Description of the demonstrator for speech output technology

Institute for Perception Research
PO Box 513, 5600 MB Eindhoven

11.11.1996

Rapport no. 1134

Description of the demonstrator for speech output technology

Paul Kaufholz
Sia van Keijsteren
Angelien Sanderman

Voor akkoord: Prof.dr. R. Collier

EoU-SOT
Paul Kaufholz, Sia van Keijsteren, Angelien Sanderman

Description of the Demonstrator for Speech Output Technology

Issue Date : 1996-23-10
Classification : Company Restricted
Reviewers : authors
Issued by : Institute for Perception Reseach
Building HWP 1
P.O. Box 513
5600 MBEindhoven
The Netherlands

Abstract

The project Speech Output Technology is a sub-project of the Ease-of-Use thematic research program for Sound and Vision. Within the project the usage of spoken annotations within an Electronic Programming Guide is investigated. To this purpose a test-environment needed to be created that allows the simulation of representative user-system interactions. In this document the usage of the final electronic programming guide is given and a software design is presented of the GUI part and the basic speech technology necessary to record, play back and navigate in, spoken annotations.

Contents

1 Introduction 2
1.1 Outline 3

2 Design and Use of the EPG 4
2.1 Remote control 4
2.2 EPG 6

3 Hardware setup 14

4 Software analyses and design 16
4.1 General software architecture 16
4.2 The Remote Control 17
4.3 The program and annotation databases 18
4.3.1 Fields of the program database 19
4.3.2 Fields of the annotation database 19
4.3.3 Fields of category database 20
4.4 The basic speech functionality 20
4.4.1 General interfacing 20
4.4.2 Architecture of the BSF 21
4.5 The timer daemon 22
4.6 EPG specific screens 23
4.6.1 Off-line via RC 23
4.6.2 On-line starting page 24
4.6.3 All channels view 24
4.6.4 Program Select screen 24
4.6.5 Category selection screen 25
4.6.6 Category view 25
4.7 The CAPI 25

5 Implementation aspects 26
5.1 Introduction 26
5.2 The EPG main program 26
5.3 Program and annotation databases 26
5.4 The Remote Control 26

5.5 Track Player 27
5.6 Timer daemon 27
5.7 Middleware CAPI 27

6 References 28

7 Distribution List 29

1 Introduction

The project Speech Output Technology (SOT) is part of the thematic research programme Ease-of-Use (EoU) for Sound and Vision. The EoU programme's overall goal is to make future multimedia home entertainment systems easy to use and consistent in their control. It is to be expected that SOT can contribute to the ease-of-use of such systems [Sanderman et. al., 1996].

Users of home entertainment systems will be confronted with a dramatic increase in the amount of information and the variety of media content. In this situation personal comments spoken by the user (voice annotations) or spoken by the system provide an opportunity to increase the usability of the entertainment system.

A particularly interesting test area for the use of annotations may be the interaction of the user with an electronically available tv-guide (EPG). It is to be expected that the growth in number of tv channels and the resulting variety of programmes will be such that the viewer will encounter problems in keeping track of his favourite programs.

While browsing through the EPG, speech is expected to be helpful to annotate the programmes one wishes to mark for later viewing. These personally spoken comments may then be listened for, to be easily reminded of the shows one intended to watch in a particular period of time, and to be warned ahead of time about the beginning of a selected programme. These personally spoken recorded annotations should allow the user maximum flexibility in the formulation of relevant comments that may facilitate later access to the information. This functionality can be increased when users have the possibility to speed up and slow down the playback of the annotations, so that they can effectively receive information at a rate that suits their own needs and capabilites. Also, the possibility to jump to the next or previous annotation will increase the functionality.

The EPG we want to use was developed by PCD (Philips) and evaluated by van der Korst, Westerink & Roberts (1993). This EPG has to be implemented and will be extended with the possibility to link voice annotations to programs and with the possibility to quickly scan the contents of the annotations. Also, non-speech audio signals were added. Therefore we have to develop the hard and software, which means:

-design and implementation of the user interface of the EPG in Visual Basic.

-design and implementation of extra functionalities (e.g. administration of stored annotations).

-design and implementation of speech technology in C (e.g. speed up button).

1.1 Outline

In section 2 the design of the user interface will be presented. We will start with a survey of the funtionalities of the Remote Control, with which the EPG can be controlled. Thereafter a description of the design and the user interface of the EPG will be given. This description should make clear what the possible functions of the EPG are. In section 3 the hardware and software architecture will be described.

2 Design and Use of the EPG

2.1 Remote control

For our project we used a RC to control the EPG. To this purpose an existing RC was chosen which supports most closely our requirements. In Figure 1 a photograph is given of this RC. On this RC we defined the following buttons:

Arrows : indicate the direction of possible navigation.

OK : confirms selection.

Back: to previous screen in the EPG.

Information : presents additional information about the content of a tv-program.

+/- : to next or previous day in the EPG.

Play : playback of the annotations.

Stop : stop the playback.

Pause : stop playing temporarily. To resume play press pause again or press play.

Next : the next annotation will be selected and played.

Previous : the previous annotation will be selected and played.

Fastforward : annotations will be played at a fast speed (10 times normal speed) in forward direction.

Fastbackward : annotations will be played at a fast speed (10 times normal speed) in backward direction.

Play faster : annotations will be played faster at an adjustable rate without change in pitch.

Play slower : annotations will be played slower at an adjustable rate without change in pitch.

Reverse play : annotions will be played in reverse direction at a slow rate.

Record : records new annotations.

Delete : deletes recorded annotations.

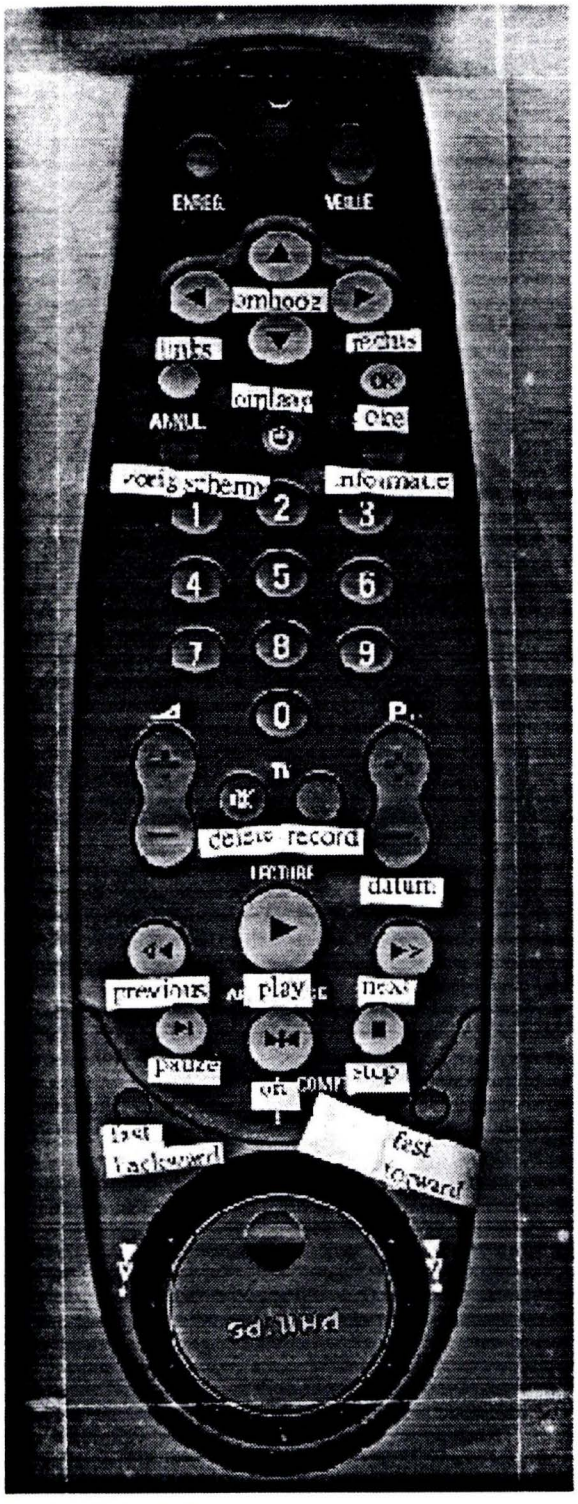
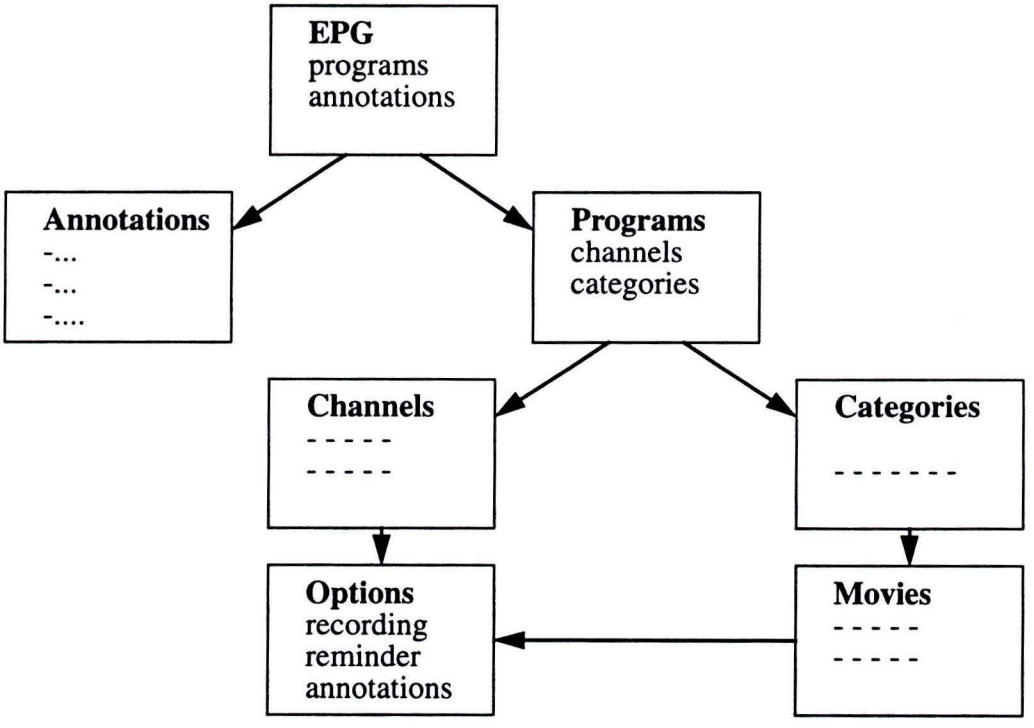


Figure 1: Photograph of the RC.

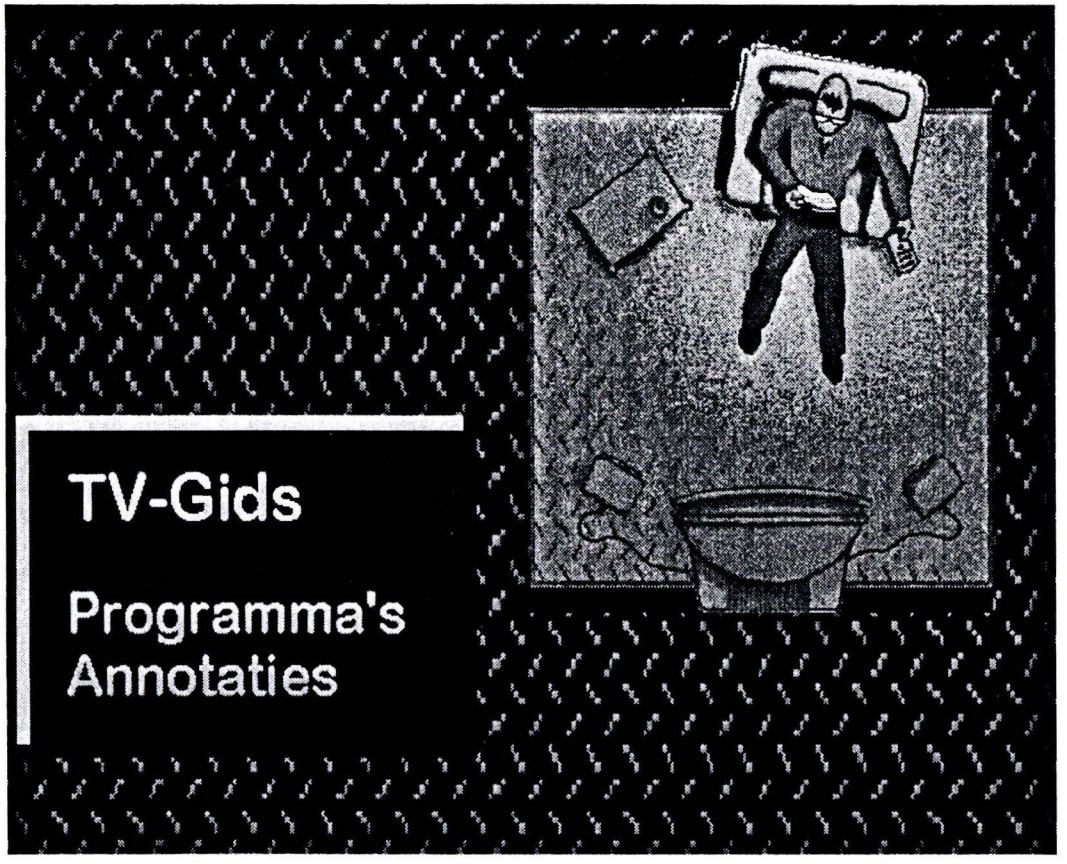
2.2 EPG

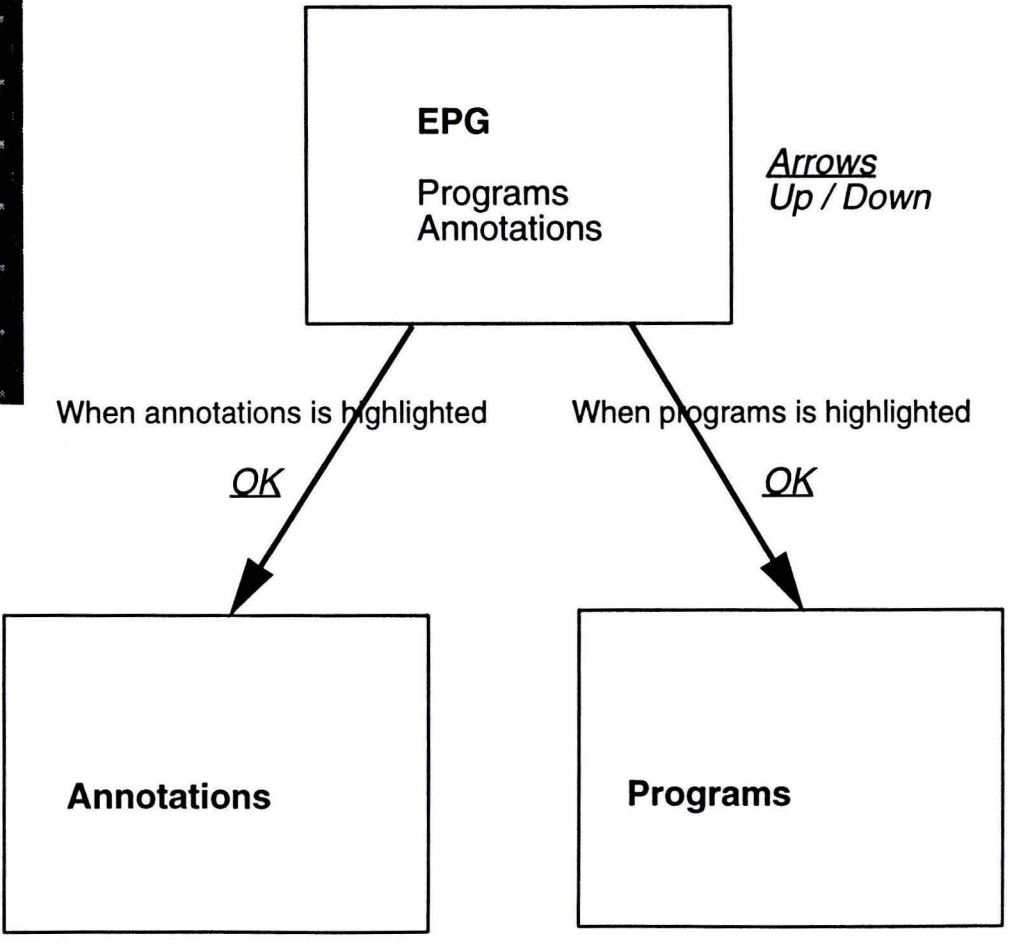
As mentioned in the introduction we implemented an EPG developed by PCD and evaluated by van der Korst, Westerink & Roberts (1993) and extended this EPG with the possibility to link voice annotations to programs. This EPG provides information on current and future television programs. This information can then be used for programming the VCR to record a specific program, to set a reminder, or to link an annotation made by the user or the system to a program. The on-line TV guide offers two ways to access the current and future programs. On the one hand programs can be accessed on a channel basis in the same way as current (paper) TV-guides provide information. This channel-based stucturing works well for a limited number of programs, but when the offer grows to 50 or more channels the information will overwhelm the user. For this purpose the EPG is extended with a possibility to select programs based on a category. By using categories, like sports and movies, the user can search for a program with a certain focus even if the total offer is large. Non-speech audio signals are added to the categories, which can be used later on as a reminder that a certain program from a particular category starts.

The EPG contains several screens whose structure is given below.

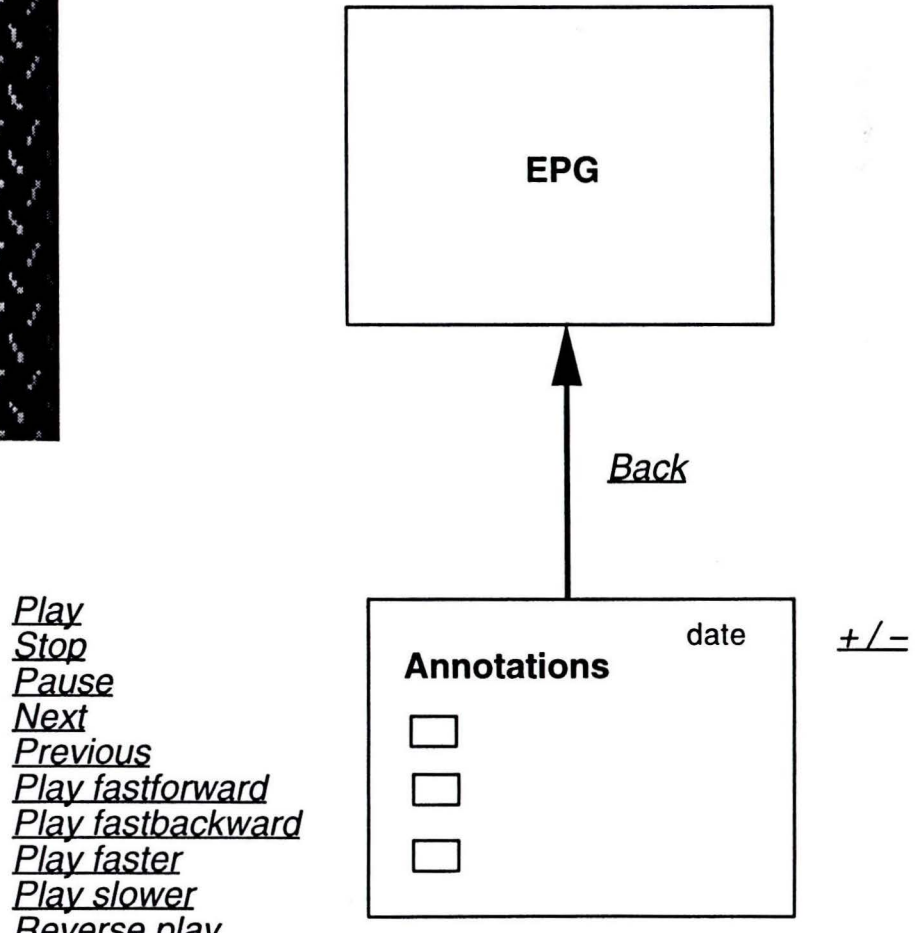


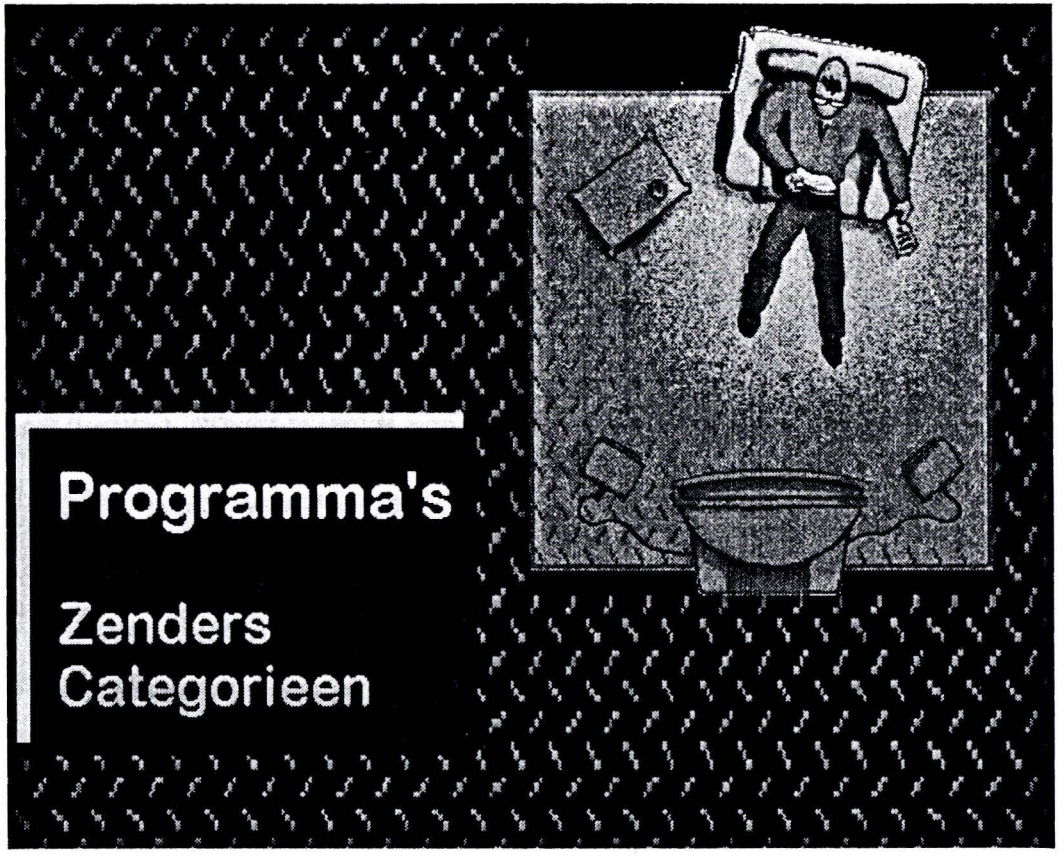
On the next pages a picture and a schematic survey of every screen are given. The schematic survey also contains the possible buttons of the RC that can be used. The RC-buttons are given underlined and in italic.

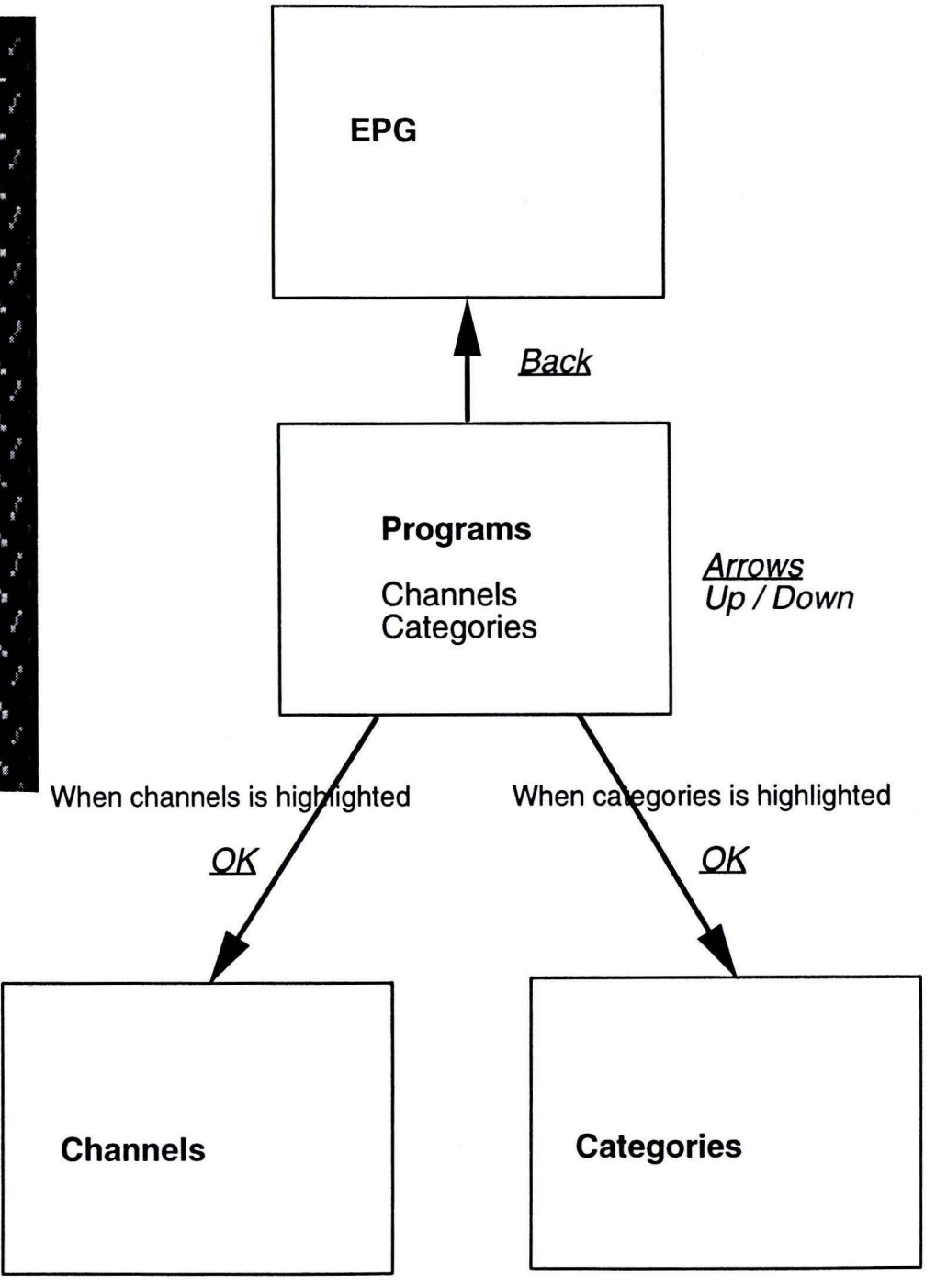

TV-Gids
Programma's
Annotaties


EPG
Programs
Annotations
Arrows
Up / Down
When annotations is highlighted
When programs is highlighted
OK
OK
Annotations
Programs

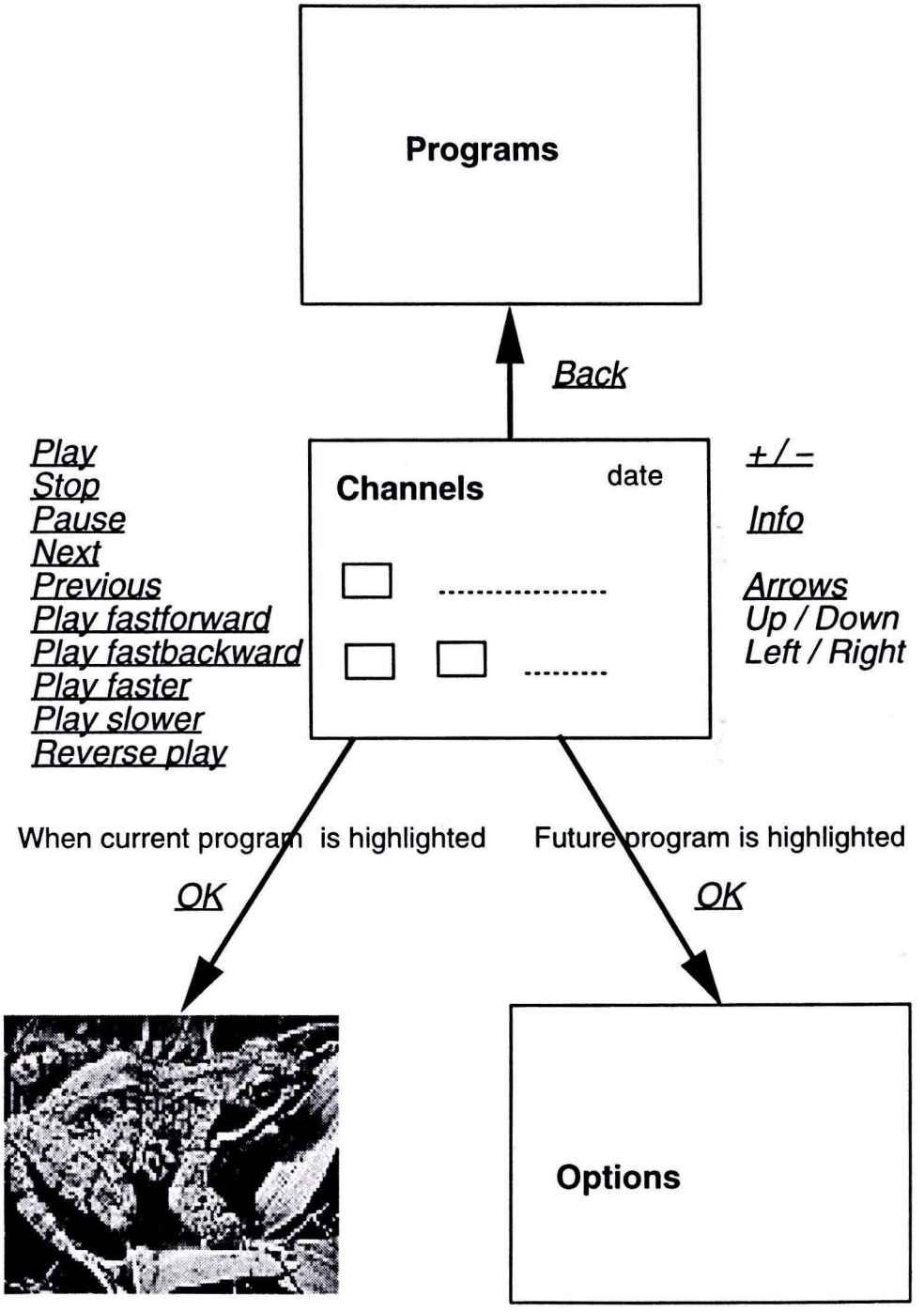

Annotaties
woensdag 24 juli 1996
Hoe is het mogelijk, Nederland 3, 18:00-18:30
Hoe is het mogelijk, Nederland 3, 18:00-18:30
Fawlty Towers, Nederland 2, 18:35-19:10
Fawlty Towers, Nederland 2, 18:35-19:10
>>


EPG
Back
Play
Stop
Pause
Next
Previous
Play fastforward
Play fastbackward
Play faster
Play slower
Reverse play
date
Annotations
+/-

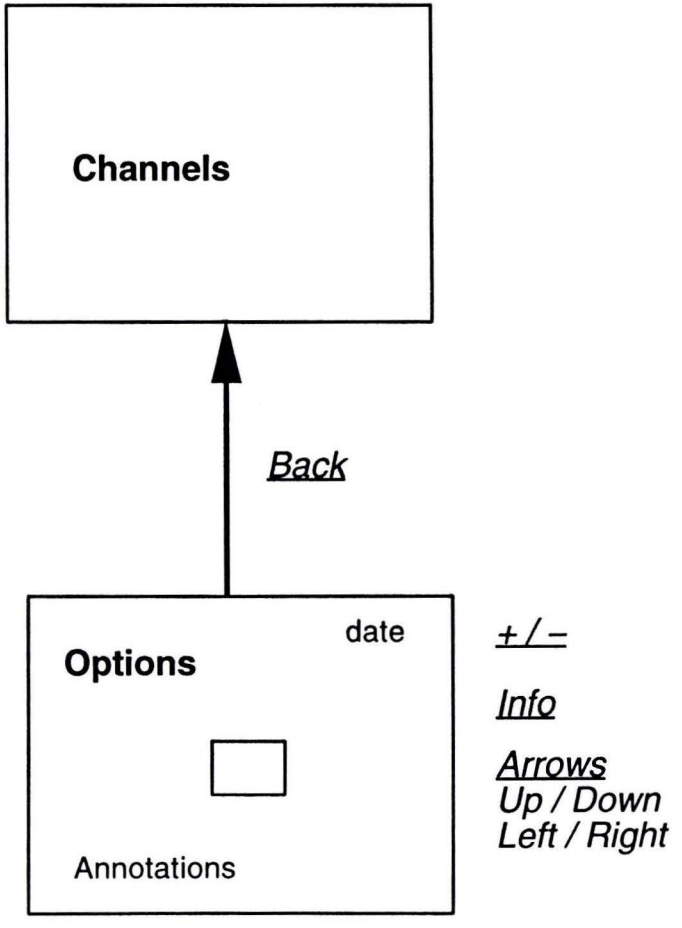

Programma's
Zenders
Categorieen

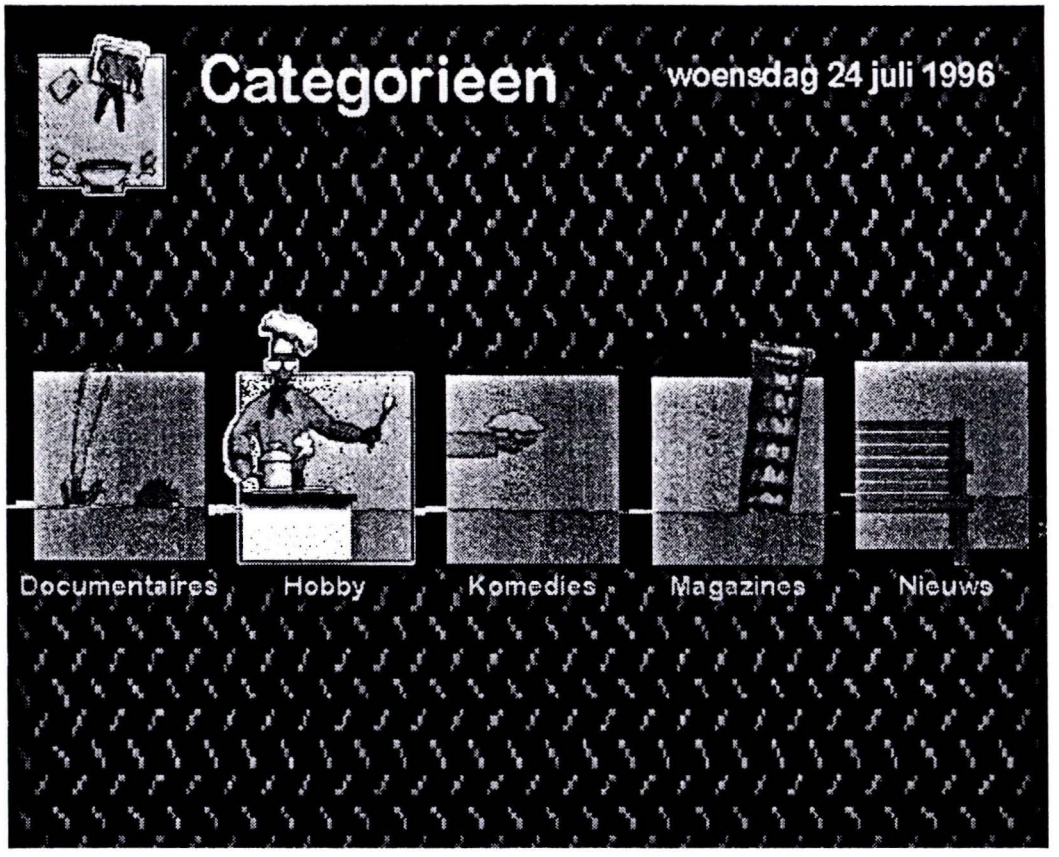

EPG
Back
Programs
Channels
Categories
Arrows
Up / Down
When channels is highlighted
OK
When categories is highlighted
OK
Channels
Categories

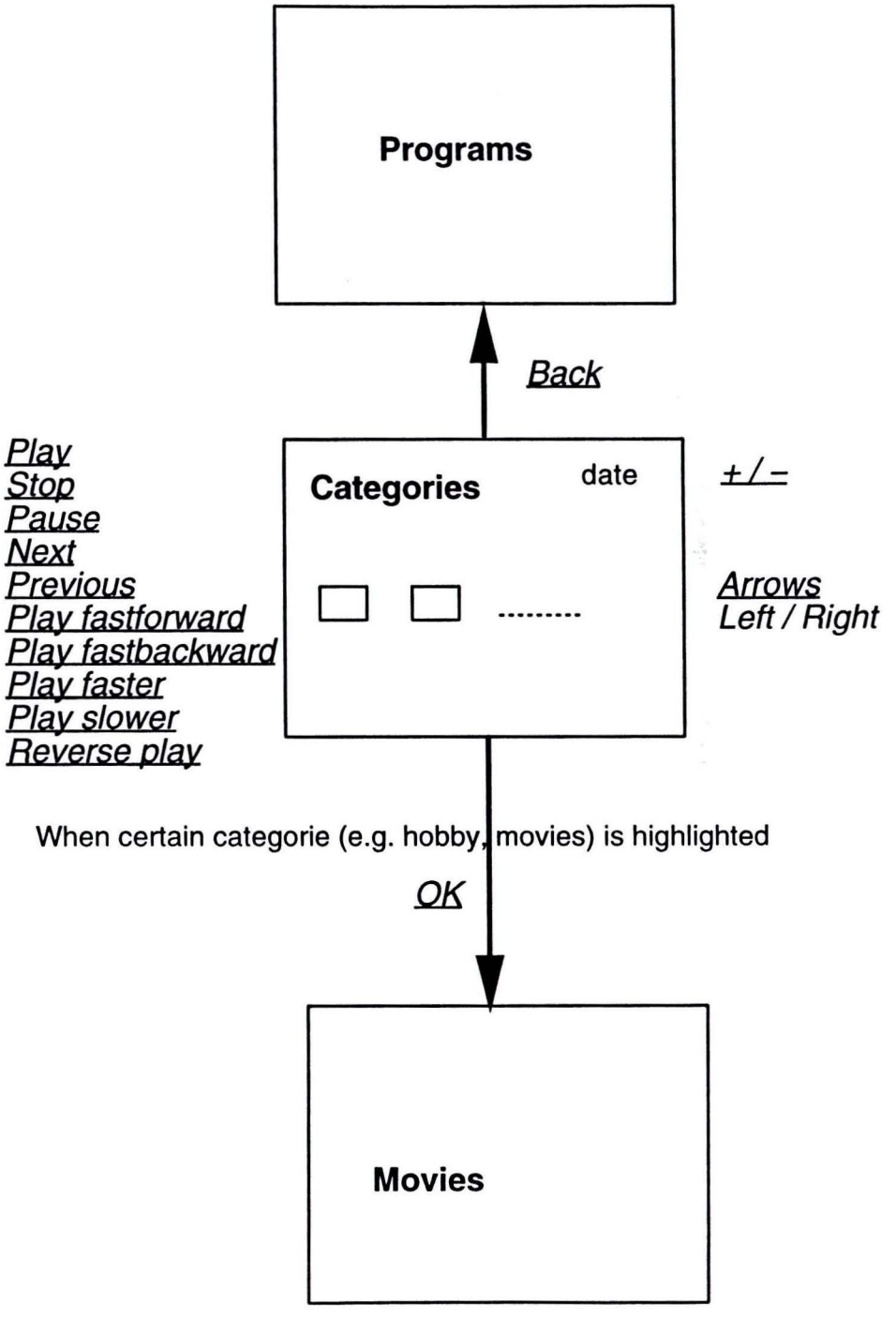

Zenders
woensdag 24 juli 1996
Nederland 1
Roseanne
NOS Journaal
Lalala Live
Duel over de grens
17:55
18:30
18:45
19:20
Nederland 2
Eyewitness
Fawlty Towers
Wings over the world
NOS Journaal
16:04
18:35
19:10
20:00


Programs
Back
Play
Stop
Pause
Next
Previous
Play fastforward
Play fastbackward
Play faster
Play slower
Reverse play
Channels
date
+ / -
Info
Arrows
Up / Down
Left / Right
When current program is highlighted
Future program is highlighted
OK
OK
Options

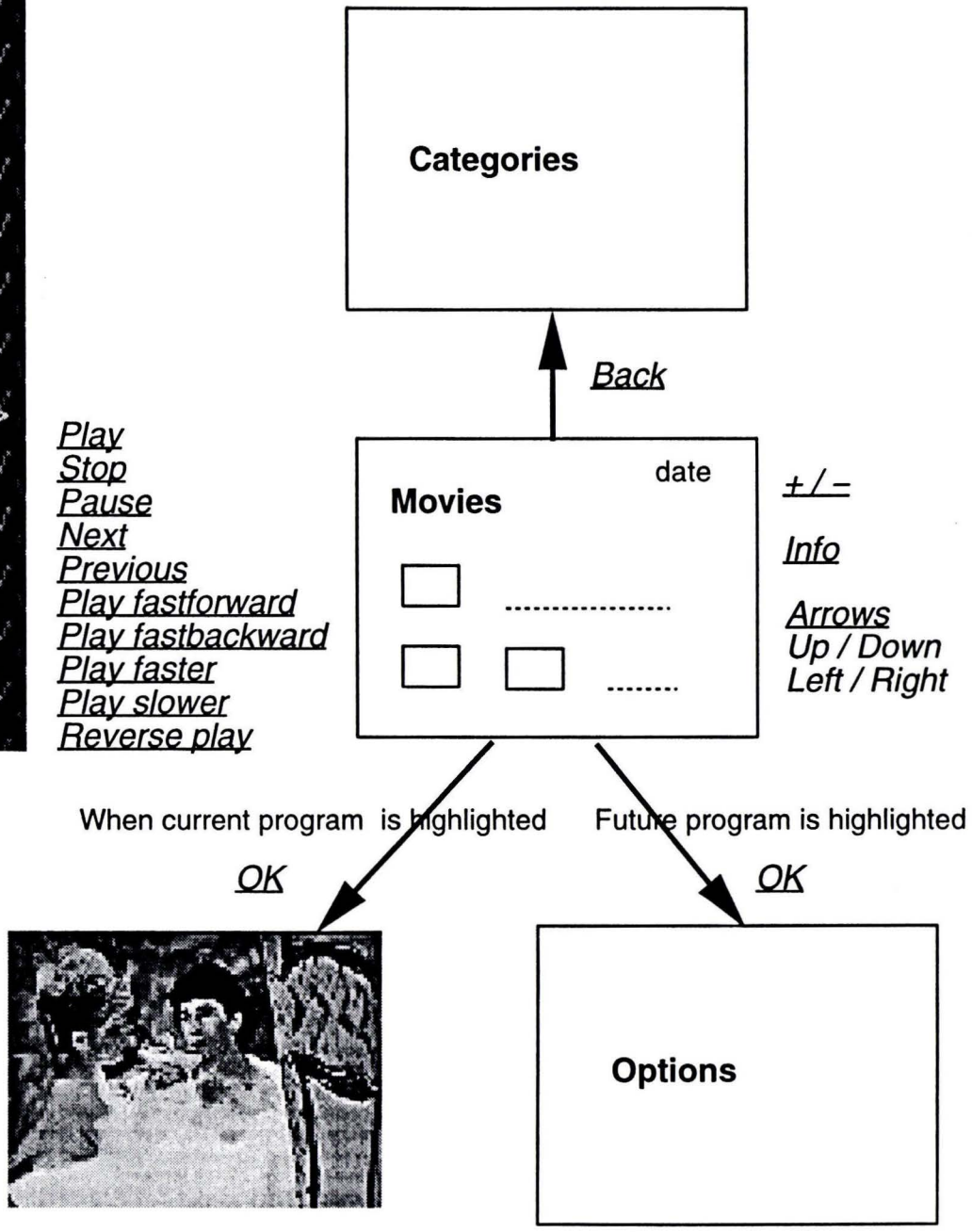

Opties
woensdag 24 juli 1996
Nederland 1
Stealing home
Uit
Uit
Aan
23:45-00:50
Annotaties


Channels
Back
Play
Stop
Pause
Next
Previous
Play fastforward
Play fastbackward
Play faster
Play slower
Reverse play
Record
Delete
Options
date
Annotations
+ / –
Info
Arrows
Up / Down
Left / Right


Categorieen
woensdag 24 juli 1996
Documentaires
Hobby
Komedies
Magazines
Nieuws


Programs
Back
Play
Stop
Pause
Next
Previous
Play fastforward
Play fastbackward
Play faster
Play slower
Reverse play
Categories
date
+ / -
Arrows
Left / Right
When certain categorie (e.g. hobby, movies) is highlighted
OK
Movies


Speelfilms
woensdag 24 juli 1996
Honey, I blew up ...
Dead before Dawn
The Waterdance
Happy Together
19:15
20:35
21:10
22:43
Night of the warrior
Young Einstein
Die Killer-Elite
Hamburger Hill
20:25
21:10
21:58
23:03


Categories
Back
Play
Stop
Pause
Next
Previous
Play fastforward
Play fastbackward
Play faster
Play slower
Reverse play
Movies
date
+ / -
Info
Arrows
Up / Down
Left / Right
When current program is highlighted
Future program is highlighted
OK
OK
Options

3 Hardware setup

The results of the EoU project, and thus of the SOT project, are meant to be possibly implemented in a MAT-like system [Vianen, 1995]. The Multi-Media Access Tower (MAT) combines several MM products and uses a TV set for visualization. A remote control is used for commanding the system.

The prototype for the SOT project is developed on a Windows-based PC. The evaluation experiments should simulate the future home-situation as much as possible. Therefore, it is strongly desirable to use the TV as monitor as well. Also, all necessary commands should be given by a remote control.

A hardware solution to the simulation problem is presented in the figure below. The prototypes run on a PC, whereas the output is transferred to a Media-Line II TV.

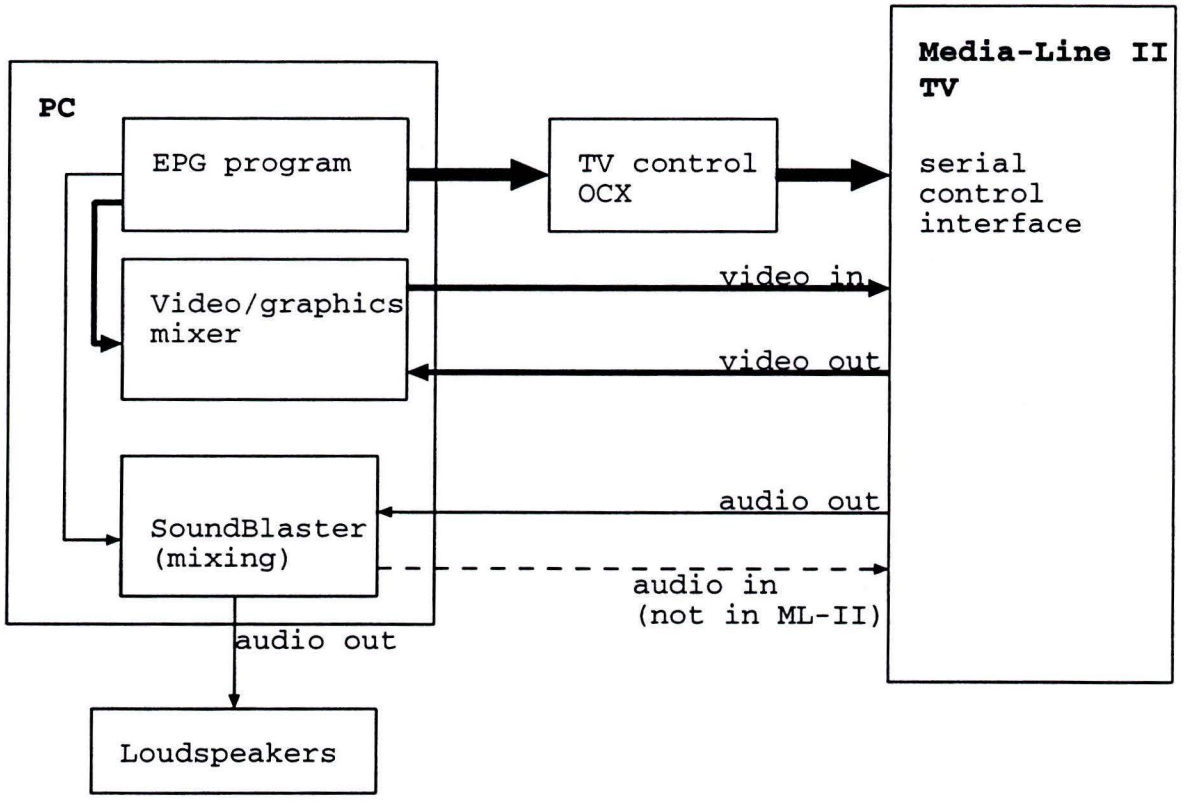


Figure 3.1 Hardware architecture for experimental setup

The ML-II TV offers the possibility to combine the PC-screen output with the video signal, and furthermore, offers an interface to control the TV settings. The prototype application can control the TV settings by the TV CONTROL OCX. The current video signal can be sent to the PC where the signal can be combined with the PC-application screen. After editing the signal is sent back to the TV and displayed on the TV screen.

The audio signal of the selected TV channel is also sent to the PC. Here, the audio signal can be mixed with speech and non-speech audio signals from the prototype application. The final audio signal can be played through the PC Soundblaster or through a direct audio-in port of the TV.

NOTE: At the start of development, the exact specifications of the ML-II TV were not available, nor was a working TV set. Until then, we concentrated on a pure PC application. Not included in the current version is the video and audio mixing. The standard VGA-signal is sent to the ML II TV set and RC control is guaranteed by a separate hardware box (Cohena from the ASA-lab, Sound & Vision) that transmits RC5 coded directly to the PC serial port. The possibility to control the TV set by RC5 codes and the use of a higher resolution VGA signal (600 x 800) is available but not implemented yet.

If all information on the ML-II TV is available the following modules have to be developed:

- TV control
- video mixer
- audio mixer

4 Software analyses and design

4.1 General software architecture

The general software setup breaks the applications down in common objects and specific objects. In the figure below the general architecture of an EPG system using spoken annotations is presented. The main EPG program uses the RC for user input. The necessary data for the TV programs of the different channels is retrieved from the program database. The timer daemon triggers all future actions: warnings, automatic video recording and automatic deletion of outdated annotations. The output devices are all necessary devices for the EPG program, like the screen, DA conversion, eventually a serial control interface to control the TV setting. All annotation related actions are performed by the basic speech functionality module. This module takes care of play back, retrieval and manipulation of the spoken annotations. Note that this module is isolated and could be used in different application domains as such. The speech annotations are standard WAV-files. The current speech manipulation algorithm needs no extra analysis during recording time. Therefore, recording and storage of annotations is done in the main EPG application part, independent from the basic speech module. The Central API (CAPI) is the middleware controlling and interfacing the abstract commands by the EPG and the underlying program database and speech functionality.

Some remarks:

- The RC is only an extra UI-item, a kind of command passing instrument, that passes commands to the non-visual part of the main applications. Another RC API must be defined for the RC module.
- The annotations database must be independent from the program database, since the annotations database must possibly be used separately in another application. Each program in the program database has references to the corresponding annotations.
- The output device will incorporate the future video and sound mixing modules and the TV control unit. For now, within the PC application only, this module takes care of the audio output.

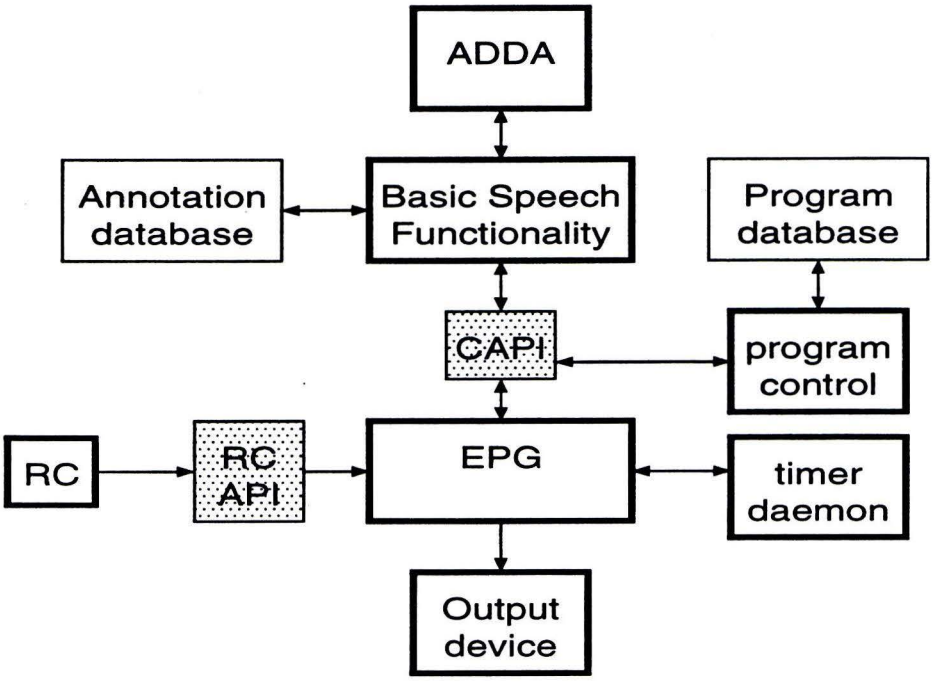


Figure 4.1 The general software architecture for the EPG application

4.2 The Remote Control

In this section the screen for the RC module will be described. Not the UI aspects (composi tion, design etc.) will be looked at, but the neccesary interfacing commands to the application that arises from the functional design of the screen. This will result in the definition of the RC API.

The Remote Control is seen as a command or message passing device. That is, it can be used within different applications. No domain specific knowledge is incorporated in the RC module. The RC passes a number of messages to the window that is receiving the RC messages. The involved ACTIONS must be triggered by the application window that uses the RC.

A *trigger once* message is a message that is sent only once each time the corresponding button is pressed. The duration of the button being pressed is not relevant. A *trigger continuously* message is a message that is passed periodically while the corresponding button is pressed. Next a list of possible messages, triggered by pressing a RC button, is given.

Trigger once RC messages:

```
RC_Play
RC_Stop
RC_Pause
RC_Next
RC_Previous
RC_FastForward
RC_FastBackward
RC_Record
RC_Delete
RC_SpeedOn
RC_Overturn_*
RC_OverturnFixed_*

RC_Back
RC_OK
RC_DateUP
RC_DateDown
RC_Right
RC_Left
RC_Info
```

Trigger continously RC messages:

```
RC_Overturn_-
RC_Overturn_+
RC_Cursor_up
RC_Cursor_down
RC_Cursor_right
RC_Cursor_left
```

4.3 The program and annotation databases

The program database contains all information about programs for the next period of time. The database contains general information about each program like all TV-guides present. Furthermore, some information specific for this EPG is stored, such as the graphical images and an possibly acomment file. Four flags indicate whether or not the program contains annotations, has to be recorded, the system should give a warning or play the comment file.

The annotations themselves are kept separately. In this database all annotation files are kept including their reference to a containing object, in this case a program.

An extra feature in this prototype EPG implementation is the support of non-speech audio for categories. When selecting a category a sound is played which can be used as a cue for the recognition of a category. The association between a category and a sound is kept in another database, including graphical images which represent a category.

4.3.1 Fields of the program database

general:

- id
- name
- start time
- end time
- channel
- category
- info

EPG specific:

- color image
- b&w image
- comment file

flags:

- play comment
- annotations
- remind
- record

4.3.2 Fields of the annotation database

- id
- filename

4.3.3 Fields of category database

- name
- active image
- passive image
- sound

4.4 The basic speech functionality

The Basic Speech Functionality (BSF) takes care of the analyses, retrieval, manipulation and play back of spoken annotations. In the current implementation the basic speech functionality has been implemented as a 'Track Player'. That is, after a list of annotations has been presented to the track player, all functionality accessible through the API resembles the functionality of a CD player.

4.4.1 General interfacing

The following interface functions are supported by the track player (Tplayer):

`TPlayerFastBackward`
Play fast backward from current track.

`TPlayerFastForward`
Play fast forward from current track.

`TPlayerSetSpeed`
Set the speed of speech play back. The maximum speed can be set. In practice up to 3 times normal speed is used.

`TPlayerGetCurrentTrack`
Get the number of the current track in the track list.

`TPlayerIsPlaying`
Boolean: Is tplayer playing right now?

`TPlayerLoadTracks`
Load a new track list.

`TPlayerNextTrack`
Start playing the next track.

`TPlayerNormalSpeed`

Continue playing at normal speed.

`TPlayerPlay`
Start playing tracks.

`TPlayerPreviousTrack`
Start playing the previous track.

`TPlayerPlayReverse`
Play reverse from the current position.

`TPlayerStop`
Stop playing.

`TPlayerPause`
Pause playing.

4.4.2 Architecture of the BSF

In the next figure the decomposition of the BSF into the interacting objects is given. A short description follows. A more detailed description about the internal structure and algorithms is out of the scope of this document.

The application using the basic speech functionality, or track player, controls what must be done by the track player. The track player returns control periodically to the calling application.

Before speed manipulation can be done, some analysis information on the speech signal is needed. The pitch and the voiced/unvoiced decision of the speech signal is calculated in real-time as a function of time. Using this information the original speech signal can be manipulated. That is, its speed is changed without affecting the perceived pitch. Pitch manipulation is also possible but not used in this project. The manipulated speech signal can be played back through the DA converter. A frame-based approach guarantees direct influence on the manipulation process, such as start, stop and change of speed.

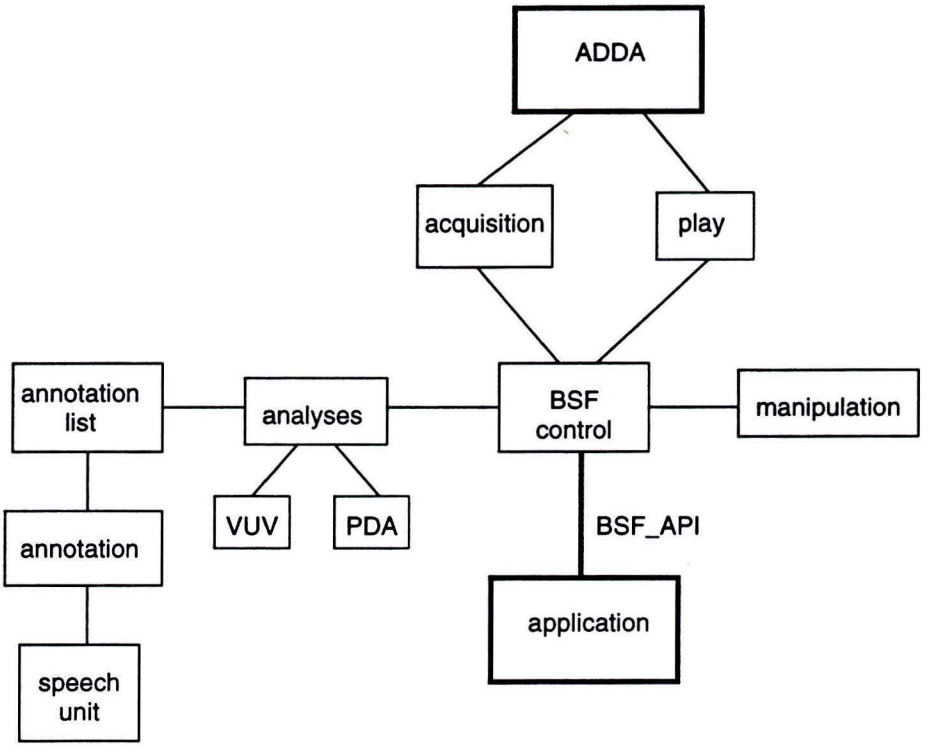


Figure 4.2 Basic Speech Functionality objects.

In the figure also recording capabilities are indicated (AD-conversion). In this project recording is the responsibility of the calling application. Track player only needs simple WAV files as input. No processing during recording time is needed for real-time performance.

4.5 The timer daemon

The timer daemon keeps track of a list of future actions. The timer daemon is checked periodically by the main program. A time stamp is the key for searching. If the checked time stamp is equal to or younger than the time stamp of an action list entry, the corresponding action is returned and activated by the polling program (EPG).

Kinds of future actions (plus action argument):

- warning (program)
- record (program)
- automatic deletion of outdated annotations (annotation)

Messages:

Timer_AddWarning(date, programID)
Timer_AddRecording(date, programID)
Timer_AddAnnotation(date, annotationID)
Timer_IsAction
Timer_GetActionType
Timer_GetProgram
Timer_GetAnnotation

4.6 EPG specific screens

In this section each screen in the EPG program will be described. Not the UI aspects (composition, design etc.) will be looked at, but the neccesary interfacing commands to the Basic Functionality with respect to annotations that arises from the functional design of the screens [van Keijsteren, 1996-1]. **Program related commands are not listed here yet!!!**

4.6.1 Off-line via RC

Ann_Play
Play all present annotations chronologically, starting with the first annotation today. The start time of the program to which the annotation is linked determines the chronological order.

Ann_Stop
Stop playing annotations.

Ann_Pause
Pause playing annotations until 'Play', 'Stop', 'Next' or 'Previous' is pressed.

Ann_Next
Start playing the chronologically next annotation.

Ann_Prev
Start playing the chronologically previous annotation.

Ann_Speed_+
Set current playing speed one unit faster. The maximum will be 4 times the normal speed.

Ann_Speed_-
Play annotations reverse, starting at normal reverse speed. Keeping the button pressed will speed up the reverse play back up to a maximum of 4 times the normal reverse play back speed.

AND/OR:

Set current playing speed one unit slower. The maximum will be 1/4 times the normal speed.

Ann_Speed_*
Play annotations (forward) in normal speed.

4.6.2 On-line starting page

No additional annotation specific commands are used.

4.6.3 All channels view

Additional messages via RC-control:

Ann_PlayProgram
Play all annotations that are related to a selected program. The order in which the annotations for one program are played is chronological. The time stamp of an annotation is the time stamp from the recording of the annotation.

Ann_NextProgram
Play the chronologically next annotation that is related to the selected program.

Ann_PrevProgram
Play the chronologically previous annotation that is related to the selected program.

4.6.4 Program Select screen

Additional messages via RC-control:

Ann_RecordProgram
Records a new annotation that is linked to the selected program.

Ann_RecordStop
Stop recording a new annotation.

Ann_Delete
Delete the annotation with the current annotation ID.

Background messages for window build up:

Ann_GetTotalInProgram
Retrieves the total number of annotation for a selected program.

Ann_PlayNrInProgram
Play the n-th annotation in the selected program.

4.6.5 Category selection screen

No additional annotation specific commands are used.

4.6.6 Category view

No additional annotation specific commands are used.

4.7 The CAPI

The control API (CAPI) is the middleware that controls the low-level functionality modules of the program database and the annotations. The message input list is the main action list as defined in the EPG specific part. That is, all general Ann_XXX messages.

These messages are translated to subsequent calls to the basic speech functionality and the program database control object. Some relatively simple algorithms have to be implemented here in order to return the right information using the low-level API's/objects.

5 Implementation aspects

5.1 Introduction

In the previous chapter the analysis and the design for the EPG using annotations has been described. In this chapter some important remarks are made on implementation aspects. Some implementation solutions need further explanation and others vary from the presented design.

5.2 The EPG main program

The main EPG program has been built in Visual Basic (VB). That is, all user interface aspects and some functional units are built within the VB environment. The timer daemon, the middleware CAPI and the remote control interface are implemented within VB.

The program and annotation databases are built with Microsoft Access. The databases are accessed by the standard database interface from within VB. The track player is linked as a DLL to the VB program and controlled using the corresponding function calls.

5.3 Program and annotation databases

Within Microsoft Access a database has been made with three related tables: Programs, Categories and Annotations. The fields correspond to the design of section 3.3. Several queries have been written to retrieve the appropriate information from within VB.

An extra advantage of this approach is that a user or a program provider can deliver a program database externally. In a commercial system this option can be used to automatically update the program database every week. A new database can be set up separately in the Access environment, using all features and tools available.

From within VB the queries can be called using the DAO interface. DAO calls are also used for updating the annotation database, eg.: insert a new annotation.

5.4 The Remote Control

An external hardware box reads RC5 codes from the remote control. These codes are transferred to the serial port of the PC. The corresponding interrupt routine translates each RC5 code to a corresponding keyboard event. This key-event is then sent to the application. This approach ensures proper functioning of the application with or without a real RC.

5.5 Track Player

The track player is compiled to a DLL. Commands can be sent to the DLL by only one interface function: TPlayerCommand(command).

The command argument can contain all messages as defined in section 3.4. For play back of speech it allocates its own DA resources.

Once more recording is done within VB. The standard MCI interface is used.

5.6 Timer daemon

The timer daemon implementation is somewhat simpler than section 3.5 describes. Originally, each event would be put on a list and the timer daemon periodically checks the starting time of the event against the current time. In the current application there is no list of events. The timer daemon periodically checks whether there are any events to happen at the current time, by directly checking the database. This solution is possible because of two reasons:

1 The number of possible kinds of events is limited to four: remind, start recording, stop recording, delete outdated annotations and programs.

2 The database contains all necessary information to collect possible events by calling pre-defined queries.

5.7 Middleware CAPI

The middleware control API has not been implemented as a separate module. This code is part of the modular VB code.

6 References

[Eggen, 1996]

Ease-of-Use thematic research for S&V (project contract).
Eggen, B. EoU-01.

[Sanderman, 1996]

Speech Output Technology (project contract).
Sanderman, A.A.. EoU-SOT-01.

[Sanderman et. al., 1996]

State of the art report Speech Output Technology (project definition).
Sanderman, A.A., Collier, R., Keijsteren van, S.. EoU-SOT-02.

[van Keijsteren, 1996]

Design and functionality of the electronic TV-Guide with the use of annotations.
Keijsteren, S. van.. In preparation.

[Vianen, 1995]

Results of MAT workshop in Knoxville.
Vianen, E. van, Lange, H. de. Philips Corporate Design.

[van der Sluis, 1996?]

EPG in Visual Basic for non-speech audio.