DESIGN AND IMPLEMENTATION OF THE ELECTRONIC PROGRAMME GUIDES FOR THE MPEG2-BASED DVB SYSTEM

Tomasz Dajda, Marcin Ciślak, Grzegorz Hełdak, Piotr Pacyna
Department of Telecommunications, University of Mining and Metallurgy
al. Mickiewicza 30, 30-059 Kraków, Poland

{dajda, cislak, heldak}@student.uci.agh.edu.pl,
pacyna@kt.agh.edu.pl

ABSTRACT

The Electronic Programme Guides (EPG) are the standard of service information delivery in MPEG2-based multimedia broadcasting networks. They were introduced in order to enable user's interaction with a diverse content carried over the multiprogramme Transport Stream in DVB-compliant networks. The structures have been standarized in ETS 300 468 document, however, the specific way of their implementation has been left outside the scope of the document, leaving some open issues to developers. This article considers several methods of inserting EPG structures into MPEG2 Transport Stream [1] along with the problems related to their creation and management. The experience has been gained during designing a simplified EPG system.

1 INTRODUCTION

The major part of the paper is focused on the EPG generation and multiplexing. These functions are covered by a broadcasting station which receives the audio-video information originating from TV content providers, video recorders, mobile units and disk arrays as well as interactive services' data produced by application servers (see fig. 1). When all the streams are multiplexed into the output MPEG2 stream EPG structures are added and the entire stream is sent over the broadcasting network to customer terminals which might be software (PC) or hardware (set-top boxes) decoders with the ability to interpret EPG information.

During the process of compression and coding, each audio or video stream is subjected to MPEG encoding, then time stamps for decoding and presentation are added and finally each stream is packetised into Packetised Elementary Stream (PES) [1]. PES stream is decodable but lacks any information on its content.

Therefore Elementary Streams may be combined into Program Stream (PS) or Transport Stream (TS) that includes Program Specific Information (PSI), a set of tables carrying the information necessary and sufficient to demultiplex and present program(s).

Program Stream carries a single program with a single time base and is used in error-free environments [1].

Transport Stream may consist of one or more programs (Single Program TS, or Multi-Prorgam TS) with one or more independent time bases for different programs and is used when errors are likely to occur during transmission [1].

In order to enable creation of the Electronic Programme Guides several tables have been added to PSI [2, 3]. Altogether they constitute Service Information (SI), necessary to implement EPG in MPEG-2 based DVB system. Through the remaining part of the paper by 'EPG structures' the authors mean the Service Information tables.

All the SI tables are carried in TS in sections, that may be of variable but strictly limited length, and are assembled in the receiver to decode the service information.

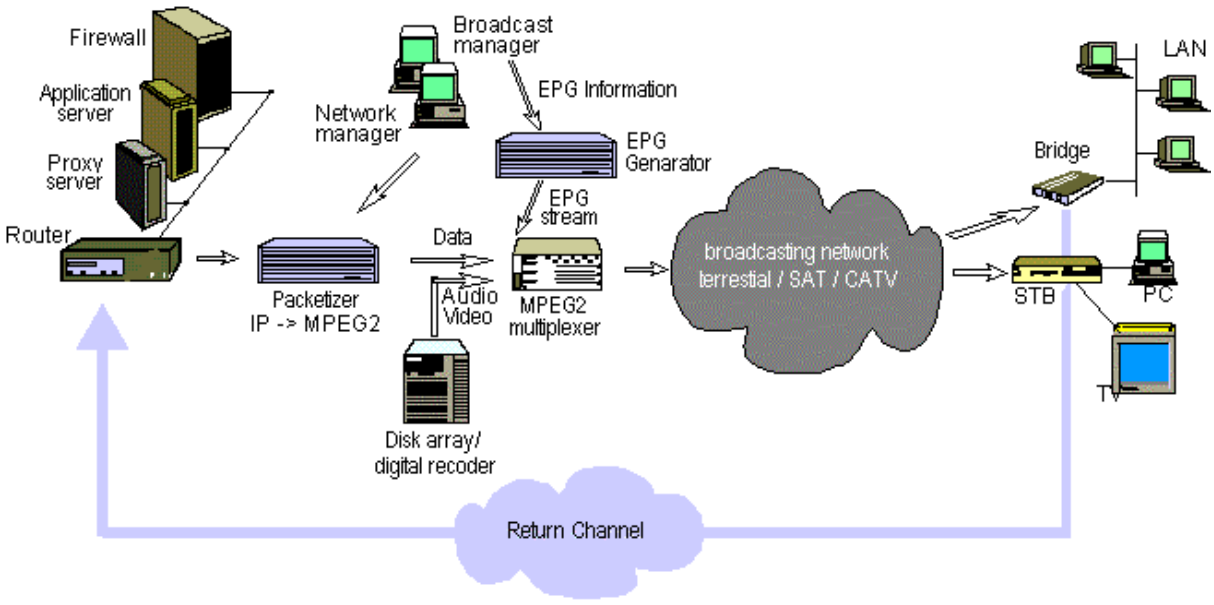

Figure 1. General DVB System.

2 SERVICE INFORMATION

Four out of eleven SI tables are essential to provide a complete description of all services grouped in theme bouquets: Network Information Table (NIT), Bouquet Association Table (BAT), Service Description Table (SDT) and Event Information Table (EIT) [2, 4].

NIT (see fig. 2) specifies ways of delivery (satellite, cable or terrestial) and frequencies for each Transport Stream. This data is to be used in complex, mixed-media broadcast networks working with Frequency Division Multiplexing. We will not describe it further, assuming that all the information needed to create NIT is stored in a system database. Other tables are not directly related to the transport layer of a distribution system.

BAT sections (see fig. 3) identify all the bouquets available in the system with a unique *bouquet_id* and describe each of them in the *bouquet_descriptors_loop*.

Inside *transport_stream_loop* all the services belonging to the given bouquet are specified there and grouped by the TS they are carried by. Moreover, each service is characterized by *service_type* and *service_id* fields. *Service_id* is identical to the *program_number* carried in PSI and will be described later on.

NIT

```
network_information_section(){
    ...
    network_id
    ...
    network_descriptors_length
    network_descriptors_loop
     {
        network_name_ descriptor
        stuffing_ descriptor
        linkage_ descriptor
        multilingual_ network_name
        private_data_specifier
      }
    Transport_stream_loop_length
    transport_stream_loop
      {
      transport_stream_id
      original_network_id
      transport_descriptors_length
      transport_descriptors_loop
        {
        satelite_delivery_system_ descriptor
        cable_delivery_system_ descriptor
        terrestial_delivery_system_descriptor
        service_list_descriptor
        frequency_list_descriptor
        }
     }
    CRC_32
}
```

Figure 2. Network Information Section.

BAT

```
Bouquet_association_section(){
    ...
    Bouquet_id
    Bouquet_descriptors_length
    Bouquet_descriptors_loop
     {
        bouquet_name_descriptor
        stuffing_descriptor
        linkage_descriptor
        multilingual_ bouquet_name_descriptor
        conditional_access_identifier
        country_availability_descriptor
     }
    Transport_stream_loop_length
    Transport_stream_loop
     {
        transport_stream_id
        original_network_id
        reserved_future_use
        transport_descriptors_length
        transport_descriptors_loop
          {
            service_list_descriptor
          }
      }
    CRC_32
}
```

Figure 3. Bouquet Association Section.

SDT

```
service_description_section(){
    ...
    Transport_stream_id
    Original_network_id
    Service_description_loop
     {
      Service_id
      EIT_schedule_flag
      EIT_present_following_flag
      Running_status
      Free_CA_mode
      Descriptors_loop_length
      Descriptors_loop
        {
          stuffing_descriptor
          bouquet_name_descriptor
          conditional_access_identifier
          country_available__descriptor
          data_broadcast_descriptor
          linkage_descriptor
          mosaic_descriptor
          multilingual_service_name
          NVOD_reference_descriptor
          service_descriptor
          telephone_descriptor
          time_shifted_descriptor
        }
      }
    CRC_32
}
```

Figure 4. Service Description Section.

EIT

```
event_information_section(){
    ...
    service_id
    ...
    transport_stream_id
    original_network_id
    event_loop
     {
      event_id
      start_time
      duration
      running_status
      free_CA_mode
      descriptors_loop_length
      descriptors_loop
        {
          stuffing_descriptor
          component_descriptor
          content_descriptor
          data_broadcast_descriptor
          extended_event_descriptor
          linkage_descriptor
          multilingual_component
          multilingual _service_name
          parental_rating_descriptor
          short_event_descriptor
          telephon_descriptor
          time_shofted_event_descriptor
        }
      }
    CRC_32
}
```

Figure 5. Event Information Section.

SDT (see fig. 4) is necessary to inform about the current status of each service and to provide its brief description and data for on-screen presentation (*mosaic_decriptor*). Furthermore, it carries information on means of conditional access, along with a phone number to be used for a dial-up return channel connection in case of interactive services.

EIT (fig. 5) contains a programme schedule for the next 64 hours (*event_loop*). Each event in the schedule is given with a list of video and audio streams related to it (*component_descriptor*), as well as other relevant information, among them text description (*event_descriptor*) and programme type (*content_descriptor*).

As mentioned before, *service_id* fields are necessary to provide linkage between the service information and logical programs within transport layer of MPTS. In order to allow it, Program Specific Information of MPEG2 standard should be used [1, 4]. *Service_id* in BAT or SDT is equal to *program_number*, which is an index to Program Association Table of PSI. In this way identificator (PID) of Program Map Table of a given service could be obtained. PMT lists all the elementary streams belonging to the program along with their PIDs, which provides access to all components of

chosen service on the level of transport layer (fig. 6).
Interestingly, it also means that, while creating EPG for a MPEG2 system, the whole structure of transmitted Transport Streams should be known. On the other hand there is no need to modify the existing PSI tables since additional SI tables have their own, unique PID numbers and, unless already implemented in the received MPTS, there is no threat that EPG being inserted will cause PID's collision.

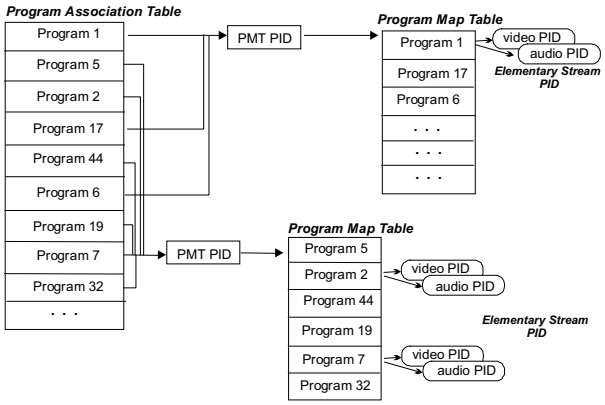

Figure 6. PAT, PMT and Elementary Streams association.

3 TRANSMITTING EPG

Basically, there are two alternatives for EPG delivery. The first is to use extra connection over additional line, usually implemented as a channel for interactive services. A more complicated but more efficient one is to send SI structures along with MPEG2 stream through broadcasting network. In both cases however, for the sake of standarization of MPEG2 decoders, it is a must to multiplex EPG into TS.

This insertion can be done in a broadcast station in two different ways. Firstly, it is possible to modify MPTS in the output multiplexer (fig. 7). The major drawback of this solution is the necessity to use additional MUX with significantly high speed requirements (i.e. quite expensive) to process a multiplex of several Transport Streams. The other option interferes the step of an output stream assembly, when EPG acts as yet another SPTS to be multiplexed (fig. 8). This method is relatively simple and for that reason we chose it for our project. However, one should be aware, that in complex broadcasting studios more than one MPTS assembly step could exist and therefore combined solution might be more appropriate.

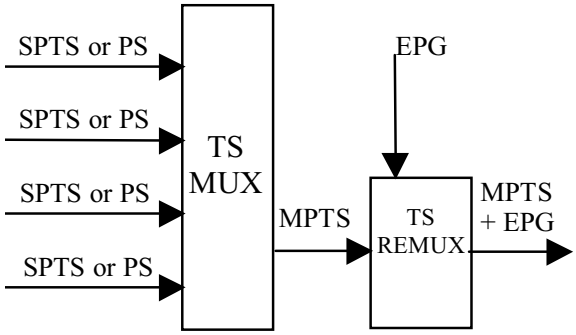

Figure 7. EPG insertion in the output MUX

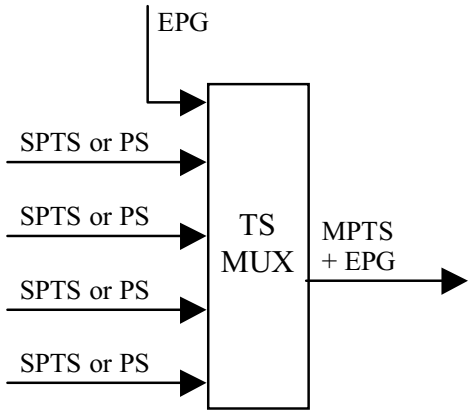

Figure 8. EPG insertion in the main MPTS MUX

4 EPG SYSTEM

A simplified EPG system (fig. 9) involves four input MPEG streams in different formats dependent on the source, which may be disk array (Program Stream) or another station, for example sports transmission provider (Single Program TS). Transmitter creates EPG tables using the information provided in the database residing in a Control Unit and multiplexes them with four input streams. This stream is then subjected to line coding and/or packetizing, and send through broadcast network to be received by set-top terminals. The main part of the receiver is a TS demultiplexer controlled by an EPG Decoder, which provides presentation and user interaction. Selected programme is then decoded by the hardware (set-top-box) or the software (PC) and presented on screen.

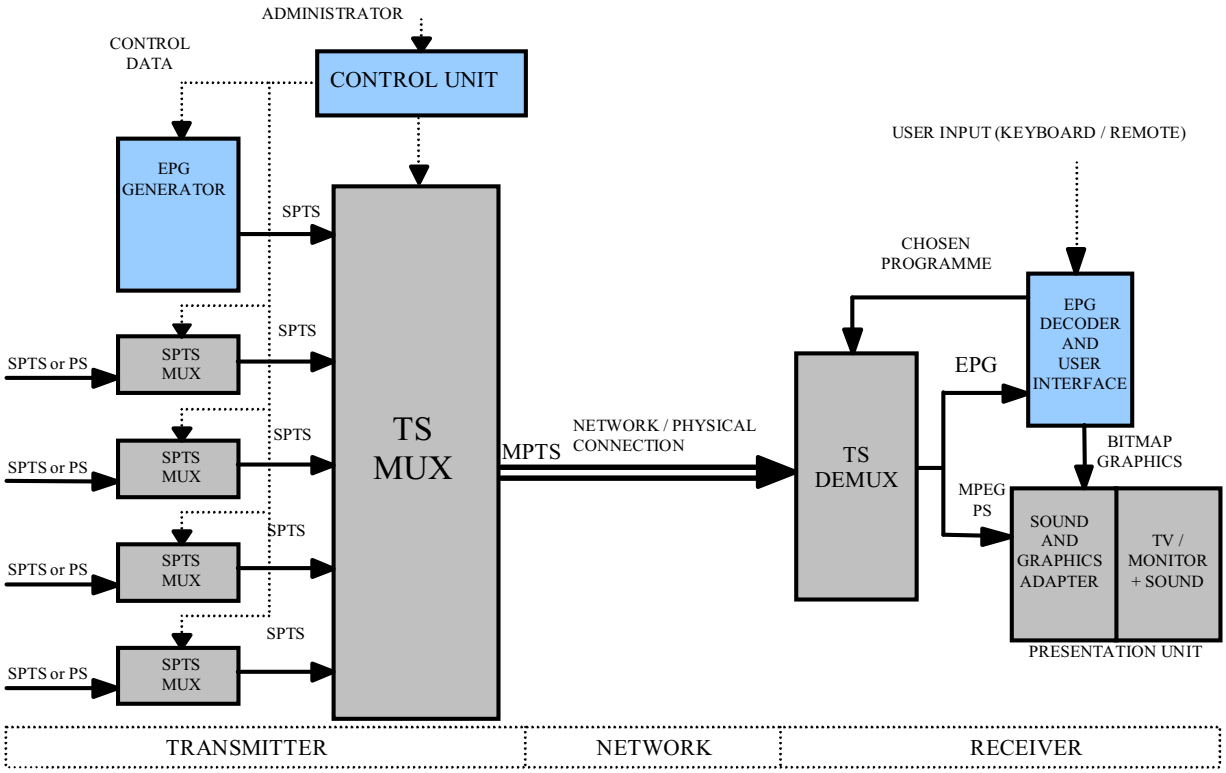

Figure 9. Simplified EPG System.

4.1 Transmitter

The most crucial for a proper functioning of the Transmitter is the Control Unit, which contains the system database. Data, especially such as programme schedule, bouquet grouping and service description, is generally managed by the administrator (fig. 10). However some information like PID numbers and PMT tables can be obtained by analysing input streams which is a task of preliminary SPTS multiplexers.

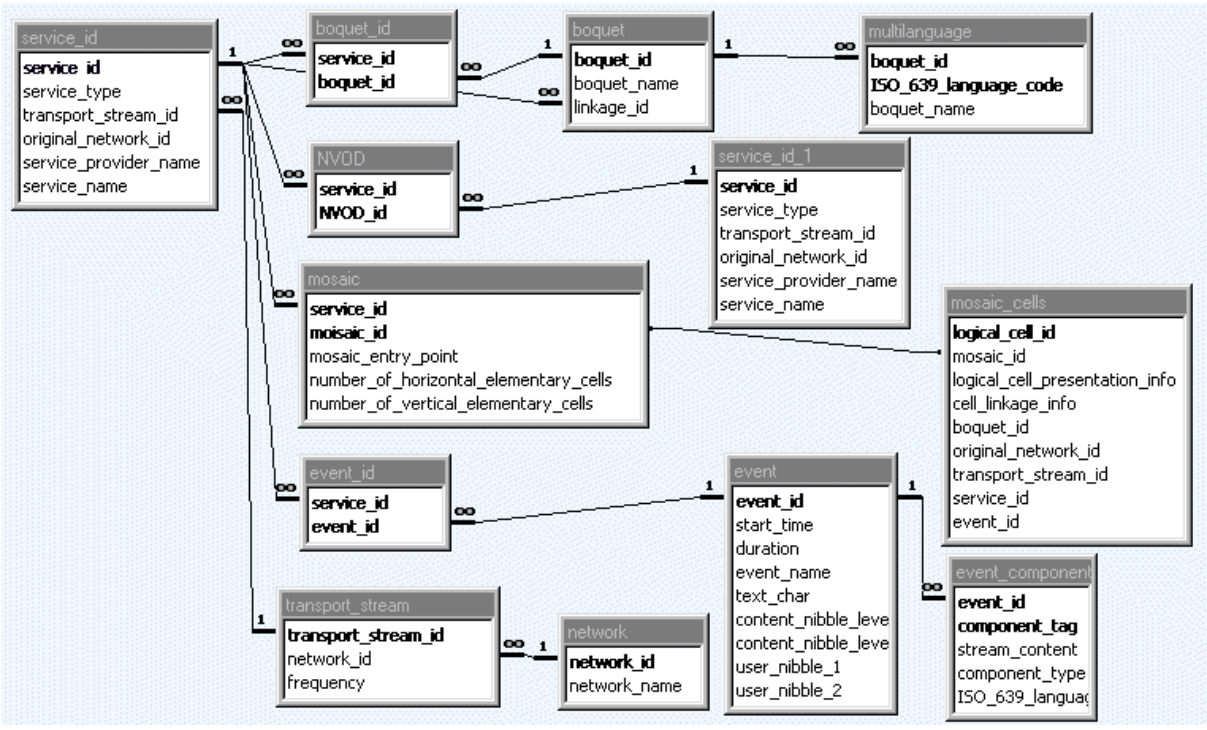

Figure 10. Relational database for EPG system.

SPTS MUXs individually synchronise with sender's transmitter using Program Clock References being transmitted and discard null or duplicated packets. Furthermore they convert any Program Streams coming from library into Single Program TSs, create required PMT tables using information form the database, and duplicate the most important packets. In case of SPTS it might be necessary to translate PID numbers of elementary streams to those used in our system. The transmission between SPTS MUX and MPTS MUX is synchronous.
Since we use relational database management system (RDBMS) the descriptors needed for EPG creation cannot be obtained directly but must be constructed by a program block within EPG Generator (fig. 11) with some help of specific SQL queries.

EPG tables are constructed from these descriptors and then packetised and kept in the memory in the output block so that the number of generation repetitions is reduced. The Generator Control Unit supervises the process of EPG creation and provides proper time restrictions for repeated EPG send-out. Moreover, in case of alteration to EPG information in the database, it starts tables generation procedure to prepare up-to-date packets.
Tables' Creator arranges disordered descriptors into dynamic linear or pseudo-linear structure such as a list with one-way pointers to classes of descriptors and counts the lengths of each descriptor loop.
The last unit reads the list and using the pointers transforms the data from descriptors into 184-byte packets containing table's sections and inserts other relevant information such as *table_id*, *section_syntax_indicator*, *section_length*, *current_next_indicator*, *section_number*, *last_section_number* and *CRC32* [2]. Eventually, the packets are stored in memory and await the moment of dispatch.

As mentioned before, it is not required to group EPG tables in one program since they have unique PIDs. However, since new PAT is created during the process of multiplexing, it would be helpful to arrange SI structures in separate program as it will enable us to use standard TS DEMUX (in receiver) for extraction of EPG stream from MPTS.
The main MPTS MUX uses five SPTS to create output Transport Stream and adds PAT constructed with a help of Control Unit. PAT's packets have the highest priority since the first section contains PCRs for the whole system. SI tables have the lowest priority and are sent last when packet queue exists. In case of queue being too long MUX can change system timing by altering PCRs through the state of clock discontinuity [1].

4.2 Receiver

The MPTS demultiplexer is controlled by EPG Decoder, which specifies the program number to extract. In more complex systems, where we would like to watch several programmes at the same time (eg. picture-in-picture) or browse the schedule along with film trailers separate EPG DEMUX might be needed.
At first EPG SPTS is demultiplexed and sent to EPG decoder (fig. 12), which splits the stream to SI tables and stores them in memory. Simultaneously Mosaic Descriptors from SDT tables are decoded and presented. While user makes the decision about programme to watch a number of mosaic decoding steps can be taken. It depends on the depth of bouquet hierarchy and the decisiveness of user. Eventually, the *service_id* number is acquired and sent to DEMUX, which decodes the chosen programme and sends it to a presentation unit. At this time EPG Decoder allows to choose between several camera views and soundtrack versions by providing presentation unit with elementary stream numbers obtained from EIT table

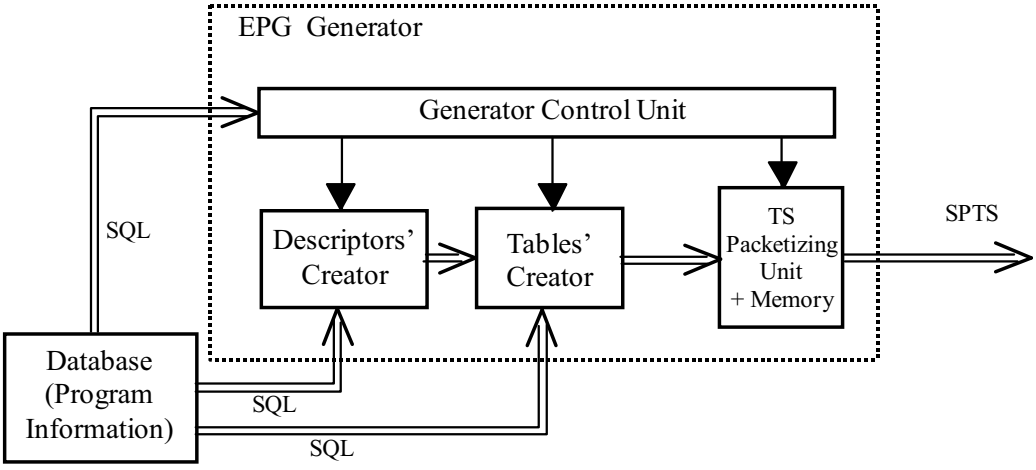

Figure 11. EPG Generator.

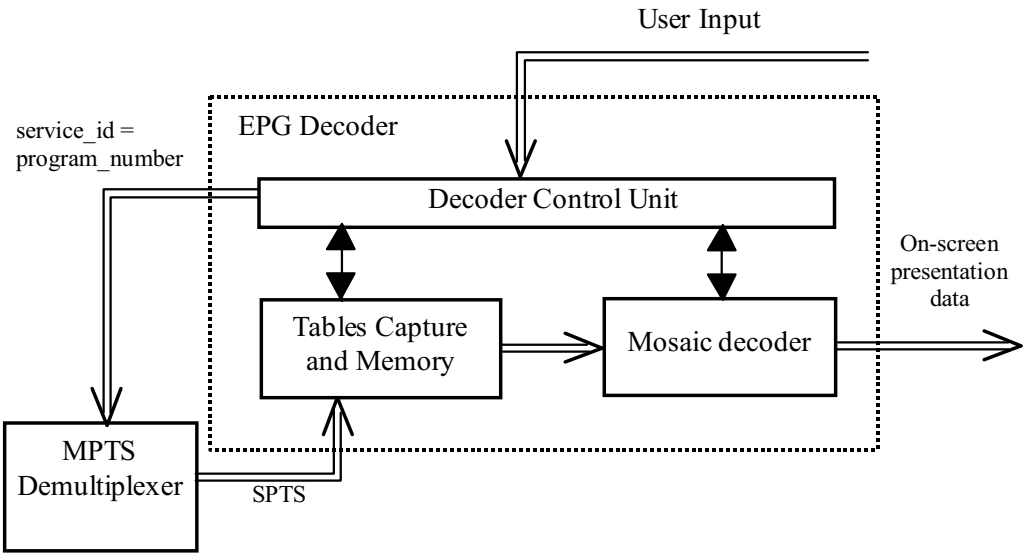

Figure 12. EPG Decoder.

5 CONCLUSION

The SI structures standarized in [2] seem to be sufficient for EPG creation in digital TV systems. However, some interactive services such as home-banking, shopping or video games, carried as private content in Transport Stream may require more sophisticated structures in order to enable interaction beyond simple choice of programme. These structures may need to be defined in the future by a separate standard in order to ensure interoperability of the devices. Therefore, several extensions to the ETS 300 468 standard might become inevitable, e.g. adding specific stream types to identify method of interactive service decoding.

The choice of a proper EPG delivery method ensures easy implementation of the insertion and decoding. Furthermore, it allows the usage of typical MPEG2 TS multiplexers and demultiplexers, and introduces as few new components to the transmission system as possible. However, some drawbacks are to be expected, especially those related to EPG database management and receiver cost reduction.
Although the scope of this article is limited to a simplified DVB system, in more complex solutions problems may appear concerning speed requirements for EPG generator as well on-line processing multi programme input streams.
In case of numerous broadcasting stations, their channels should always be accompanied by adequate EPG structures. Therefore, in the network of numerous TV stations, news providers, interactive services sources, EPG management is not a simple issue as it involves monitoring and transforming EPG information. MPTS would then be remultiplexed, hence EPG creation should partially be based on the data received in the transport streams. The solution here would be a robust database with two methods for updates. Some information, like bouquet hierarchy or network description can only be inserted locally while other information, such as service and event description or programme schedule may partially be derived from the received structures.

To summarise, EPG system should be present in every broadcasting station that transmits its own, or just remultiplexes programmes of other stations. It is up to the proper generator and database construction how efficient and reliable EPG and all the system will be.

6 PAST AND FUTURE WORK

During the process of designing several implementation problems have been successfully coped with, especially the method and the place of EPG inserting and structure of the system database. Moreover, the question on the range of independence of the EPG Generator has been discussed, concluding that database management system should assist EPG generation by constructing substitute views of descriptors as a base for SQL queries from the EPG Generator.

The paper discusses a simplified EPG system, which is currently being implemented as an application for Windows NT. Therefore, the system is actually an emulator of a real broadcasting network focused strictly on EPG management. At present the development effort targets database administrator interface and EPG decoder construction.

REFERENCES

[1] ITU-T Recommendation H.222.0, ‘Generic coding of moving pictures and associated audio information: systems‘, July 1995.

[2] ETS 300 468, 'Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems', January 1997.
[3] J. Derkacz, J. Gozdecki, Z. Hulicki, K. Juszkiewicz, P. Pacyna, J. Roman, "Interaktywne usługi multimedialne na platformie DVB", Fundacja Postępu Telekomunikacji, 210 pages, Kraków, 1999.
[4] P. A. Sarginson, MPEG-2: Overview of the systems layer, BBC Technical Report, BBC RD 1996/2.