Integrated Personal Recommender Systems

Ronald Chung
University of Auckland
Department of Information Systems & Operations Management
Private Bag 92019,
Auckland, New Zealand
+64 9 3737999
r.chung@auckland.ac.nz

David Sundaram
University of Auckland
Department of Information Systems & Operations Management
Private Bag 92019,
Auckland, New Zealand
+64 9 3737999
d.sundaram@auckland.ac.nz

Ananth Srinivasan
University of Auckland
Department of Information Systems & Operations Management
Private Bag 92019,
Auckland, New Zealand
+64 9 3737999
a.srinivasan@auckland.ac.nz

ABSTRACT
Recommender Systems belong to a class of systems intended to assist individuals make evaluations about entities in meaningful ways. In this paper we discuss the issues in the design of integrated recommender systems and suggest a framework that takes the perspective of an individual functioning in multiple domains. This is particularly applicable today with the rapidly increasing diffusion of personalized, networked mobile devices. We present some preliminary design ideas in the form of a functional prototype.

Categories and Subject Descriptors
J.4 [**Social and Behavioral Sciences**]: Applications to assist personal decision making; H.4 [**Information Systems and Applications**]: Recommender Systems

General Terms: Design, Economics, Management

Keywords
Personal Decision Making; Integrated Recommender Systems

1. INTRODUCTION
Recommender Systems (RS) represent a class of systems designed to help individuals deal with information overload, incomplete information, and their capacity to make evaluative judgments. Such systems help individuals by providing recommendations through the use of various personalization techniques (Adomavicius and Tuzhilin, 2005). These systems have long been employed as an integral component of e-commerce websites where they are treated as business tools for generating competitive advantage (Schafer, Konstan and Riedl, 2000). More recently, the focus of the RS research community has shifted from an organizationally driven perspective to one of enhanced effectiveness in personal decision making; i.e., how can RS applications support personal decision making rather than being an addendum to an e-commerce website with the intent of converting a browser to a buyer (e.g. Niinivaara (2004)).

Miller (2005) argues that to truly exploit the potential of RS, generated recommendations should serve users - the individuals, not commercial sites. Likewise, Schafer et al. (2000) have argued that the success of a RS should really be measured by how effectively the system helps its users make decisions, rather than measuring how much profit it generates for a commercial website. Having a RS that is fully capable of supporting personal decision making seems to be the goal of the current wave of RS research, yet many researchers have hinted that we are still far away from reaching it. Yu (2004) has observed that most current-generation recommender systems are separate systems, each focusing only on a single area of interest – one application domain and typically provides recommendations for only one item class (or type). Likewise, Niinivaara (2004) has commented that most RS are monolithic and *ad-hoc* systems designed for restricted purposes. González, Lopez and de la Rosa (2005) have reasoned that using RS as a decision-support tool is inconvenient as users would have to access different systems for different decisions and investment in personalizing one RS is not transferable to another system. Adomavicius and Tuzhilin (2005) explain that in most RS, recommendation methods are hard-wired into the systems by vendors. Therefore, they support only a predefined and fixed set of recommendations and users cannot customize them according to their needs in real time.

In an attempt to address some of these issues, in this research, our objective is to propose a framework for recommender systems applications with a focus on supporting personal decision making. In order to construct such a framework, we draw from the literature on decision making, decision support, as well as recommender systems. We then show a proof-of-concept of the framework through a prototype that implements many aspects of the framework. In essence, we discuss a design approach to the implementation of recommender systems that we think will consider an individual's decision making preference structure that potentially spans multiple domains and is inclusive of the inputs of a relevant community of personal decision makers. With the advent of the widespread adoption of personalized and networked mobile devices, this approach to recommender systems may prove to be particularly useful. In the remainder of the paper we discuss the relevant literature that helps us distil concepts to assist in the development of a design framework. We then present the essentials of a prototype based on the framework. We conclude with some comments on the potential future direction for this line of research.

2. BACKGROUND LITERATURE

In this section we review salient highlights from the literature on decision making processes. We then look at the development of recommender systems, consider the design implications and briefly assess the state of the art of current RS applications.

2.1 Decision Making Process: Some Proposals

Building on Simon's (1960) classic model of decision making, Mintzberg, Raisinghani and Theoret (1976) have enriched it to further capture and emphasize some of the complexities of personal decision making. Such a view is endorsed by a number of subsequent researchers such as Marakas, (2003), Rowe and Boulgarides (1994), etc. The highlights of the model are described briefly:

- Identification - In this stage, an individual recognizes a problem or an opportunity and diagnoses the situation, which may trigger a decision.
- Development - If a decision is required, at the development stage the individual may search for readily available existing solutions, design a new solution set or modify existing solutions to suit the current decision.
- Selection - When the individual reaches the selection stage, he/she iteratively evaluates the identified choices by judgment, analysis and bargaining, before the final decision is authorized. Various analytical techniques used in this phase are comparing, ranking, weighing and sensitivity analysis (Marakas, 2003).

Marakas (2003) has defined a "good" decision as a decision that results in the attainment of the objective or objectives that gave rise to the need for a decision within the boundaries and constraints imposed by the problem's context. These objectives, boundaries or constraints relating to a personal decision can broadly be interpreted as the various dimensions of the decision.

In summary, the decision making literature suggests that we need to cater for a) a rich sense of individuality b) multiple decision making processes and styles c) multiple theoretical models to understand decision making and d) the fact that personal decisions can be complex, interrelated, multi-faceted and bounded by various perspectives.

2.2 Recommender Systems

Early versions of RS were termed "Collaborative Filtering" (Goldberg, Nicholas, Oki and Terry, 1992) since collaborative filtering was the algorithm used for generating recommendations on items. However, Resnick and Varian (1997) later argued that such a system not only filters but also makes suggestions on new items. They have suggested that it should be called "Recommender System" and re-defined it as a system that takes people's recommendations as inputs, aggregates them and directs the results to appropriate recipients. This definition has later been modified, broadened or tightened further as the RS concept has evolved. For example, Burke (2000) broadly defined a RS as a computer system that provides advice to users about items they might wish to purchase or examine. Similarly, Huang, Chung, Ong and Chen (2002) stated that RS advise users on relevant products and information by predicting the users' interests in product, based on various types of information such as past purchases and product features. More recently, Kostan (2004) has once again tightened the definition of a RS as "a system that uses the opinions of members of a community to help individuals in that community identify the information or products most likely to be interesting to them or relevant to their needs". For the purposes of this paper, our working definition of recommender systems is: *a class of systems that attempt to assist people in making personal decisions in various domains, by offering recommendations about items to users employing various methods*.

2.3 Design Considerations for Recommender Systems

Reduce information overload - RS are expected to help reduce information overload through personalization (van Alstyne and Brynjolfsson, 1997). The objective is to display those items that are relevant to a user in various ways such as employing collaborative-filtering or attributes-based filtering. While attribute-filtering refines a list of alternative items, collaborative-filtering is capable of exploring new relevant items.

Leverage the strength of online communities - The majority of RS make use of online communities for supporting personal decision making (Schafer et al., 2000). Some typical mechanisms are social-filtering and statistical summarization of community opinions. This approach exploits the fact that we tend to rely on recommendations from others by means such as word of mouth, recommendation letters or item reviews.

Design Components and Elements of RS - Although all RS share the same objective (to recommend items to users), the design and implementation of a RS are nevertheless complex and variable (Resnick and Varian, 1997). We present a conceptual framework (Figure 1) that illustrates the main components of a typical RS and the relationships between its components. The framework also depicts the general process which a typical RS follows for generating recommendations. Two main elements pertinent to the design of an RS are also included in the framework: the delivery method and the degree of personalization.

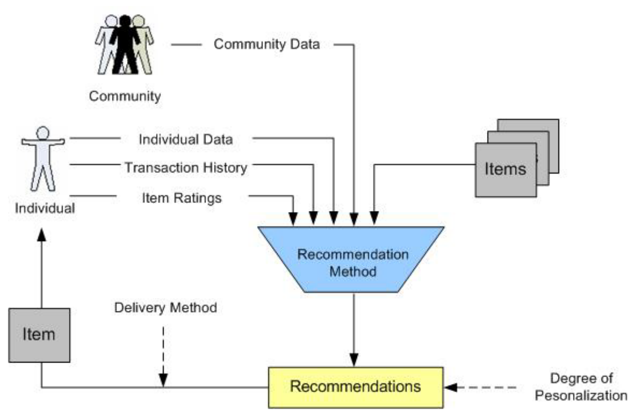

Figure 1. The Common Components and Elements of RS (Adapted from: Schafer et al. (2000))

Input Data - Schafer et al. (2000) have identified the following most compelling types of input data:

Inidvidual data – This includes any information pertinent to an individual such as name, gender, qualification level, nationality, personal interests, goals, hobbies, values, beliefs, preferences, health conditions and financial status.

Item ratings – This is often used in collaborative-filtering RS whereby the individuals are either explicitly or implicitly requested to rate some items to indicate their preferences (Rashid, Albert, Cosley, Lam, McNee, Konstan and Riedl, 2002).

Transaction history – Occasionally, RS may implicitly derive the preferences of their users by analyzing their transaction histories such as purchase histories and internet browsing histories (Schafer et al., 2000).

Community data – RS can also make recommendations based on calculated summary data. Highly rated items and best selling products are typical examples of this type of data.

RS generate recommendations to users by using various forms of recommendation methods. On the one hand, a RS can be as simple as a search mechanism (that is user-oriented); on the other hand, it can be entirely automated (system-oriented) by the use of data-mining algorithms (Shardanand et al, 1995, Good et al, 1999, Wolf et al, 1999).

Each method may require different types of input data. Amongst the various types of recommendation methods, Schafer et al. (2000) have listed the following most prominent ones:

Raw-retrieval – The RS typically provides a search interface through which a user can search for items that meet certain requirements (e.g. by search term or matching item attributes).

Statistical summarization – Aggregates or summarizes community opinions statistically and provides recommendations to users based on those opinions (Schafer et al., 2000).

Attribute-based – This method generates recommendations based on syntactic properties of items and user interests in those properties (O'Donovan and Smyth, 2005).

User-to-user correlation – This recommendation method, often implemented with collaborative-filtering, is perhaps the most popular method used in RS (Carenini, Smith and Poole, 2003). Using this method, recommendations are prepared for a user based on the correlation between a particular user and other users of the system (Shardanand and Maes, 1995; Srikumar and Bhasker, 2004; Goldberg, Nicholas, Oki and Terry, 1992).

Item-to-item correlation – This method, sometimes called content-based, generates recommendations based on constructed relationships between items of the same class (item type) (Burke, 2000).

Recommendation Outputs - After input data are gathered and processed with a recommendation method, a RS then generates a set of recommendations accordingly. The items that are recommended to users by the same RS are usually of the same item type. For example, one may expect that a list of restaurants will be generated as recommendations by a restaurant recommender. Rarely, does a RS produce a list of items that are of a mix of item types (cross-item) as recommendations.

Depending on the context and the reason for producing recommendations, recommended outputs generated by a RS may be items that the user has seen or experienced in the past, or items that the user has no knowledge of. For instance, RS that implement a collaborative-filtering algorithm often generate recommendations for an individual on items that he/she has not seen or experienced before (Resnick and Varian, 1997).

2.4 Assessing RS Applications

To illustrate the current status of RS applications we mapped typical systems on a grid as shown in Figure 2. The X axis corresponds to the extent to which a recommender system supports multiple domains. The Y axis represents the average diversity of recommended item types across all domains covered by a RS.

Overall, the majority of reviewed recommender systems are located on the top-left, bottom-left or bottom-right of the quadrant. The top right corner of this matrix offers the greatest opportunity for pursuing ideas related to the development of new generation RS applications. It involves the powerful and potentially useful combination of multiple item types encompassing multiple domains.

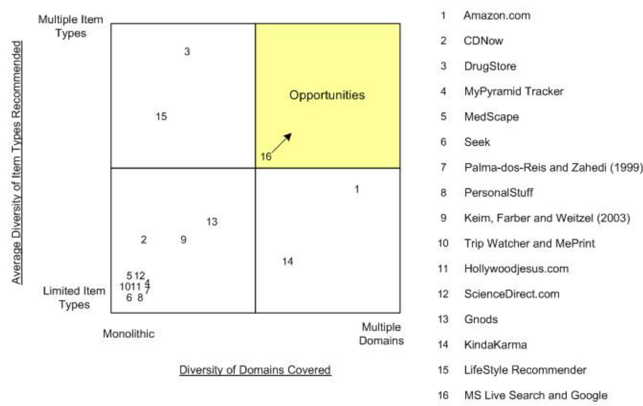

Figure 2. Assessment of Recommender Systems

3. Framework for Integrated Personal Recommender System (IPRS)

In order to meet the above high-level requirements, an enhanced-RS should satisfy a number of system requirements as detailed in this section. Figure 3 illustrates the requirements as well as how they can help address personal decision making issues that we have identified.

The identified issues of personal decision making are represented in the pink (heavily shaded) boxes in Figure 3. While the system requirements are grouped into yellow (lightly shaded) boxes, the arrows indicate which group of requirements addresses which issues or sub-issues of personal decision making. For example, Integrated Profile Management is required to address the rich sense of individuality in personal decision making. To address multi-dimensional decision making, Modeling and Model Management, Scenario Modeling and Scenario Management, and Integrated Recommendations are collectively required.

Integrated Profile Management – In order to capture a rich sense of individuality of every user, the enhanced-RS should support flexible profile management. Users should be given the ability to create, modify and delete profiles freely. In such a profile, the individualities represented in various forms such as characteristics, preferences, values, beliefs and background are captured. More importantly, a user should also be able to apply his/her individual preference structure to a decision. For example, if a user is engaged in a decision wherein he/she would like to select a restaurant, the system should allow the user to apply his/her

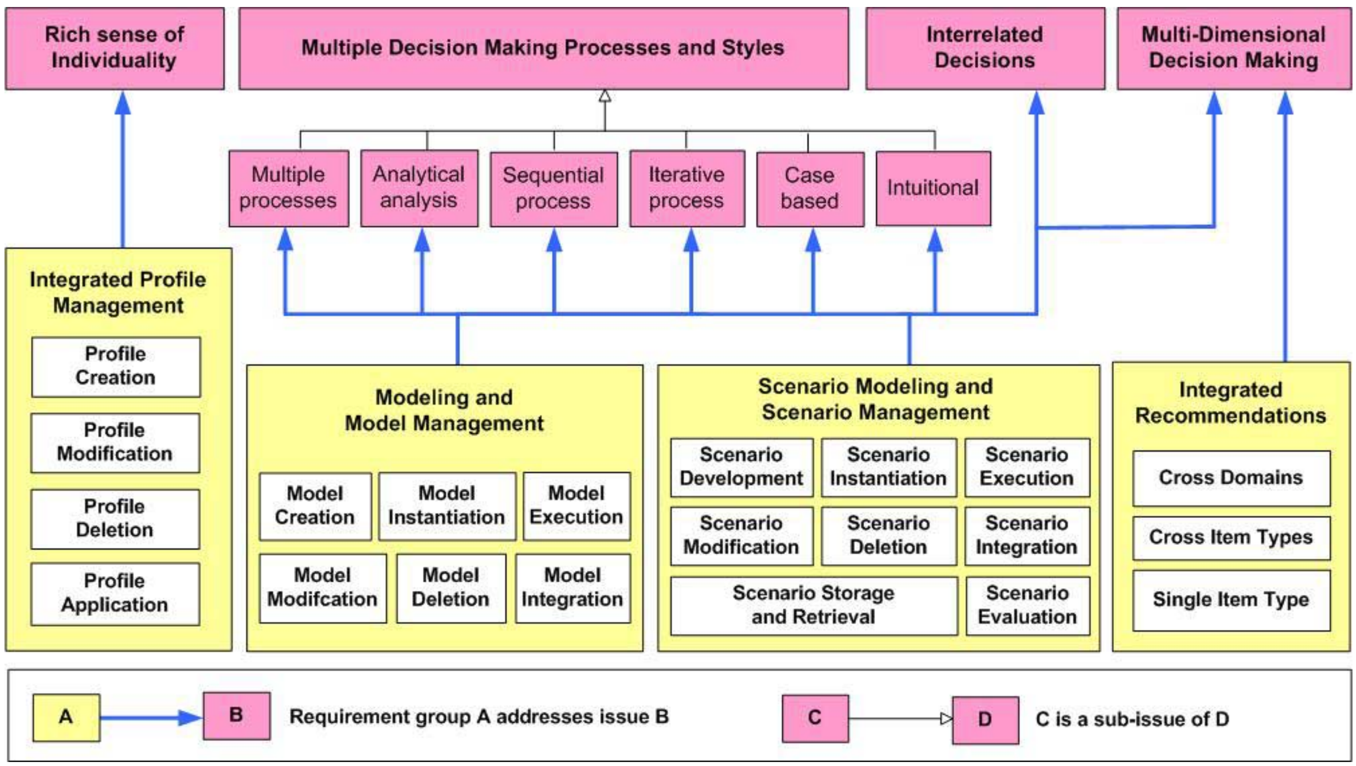

Figure 3. The System Requirements

personal preference, taste and requirements of the decision for selecting a suitable restaurant. The system must facilitate profile management in a flexible manner as these preferences can be cross-domain and diverse.

Modeling and Model Management – To allow users to perform decision analyses, the enhanced-RS should provide a variety of models, which will abstract the various activities performed by users throughout the personal decision making process. For example, a model should be provided to users for assigning weights to multiple criteria of a decision, and another model provided for extracting community opinions. The system should allow users to create, instantiate, execute, modify, delete and most importantly, integrate models together to represent complex situations. Users should not only be able to perform such activities at run-time, but should also be able to persistently store models at any stage during the modeling life cycle (Dolk and Kottemann, 1993). This will facilitate re-use of models and hence support case-based decision making.

Scenario Modeling and Scenario Management – To facilitate scenario analysis (Klein, 1995), the system should enable users to develop, instantiate, execute, modify, integrate, evaluate and delete scenarios at run-time (Ahmed, 2002). Users should also be able to retrieve existing scenarios for re-use or adaptation and be able to save current scenarios on a persistent basis.

Integrated Recommendations – In situations where users do not know exactly what item types they should obtain recommendations for, or what the multiple dimensions of a decision are, the enhanced-RS should be able to provide useful support by generating integrated recommendations. Such an integrated recommendation should contain a set of items that can be of the same item type or of different item types, and should collectively address a single underlying decision.

Integrated recommendations can be generated at three levels. Recommendations maybe simple and consist of items that belong to a single item type. Recommendations can also be of cross-item type even though the item types may belong to the same domain. At the highest level, recommendations can be cross-domain when the recommended item types belong to completely different domains. For example, when a user is looking for a book on stress management, the enhanced-RS may also provide recommendations to the user on various other items that also can help with handling stress, such as counseling services, psychologists, parks, activities and medicines. The enhanced-RS should be capable of generating integrated recommendations at all three integration levels. *The working definition of a RS provided in Section 2.2 is qualified in the case of IPRS to focus on personal decision making.*

4. Implementation of Integrated Recommendations

Integrated recommendations can be generated at three different levels (single item type, cross-item type and cross-domain) in at least three different ways.

General Filtering - One simple way to generate integrated recommendations is to instantiate a *General Filtering* Model with multiple item types that may belong to multiple domains and execute the model. In this case, all the attributes of all the items of different item types will collectively be searched by a search term.

Community Filtering - Another possible way to generate integrated recommendations is to utilize ratings being shared within particular communities. A personal decision maker may obtain integrated recommendations by creating a *Community Filtering* model, instantiate it by multiple item types that may belong to multiple domains and execute it by the community-average solver to retrieve ratings from a specific community. Items that have been rated highly by other people within a particular community will be collectively recommended to a personal decision maker. The items may be of different item types and belong to multiple domains.

Market Basket Analysis - Apart from *General Filtering* and *Community Filtering*, there are other ways that can potentially be used to generate integrated recommendations. One is to rely on data-mining algorithms such as Market Basket Analysis. Market Basket Analysis, also called Association Rules, is a data-mining algorithm that can extrapolate hidden relationships between items to form a relationship model (Berry and Linoff, 1997). Various types of data can potentially be fed into a Market Basket Analysis for building the relationship model. This includes item ratings and search histories in IPRS. Table 1 Summarizes how the three levels of integration can be implemented with the above three methods.

Table 1 Implementing Integrated Recommendations

Levels of Integration \ Methods	General Filtering	Community Filtering	Market Basket Analysis
Single Item Type	Instantiate a general filtering model with one item type	Instantiate a community filtering model with one item type	Build a cross-item relationship model for each item type
Cross-Item Type	Instantiate a general filtering model by selecting multiple item types in one domain	Instantiate a community filtering model by selecting multiple item types in one domain	Build a cross-item relationship model for each domain
Cross-Domain	Instantiate a general filtering model by selecting multiple domains	Instantiate a community filtering model by selecting multiple domains	Build one integrated cross-item relationship model using all domains and item types

To provide a proof-of-concept and to highlight an implementation of the key capabilities discussed thus far, a prototypical implementation of IPRS is developed. The architecture of the IPRS and sample sessions with the IPRS prototype are introduced briefly in the following sections.

5. Generic IPRS Architecture

In the Generic Integrated Personal Recommender System Architecture (Figure 4), we present each layer of IPRS, key stakeholders, and reveal the components of the kernel as well as the relationships between them. Salient aspects of the architecture are discussed below.

Personal decision maker – is a person, a registered user of IPRS who utilizes the system for supporting personal decision making. The key activity that a personal decision maker would perform is to model and analyze a decision that he/she encounters and to obtain some recommendations regarding the decision. In order to obtain personalized recommendations, a personal decision maker may create and maintain his/her personal profile with the *Profile Manager* (explained later in this section). To capture his/her preferences, and hence obtain recommendations based on user-to-user correlation, a personal decision maker rates certain items that he/she has experience with, in the *Ratings Manager* (explained later in this section).

The Kernel - The kernel is the core of the IPRS and it is the application itself that delivers the key functionalities to the users. We describe the sub-components of it as well as the interactions between them in this section as follows:

Profile Manager - To address the issue of *rich sense of individuality*, IPRS contains a component called *Profile Manager* that is used by a personal decision maker to maintain his/her personal profile in various domains. To capture the individualities, a personal decision maker can enter personal data such as preferences for certain item type, values, beliefs, hobbies, interests, experiences and goals. A personal decision maker is able to freely create any item in a profile so that he/she can refer back to the profile items when he/she is modeling a decision with the *Integration Manager*. For example, a personal decision maker may create a profile called "Investment". In such a profile, he/she can create a new item called "Investment Limit" that contains the value of $650,000.

Recommendation Engine - is essentially the heart of the kernel as it generates various types of recommendations to a personal decision maker. After a personal decision maker has modeled a decision with the *Integration Manager*, the *Recommendation Engine* would fetch the selected model(s) from the *Integration Manager* and execute the model(s) with the solver(s) retrieved from the solver base. After the recommendations are generated, control is passed back to the *Integration Manager*, which displays the executed model output to the personal decision maker.

Integration Manager - is a significant component in IPRS that integrates various personal decision components when a personal decision maker models a decision. It is responsible for retrieving the available model structures from the model base so that the personal decision maker can create new instances of model structures for modeling a decision. To instantiate a model created, the *Integration Manager* also integrates a model with items and possibly some predefined profile items. To generate recommendations, the *Integration Manager* may also need to retrieve ratings given by the personal decision maker as well as those ratings given by others who belong to the same community. The *Integration Manager* collects ratings of other personal decision makers via the *Communication & Community Manager*. The *Integration Manager* retrieves the model structures and the items for integration from persistent storage via the *Model Manager* and the *Items Manager*. To support a case-based approach, the *Integration Manager* may pass a created scenario to the *Scenario Manager* to be saved to persistent storage. Also, when a personal decision maker is modeling a decision, via the *Scenario Manager*, he/she may retrieve saved scenarios to the *Integration Manager*. The retrieved scenarios may then be integrated with each other or with other models to model a complex decision.

Communication and community manager - is mainly responsible for establishing a connection between different personal decision makers. When a personal decision maker wants to generate a recommendation based on user-to-user correlation, on behalf of the *Integration Manager*, the *Communication & Community Manager* retrieves and consolidates the ratings of the neighbors who belong to the same community as the personal decision maker. Using the *Communication & Community Manager*, a personal decision maker may define community

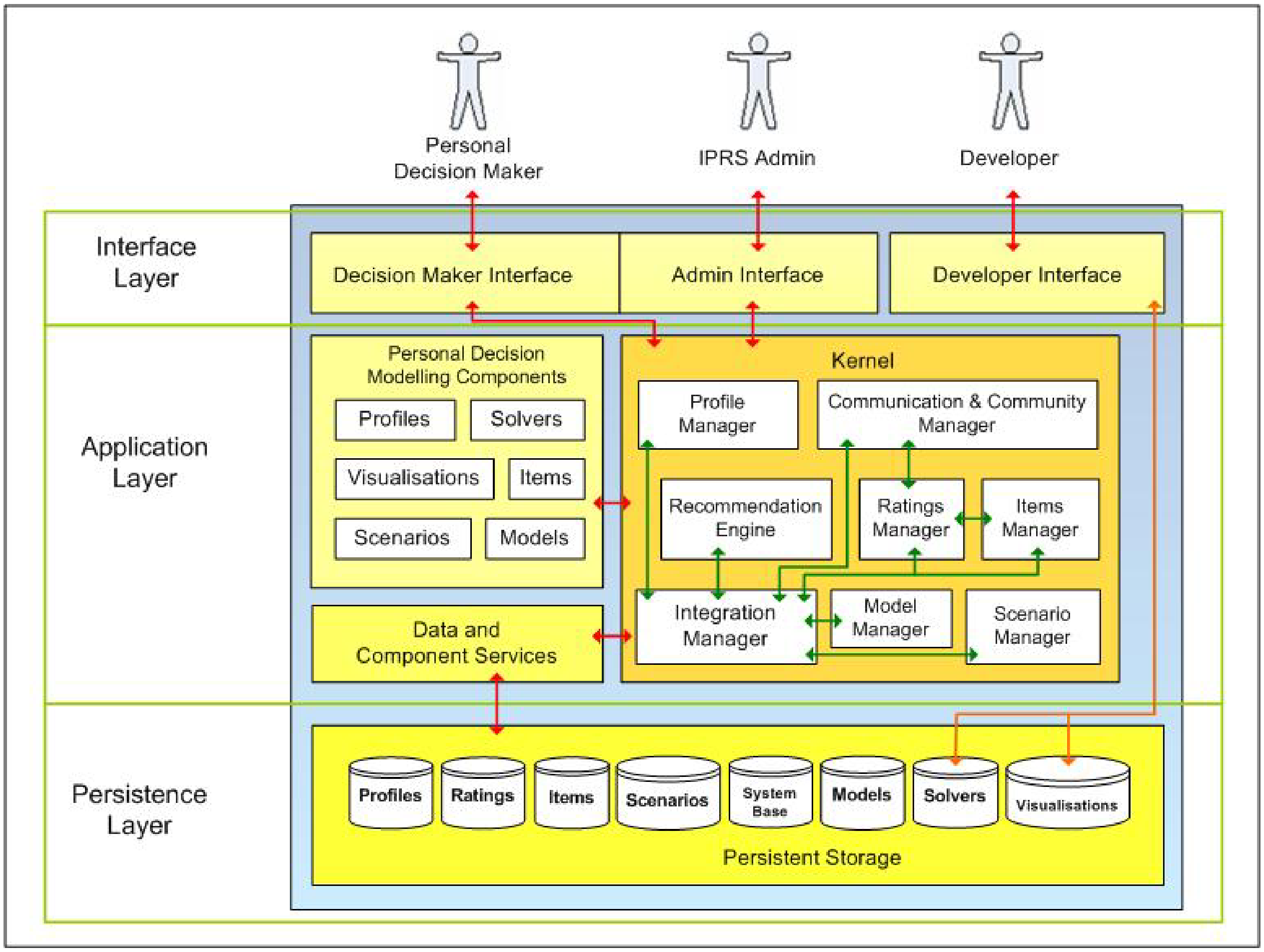

Figure 4. The IPRS Architecture

membership, create a new community or delete a community that he/she has created.

Ratings Manager - a personal decision maker may selectively rate items to indicate his/her preferences after the items are fetched from the items base by the items manager. When a personal decision maker demands recommendations by collaborative-filtering, the ratings manager retrieves the ratings of each neighbor for the specified item types or domains on behalf of the communication and community manager.

Items Manager - is responsible for retrieving and maintaining items. When a personal decision maker models a decision, he/she may instantiate a model with items, which would be retrieved by the *Items Manager*. When a personal decision maker provides ratings to IPRS, the *Items Manager* retrieves the target items for the *Ratings Manager*. Through the *Items Manager*, a personal decision maker can also control the domain(s) to which each item type would belong according to his/her preferences, beliefs and values. The IPRS admin may use the *Items Manager* for approving any new item type or item that has been requested by personal decision makers. The IPRS admin also uses the *Items Manager* simply for administering the items base

Model Manager - when a personal decision maker creates a model in the *Integration Manager* at run-time, he/she is actually creating an instance of a model based on a model structure or a model template. In such a case, the *Model Manager* is responsible for retrieving the model structure that a personal decision maker is using. Alternatively, a personal decision maker may also create a new model structure, for example forming a complex model structure by linking multiple model structures together (e.g. pipelining, splicing or consolidation) and save the new model structure to the model base. When the personal decision maker models a decision, a case-based approach may be adopted by searching, screening, selecting and adapting a customized model structure from the model base.

Scenario Manager - In IPRS, a scenario is an executed model with particular parameter settings. The *Scenario Manager* is responsible for saving any scenario created in the *Integration Manager* to the scenario base. Further, when a personal decision maker is modeling a decision, he/she may retrieve the saved scenarios from the scenario base via the *Scenario Manager*.

Data and Component Services - The *Data and Component Services* are an abstraction of services at the application layer that can be used by the components in the kernel to establish connections with the data and components at the persistence level. This includes saving and retrieving the data and components. To emphasize the difference between profile items (such as age,

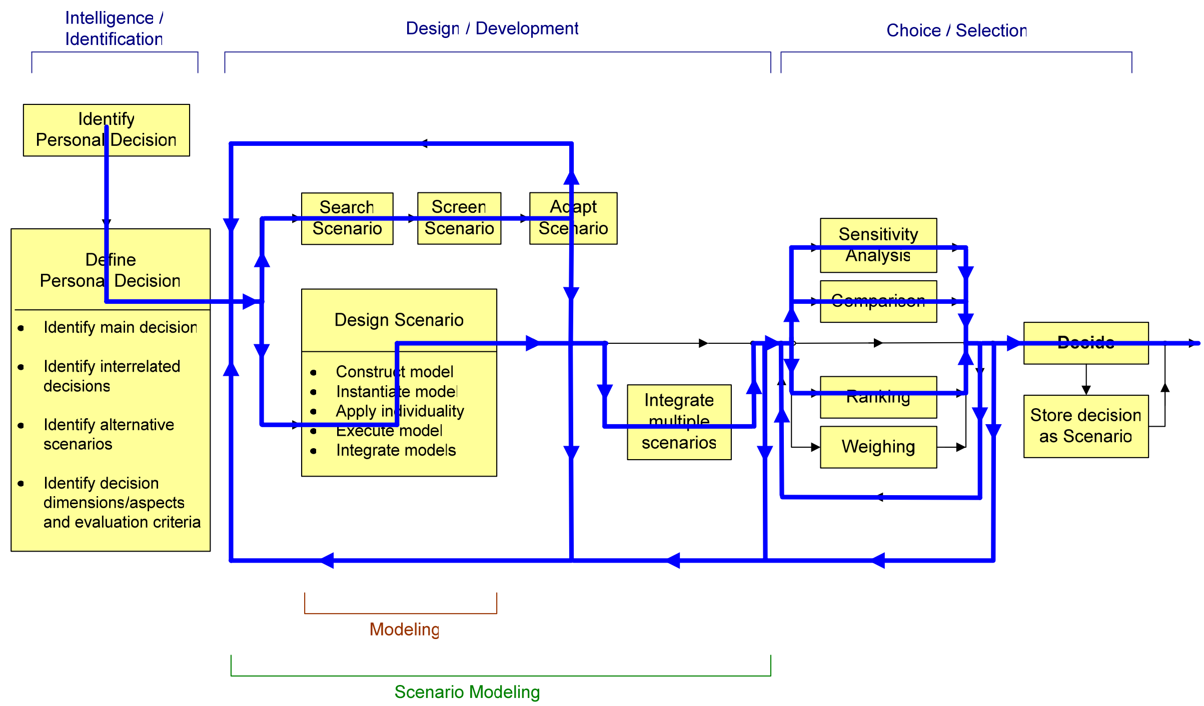

Figure 5. Implementing the personal decision making process

preferred colors, nationality, and investment amount) and item ratings, the *Ratings Base* has been logically separated from the *Profile Base*. The *Ratings Base* stores the ratings on various items from different domains rated by all personal decision makers in IPRS. The *System Base* stores any data in IPRS that are not captured by the other components at the persistence layer such as network address of computers used by the personal decision makers and the usage patterns of them. A link has also been added between the *Developer Interface* and the *Solver Base*, and between the *Developer Interface* and the *Visualization Base* to indicate that the solvers and the scenarios are developed and maintained by the developer.

6. A sample session illustrating interrelated decisions

Consider an individual who would like to visit a particular city for one day. With only a limited amount of time, he/she would like to obtain recommendations about which restaurant to dine in, what activities to consider or what tourist attractions to visit and where to find suitable accommodation (the "Identify Personal Decision" step). This is the identification stage.

This demonstration illustrates how interrelated decisions (which belong to different domains and involve different item types) can be modeled with model integration and scenario modeling. Figure 5 shows the steps that the traveler may go through with the IPDM Process in the sample session in order to reach the final decision.

The development phase of the personal decision making model is now illustrated. In the travel plan, he/she needs to make three sub-decisions (the "Define Personal Decision" step): 1) Which restaurant to select 2) What activities to consider or what tourist attractions to visit and 3) Where should he/she stay for the night. These three sub-decisions are interrelated, such as by location because the traveler would like to optimize the plan by selecting the restaurants, activities or tourist attractions and accommodations that are close to each other. One way to tackle this problem is to construct multiple scenarios ("Identify alternative scenarios" in the "Define Personal Decision" step) and to explore the relationships between the three sub-decisions as follows. First select the most suitable restaurant. Then based on the suitable restaurant, activities or tourist attractions that the traveling would enjoy the most. Finally, based on where the selected restaurant and the activities or tourist attractions are, choose accommodation that is appropriately located.

To model the above scenario, the traveler will have to iteratively go through the "Design Scenario" step and the "Integrate multiple scenarios" step of the IPDM Process. At any stage, the traveler can re-use any existing scenarios that he/she may have saved previously ("Search Scenario", "Screen Scenario" and "Adapt Scenario").

The traveler can first create and execute an *Attributes Filtering* model as in Figure 6: [1] and [2] to obtain some suitable restaurants (the "Design Scenario" step).

Then, to find the tourist attractions that are close to these two restaurants (as displayed in Figure 6. [3], the traveler can create, instantiate and execute a second *Attributes Filtering* model as in Figure 7. [1] and [2]. To search for tourist attractions that are close to the restaurants that have been recommended in the first model, the traveler can set the attribute "Suburb" of the tourist attractions to be close to any of the suburbs in model 1 (as indicated in the rectangular area in Figure 7. [2]). Thus, model 1 and model 2 are integrated structurally and the execution output of model 2 reveals that there are three tourist attractions (Figure 7. [3]).

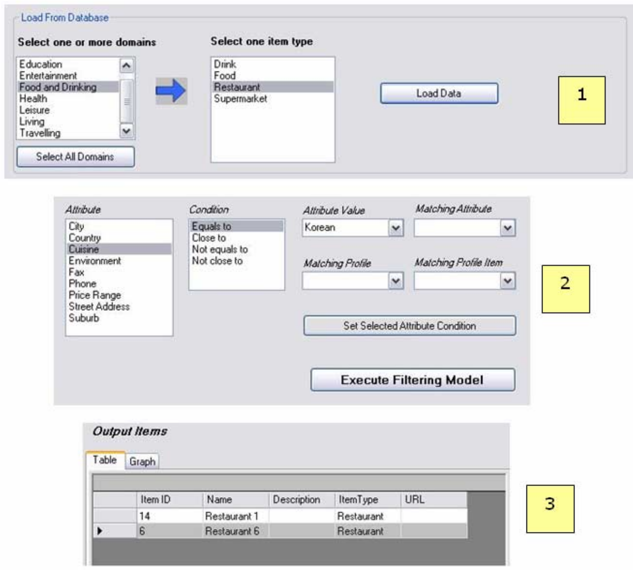

Figure 6. Finding a Restaurant

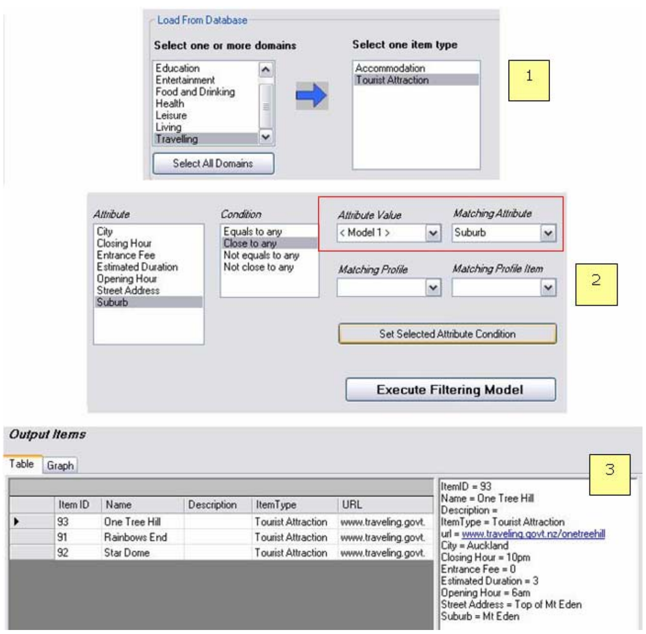

Figure 7. Finding Suitable Tourist Attractions Based on the Selected Restaurants

Figure 8. illustrates what the modeling panel of the *Integration Manager* looks like after integrating the two models. After the traveler has worked out which activities to consider, he/she can then repeat the procedure to search for some accommodations that are close to the restaurants. The traveler needs to create another *Attributes Filtering* model (model 3), instantiate it with the accommodations available (Figure 9. [1]) and integrate model 3 with model 1 by setting the "Suburb" attribute of the accommodations to equals to the suburbs in model 1 (Figure 9. [2]). Executing such a model shows that there are two suitable accommodations (Figure 9. [3]). The selection phase of the personal decision making model can now be executed.

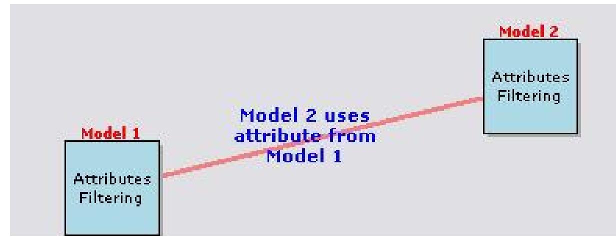

Figure 8. Modeling Plan Option 1 – Integrating Model 1 and 2

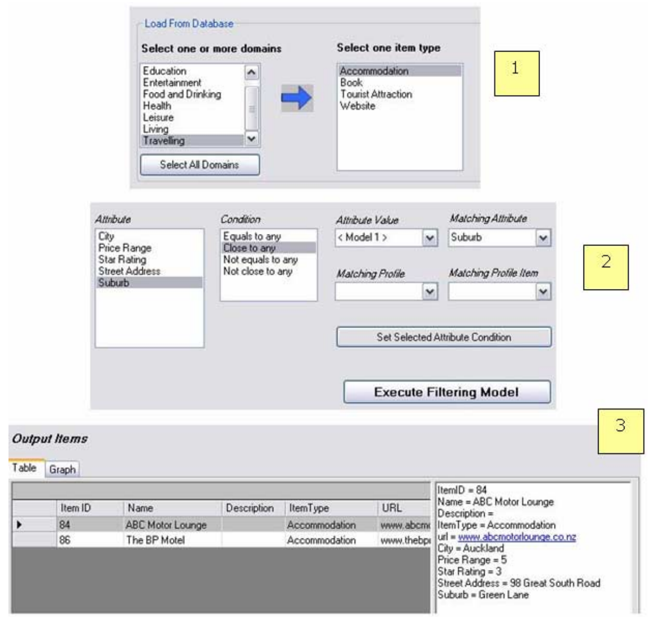

Figure 9. Finding Suitable Accommodations Based on the Selected Restaurants

Figure 10. illustrates what the modeling panel of the *Integration Manager* looks like after integrating Model 1 with Model 2 and Model 3.

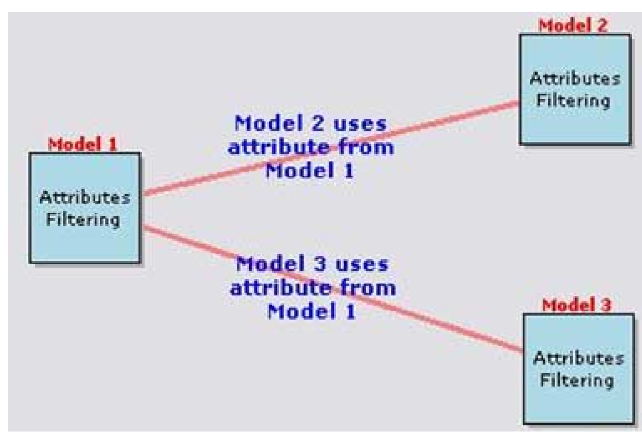

Figure 10. Integrating Model 1 with Model 2 and 3

The model created by the individual can now be saved to the scenario base.

7. CONCLUSION

Recommender Systems serve a useful function in their role of being a highly functional component in current e-commerce applications. Typically the approach taken by these systems is focused on the application domain of the "vendor" as opposed to that of the individual. In this paper we take the perspective of the individual decision maker and consider typical aspects of personal decision making processes. This necessitates the consideration of multiple domains and inter-related decisions within the context of what we call an integrated personal recommender system. We believe that the utility of such systems will be greatly enhanced by taking such a perspective. Further the widespread diffusion of networked mobile devices make this perspective particularly useful for future development. Future work needs to consider how the provider and the consumer of outputs of recommender systems can have their requirements concurrently addressed within the umbrella of a single integrated system.

8. REFERENCES

[1] Adomavicius, B. and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on knowledge and Data Engineering*., 17, 6 (2005), 734-748.

[2] Ahmed, M. D. *Design and Implementation of Scenario Driven Flexible Decision Support Systems Generator*. Masters Thesis, Department of Management Science and Information Systems, University of Auckland, Auckland, New Zealand, 2002.

[3] Berry, M. J. A. and Linoff, G. *Data Mining Techniques*, John Wiley & Sons, 1997.

[4] Burke, R. Knowledge-based Recommender Systems. *IEEE Transactions on. Knowledge and Data Engineering*., 17, 6 (2000), 734-749.

[5] Carenini, G., Smith, J. and Poole, D. Towards more Conversational and Collaborative Recommender Systems. *Proceedings of the 8th international conference on intelligent user interfaces*, 2003.

[6] Dolk, D. R. and Kottemann, J. E. Model integration and a theory of models. *Decision Support System.,* 9, 1 (1993), 51-63.

[7] Goldberg, D., Nicholas, D., Oki, B. M. And Terry, D. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*., 35, 12 (1992), 61-70.

[8] González, G., López, B. and Lluís de la Rosa, J. A Multi-agent Smart User Model for Cross-domain Recommender systems. *IUI'05 Workshop: Beyond Personalization 2005*, San Diego, USA, 2005.

[9] Good, N., Schafer, J. B., Kostan, J. A., Borchers, A., Sarwar, B., Herlocker, J. and Riedl, J. Combining Collaborative Filtering with Personal Agents for Better Recommendations. *Proceedings of AAAI-99*, AAAI Press., (1999), 439-446.

[10] Huang, Z., Chung, W., Ong, T., Chen, H. A Graph-based Recommender System for Digital Library. *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*., (2002).

[11] Klein, G. A. Automated Aids for the Proficient Decision Maker. *Proceedings of the 1980 IEEE Systems, Man and Cybernetics Conference.*, (1980), 301-304.

[12] Konstan, J. A. Introduction To Recommender Systems: Algorithms and Evaluation. *ACM Transactions on Information Systems.,* 22, 1 (2004), 1-4.

[13] Marakas, G. M. *Decision Support Systems In the 21st Century (2nd Ed)*, Upper Saddle River, NJ: Prentice Hall, 2003.

[14] Miller, B. Toward a personalized recommender system. *Ph.D. thesis*, University of Minnesota, Minneapolis, MA, USA, 2003.

[15] Mintzberg, H., D. Raisinghani and A. Theoret. The Structure of "Unstructured" Decision Processes. *Administrative Science Quarterly.*, 21 (1976), 246-274.

[16] Niinivaara, O. Agent-Based Recommender Systems, University of Helsinki, Department of Computer Science., (2004), 1-48.

[17] O'Donovan, J., Smyth, B. Trust in recommender systems, *Proceedings of the 10th international conference on intelligent user interfaces.*, (2005).

[18] Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., Riedl, J. Getting to Know You: Learning New User Preferences in Recommender Systems." *Proceedings of the 2002 international conference on intelligent user interfaces., (2002).*

[19] Resnick, P., Varian, H. R. Recommender systems. *Communications of the ACM.,* 40, 3 (1997), 56-58.

[20] Rowe, A. J. and Boulgarides, J. D. *Managerial Decision Making*, Upper Saddle River, NJ: Prentice Hall, 1994.

[21] Schafer, J. B., Kostan, J. A., Riedl, J. Electronic Commerce Recommender Applications. *Journal of Data Mining and Knowledge Discovery.,* 5, ½ (2000), 115-152.

[22] Shardanand, U. and Maes, P. Social information filtering algorithms for automating 'word of mouth. *CHI-95: Conference proceedings on Human factors in computing systems.*, New York: ACM Press, 1995, 210-217.

[23] Simon, H. A. *The New Science of Management Decision*. New York: Harper & Row, 1960.

[24] Srikumar, K., Bhasker, B. Personalized Product Selection in Internet Business. *Journal of Electronic Commerce Research*., 5 (2004), 216-227.

[25] Van Alstyne, M. and Brynjolfsson, E. Electronic Communities: Global Village or Cyberbalkans?. MIT Sloan School., (1997), 1-32.

[26] Wolf, J., Aggarwal, C., Wu, K-L., and Yu, P. Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*, San Diego, CA, 1999.

[27] Yu, C.-C. A Web-Based Consumer-Oriented Intelligent Decision Support System for Personalized E Services. Proceedings of the Sixth International Conference on Electronic Commerce., (2004)