Adaptive User Preference Modeling and Its Application to In-flight Entertainment

Hao Liu
Technische Universiteit Eindhoven
P.O. Box 513
5600 MB Eindhoven, Nederlands
Tel: +31 - (0)40 - 247 2514
hao.liu@tue.nl

Ben Salem
Technische Universiteit Eindhoven
P.O. Box 513
5600 MB Eindhoven, Nederlands
Tel: - (0) 40 - 247 5915
b.i.salem@tue.nl

Matthias Rauterberg
Technische Universiteit Eindhoven
P.O. Box 513
5600 MB Eindhoven, Nederlands
Tel: +31 - (0)40 - 247 5215
g.w.m. rauterberg@tue.nl

ABSTRACT

This paper first presents an adaptive user preference model for personalized service delivery systems. In this model, user preference is modeled by a two-layer tree with dynamic changeable structures. The top layer of the tree is used for modeling user's long term service preference. Each node represents user's long term evolving commitment to certain categories of service. The lower layer of the tree is used for modeling user spontaneous service requirement which depends on context of use. Each node relates one context of use to one or more desired service requirements. The tree is dynamically constructed by the formal relation definitions among nodes. The advantage of this structure is three folds: (1) it can not only model the user's long term but also spontaneous preference items. More over, the relations between all preference items are formally defined; (2) if the number of preference items is many, it is more efficient and easier to find the right preference items; (3) if the user desired service has been removed, the system can utilize the personalized hierarchy service structure of the preference tree to calculate and recommend similar services. After the introduction of the user preference modeling, an algorithm of how to build it is presented. Finally, we customized the user preference for personalized in-flight entertainment recommendation to validate its features.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems – *Human factors, Human information processing.*

General Terms

Algorithms, Design, Human Factors, Languages, Verification.

Keywords

User preference, context-awareness, in-flight entertainment.

DIMEA'08, September 10–12, 2008, Athens, Greece.

1. INTRODUCTION

User preference was suggested as an improvement for a variety of applications. Current trends are the integration of user preference in the delivery of personalized services for an aware environment [1] [2]. According to the previous investigation [3] [4], user preference involves not only user's evolving long-term commitment to certain categories of service, but also user's instantaneous service requirements which depend on context of use. The user's instantaneous service requirements subordinate to the user's long-term service preference.

Currently, user preference modeling approaches for personalized service delivery systems could fall into two categories: (1) hierarchical tree modeling approach where the user preference is modeled by dimensions (e.g., sports, reading), each dimension can be further refined with sub dimensions [5]; (2) rule-based language modeling approach where the delivery of services relates to the context of use with if-then logic [1] [6]. The advantage of the hierarchical tree approach is that it is well organized and easier to find user desired preference items. Moreover, if the user desired service has been removed, it is easier to recommend a similar service according to the personalized service category structure of the tree; the disadvantage aspect is that it can only express user's long-term static preference. The advantage of the rule-based language approach is that it is based on a clear formalism and can be used to express some of the user's dynamic characteristics such as context-aware user preferences. The disadvantage side is that its expressive power is limited and not able to model the relationships among rules. So, if the number of user preference items is many, organizing, updating, pinpointing, etc. the preference items are not easy.

In this paper, we first present an adaptive user preference model which is represented by a two-layer tree with dynamic changeable structures. The top layer of the tree is used for modeling user's hierarchical static service preference. Each node represents user's long term commitment to certain categories of service. The lower layer of the tree is used for modeling user's dynamic service preference which depends on context of use. Each node relates one context of use to one or more desired service requirements. The tree is dynamically constructed by the formal relation definitions among nodes. It combines the current hierarchical tree and rule-based language user preference modeling approaches' advantages while overcomes their shortcomings; after that, an algorithm of how to build a user preference model is given;

finally, we customized the user preference for personalized in-flight entertainment recommendation to showcase and validate its features.

The rest of this paper is organized as follows. Related works and their limitations are discussed in Section 2. Our approach for adaptive user preference modeling and an algorithm of how to build it are presented in Section 3. In section 4, we first introduce the architecture of a new adaptive in-flight entertainment system. Then, an algorithm of user preference based music recommendation is presented to clarify and validate features of our user preference modeling approach. Conclusions and future works are presented in Section 5.

2. RELATED WORKS

User preference has been an essential component of personalized service delivery systems for a long time such as personalized entertainment [7], query enhancement [8] [9], digital libraries [10] and the personalization of websites [5]. Current trends are the integration of user preference in the delivery of personalized services for an aware environment [1] [2] [6] [11]. These systems model user's preference with either the hierarchical tree modeling approach [5] [7] [9] or the rule-based language modeling approach [1] [2] [6].

Hierarchical tree user preference modeling approach presented in [5] [7] [9] can model user's long-term static service preference without considering user's spontaneous service preference which depends on context of use. The user preference models in these applications are hierarchically organized based on the service ontology of the systems [5] [9] or domain ontology [7] [12]. Due to the hierarchical tree user preference modeling approach can't model user's spontaneous service preference, its express power is not enough to be used to deliver personalized services for an aware environment.

The rule-based user preference modeling approaches introduced in [1] [2] [6] [11] relate the context of use to user's desired services. The user preference models in these applications are modeled with unrelated preference items with if-then logic. However, due to these models can't express the relationships among rules, so it is not easy to organize and manage the unrelated preference items. If the number of preference items is many, it will cost more system performances to find the right preference items than hierarchical tree user preference models. Moreover, because these rule-based user preference models are based on the common service structure of the system and does not consider building personalized service structure which relates to the user's personalized decision tree [3] [4], so, if the user desired service has been removed, it is difficult for the system to recommend alternative services to the user without interruption.

Although user preference model in [10] claims that it can express both long-term and short-term service preference of the user, we found that it just used the before mentioned hierarchical tree approach to model user's long-term service preference and rule-based approach to model user's short-term service preference over a period of time separately. Because of this reason, we do not treat the modeling approach in [10] as a new user preference modeling approach.

3. USER PREFERENCE

In this section, we first present an adaptive user preference model. Then, an algorithm of how to build a user preference model is introduced.

3.1 Describing User Preference

In order to model user long-term static and spontaneous dynamic preference items and the relations between them, we first need to give some formal definitions as follows.

Definition 1: A service is described by a set of attribute (a)/ value (v) pairs. It can be expressed formally by an ordered vector $E_m = ((a_{m1}, v_{m1}), (a_{m2}, v_{m2}), ...(a_{mn}, v_{mn}))$ where (a_{mn}, v_{mn}) is the n^{th} attribute/value pair. In this paper, for simplicity reasons, sometimes we represent $E_m = (v_{m1}, v_{m2}, ...v_{mn})$ since E_m is an ordered vector.

Definition 2: Context of use S_n is defined as a categorization of the actual situation under which the service is delivered by the system. It is expressed formally as an ordered vector $S_m = (v_{m1}, v_{m2}, ...v_{mn})$ where v_{mn} is n^{th} category of the actual situation under which the service is delivered by the system.

Definition 3: User static service preference item T_n is defined as an attribute/value vector (a_n, v_n) where the attribute is a_n and the value is v_n.

Definition 4: For a piece of user static service preference item $T_m = (a_m, v_m)$, if $v_m = \{v_{m1}, v_{m2}, ...v_{mn}\}$, it could then be further refined into sub static service preference items. If it does, we defined the relation between the preference item (a_m, v_m) and its possible sub preference item (a_p, v_{mn}) as refinement $R((a_m, v_m), (a_p, v_{mn}))$ from (a_m, v_m) point of view or composition $C((a_m, v_m), (a_p, v_{mn}))$ from (a_p, v_{mn}) point of view.

Definition 5: User dynamic preference item P_m is defined as $(s_m, w_m * e_m)$ where s_m is the context of use and $w_m * e_m$ is defined as $(w_{m1} * v_{m1}, w_{m2} * v_{m2}, ...w_{mn} * v_{mn})$, w_m is the traditional VSM (Vector space model) to describe the attributes with different weighting schemes where $\sum_{i=1}^{n} w_{mi} = 1$.

Definition 6: For two pieces of user dynamic preference items $P_m = (s_m, w_m * e_m)$ and $P_n = (s_n, w_n * e_n)$, if $s_m \supseteq s_n$ and $e_m \supseteq e_n$, we define the relation between these two user preference items as preference item $(s_m, w_m * e_m)$ could be further refined, and preference item $(s_n, w_n * e_n)$ could be one of the sub preference items to compose user preference item $(s_m, w_m * e_m)$. Formally, we represent the relations as $R(P_m, P_n)$ or $C(P_n, P_m)$.

Definition 7: For a piece of user dynamic preference item $P_m = (s_m, w_m * e_m)$ and a piece of user static preference item $T_n = (a_n, v_n)$, if $v_n \supseteq e_m$, we define the relation between them as preference item P_m subordinates preference item T_n. We represent the relation formally as $O(T_n, P_m)$.

Based on the above definitions, figure 1 depicts our user preference's Meta model. A user preference for personalized service delivery system is composed by the user's long-term static commitment to certain kinds of service and spontaneous service requirements which depends on context of use. User static preference is composed by a set of preference items. Each preference item is represented as an attribute/value pair (e.g. $(entertainmentType, music)$) and it could be further refined (e.g. $(genre, jazz)$) into sub preference items from top-down point of view or be composed by sub preference items from down-top point of view. Each user's dynamic preference item relates one context of use to his/her desired service requirement. It could also be further refined to sub preference items or be composed by sub preference items. Some of the user's dynamic preference items subordinate the user's static preference items as we defined in definition 7.

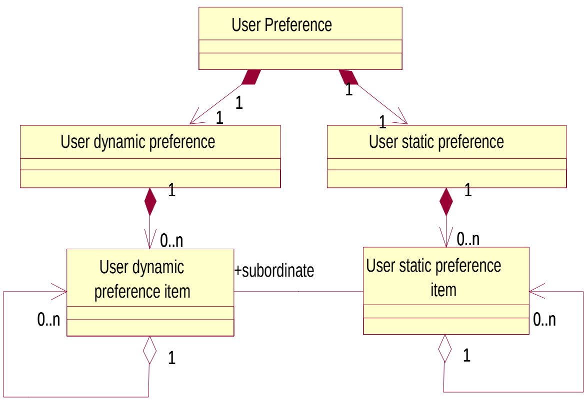

Figure 1. User preference Meta model

A detailed user preference model is illustrated in figure 2. This model consists of two layers, one layer is used to model user's long-term entertainment service preference, and the other is used to model user's spontaneous entertainment preference which depends on context of use. In the user static entertainment preference layer, the entertainment could be further refined into music, game, etc. And music could be further refined into jazz, folk, etc. In figure 2, we assume that Jazz is described by two attributes tempo and author. The user's preference item (nervous, (tempo, 60)) subordinates to his jazz music preference and could also be further refined into sub preference items. In figure 2, one of its sub preference items is ((working, nervous), ((tempo, 60), (author, John)) where (working, nervous) is one context of use.

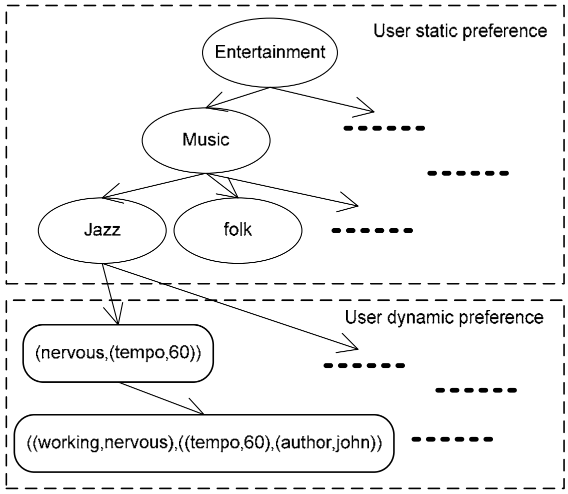

Figure 2. A user preference instance model

3.2 Building User Preference Tree

We represent P as the set of a user's dynamic preference items and T as the set of his/her static preference items. For a specific personalized service delivery system, the relations among user static preference items depend on the service definition of the system. In this sub section, we assume that relations among user static preference items have been built. The following algorithm is used to build a user preference tree. In figure 3, based on the definitions in sub section 3.1, the formal refinement/composition relations among all the dynamic preference items are built from line 1 to line 4. After that, from line 5 to line 15, the subordinate relations between dynamic preference items and static preference items are set up.

Function User_preference_tree_buildup

Input: User static preference set *T*, User dynamic preference set *P*.

Output: User preference tree

1. $for\ \ i=1, i \le p, i++;$ *Build the refinement or composition*
2. $\quad for\ \ j=1, j \le p, j++;$ *relations among preference items*
3. $\quad if\,(S_i \supseteq S_j\ and\ V_i \supseteq V_j);$
4. $\qquad R(P_i, P_j);$
5. $for\ \ i=1, i \le p, i++;$ *Build the subordinate relations*
6. $\{Boolean\ \ r = true;$ *between dynamic and static*
7. $\quad for\ \ j=1, j \le p, j++;$ *preference items*

$\{$ *if* $C(P_i, P_j)$;

$r = false$;

$\}$

if $r == true$;

$for\ k = 1, k \leq t, k++;$

if $V_k \supseteq E_j$; $T_k = (A_k, V_k)$

$O(T_k, P_j)$;

$\}$

Figure 3. User preference tree build-up process

4. APPLICATION TO IN-FLIGHT ENTERTAINMENT

In this section, we first introduce the architecture of a new adaptive in-flight entertainment system. Then, an algorithm of user preference based music recommendation is presented. By this algorithm, two of advantages of our user preference modeling: (1) if the number of preference items is many, it is more efficient and easier to find the right preference items; (2) if the user desired service has been removed, the system can utilize the user preference's personalized hierarchy structure to calculate and recommend similar services, will be validated.

4.1 Architecture of an Adaptive In-flight Entertainment System

Figure 4 presents the main components that make up an adaptive in-flight entertainment system [13]. In the figure, the entertainment service manager is responsible for the in-flight entertainment service (such as music, image and game) registration, categorization, un-registration, etc service management functions. The user context manager collects and models signals from the sensors and updates the context information in the database. The user profile manager models, and updates the user profile information. The user preference learning manger is responsible for user preference tracking and learning. It forwards its result to the user profile manager for updating. The adaptive inference unit is the core component of the whole architecture. It is used to mediate between entertainment services, user context and user profile information according to a set of algorithms to: (1) provide the passenger preferred entertainment intelligently; (2) present personalized entertainment service contents according to the passenger's demographic information and context information such as psychological states if the passenger wants to select entertainment services himself/herself. The passenger interacts with the adaptive in-flight entertainment interface to select his/her preferred entertainment services. The coordination mechanism between the above introduced components is based on Event-Control-Action. For example, once the passenger was in negative stress psychologically and he/ she was not sleeping, chattering with others, working or entertaining, the adaptive inference unit will get the "calming" music list, select personalized music according to the passenger's preference information and play the music to the passenger to reduce his/ her negative stress.

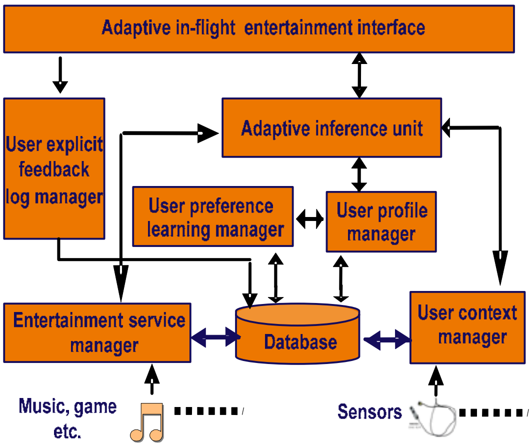

Figure 4. The architecture of an adaptive in-flight entertainment system

4.2 Music Selection Algorithm Based on User Preference

The adaptive in-flight entertainment system recommends personalized music to the passenger according to the context of use, user preference and the available music collection. In this sub section, we will introduce two main process clips of a music recommendation algorithm. The first one is the process of searching appropriate user music preference items on the user preference tree according to the current context of use. The second one depicts the process of recommending similar music if the user preferred pieces of music under one context of use have been moved.

Figure 5 depicts a recursive process of searching for appropriate preference items over the user's preference tree according to the current context of use. We assume T as the stack containing the all the dynamic preference items which subordinate to the static preference items; $R(p)$ is the function of judging whether preference item p is composed by sub preference items; Sub_preference_items is the function that can get all p 's sub preference items and push them into a stack. The function of figure 5 returns a stack containing all the searched preference items under context of use S_m over preference tree P .

Function SFPI (Searching for preference items)

Input: User preference item *P*, Context of use S_m , Stack *T*.

Output: Stack O containing searched preference items.

Begin of function SFPI

$p = T_{.size}$;

$for\ i = 1, i \leq p, i++;$ *Check each node and its sub*

$P = T_{.pop()}$; nodes *in the stack*

$if\,(S \supseteq S_m)$;

$if\,(S = S_m)$;

$O_{.push(P)}$;

$else\ \ if\ \ \ R(P)$; *if* P *is composed by sub*

{ $T = sub\_preference\_items(P)$;

Call Function SFPI; } *preference items*

End of function SFPI

Figure 5. Searching for appropriate preference items

After the appropriate preference items have been found, the next step is to search the required music from the music collection. However, like most of the personalized service delivery system, the in-flight music collection is dynamic. The reason is that different flights have different music collections. Even for the same flight, the airline may update their music collections periodically. So it is quite possible that the system can't find the exact pieces of music required by user preference items. Under this circumstance, in order to increase the robustness the system should be able to recommend similar pieces of music to the user. To do that a similarity measure between music options is necessary. The following definitions give the similarity measurement between two pieces of music.

Definition 8: The attribute/value similarity measure $S(v_m, v_n)$ is defined as the similarity between attribute/value (a_n, v_n) and (a_n, v_m). For nominal, binary and categorical attributes, $S(v_m, v_n)$ is either 1 if the attribute values are identical, or 0 if the value does not match. Formally,

$$S(v_m, v_n) = \begin{cases} 1, v_n = v_m \\ 0, v_m \neq v_n \end{cases}$$

For numeric attributes, the value of $S(v_m, v_n)$ is one minus the value of the ratio between the absolute value difference and the total span of the attribute value domain. More precisely, $S(v_m, v_n) = 1 - \frac{|v_m - v_n|}{r_k}$ where r_k is the total value span of attribute a_n [3].

Definition 9: The music similarity measure $S(e_m, e_n)$ is defined as the similarity between music options e_m and e_n. Combining the definition 8 and definitions in section 3, it is then essentially a normalized weighted sum of the value similarity of attributes. Formally,

$S(e_m, e_n) = \sum_{i=1}^{i} w_i * S(v_{mi}, v_{ni})$ where w_i is the weight of attribute a_i in music selection with $\sum_{i=1}^{i} w_i = 1$ [3].

Figure 6 depicts a recursive process of searching for user desired music according to preference items. We assume M is the available music collection of the in-flight entertainment system and $Cmp\_music\_col(p, M)$ is the function of computing the sub collection of M that may relate to preference item p. V_t in figure 6 is the personalized thresh hold value of the user's acceptance of the similarity difference between his/her desired music and others. This value is learned and set by the user preference learning manger component in the in-flight entertainment system. Function $Cmp\_ft\_nd(p)$ in line 9 of figure 6 is used to find the preference item with which p has composition or subordination relation.

Function: Music recommendation

Input: User preference item p, music collection list M.

Output: List of desired music L.

1. Begin of function Music_recommendation
2. $M = Cmp\_music\_col(p, M)$;
3. $n = M_{.size()}$;
4. $for\ \ i = 1, i \le n, i{+}{+};$ *Get the desired music list*
5. { $E = M_{.get(i)}$;
6. $if\,(comp\_Similarity(E_p, E) \le V_t)$;
7. $L_{.add(E)}$; }
8. $if\ \ \ L_{isEmpty()}$;
9. $p = Cmp\_ft\_nd(p);$ *Get the preference item with*
10. $if\ \ \ p\,!{=}\,null$; *which p has subordination or*
11. *Call Music_recommendation*;
12. End of function Music_recommendation

Figure 6. Process of recommending music

4.3 Validation

In this sub section, two advantages of our user preference modeling: (1) if the number of preference items is many, it is more efficient and easier to find the right preference items; (2) if the user desired service has been removed, the system can utilize the user preference's personalized hierarchy structure to calculate and recommend similar services, are going to be validated by analyzing two main process clips presented in sub section 4.2.

Assuming a user has n pieces of preference items, if they are un-organized, the computing complexity of finding the exact preference items under one context of use is $O(n)$; However, using our user preference modeling approach, as we can see from figure 5, the computing complexity is $O(\log(n))$. So, if the number of preference items is many, the system performance could be improved significantly by using our user preference modeling approach.

According to [3] [4], user prefers or rejects a service according to his/her own personalized decision tree which means different attributes of the service have different priority sequence and importance in selection process. In our method, the composition relations between preference items embody the priority sequences of the attributes and the weighing mechanism on attributes embodies the importance of attributes in the user's selection process. So, when the exact music required by a preference item has been removed, our system can roll back the user's selection process of the music to recommend possible alternatives music as we described in figure 6. In this way, the user requirement could be fulfilled and the system could be more robust.

5. CONCLUSION

This paper has presented a new user preference modeling approach for personalized service delivery system. In our model, the user preference is modeled by a two-layer tree with dynamic changeable structures. Compared with the traditional hierarchical tree and rule-based user preference modeling approaches, it has the following advantages: (1) it can not only model the user's long term static but also spontaneous preference items, more over the relations between all the preference items are formally defined; (2) if the number of preference items is many, it is more efficient and easier to find the right preference items; (3) if the user desired service has been removed, the system can utilize the profile's personalized hierarchy structure to calculate and recommend user similar services. After the introduction of the user preference modeling and an algorithm of how to build it, we customized the user preference for personalized in-flight music recommendation to showcase and validate our claimed advantages.

In the future, we planned to do the real world tests to give more quantity measurements to analysis and validate the advantages of our user preference model.

6. ACKNOWLEDGMENTS

This project is sponsored by the European Commission DG H.3 Research, Aeronautics Unit under the 6th Framework Programme SEAT, under contract Number: AST5-CT-2006-030958.

7. REFERENCES

[1] Oppermann R. 2005. From user-adaptive to context-adaptive information systems. i-com 3, 4-14.

[2] Bartneck C., Masuoka A., Takahashi T., Fukaya T. 2006. The Learning Experience with Electronic Museum Guides. Psychology of Aesthetics, Creativity, and the Arts, 0(1), 18-25.

[3] Pawn S. 2000. Music and Choice: adaptive systems and multimodal interaction, PhD thesis, Technical university of Eindhoven.

[4] Pawn S., Eggen B. 2003. Realization and user evaluation of an automatic playlist generator, Journal of new music research, 2003, Vol. 32, No.2. pp179-192.

[5] Goel M., Sarkar S. 2002. Web Site Personalization Using User Profile Information. In AH 2002, LNCS 2347, De Bra, P, Brusilovsky, P, Conejo, R, (Eds.), pp. 510-513.

[6] Etter, R., Costa, P.D., Broens, T. 2006. A Rule-Based Approach Towards Context-Aware User Notification Services, In ICPS06, IEEE, ISBN: 1-4244-0237-9, pp.281-284.

[7] Shi X., Hua J. 2005. An adaptive preference learning method for future personalized TV. In proceeding of KIMAS 2005, Craig Thompson and Henry Hexmoor (Eds.), pp. 260-264.

[8] Korfhage R. 1984. Query Enhancement by User Profiles. In Proc. of the joint BCS & ACM Symposium on Research & Development in Information Retrieval, pp. 111-121.

[9] Pretschner A., and Gauch S. 1999. Ontology Based Personalized Search. In 11th IEEE Intl. Conf. On Tools with Artificial Intelligence, 1999, pp. 391-398.

[10] Amato G., Straccia U. 1999. User Profile Modelling and Applications to Digital Libraries. In ECDL '99, Springer Verlag LNCS 1696, Abiteboul,S, Vercoustre, A-M, (Eds.), pp. 184-197.

[11] Kraemer R., Schwander P. 2003. Bluetooth based wireless Internet applications for indoor hot spots: experience of a successful experiment during CeBIT 2001. Computer Networks, 41(3), pp. 303-312.

[12] Kim J.Y., Kim J.W., and Kim C. 2007. Ontology-Based User Preference Modeling for Enhancing Interoperability in Personalized Services, C. Stephanidis (Ed.): Universal Access in HCI, Part III, HCII 2007, LNCS 4556, pp. 903–912.

[13] Liu H., Rauterberg M. 2007. Context-aware in-flight entertainment system. In: M.J. Dainoff (Ed.) Proceedings of Posters at HCI International: Part X (LNCS CD-Rom [ISBN 978-3-540-73332-4], pp. 1249-1254), Springer.