In: Proc. 5th Int. Conf. on Information and Knowledge Management (CIKM'96), Rockville, MD., 1996, . pp. 243-250.

The Personal Electronic Program Guide — Towards the Pre-selection of Individual TV Programs

Draft Version 3.0

Michael Ehrmantraut, Theo Härder (University of Kaiserslautern),
Hartmut Wittig, Ralf Steinmetz (IBM European Networking Center)

{ehrman, haerder}@informatik.uni-kl.de
{wittig, steinmetz}@vnet.ibm.com

Abstract:
Digital and interactive TV systems will bring hundreds of TV program channels to the people. Then, the use of existing strategies for channel selection will fail: It is neither possible to select the program by "surfing" through the channels, nor by using printed program guides.

This paper describes a new approach towards creation of a Personal Electronic Program Guide (EPG) regarding the individual view preferences of a user. The foundation of individual TV program pre-selection is a classified and categorized TV program description and the maintenance of user profiles describing his viewing preferences. Besides creation of user profiles, an automatic update technique will be introduced. Channel selection events are taken as parameters for an adjustment of user profiles. Personal program pre-selection is done by using intelligent filtering techniques which apply individual user profiles to the TV program description (so-called matching). An approach to design and realize an EPG prototype system will be described in this paper.

1 Introduction

Today, there are two ways users obtain TV program information: (1) *Channel surfing:* The user switches from channel to channel to get an overview of the programs which are currently running. (2) *Using Printed Program Guides:* The user studies paper guides including descriptions and indices to the TV program in order to get (a priori) TV program information.

Assuming there is a TV service with 500 channels broadcasted, channel surfing takes 83 minutes before every channel has been glanced once[1]. In such an environment, a printed program guide would comprise about 350 pages a week (10 channels per page and day). Both methods will fail in practical use. Instead, we propose a filter-based approach providing only that program information to the user which is in accordance with his personal preferences. The resulting personal TV program guide facilitates desired program selections.

For this purpose, user profiles have to be installed based on the preferences of the viewers to be supported. Furthermore, administration procedures are necessary to maintain these personal profiles that they continue to reflect the current viewing habits. Otherwise, changes concerning these habits would quickly make them obsolete. To cope with this problem we pursue as follows: While viewers watch "their TV program", statistic information on channel accesses is collected in a heuristic manner. From time to time, the personal EPG uses such accumulated personal access statistics as input parameters for an automatic profile adaptation. The resulting profile is then used to perform the calculation of the new program guide. Hence, no administration is required by the viewer.

1. It is assumed that the user needs 10 seconds per program.

This paper outlines the concepts, architecture, and implementation of the Personal EPG system. Sect. 2 introduces the structure of TV channel and program descriptors whereas Sect. 3 sketches important aspects of user profiles. In Sect. 4 and 5, we explain the algorithms performing the *matching* of program offers and user profiles; here, the most important step is to achieve *clustering* as feedback interpretation method. Finally, implementation details are discussed in Sect. 6.

2 TV Program Descriptors

In order to compare viewer interests and TV programs offered, two basic elements of a personal EPG can be identified:

- *Program descriptors* containing information about the TV program.
- *User profiles* containing a specification of the user needs, desires and demands.

Program Descriptors

A program descriptor may either contain a specification of a TV program or a TV channel. In this paper, we concentrate on descriptors characterizing TV programs. An analysis of existing printed and digital TV guides yielded the following classes of program attributes:

- Statistics
- Genre / Category
- Quality

Statistics

Table 1 shows the statistical description of the program genres “Show”, “Movie” and “Sport” which are based on attributes used in existing movie databases (e.g. Movie Select).

Table 1: Statistical attributes of program categories

Program Category	Attributes	Example
Show	- Title - Moderator, producer - Prize - Goal - Participants - Interactivity - Advertisement - Live	100.000 Dollar Show John Little 100.000 Dollar to win games against 3 others yes no
Movie	- Script author, producer, actors - Title, year, country - Language - Epoch - Minimum age of the viewer - Interactivity, number of branches - Cost	King, Spielberg, Costner Dark Christmas English Future 18 2 Dollar
Sport	- Title - Moderator - Importance - Player, opponents - Place, local spectators - Live - Cost	Sport News G. Jauch World Championship Germany vs. Brazil Berlin, Germany yes 5 Dollar

As indicated by the examples, some of the attribute values are of the following types (statistical attributes): integer, real numbers, strings, enumerations. However, a number of attributes have "fuzzy" values such as the epoch a movie is playing in.

Genre / Category

The Service Information (SI) part of the Digital Video Broadcast (DVB) standard [DVB94] contains a classification of TV program genres. The genre information specifies one or more categories characterizing the content of a program. A so-called *content identifier* is a descriptor of the digital program and used to provide classification information of an event or a program. To describe an event or a program which are characterized by multiple facets or aspects, it is possible to define a set of content identifiers.

The DVB SI specification consists of the following genres:

- Movie
- News / Current affairs
- Show / Game show
- Sports
- Children / Youth programs
- Music / Ballet / Dance
- Arts / Culture (without music)
- Social / Political issues / Economics
- Education science / Factual topics
- Leisure hobbies

At the first sight, this spectrum of genres seems to cover the desired broad range in sufficient depth. However, when analyzing the DVB SI description in more detail, the following drawbacks become obvious: (1) Just two hierarchy levels are defined in the standard, e. g., the category "Movie" can only be further refined using one level of subclasses. (2) There are only 16 alternatives for each of the two levels of categories. For example, no specific identifiers exist for basketball, golf, or car racing. (3) It is not possible to define weights when multiple categories apply. For example, in case of a movie about car racing sport it may be helpful to have a quantification of the dominance of a specific sub-genre (Movie or Sports). (4) The information in the classification tree is overloaded. For example, it contains information about program attributes like rating information in addition to the basic genre information.

For these reasons, DVB SI is not as precise as required for the personal EPG. Accordingly, we propose the following improvements and refinements:

(1) A genre tree may contain as many sub-genres as required for the classification of the TV program, as illustrated by Fig. 1.

(2) The number of content identifiers per genre/sub-genre has to be increased to achieve a fine granularity.

(3) Each genre/sub-genre node contains an additional parameter which expresses its importance which is used in order to attach a weight to a node.

(4) The genre tree contains genre parameters only. For example, the recommended age of the viewer has been extracted from the genre information and is stored autonomously.

(5) As a result of experimental studies based on existing printed TV guides, additional genres and sub-genres are introduced.

As a consequence of these refinements, our genre tree defined for the Personal EPG system

contains the following top genres: Movie, News, Sport, Music, Education, Travel, Advertisement, and Show. Each top genre is refined by a sub-tree of sub-genres. To illustrate this improvement, Fig. 1 shows the sub-tree of the genre “Movie”. A parameter “weight” is attached to every node in the genre tree. The weight of a node tries to quantify the correspondence of the program with the specific genre represented by this node.

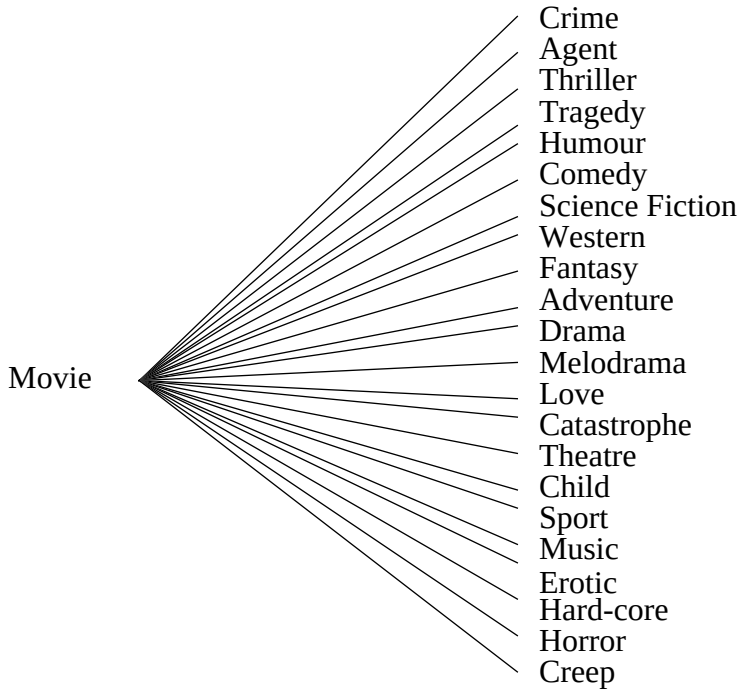

Fig. 1: Genres of movies

Quality

Printed TV guides contain qualitative information, e.g. ranking lists, tips of the day, or reviewer annotations. Though these parameters are not objective, they may help users as additional assessment criteria. To capture this kind of information we analyzed existing TV guides. As a result, we introduced the following attributes to describe the quality of a TV program:

- suspense
- humor
- action
- sexuality

Attribute types

Up to now, we have characterized the kind of TV program information needed for the preselection process. To capture this information, we used the following attribute data types defining the value ranges allowed for the program descriptors:

- *Percent Attribute Type (PAT):* It contains floating point numbers between 0 and 100 (i. e. quality in percent).
- *Number Attribute Type (NAT)*: To store integer numbers (i. e. production year) a NAT is required.
- *Discrete Attribute Type (DAT):* Text strings can be stored by using the DAT. The comparison of DATs is a boolean result.

- *Hierarchy Attribute Type (HAT)*: This attribute type is used to express hierarchical dependencies between attribute values, e. g., they can be used to specify genre trees.
- *Multi-Attribute Types (MAT)*: A MAT consists of more than one attribute of the above types.

3 User Profile Description

So far, we have outlined the structure of the program descriptions which embody the various program offers in the Personal EPG system. To accomplish an automatic matching of user preferences with the best available offer, user profiles are needed. User profiles should express the user's wishes, likes, dislikes and demands which often appear to be fuzzy. Therefore, they are modeled by using fuzzy sets.

The specification of user profiles is similar to the specification of program descriptors with the following exceptions: Each attribute contains a *priority information* data field. This priority information is used to express the importance of attributes w.r.t. the user's preferences. For example, a value of zero means that the specific attribute is not significant. If the priority is 100, the attribute is of high interest for the user.

The comparison of two values depends on the priority of the attribute. Accordingly, a comparison of two fuzzy sets yields a fuzzy result. To illustrate these issues, we give some examples concerning the use of the priority information and fuzzy specification of attributes.

Example 1: A user likes humorous movies independent of the performing actors. Such a preference is expressed in the user profile by giving the genre attribute a high priority (e.g. 90) and setting the actors attribute priority to zero.

Example 2: A user prefers the early *Rock and Roll* movies. This predilection is expressed by assigning the sub-genre *Music - Rock and Roll* a high priority and high value. Furthermore, the epoch of the movie should be specified by using a fuzzy attribute value "60ies".

4 Comparison of Program Descriptors and User Profiles

After having sketched structure and meaning of program descriptors and user profiles, we will explain how we have designed the matching algorithm. The comparison of program descriptors and user profiles - both contain fuzzy values - is the basic operation in the program filtering process. The desired result of this matching or filtering operation is to determine the "best" programs related to the personal interests of a specific user. For this purpose, a program profile and a user profile are compared by using the so-called function *MATCH*. The definition of the MATCH function is shown below:

Definition: MATCH: $R^n \rightarrow R$ with

$$\text{MATCH}(V) = \frac{\|V\|}{\|\xi\|}$$

with: $$\xi = \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}, |\xi| = n \Rightarrow \|\xi\| = \sqrt{n}$$

The result of MATCH indicates the degree of similarity between program descriptor and user profile. The components of the vector V and the result of MATCH are in the interval [0, 100]. $V \subseteq R^n$ and $\| \|$ are a norm to the function MATCH (e.g., Euclidian norm). The vector V is the result of sub-function $MATCH_v$. This vector contains the result values of the attribute-to-attribute comparison of complete profiles.

$$MATCH_v : PB \times P \times W \rightarrow R^n \text{ with}$$

P : Program profile
B : User profile
PB : $P \cup B$
W : Similarity vector $\subseteq R^n$
R : [0..100]

$$MATCH_v (pb , p , w) := (MATCH_A (pb_i , p_i , w_i)) \text{ with}$$

Attribute at position i is of type A
$pb = (pb_0 , ... , pb_n)$ $pb \in PB$
$p = (p_0 , ... , p_n)$ $p \in P$
$w = (w_0 , ... , w_n)$ $w \in W$

and $MATCH_A : X \times Y \times G \rightarrow R$

with: $X \in pb_i , Y \in p_i , G \in w_i , i \in [0..n] , n=card(attributes)$

Since we have multiple attribute types, specific functions for each attribute type are required. Hence, we need a number of comparison functions defined in a similar way like the function $MATCH_A$. For example, $MATCH_A$ compares the values of two similar attributes in the program and user profile. An additional one is shown below in the next section.

Fuzzy Set Matching

The fuzzy method allows the comparison of fuzzy attribute sets. Attributes of the program description, the user profile and the priority of the user profile are input parameters of the fuzzy set matching process. The fuzzy set matching function is defined by:

$$MATCH_{AF} : X \times Y \times G \rightarrow R$$

$$MATCH_{AF} (x , y , g) := f_{A,x} (y)$$

Fig. 2 shows an exemplary function f_{AF}. For each attribute A, the values t_1, t_2 , t_3 and t_4 must be defined as a product of the factors a_1, a_2 , a_3 and a_4 and the weight:

$$\bar{g} = 100\% - g$$

$$t_i := a_i * \bar{g}$$

This definition allows to attach specific parameters to the attributes. By using this property, we gain the ability to model relations between attributes of different types (i.e., with small factors a_i the comparison of two values is more stronger than by using big factors).

To enable the comparison of NAT's similar evaluation schemes have been developed. The other types occurring in the matching process (HAT, MAT) are based on complex computation methods. Details can be found in [Ehr95].

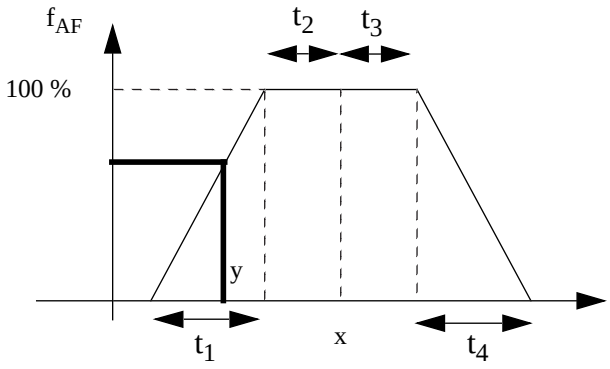

Fig. 2: Specification Model for Attributes using fuzzy sets

5 User Profile Initialization and Adaptation

So far, we have introduced the basic structure of program descriptors and user profiles as well as an algorithm for filtering and matching. To implement these concepts, mechanisms for initialization and update of user profiles are required.

Initialization

User profiles are initialized when users subscribe to the Personal EPG service. The following modes of user profile initialization are provided: *manual*, *semi-automatic*, and *automatic*.

Manual and Semi-Automatic Modes
In the manual mode of profile initialization, the user is required to fill out the initial user profile by himself. Administration of user profiles is very complex due to a high number of parameters to be specified. For this reason, a more convenient *semi-automatic* mode has been provided: A set of initial user profiles is made available. Based on age and profession, users choose their initial profile. In the existing Personal EPG prototype, initial profiles have been derived from usage statistics given by Media Control[2]. In both manual and semi-automatic mode, every time the user preferences will change, a re-adjustment requires explicit update operations. This update must be done “by hand”.

Automatic Mode
An automatic initialization and adaptation mode was developed in addition to the above modes. By observing the program selections (e.g., done by a remote control) program usage data is traced. Based on the usage data, preferences and demands of the user are analyzed. As a result, a history list containing all program selection events is kept. As soon as a minimum set of control operations has been registered, an initial user profile is calculated based on the initial interactions. For example, programs which have often been seen are assumed to be of interest and be part of the initial user profile, whereas programs which were not selected or often interrupted will disappear.
In the automatic mode, users are not forced by the system to give a complete specification of their own profiles and are not strained by complex setup operations. However, the given specifications can not be as precise as required for perfect adjustment.

Dynamic Adaptation of User Profiles
In addition to the automatic initialization of user profiles, the automatic mode can be used for a

2. Media Control is a German company which is specialized in tracking the watching behaviour.

dynamic adaptation of user profiles when changes in the user preferences occur. In such cases, the re-adjustment is automatically performed by the personal EPG system. In the following, the mechanism of automatic re-adjustment will be explained.

Event History List

Cannel number and time are the primary key to access the description of the TV program. The *event history list* contains all program selections identified by channel number and time. A sliding window monitoring mechanism is used for the interpretation of events in the history list and the adaptation of user profiles in the so-called *clustering* phase.

Clustering

Clustering methods are the basic algorithms for the adaptation of profiles. In the context of the Personal EPG, application clustering methods are used to interpret the feedback given by users when watching TV programs. Similar profiles have to be clustered, new interests have to be detected, new sub-profiles have to be merged with existing profiles, existing profiles must be updated and old interests are to be removed.

In general, clustering methods combine a set of profiles to a superset containing similar elements. The clustering methods used in the Personal EPG system were derived from clustering methods used in document retrieval systems:

Clustering methods are divided into non-hierarchical and hierarchical methods. The non-hierarchical ones typically have a low complexity, e. g. O(n) or O(n log n) [Wil88]. As a disadvantage, their results depend on the sequence of the elements processed. In the context of document retrieval, precise results are more important than computation time [Ras93]. Therefore, most document retrieval systems use *hierarchical, agglomerate clustering methods (HACM)*. In the following, we present a list of the most popular HACMs:

- The *single-linkage-method* [JaP73] sequentially groups elements with their nearest neighbors. Their essential characteristics are high stability and a tendency to produce chains of elements *(chaining effect*). Such a property includes the risk to establish large clusters.
- Contrary to the single-linkage-method, the *complete-linkage-method* calculates the elements with the lowest similarity (*Furthest-Neighbor-method)*. Usually, the result is a set of many small clusters.
- The *average-linkage-method* produces clusters that contain documents which have a higher average similarity to the remaining members of their cluster than to all other documents in other clusters [EHW89].

High stability and high precision of results are key requirements for the Personal EPG clustering method. The research results concerning clustering approaches indicate that *the single-linkage-method* clustering algorithm is best suited to meet the above criteria.

Fig. 4 shows some sample clusters of two-dimensional profiles where the characters symbolize the elements to be clustered. The relation “next neighbor” is expressed using arrows. This relation is based on similarity functions [BEP+87] which compute the degree of similarity of two multi-dimensional profiles . In the context of the personal EPG, we use the function MATCH introduced above to determine the similarity of profiles. For this purpose, MATCH is applied to each pair of elements. In a following step, the elements are grouped in such a way that centroids of neighboring relationships are formed. The rectangles in Fig. 4 illustrate the center of gravity of the resulting clusters, i.e., the centroids (C_1, ... C_4). If two elements are their nearest

neighbors they are both connected to their second next neighbor. These computations are repeated until a minimum set of clusters has been achieved (depending on given threshold values). In our cluster example, the four sets (F,A,C,B,K), (G,I), (L,J,D,E,H,N), and (M,P,O,Q) are grouped. The resulting centroids (C_1, ... C_4) are used to adapt the user profile.

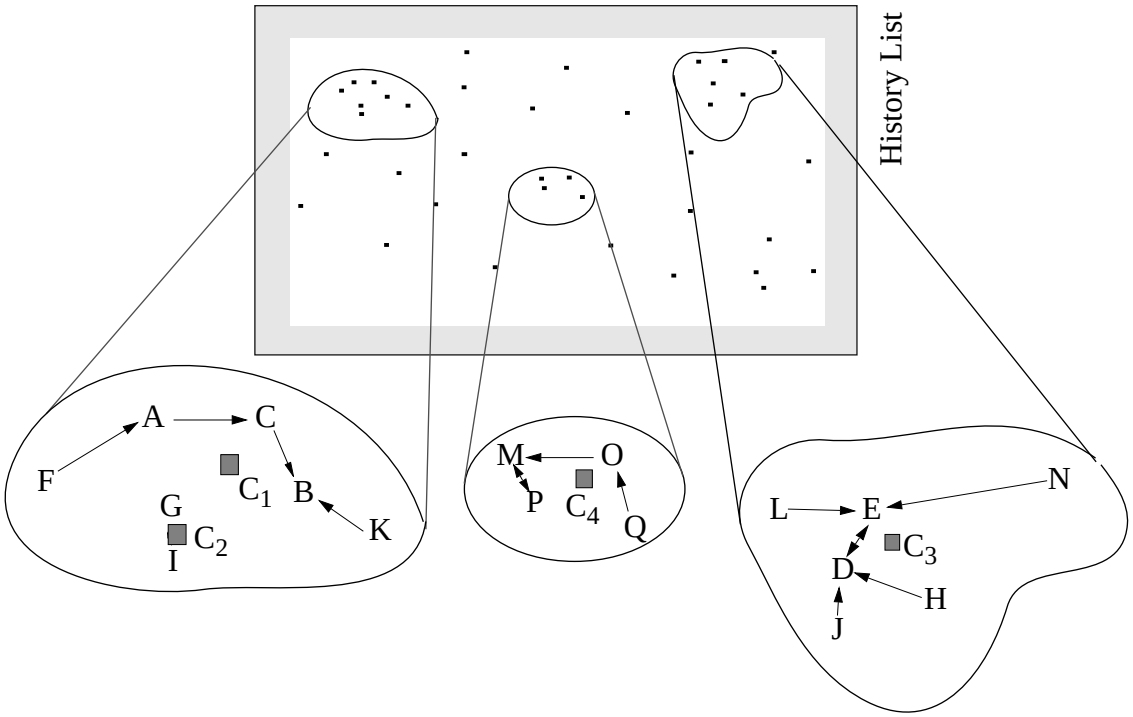

Fig. 3: Clustering method used in Personal EPG

6 Implementation and Results

Personal EPG has been implemented as a prototype system in order to investigate the system and user behavior, and to stimulate the intelligent filtering development related to digital and interactive TV systems. Our prototype system is based on the POSTGRES database system and CLIPS - a decision support system running on the AIX operating system[3].

The following components have been implemented:

1) A metadata base containing attribute names, types and parameter descriptions of the program descriptor and user profile. Add, change or delete operations are implemented.
2) This metadata base is used by an administration tool to generate the program description and user profile database.
3) The functions MATCH and CLUSTER are implemented as specified in Sect. 4.
4) An Administration Test Demonstration Center was implemented based on the Tcl/Tk scripting language. An administrator of the Personal EPG prototype can access program descriptors, test user profiles and calculation parameters.
5) An approximate number of 3.000 movies classified by genre, 100 movies classified by

3. POSTGRES was developed at Berkeley University, California, CLIPS by the US NASA, AIX is trademark of IBM.

quality properties, as well as 10 test user profiles have been added.

The evaluation of the prototype revealed the following results:
By applying program filters, users get a program of manageable size created by taking their personal preferences into account. This program restricted to the interesting offers and the introduction of priority parameters expressing the degree of interest allow to build a hot-list of favorite TV programs for an individual user. Based on this structure, Personal EPG helps to find the program faster and in a more convenient way.

Fig. 4 shows the user interface of our first prototype implementation. The implementation is based on the *Globally Accessible Services (GLASS)* System which is a run-time environment based on the Multimedia and Hypermedia Experts Group standard (details see [BER94], [CGG+96]). The hot-list illustrated in Fig. 4 has been calculated based on a test user profile containing the genres *crime* and *adventure* as preference categories.

Fig. 4: Personal EPG Prototype

Problems

We experienced the problem of deriving small clusters because the “next neighbor” relation is transitive in most cases due to a large number of attributes (= dimensions) in the clustering process. To avoid the effect of small clusters a multi-phase-clustering function has been developed. It allows to define specific attributes as parameters for the first clustering loop (e.g. genre of programs). In the next loops, traditional clustering algorithms are applied.

Furthermore, we have observed a trade-off between the power of small footprint iTV terminals and time-consumption of the matching and clustering algorithms. Therefore, we recommend the following alternative: Off-line generation of the TV guide at a more powerful application server. In this case, however, protection issues concerning private information become more serious.

7 Conclusion and Further Work

We have presented an investigation of concepts and implementation methods towards building

a personal electronic TV program guide. The primary focus of our article has been on structuring and specifying TV program descriptors and user profiles as well as on design of the corresponding matching algorithm for them. Furthermore, we have realized a first prototype of a Personal EPG system including individual user initialization and dynamic user profile adaptation to preference changes. To refine our approach, exploration of the following aspects plays an important role in the future.

Applicability to DAVIC and DVB Compliant Systems

Currently, two system standards for digital and interactive multimedia systems exist: (1) Digital Video Broadcast — DVB and (2) Digital Audio-Visual Council (DAVIC). Our Personal EPG is applicable to both. Based on Personal EPG, a proposal [Wi95] defining the open interfaces required for personalized EPG's has been submitted to and is under consideration of DAVIC.

Knowledge Bases

Our experiments with the Personal EPG prototype have shown that the basic algorithms are not sufficient to recognize specific patterns of user behavior. Therefore, additional aspects have to be considered when exploiting user interactions as input to the automatic update of user profiles:

- Periodic events (e. g., series, christmas, championships, olympic games)
- Zapping
- Sleeping in front of the TV set
- Vacation (no user activities)
- Visits (activities of new users)

To be able to capture such events, our Personal EPG will be extended by knowledge bases approximating and implementing intelligent functions for user profile management and retrieval of structural information. In addition, formal specifications of such knowledge are required as well as the implementation of tailored strategies to derive the best heuristic decisions. Therefore, our Personal EPG system will be extended by an intelligent agent providing knowledge bases for the filtering and clustering tasks.

Semi-automatic Initialization Mode

As an extension of the manual and automatic initialization modes, the implementation of the semi-automatic mode is in progress.

Description of Channels, Scenes, and Cuts

The description of channels, scenes and cuts has been regarded in the design of clustering and matching algorithms. Results of ongoing research in the area of content-based retrieval in vidoes (see [CLP94], [DiG94], and [ZGS+94] for automatic classification and semantic description of scenes and clips) are to be examined. It is expected that similar filtering and adaptation methods can be applied to channels, scenes, and clips.

Acknowledgments

The authors express their gratitude to W. Kalfa and N. Ritter for the fruitful discussion of concepts and improvement of readability as well as to O. Rehders for the integration of the proto-

type application into the GLASS environment.

Literature

[BER94] DEC, GMD Fokus, Grundig Multimedia Solutions, IBM ENC, Tech. Univ. Berlin: *BERKOM Globally Accessible Services: System Specification 1.0.* BERKOM, Berlin, May 1994.

[BEP+87] K. Backhaus, B. Erichson, W. Plinke, C. Schuchard-Ficher, R. Weiber: *Multivariate Analysemethoden.* Springer, 1987, pp. 115-159

[CGG+96] H.Cossmann, C.Griwodz, G.Grassel, M.Puehlhoefer, M. Schreiber, R. Steinmetz, H.Wittig, L.Wolf: *Interoperable iTV Systems based on MHEG*. Proc. Multimedia Computing and Networking (MMCN), San Jose, 1996.

[Chi91] D. Chin: *Intelligent interfaces as agents*. Intelligent User Interfaces, ACM Press, 1991.

[ChS94] S.-S. Chen, S.S. Shen: *Multimedia Computing for Digital Libraries*. Newsletter of the Technical Committee on Multimedia Computing of IEEE Multimedia, June 1994.

[CLP94] T.-S. Chua, S.-K. Lim, H.-K. Pung: *Content-based retrieval of Segmented Images*. Proc. 2nd ACM Multimedia, San Francisco, Oct. 1994.

[DiG94] N. Dimitrova, F. Golshani: *Rx for Semantic Video Database Retrieval*. Proc. 2nd ACM Multimedia, San Francisco, Oct. 1994.

[DVB94] DVB: *Digital broadcasting systems for television, sound and data services; Specification for Service Information (SI) in Digital Video Broadcasting (DVB) systems*. ETS 300 468, 1995.

[EHW89] A. El-Hadouchi, P. Willett: *Comparison of hierarchical agglomerative clustering methods for document retrieval.* Computer Journal 32(3), 1989, pp. 220-227.

[Ehr95] M. Ehrmantraut: *Verfahren zur benutzerspezifischen Informationsrepräsentation in interaktiven Fernsehsystemen*. Diploma Thesis. Univ. Kaiserslautern, IBM ENC Heidelberg, 1995.

[GGR94] J.D. Gabbe, A. Ginsberg, B.S. Robinson: *Towards Intelligent Recognition of Multimedia Episodes in Real-Time Applications.* Proc. 2nd ACM Multimedia, San Francisco, Oct. 1994.

[JaP73] R. A. Jarvis, Edward A. Patrick: *Clustering Using a Similarity Measure Based on Shared Near Neighbors.* Proc. IEEE Transactions on Computers, Vol. C-22, No. 11, Nov. 1973

[Mar93] A. Markus: *Future Directions in Advanced Interface Design.* Communication with Virtual Worlds, Springer, 1993.

[Ras93] E. Rasmussen: *Clustering algorithms*. Information Retrieval - Data Structures and Algorithms (W. Frakes and R. Beat-Yates, Eds.), Chapter 16, Prentice Hall, Englewood-Cliffs, New Jersey, 1993.

[Ril91] G. Riley: *CLIPS: An Expert System Building Tool*. Proc. Technology 2001 Conference, Washington, DC, Nov. 1991.

[SHH86] M. Stonebraker, C. Hanson, E. Hong: *The design of the POSTGRES Rule System*. Proc. Data Engineering Conference, Los Angeles, Feb. 1986.

[Wi95] H. Wittig: *Personal EPG's as DAVIC Applications.* 7th DAVIC Meeting, Application Technical Committee, London, March 1995.

[WiGr95] H. Wittig, C. Griwodz: *Intelligent Media Agents in Interactive Television Systems.* ICMCS ´95, Boston, May 1995.

[Wil88] P. Willett: *Recent trends in hierarchic document clustering: A critical review.* Information Precessing and Management 24(5), 1988, pp. 577-597.

[ZGS+94] H. Zhang, Y. Gong, S.W. Smoliar, S.Y. Tan: *Automatic Parsing of News Video*. Proc. 1st IEEE Int. Conf. on Multimedia Computing and Systems, Boston, May 1994.