PROVIDING RECOMMENDATIONS IN AN AGENT-BASED TRANSPORTATION TRANSACTIONS MANAGEMENT PLATFORM

Alexis Lazanas, Nikos Karacapilidis and Yiannis Pirovolakis
Industrial Management and Information Systems Lab, MEAD
University of Patras, 26504 Rio Patras, Greece

Keywords: Recommendation techniques, recommender systems, transportation, web-based systems, intelligent agents.

Abstract: Diverse recommendation techniques have been already proposed and encapsulated into several e-business systems aiming to perform a more accurate evaluation of the existing alternatives and accordingly augment the assistance provided to the users involved. Extending previous work, this paper focuses on the development of a recommendation module for transportation transactions purposes and its integration in a web-based platform. The module is built according to a hybrid recommendation technique, which combines the advantages of collaborative filtering and knowledge-based recommendations. The proposed technique and supporting module enable customers to consider in detail alternative transportation transactions satisfying their requests, as well as to evaluate such transactions after their completion.

1 INTRODUCTION

Transportation management involves diverse decision making issues, which are basically related to the choice of route and carrier selection. Such issues raise due to the differentiation between the dispatcher's preferences (e.g. cost limitation, loading, and delivery dates) and the carrier's service resources (e.g. transportation media, itinerary, and capacity). The matching of the above preferences and services cannot be easily handled manually, as in most cases a plethora of alternative options exist, while time and money limitations are ubiquitous. Therefore, the field of transportation management requires quick and cost-effective solutions to the customers' demands for distribution and shipping operations.

This paper extends our previous work on the exploitation of software agent technology in transportation management (Karacapilidis *et al.*, 2006; Lazanas *et al.*, 2005). More specifically, we have addressed analysis, design and implementation issues raised during the development of an innovative agent-mediated electronic marketplace, which is able to efficiently handle transportation transactions of various types. Agents of the proposed system represent and act for any user involved in a transportation scenario, such as customers who look for efficient ways to ship their products and transport companies that may - fully or partially - carry out such requests, while they cooperate and get the related information in real-time mode. Our overall approach is based on flexible models that achieve efficient communication among all parties involved, coordinate the overall process, construct possible alternative solutions and perform the required decision-making. In addition, the supporting web-based system is able to handle the complexity that is inherent in such environments, which is mainly due to the frequent need of finding a "modular transportation solution". To further explain this concept, consider the case where a customer wants to convey some goods from place A to place B, while there is no transport company acting directly between these two places. Supposing that two available carriers X and Y have some scheduled itineraries from A to C and from C to B, respectively, it is obvious that a possible solution to the above customer's request is to involve both X and Y and fragment the intended overall itinerary to the related sub-routes. It is also noted that these carriers may be associated with diverse transportation means, such as trains, trucks, ships and airplanes. Our system is able to manage all the necessary freighting and fleet scheduling processes in wide-area transportation networks. Its agents

Lazanas A., Karacapilidis N. and Pirovolakis Y. (2006).
PROVIDING RECOMMENDATIONS IN AN AGENT-BASED TRANSPORTATION TRANSACTIONS MANAGEMENT PLATFORM.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - SAIC*, pages 87-92
DOI: 10.5220/0002461900870092

cooperate upon well-specified business models, which may efficiently carry out diverse processes.

This paper focuses on the features and functionalities of a new module integrated in the above system, namely the *recommendation module*, which aims at enhancing the quality of the associated decisions. Recommender systems have been described as systems that produce individualized recommendations as output or have the effect of guiding the user in a personalized way, in environments where the amount of on-line information vastly outstrips any individual's capability to survey it (Burke, 2002). Alternative techniques have been also proposed in the literature in order to handle the above issues. Having thoroughly considered their pros and cons, our approach follows a hybrid recommendation technique.

The remainder of this paper is structured as follows. Section 2 discusses approaches and related work from the area of recommender systems. Section 3 describes the basic aspects of our approach, namely transportation plan selection, alternative solution evaluation and recommendation methodology. Section 4 deals with the evaluation of a transaction and the exploitation of agent technologies. Finally, Section 5 concludes the paper.

2 RECOMMENDER SYSTEMS

Recommender systems apply data analysis techniques to assist users finding the items they need by producing a predicted *likeness score* or a list of *top-N recommended items*. Two of the most widely adopted recommendation techniques are Collaborative Filtering (CF) and Knowledge Based Recommendation (KBR), each one possessing its own strengths and weaknesses. Collaborative Filtering (CF) (Resnick *et al.*, 1994) is the most commonly used recommendation technique to date. The basic idea of CF-based algorithms is to provide item recommendations or predictions, based on the opinion of other like-minded users. In a typical CF scenario, there is a list of m users $U = \{u_1,u_2,\ldots,u_m\}$ and a list of n items $I = \{i_1,i_2,\ldots,i_n\}$. Each user u_i is associated with a list of items I_{Ui}, for which the user has expressed his/her opinion. Opinions can be explicitly given by the user as a rating score, generally within a certain numerical scale, or can be implicitly derived from transaction records, by analyzing timing logs, mining web hyperlinks and so on. For a particular user U_a, the task of a collaborative filtering algorithm is to find an *item likeness* that can be of two forms:

- *Prediction:* this is a numerical value, $P_{a,j}$, expressing the predicted likeness of item i_j (i_j does not belong in I_{Ua}) for the user. The predicted value is within the same scale (e.g. 1 to 5) as the opinion values provided by U_a.
- *Recommendation:* this is a list of N items I_r (I_r is a subset of I) that the user will like most (the recommended list must contain items not already selected by the user). This outcome of CF algorithms is also known as Top-N recommendation. (Sarwar *et al.*, 2000).

On the other hand, Knowledge-Based Recommendation attempts to suggest objects based on inferences about a user's needs and preferences. In some sense, all recommendation techniques could be described as doing some kind of inference. Knowledge-based approaches are distinguished in that they utilize functional knowledge: that is, they have knowledge about how a particular item meets a particular user need, and can therefore reason about the relationship between a need and a possible recommendation. The user profile can be any knowledge structure that supports this inference. In the simplest case, as in Google (www.google.com), it may simply be the query that the user has formulated. The Entrée system and several other recent systems (for an overview, see (Schmitt & Bergmann, 1999)) employ techniques from case-based reasoning for knowledge-based recommendations.

The knowledge used by a knowledge-based recommender system can also take many forms. Google uses information about the links between web pages to infer popularity and authoritative value (Brin & Page, 1998). Entrée uses knowledge of cuisines to infer similarity between restaurants. Utility-based approaches calculate a utility value for objects to be recommended; in principle, such calculations could be based on functional knowledge. However, existing systems do not use such inference mechanisms, thus requiring users to do their own mapping between their needs and the features of products, either in the form of preference functions for each feature, as in the case of Tête-à-Tête, or answers to a detailed questionnaire, as in the case of PersonaLogic (Burke, 2002). Knowledge-based recommender systems are prone to the drawback of all knowledge-based systems: the need for knowledge acquisition. More specifically, there are three types of knowledge that are involved in such a system:

- *Catalog knowledge:* Knowledge about the objects being recommended and their features. For example, the Entrée recommender should know that "Thai" cuisine is a kind of "Asian" cuisine.
- *Functional knowledge:* The system must be able to match the user's needs with the object that might satisfy those needs. For example, Entrée knows that a need for a romantic dinner spot could be met by a restaurant that is "quiet with an ocean view".
- *User knowledge:* To provide good recommendations, the system must have some knowledge about the user. This might take the form of general demographic information or specific information about the need for which a recommendation is sought.

Of these knowledge types, the last one is the most challenging, as it is, in the worst case, an instance of the general user-modeling problem. Despite this drawback, knowledge-based recommendation has some beneficial characteristics. It is appropriate for casual exploration, because it demands less from the user compared to the utility-based recommendation. Moreover, it does not involve a start-up period during which its suggestions are of low quality. On the other hand, a knowledge-based recommender cannot "discover" user niches, the way collaborative systems can. However, it can make recommendations as wide-ranging as its knowledge base allows.

3 OUR APPROACH

3.1 Transportation Plans and Evaluation of Alternative Solutions

The recommendation procedure adopted in our approach is highly associated with the selection (by the user) of the appropriate "transportation plan" (see Figure 1). A transportation plan typically defines the user preferences for the upcoming transactions. The five alternative plans offered are:

- *Express*
- *Economic*
- *Safe*
- *Dependable*
- *User Defined*

It can be easily observed that each of the first four plans declares a specific tension in the recommendation strategy to be followed by the system, in that it either minimizes the overall duration or cost (first two plans) or it retains a high level of safety or dependability (third and fourth plans) of the suggested itineraries. The last choice offers the possibility for a customized plan definition. A user defined plan may combine parameters from all the above four plans. The selection of one of these plans will influence the recommendations processes of our approach for the particular user.

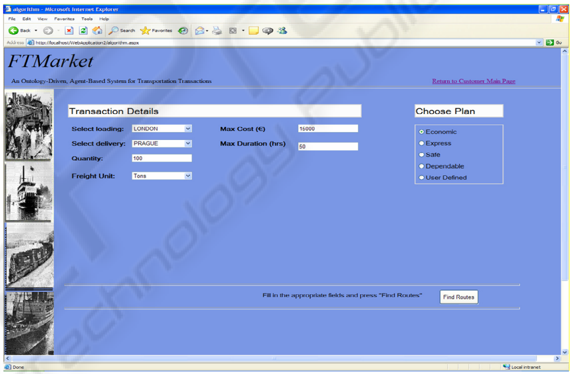

Figure 1: User request for a transaction.

Having constructed all possible transportation solutions, our approach proceeds to the evaluation phase. In this phase, our approach's Decision_Maker agent performs two distinct tasks. First, a refinement of the set of alternative solutions constructed takes place, by excluding solutions that do not comply with the customer's requirements (the algorithm developed for the construction of alternative routes takes into account only the duration and cost parameters as upper bounds of the proposed solutions). In this phase, a set of predefined rules are employed to exclude the alternative solutions that do not correspond to the specific freight transportation's requirements and customer preferences. For instance, if the customer had chosen the "*Safe*" transportation plan, the following set of rules will be deployed:

```
for each constructed_solution
{
if (safety_level < AVERAGE) OR
((safety_level = AVERAGE) AND
```

```
(dependability_level < LOW)) then
discard constructed_solution;
else if (safety_level >= HIGH) OR
((safety_level = AVERAGE) AND
(dependability_level >= LOW)) then
solution ( constructed_solution;
}
```

Table 1 presents the constraints to be met for each transportation plan (for the User Defined plan, this process takes into account the constraints set by the user). In all cases, solutions that do not satisfy these constraints are discarded.

Table 1: Selection Criteria for each plan.

Plan[1]	Cost	Duration	Safety	Dependa-bility
1	Any	Min	Any	Any
2	Min	Any	Any	Any
3	Any	Any	>Average	>=Low
4	Any	Any	>= Low	> Average
5	UD[2]	UD	UD	UD

[1] *Plans 1 to 5 correspond to: Express, Economic, Safe, Dependable and Hybrid.*
[2] *User Defined Criterion*

3.2 A Methodology for the Selection of Alternative Route Paths

In order to construct optimal and sub-optimal solutions, our approach uses an elaborated version of Dijkstra's shortest path algorithm (Crauser *et al.*, 1998). The majority of shortest path algorithms in the literature uses a bidirectional, single-weighted graph to represent a connected set of vertices (V_i) through a number of arcs A_{ij} (from V_i to V_j). Our algorithm takes into consideration each A_{ij} and its correspondent weight (w_{ij}) in order to produce a route path from a starting point (S) to an ending point (E) that minimizes the total weight (w_{SE}). The complexity in our approach consists in the presence of a pair of factors that affect each arc's weight, namely the cost and the duration. Due to the fact that there exist two weights for each arc, we confronted the problem of unifying these weights into a single one, in order to proceed with the ranking of the solutions.

More specifically, each arc's A_{ij} weight (w_{ij}) consists of a cost weight ($w_{cost-ij}$) and a duration weight ($w_{duration-ij}$). Obviously, $w_{ij} = w_{cost-ij} + w_{duration-ij}$.

Having defined the total weight for each arc we encountered the problem of adding these two parameters that are measured in different units (Euros and hours, respectively). This problem was confronted by applying a normalization technique that divided both the cost and duration terms of an A_{ij} route with the correspondent maximum cost and duration. It is:

$$w_{duration-ij} = duration_{ij}/max(duration)$$
$$w_{cost-ij} = cost_{ij}/max(cost)$$

Another issue that came up after the weight normalization procedure concerned the solutions ranking. To address this problem, our approach provides the user with different solutions by using a pair of weight coefficients (costCoef and durationCoef) and calculating solutions corresponding to alternative combinations of the weights of the cost and duration criteria (see Figure 2), according to the formula:

$$w_{ij} = (costCoef * w_{cost-ij}) + (durationCoef * w_{duration-ij})$$

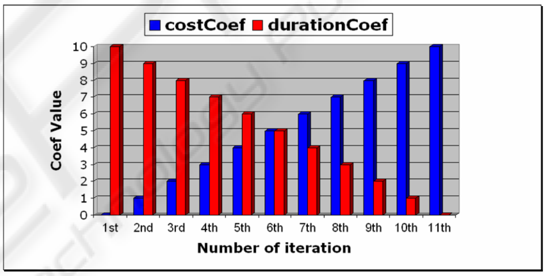

Figure 2: Coefficients' variation.

This process is described in pseudo-code below:

```
{
costCoef ← 0.0;
durationCoef ← 1.0;
step ( 0.0;
for each step calculate
  {
       weight[i][j] ( costCoef*Wcost +
       durationCoef * Wduration;
       costCoef ( step;
       durationCoef = 1-step;
  }
  perform shortest path algorithm;
  step ( step + 0.1;
}
```

The outcome of this process is then provided to the user. An instance of the related system interface is shown in Figure 3.

3.3 Recommendation Methodology

The recommendation procedure begins immediately after the ranking of the alternatives. It is a complex process which is carried out in three basic stages: the evaluation of the carriers and the transactions data, the exploitation of transaction data through a data mining process, and the recommendation methodology selection or synthesis. At the beginning of the process, the system stores all the appropriate data that are submitted by the user and are related with scheduled or completed transactions. These data are of significant importance and will be further exploited by the data mining process. Moreover, in this stage the user evaluates the carrier(s) involved in a transaction through an interface since the evaluation process concerns assigning a score to each carrier (this is discussed in more detail in Section 4.1).

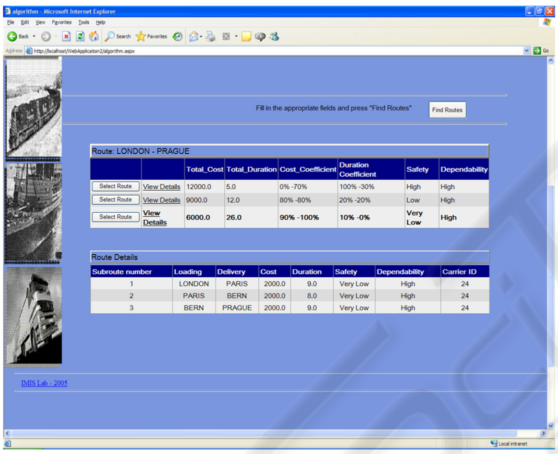

Figure 3: Solution ranking produced by the system.

The second stage of recommendation concerns the data mining process. At this stage, transactions data are gathered through knowledge construction processes (Cho *et al.*, 2002). Although important, the description of these processes goes out of the scope of this paper. We only mention here that knowledge construction in our case refers to the formulation of rules that could be used when a knowledge based recommendation strategy is performed.

The last phase of the recommendation refers to the selection or synthesis of the appropriate recommendation technique. This objective will be reached through the definition of well structured rules that will be applied for each transaction. The recommender agent of our system takes the initiative to look for the most appropriate recommendation technique. For example, for a particular itinerary from point i to j, taking into consideration that the customer has selected a certain plan, a rule for the specific itinerary could recommend a carrier that is different than the one suggested by the CF technique, based on the carriers' evaluation process described earlier in this section.

4 IMPLEMENTATION ISSUES

The recommender module has been integrated into our platform as an add-on tool and has been thoroughly tested. After the completion of a transportation transaction, the customer is able to evaluate it by providing his/her opinion about the carrier(s) who were involved. Since transactions handled by our platform often concern modular solutions, multiple carriers may be involved. Thus, the rating procedure concerns each individual carrier.

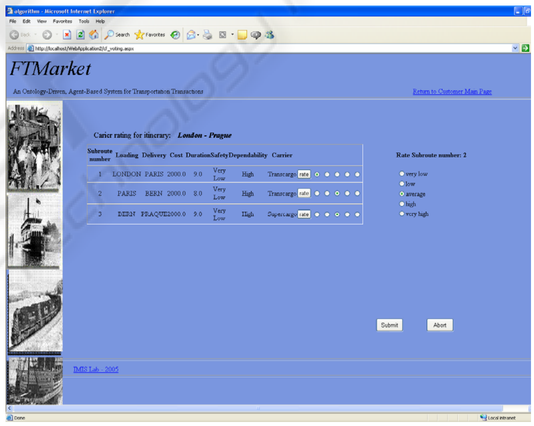

Figure 4: The carriers' evaluation interface.

As shown in Figure 4, a customer expresses his/her position for a completed transaction by selecting a value for each feature, following a typical Likert scale (very low, low, average, high, very high). Each value is stored in the appropriate table in the systems SQL Server database. This rating is very helpful during the main phase of the recommendation process presented in Figure 5, which takes place after the ranking of the alternative solutions described earlier. As shown in the figure, for the itinerary "London–Prague" the customer has selected one of the proposed routes provided by the system (see Figure 3). After selecting the specific route, the customer may request a recommendation for the involved carriers. The system provides different types of recommendations features such as:

an average score ranking, top-10 carriers, “red card” carrier (carriers with under average scoring), and a “suggestion” functionality. Each suggestion made by the system is accompanied by a prediction status label P_i = {critical, acceptable, indifferent}, which refers to the significance level of the system’s suggestion. In the example shown in Figure 5, the suggestion is concerned “critical” because the system retains some “negative” knowledge for the carrier under consideration.

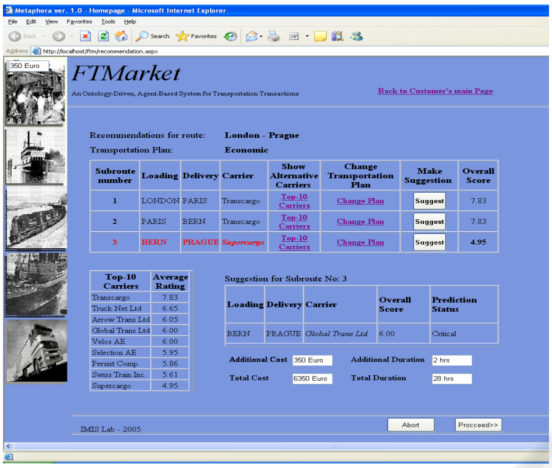

Figure 5: The recommendation module of the system.

For the implementation of the proposed recommendation technique, a recommender module has been developed using agent and data warehousing technologies. More specifically, a new agent, namely the Recommender Agent, has been implemented in order to coordinate the overall process of recommendation and has been assigned with two main tasks:

- *The recommendation technique selection,* in that the agent is capable of switching among alternative recommendation algorithms, depending on the characteristics of each transaction. The agent carries out data mining tasks and generates the appropriate rules or models.
- *The characterization of each carrier* who is involved in a transportation transaction and the *recommendation of alternative carriers* capable of fulfilling a particular transportation request.

Through these formally modeled tasks, the recommender agent provides continuous assistance to customers, while it remains active and capable to adapt its “behavior” into a rapidly changing environment.

5 CONCLUSIONS

This paper presents our approach on the integration of hybrid recommendation techniques into an agent–based transportations transaction management platform. We proposed a hybrid approach that combines different recommendation techniques, in order to provide the user with more accurate suggestions. The overall process is coordinated by a recommender agent, who is responsible of performing multiple tasks. The presence of the agent guarantees that the user will be provided with continuous recommendation, dynamic update and recommendation techniques synthesis.

REFERENCES

Brin, S. and Page, L.: 1998, The anatomy of a large-scale hyper textual {Web} search engine. Computer Networks and ISDN Systems, 30(1-7), 107-117.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, 12, pp. 331-370.

Cho, Y., Kim J. and Kim, S. (2002). A personalized recommender system based on web usage mining and decision tree induction. Expert Systems with Applications, 23, pp. 329-342.

Crauser A., Mehlhorn K., Meyer U. and Sanders P., 1998. A Parallelization of Dijkstra's Shortest Path Algorithm. In Proceedings of 23rd International Symposium, MFCS'98, Brno, Czech Republic.

Karacapilidis N., Lazanas A., Megalokonomos G. and Moraitis P., 2006: On the Development of a Web-based System for Transportation Services. Information Sciences (in press).

Lazanas A., Evangelou C., and Karacapilidis N. (2005): Ontology-Driven Decision Making in Transportation Transactions Management. In W. Abramowicz (ed.), Proc. of the 8th Int. Conf. on Business Information Systems, Poznan, Poland, April 20-22, pp. 228-241.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of CSCW '94, Chapel Hill, NC.

Sarwar, B.M., Karypis, G., Konstan, J.A. and Riedl, J. (2000). Analysis of Recommender Algorithms for E-Commerce. In Proceedings of the ACM Conference on e-Commerce, Minneapolis, MI, October 17-20, 2000, ACM Press, pp. 158-167.

Schmitt, S. and Bergmann, R.: 1999, Applying case-based reasoning technology for product selection and customization in electronic commerce environments. 12th Bled Electronic Commerce Conference. Bled, Slovenia, June 7-9, 1999.