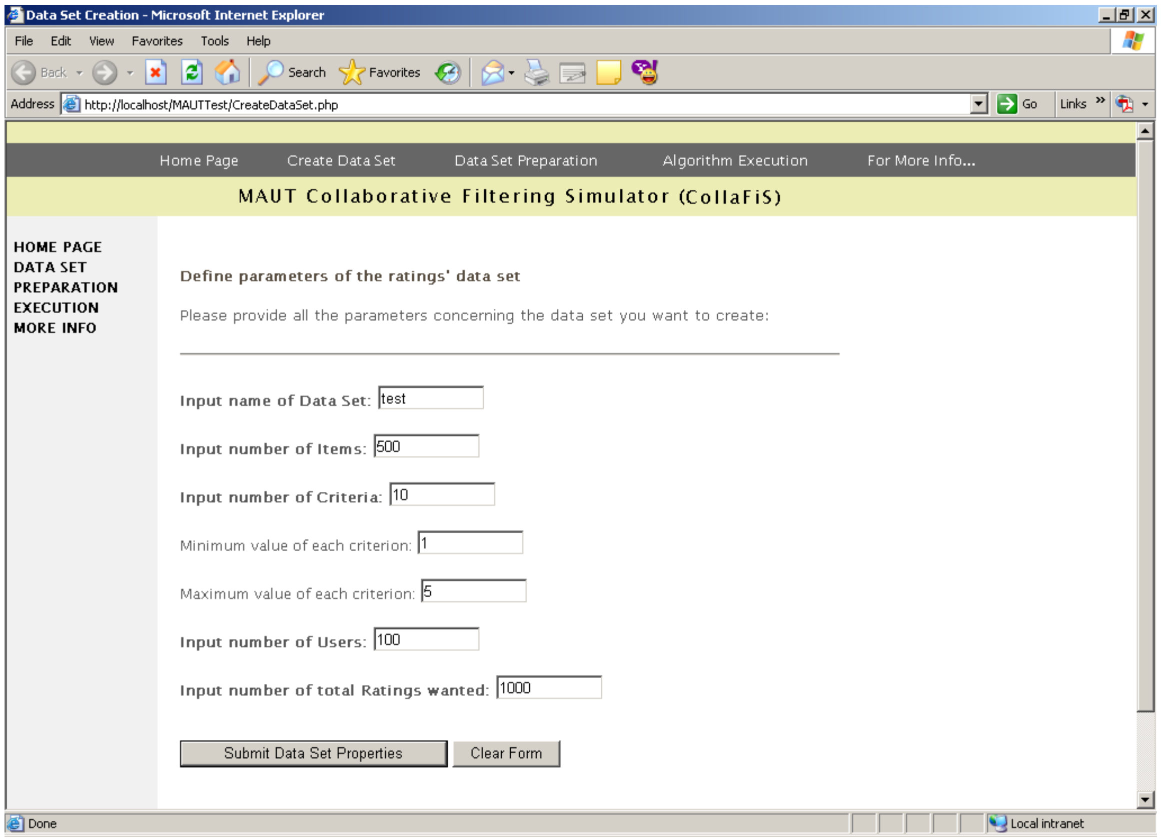
Step 3: Preparation of data set. The created data set has been processed by CollaFiS and split into two components to be used for experimentation. In particular, a "50%-50%" splitting has been adopted, so that the training component includes 150 wine evaluations (as estimated in step 1), and the testing component includes 150 evaluations as well.

Step 4: Algorithm selection and execution. The prepared data set has been used in order to experimentally test the performance of a selected algorithm, for various parameter values. More specifically, the Similarity Per Priority (PW) algorithm has been chosen, considering two variations, and ranging the maximum number of neighbors (MNN) considered from "1 to 20". In addition, the selected algorithm variations have been compared to four simple ones (two random-valued ones and two calculating simple mean values), used as control measures.

Step 5: Results collection and analysis. The selected algorithm variations have all been executed on the created data set components, and results of their performance measured in terms of MAE, coverage, and execution time. These results allowed for the selection of the most appropriate algorithm variation for the studied data set. In particular, a Pearson PW algorithm variation with MNN=8 has been selected, providing a combination of low MAE, high coverage, and fast execution for the examined data set. Fig. 9 illustrates an example of this analysis, where the MAE results from all algorithm variations are compared.

V.Conclusion

The systematic evaluation of recommender systems often requires their simulated experimental investigation for particular application domains, using data sets from actual usage and operation [13]. Until today, there have not been in the literature any proposals of Web-based testing tools that will support this task for multi-criteria recommenders.


Data Set Creation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/MAUTTest/CreateDataSet.php
Go Links
Home Page Create Data Set Data Set Preparation Algorithm Execution For More Info...
HOME PAGE
DATA SET
PREPARATION
EXECUTION
MORE INFO
Done
Local intranet

MAUT Collaborative Filtering Simulator (CollaFiS)

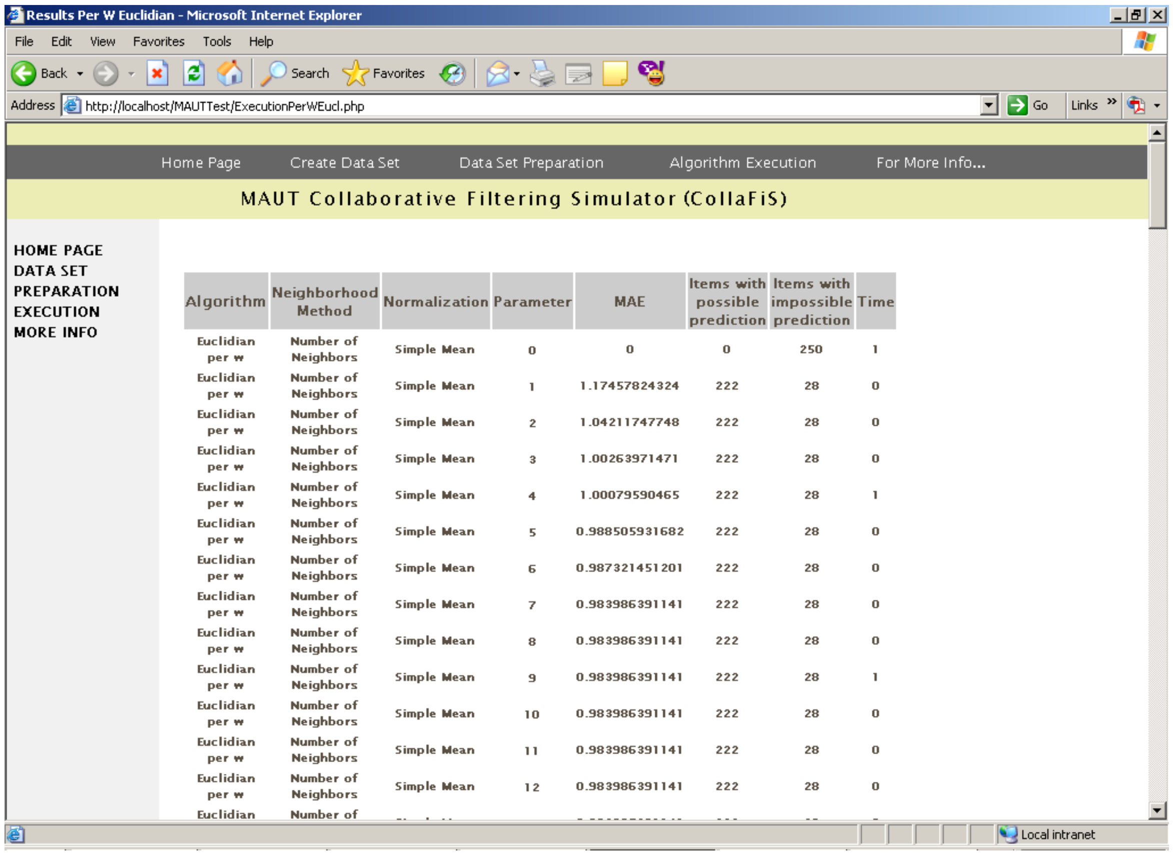
Define parameters of the ratings' data set

Please provide all the parameters concerning the data set you want to create:

Input name of Data Set:	test
Input number of Items:	500
Input number of Criteria:	10
Minimum value of each criterion:	1
Maximum value of each criterion:	5
Input number of Users:	100
Input number of total Ratings wanted:	1000

Submit Data Set Properties | Clear Form

Fig. 7: A CollaFiS screenshot during the creation of a synthetic data set.


Results Per W Euclidian - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/MAUTTest/ExecutionPerWEucl.php
Go Links
Home Page Create Data Set Data Set Preparation Algorithm Execution For More Info...
HOME PAGE
DATA SET
PREPARATION
EXECUTION
MORE INFO
Local intranet

MAUT Collaborative Filtering Simulator (CollaFiS)

Algorithm	Neighborhood Method	Normalization	Parameter	MAE	Items with possible prediction	Items with impossible prediction	Time
Euclidian per w	Number of Neighbors	Simple Mean	0	0	0	250	1
Euclidian per w	Number of Neighbors	Simple Mean	1	1.17457824324	222	28	0
Euclidian per w	Number of Neighbors	Simple Mean	2	1.04211747748	222	28	0
Euclidian per w	Number of Neighbors	Simple Mean	3	1.00263971471	222	28	0
Euclidian per w	Number of Neighbors	Simple Mean	4	1.00079590465	222	28	1
Euclidian per w	Number of Neighbors	Simple Mean	5	0.988505931682	222	28	0
Euclidian per w	Number of Neighbors	Simple Mean	6	0.987321451201	222	28	0
Euclidian per w	Number of Neighbors	Simple Mean	7	0.983986391141	222	28	0
Euclidian per w	Number of Neighbors	Simple Mean	8	0.983986391141	222	28	0
Euclidian per w	Number of Neighbors	Simple Mean	9	0.983986391141	222	28	1
Euclidian per w	Number of Neighbors	Simple Mean	10	0.983986391141	222	28	0
Euclidian per w	Number of Neighbors	Simple Mean	11	0.983986391141	222	28	0
Euclidian per w	Number of Neighbors	Simple Mean	12	0.983986391141	222	28	0
Euclidian	Number of						

Fig. 8: A CollaFiS screenshot after the execution of the algorithm, with the produced simulation results.

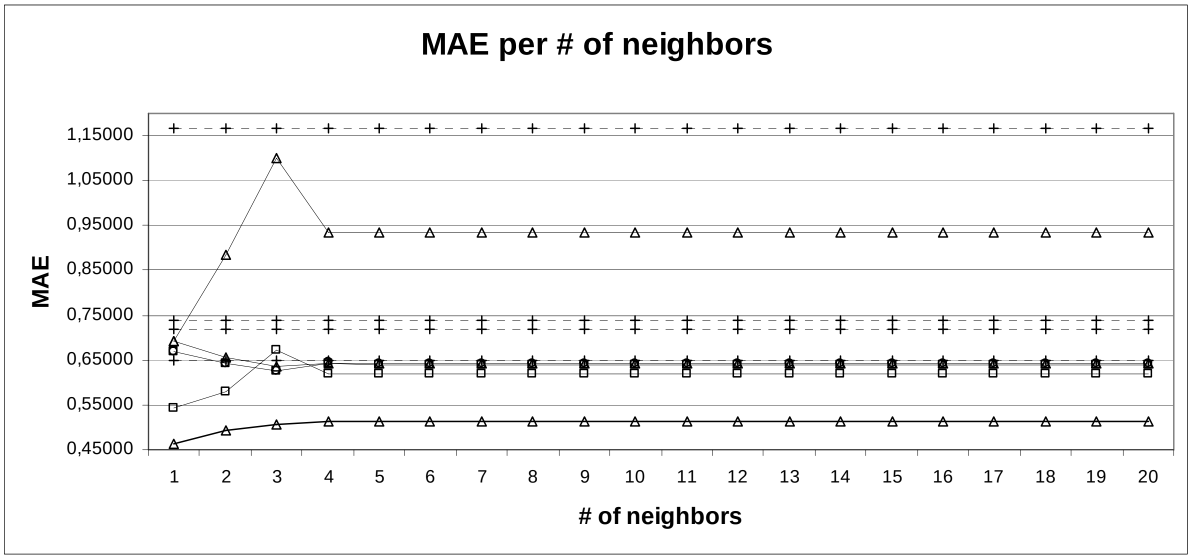


Fig. 9: MAE experimentation results. The values for the simple algorithms appear using a cross "+"and a dotted line, for the first variation using a triangle, and for the second variation using a square.

In addition, for multi-attribute collaborative filtering systems, there are not yet any publicly available multi-criteria evaluation data sets from real-life applications, nor any testing methods or tools before their actual online deployment. For this purpose, this paper presented the specification, implementation, and usage of CollaFiS, a Web-based testing tool that may greatly facilitate the simulation and experimental evaluation of multi-attribute collaborative filtering algorithms.

CollaFiS allows a recommender system researcher or developer to create a synthetic data set, by defining data set properties such as the number of criteria, items, users, evaluations, etc. When the simulator is used to create a synthetic set, these variables take randomly values from the whole spectrum of permitted values. On the other hand, this is not the case for real data sets. For example, in particular problem domains the evaluators may be using only the higher values of the criteria evaluation scales. In order for the synthesized sets to better mimic the expected real ones, it would be desirable to allow the users of the tool to also define this type of properties for the produced data sets. It is our intention to include such functionalities in a future version of CollaFiS. It may also be possible to allow the users define groups of evaluators with similar "evaluation profiles" (ways that they are expected to evaluate). These evaluation profiles can be defined based on the results of pilot studies that collect real item evaluations from focus groups.

References

[1] G. Adomavicius, and A. Tuzhilin, "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. Knowl. Data Engin.*, vol. 17(6), 2005, pp. 734-749.

[2] D. Ariely, J. G. Jr. Lynch, and M. Aparicio, "Learning by Collaborative and Individual-based recommendation Agents," *J. Consum. Psych.*, vol. 14(1/2), 2004, pp. 81-94.

[3] G. Baourakis, N. F. Matsatsinis, and Y. Siskos, "Agricultural product development using multidimensional and multicriteria analyses: The case of wine," *Eur. J. Oper. Res.*, vol. 94 (2), 1996, pp. 321-334.

[4] W. Boggs, and M. Boggs, *Mastering UML with Rational Rose*. Alameda CA: SYBEX Inc., 2002.

[5] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modelling Language User Guide*. Boston, MA: Addison-Wesley, 1998.

[6] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *14th Conf. Uncertainty in Artificial Intelligence*, Madison WI, USA, July 1998.

[7] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Model. User Adapt. Inter.*, vol. 12, 2002, pp. 331-370.

[8] S. H. Choi, and Y. H. Cho, "An utility range-based similar product recommendation algorithm for collaborative companies," *Exp. Syst. Appl.*, vol. 27, 2004, pp. 549-557.

[9] M. Deshpande, and G. Karypis, "Item-based Top-N Recommendation Algorithms," *ACM Trans. Inf. Syst.*, vol. 22(1), 2004, pp. 143-177.

[10] S. Guan, C. S. Ngoo, and F. Zhu, "Handy broker: an intelligent product-brokering agent for m-commerce applications with user preference tracking," *Electr. Comm. Res. App.*, vol. 1, 2002, pp. 314-330.

[11] U. Hanani, B. Shapira, and P. Shoval, "Information Filtering: Overview of Issues, Research and Systems," *User Model. User Adapt. Inter,*. vol. 11, 2001, pp. 203-259.

[12] J. Herlocker, J. A. Konstan, and J. Riedl, "An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms," *Inf. Retr.*, vol. 5, 2002, pp. 287-310.

[13] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Trans. Inf. Syst.*, vol. 22(1), 2004, pp. 5-53.

[14] N. Karacapilidis, and L. Hatzieleutheriou, "A hybrid framework for similarity-based recommendations," *Int. J. Bus. Intell. Data Min.*, vol. 1(1), 2005.

[15] R. L. Keeney, *Value-focused Thinking: A Path to Creative Decisionmaking*. Cambridge MA: Harvard University Press, 1992.

[16] J. A. Konstan, "Introduction To Recommender Systems: Algorithms and Evaluation," *ACM Trans. Inf. Syst.*, vol. 22(1), 2004, pp. 1-4.

[17] W.-P. Lee, "Towards agent-based decision making in the electronic marketplace: interactive recommendation and automated negotiation," *Exp. Syst. Appl.*, vol. 27, 2004, pp. 665-679.

[18] W.-P. Lee, C.-H. Liu, and C.-C. Lu, “Intelligent agent-based systems for personalized recommendations in Internet commerce,” *Exp. Syst. Appl.*, vol. 22, 2002, pp. 275-284.

[19] D.-R. Liu, and Y.-Y. Shih, “Integrating AHP and data mining for product recommendation based on customer lifetime value,” *Inf. Manag*., vol. 42, 2005, pp. 387-400.

[20] N. Manouselis, and D. Sampson, “A Multi-criteria Model to Support Automatic Recommendation of e-Learning Quality Approaches,” *16th World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-MEDIA 2004*, Lugano, Switzerland, June 2004.

[21] N. Manouselis, and C. Costopoulou, “Experimental Analysis of Design Choices in a Multi-Criteria Recommender System”, *Int. J. Pattern Rec. and Art. Int.*, Sp. Issue on "Personalization Techniques for Recommender Systems and Intelligent User Interfaces", in press.

[22] L. Maritza, C. N. Gonzalez-Caro, J. J. Perez-Alcazar, J. C. Garcia-Diaz, and J. Delgado, “A Comparison of Several Predictive Algorithms for Collaborative Filtering on Multi-Valued Ratings,” *2004 ACM Symposium on Applied Computing (SAC’04)*, Nicosia, Cyprus, March 2004.

[23] B. N. Miller, J. A. Konstan, and J. Riedl, “PocketLens: Toward a Personal Recommender System,” *ACM Trans. Inf. Syst*., vol. 22(3), 2004, pp. 437-476.

[24] M. Montaner, B. Lopez, and J. L. de la Rosa, “Evaluation of Recommender Systems through Simulated Users,” *ICEIS 2004*.

[25] H. Nguyen, and P. Haddawy, “DIVA: Applying Decision Theory to Collaborative Filtering,” *AAAI Worksh. Recomm. Syst*., Madison, WI, July 1998.

[26] H. Nguyen, and P. Haddawy, “The Decision-Theoretic Video Advisor,” *15th Conf. Uncert. Artif. Intell.*, Stockholm, Sweden, 1999, pp. 494-501.

[27] S. Noh, “Implementing Purchasing Assistant Using Personal Profile,” *IADIS Int. Conf. App. Comp.*, Lisbon, Portugal, March 2004.

[28] M. Papagelis and D. Plexousakis, “Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents,” *Eng. Apps Art. Int*., vol. 18, 2005, pp. 781-789.

[29] P. Perny, and J.-D. Zucker, “Preference-based Search and Machine Learning for Collaborative Filtering: the ‘Film-Conseil’ Movie Recommender System,” *Inform. Interact. Intell*., vol. 1(1), 2001, pp. 9-48.

[30] M. Plantie, J. Montmain, and G. Dray, “Movies Recommenders Systems: Automation of the Information and Evaluation Phases in a Multi-criteria Decision-Making Process,” *DEXA*, K.V. Andersen, J. Debenham, and R. Wagner, Eds. Berlin Heidelberg: Springer-Verlag, 2005, pp. 633-644.

[31] B. Price, and P. R. Messinger, “Optimal Recommendation Sets: Covering Uncertainty over User Preferences,” *Informs Ann. Meet*., Denver 2004, AAAI Press, 2005.

[32] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An Open Architecture for Collaborative Filtering,” *ACM CSCW*, 1994, pp.175-186.

[33] K. Saleh, “Document electronic commerce systems and software using the unified modeling language,” *Inf. Soft. Techn.*, vol. 44, 2002, pp. 303-311.

[34] C. Schmitt, D. Dengler, and M. Bauer, “The MAUT-Machine: An Adaptive Recommender System,” *ABIS Worksh. Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen*, Hannover, Germany, October 2002.

[35] U. Shardanand, and P. Maes, “Social Information Filtering: Algorithms for Automatic ‘Word of Mouth’,” *Proc. Conf. Hum. Fact. Comput. Syst.*, Denver CO, USA, 1995.

[36] K. Srikumar, and B. Bhasker, “Personalized Product Selection in Internet Business,” *J. Electr. Comm. Res*., vol. 5(4), 2004, pp. 216-227.

[37] M. Stolze, and M. Stroebel, “Dealing with Learning in eCommerce Product Navigation and Decision Support: The Teaching Salesman Problem,” *2nd World Congr. Mass Custom. Person.*, Munich, Germany, 2003.

[38] G. Tewari, J. Youll, and P. Maes, “Personalized location-based brokering using an agent-based intermediary architecture,” *Dec. Supp. Syst*., vol. 34, 2002, pp. 127-137.

N. G. Manouselis was born in Toronto, Canada. He holds a diploma in Electronics & Computer Engineering from the Technical University of Crete, Greece (2000). He also holds a Master in Operational Research from the Dept. of Production & Management Engineering (2002), as well as, a Master in Electronics & Computer Engineering from the Dept. of Electronics & Computers Engineering (2003), both from the Technical University of Crete, Greece.

Before joining the Informatics Laboratory of the Agricultural University of Athens, Greece (2005), he has been a member of the Informatics & Telematics Institute of the Centre for Research & Technology Hellas, Greece (2001-2004). During 2000-2001 he has been a member of the Decision Support Systems Laboratory of the Technical University of Crete, Greece. He has participated in several European and National Research Projects. His research interests involve the design, development and evaluation of electronic services, as well as their applications for the agricultural sector.

Mr. Manouselis is a member of the IEEE, the Greek Computer Society, and the Technical Chamber of Greece. His diploma thesis has been awarded an Honorary Distinction from the Technical Chamber of Greece (2nd Place 2000). In 2005, he has received a PhD Presentation Distinction at the International Congress on Information Technologies in Agriculture, Food and Environment (ITAFE05).

C. I. Costopoulou was born in Athens, Greece. She holds a BSc in Mathematics from the National and Kapodistrian University of Athens, Greece (1986), an MSc in Software Engineering from Cranfield Institute of Technology, UK (1989), and a PhD in Electrical and Computer Engineering from the National Technical University of Athens, Greece (1992). From 1992 until 1995 she was member at the Telecommunications Laboratory of the National Technical University of Athens, Greece. Since 1995 is an Assistant Professor at the Informatics Laboratory of the Agricultural University of Athens, Greece.

Her research interests include communication networks, formal methods, Web services, intelligent agents and e-commerce technology. Her latest work focused on portal technology, semantic Web, e-government and e-business services. She has published more than 60 articles in scientific journals, books and conferences focusing on the aforementioned issues.

Dr. Costopoulou is currently the scientific coordinator or member of the working group of four European and Greek government funded projects in the areas of portal technology and the development of e-content and e-government services. Over the past years she has participated in over fifteen European and Greek government funded research projects.