# DATA PROCESSING SYSTEM

The present invention relates to receiver/decoders for digital television systems.

The advent of digital transmission systems intended primarily for broadcasting television signals, in particular but not exclusively satellite television systems, has opened up the possibility of using such systems for other purposes. One of these is to provide interactivity with the end user.

The present invention finds specific application in a broadcast digital television system in which received signals are passed through a receiver to a receiver/decoder and thence to a television set. The receiver/decoder decodes a compressed MPEG-type signal into a television signal for the television set. It is controlled by a remote controller handset, through an interface in the receiver/decoder.

One way of providing the interactivity described above is to run an application on the receiver/decoder through which the television signal is received. The code for the application could be permanently stored in the receiver/decoder. However, this would be rather limiting. Preferably, the receiver/decoder should be able to download the code for a required application. In this way, more variety may be provided, and applications can be updated as required without any action on the part of the user.

It is desirable to allow a number of different manufacturers to manufacture the receiver/decoder. Obviously all the manufacturers must satisfy certain common functional specifications, but subject to that, it is desirable to permit them to make their own choices with regard to the details of their designs, including the details of the hardware. However, the overall functioning of the receiver/decoder, and in particular the functional manner in which it processes applications, must be identical for all decoder/receivers. The design of those parts of its control system concerned with processing applications must therefore be common to all receiver/decoders, and should be the responsibility of the system operator rather than the hardware manufacturers.

These two objectives (freedom in hardware design and common system functionality) are obviously difficult to harmonize. The object of the present invention is to overcome this difficulty.

According to its main aspect, the present invention provides a receiver/decoder for a digital television system, comprising: means for receiving a compressed MPEG-type signal; means for decoding the received signal to provide a television signal; means for supplying the television signal to a television; a user input interface; and a computer system for controlling the receiver/decoder in accordance with input signals received through the user input interface, the computer system comprising: device manager means for receiving signals from a plurality of ports and providing data to said ports; and virtual machine means, coupled to the device manager means, comprising means for processing data received from the device manager means and returning data thereto.

Preferably the virtual machine means comprise an operating engine, a library of routines, an interpreter, storage means for storing a plurality of instruction sequences, queue management means, and buffer management means.

Preferably, the receiver/decoder further comprises a plurality of interfaces for coupling to external units, said device manager means comprising at least one device means associated with either at least one interface or with at least one function of the receiver/decoder. Preferably also the device manager means include at least one device means coupled to at least one device driver.

Preferably the receiver/decoder includes a plurality of application sources each for providing an application for controlling the receiver/decoder and/or the television, and wherein the operating engine is located between the application sources and the logical device means, and includes queue means for receiving events generated through the device manager means and passing them to the engine. Preferably the operating engine comprises a plurality of process sequencer units associated with the device means, and means for extracting events from the queue means and actuating the

associated process sequencer units accordingly. Preferably also there are filter means for matching an event against a list of event types before entering the event in the queue means. Preferably also the receiver/decoder further includes means for storing at least one application.

A further aspect of the present invention provides a receiver/decoder for receiving broadcast signals, said receiver/decoder comprising:

means for controlling the receiver/decoder in accordance with received signals, said controlling means comprising:

means for receiving signals from a plurality of ports and providing data to said ports; and

virtual machine means for processing data received from the receiving means and returning data thereto.

Another aspect of the present invention comprises a system comprising a plurality of receiver/decoders each as described above, wherein the different receiver/decoders have a common virtual machine means and respective different device manager means coupled to respective different circuitry.

Preferred features of the present invention will now be described, purely by way of example, with reference to the accompanying drawings, in which:-

Figure 1 shows the overall architecture of a digital television system according to the preferred embodiment of the present invention;

Figure 2 shows the architecture of an interactive system of the digital television system;

Figure 3 shows the arrangement of files within a module downloaded into the memory of an interactive receiver/decoder;

Figure 4 is a schematic diagram of interfaces of the receiver/decoder;

Figure 5 shows the arrangement of memory volumes of the memory of the interactive receiver/decoder;

Figure 6 is a functional block diagram of the receiver/decoder; and

Figure 7 shows certain components of the virtual machine and run time engine in more detail.

An overview of a digital television system 1000 is shown in Figure 1. The invention includes a mostly conventional digital television system 2000 which uses the known MPEG-2 compression system to transmit compressed digital signals. In more detail, MPEG-2 compressor 2002 in a broadcast centre receives a digital signal stream (typically a stream of video signals). The compressor 2002 is connected to a multiplexer and scrambler 2004 by linkage 2006. The multiplexer 2004 receives a plurality of further input signals, assembles one or more transport streams and transmits compressed digital signals to a transmitter 2008 of the broadcast centre via linkage 2010, which can of course take a wide variety of forms including telecom links. The transmitter 2008 transmits electromagnetic signals via uplink 2012 towards a satellite transponder 2014, where they are electronically processed and broadcast via notional downlink 2016 to earth receiver 2018, conventionally in the form of a dish owned or rented by the end user. The signals received by receiver 2018 are transmitted to an integrated receiver/decoder 2020 owned or rented by the end user and connected to the end user's television set 2022. The receiver/decoder 2020 decodes the compressed MPEG-2 signal into a television signal for the television set 2022.

A conditional access system 3000 is connected to the multiplexer 2004 and the receiver/decoder 2020, and is located partly in the broadcast centre and partly in the decoder. It enables the end user to access digital television broadcasts from one or more broadcast suppliers. A smartcard, capable of deciphering messages relating to commercial offers (that is, one or several television programmes sold by the broadcast supplier), can be inserted into the receiver/decoder 2020. Using the decoder 2020 and

smartcard, the end user may purchase commercial offers in either a subscription mode or a pay-per-view mode.

An interactive system 4000, also connected to the multiplexer 2004 and the receiver/decoder 2020 and again located partly in the broadcast centre and partly in the decoder, enables the end user to interact with various applications via a modemmed back channel 4002.

Figure 2 shows the general architecture of the interactive television system 4000 of the digital television system 1000 of the present invention.

For example, the interactive system 4000 allows an end user to buy items from on-screen catalogues, consult local news and weather maps on demand and play games through his television set.

The interactive system 4000 comprises in overview four main elements:

an authoring tool 4004 at the broadcast centre (or elsewhere) for enabling a broadcast supplier to create, develop, debug and test applications;

an application and data server 4006 the broadcast centre, connected to the authoring tool 4004 for enabling a broadcast supplier to prepare, authenticate and format applications and data for delivery to the multiplexer and scrambler 2004 for insertion into the MPEG-2 transport stream (typically the private section thereof) to be broadcast to the end user;

a virtual machine including a run time engine (RTE) 4008, which is an executable code installed in the receiver/decoder 2020 owned or rented by the end user for enabling an end user to receive, authenticate, decompress, and load applications into the working memory 2024 of the receiver/decoder 2020 for execution. The engine 4008 also runs resident, general-purpose applications. The engine 4008 is independent of the hardware and operating system; and

a modemmed back channel 4002 between the receiver/decoder 2020 and the application and data server 4006 to enable signals instructing the server 4006 to insert data and applications into the MPEG-2 transport stream at the request of the end user.

The interactive television system operates using "applications" which control the functions of the receiver/decoder and various devices contained therein. Applications are represented in the engine 4008 as "resource files". A "module" is a set of resource files and data. Several modules may be required to make up an application. A "memory volume" of the receiver/decoder is a storage space for modules. An "interface" is used to download modules. Modules may be downloaded into the receiver/decode 2020 from the MPEG-2 transport stream.

The elements mentioned in the previous paragraph are now described in more detail.

For the purposes of this specification, an application is a piece of computer code for controlling high level functions of preferably the receiver/decoder 2020. For example, when the end user positions the focus of a remote controller on a button object seen on the screen of the television set 2022 and presses a validation key, the instruction sequence associated with the button is run.

An interactive application proposes menus and executes commands at the request of the end user and provides data related to the purpose of the application. Applications may be either resident applications, that is, stored in the ROM (or FLASH or other non-volatile memory) of the receiver/decoder 2020, or broadcast and downloaded into the RAM or FLASH memory of the receiver/decoder 2020.

Examples of applications are:-

- An Initiating Application. The receiver/decoder 2020 is equipped with a resident initiating application which is an adaptable collection of modules (this term being defined in more detail hereunder) enabling the receiver/decoder 2020 to be immediately operative in the MPEG-2 environment. The application provides core features which can be modified by the broadcast supplier if required. It also provides an interface between the resident application and downloaded applications.
- A Startup Application. The startup application allows any application, either

downloaded or resident, to run on the receiver/decoder 2020. This application acts as a bootstrap executed on arrival of a service in order to start the application. Startup is downloaded into RAM and therefore can be updated easily. It can be configured so that the interactive applications available on each channel can be selected and run, either immediately after downloading or after preloading. In the case of preloading, the application is loaded into the memory 2024 and is activated by the startup when required.

- A Program Guide. The Program Guide is an interactive application which gives full information about programming. For example, it may give information about, say, one week's television programmes provided on each channel of a digital television bouquet. By depressing a key on the remote controller 2026, the end user accesses an add–on screen, overlaid on the event shown on the screen of the television set 2022. This add–on screen is a browser giving information on the current and next events of each channel of the digital TV bouquet. By depressing another key on the remote controller 2026, the end user accesses an application which displays a list of information on events over one week. The end user can also search and sort events with simple and customised criteria. The end user can also access directly a selected channel.
- A Pay Per View application. The Pay Per View Application is an interactive service available on each PPV channel of the digital TV bouquet in conjunction with the conditional access system 3000. The end user can access the application using a TV guide or channel browser. Additionally, the application starts automatically as soon as a PPV event is detected on the PPV channel. The end user is then able to buy the current event either through his daughter smartcard 3020 or via the communication server 3022 (using a modem, a telephone and DTMF codes, MINITEL or the like). The application may be either resident in the ROM of the receiver/decoder 2020 or downloadable into the RAM of the decoder 2020.
- A PC Download application. On request, an end user can download computer software using the PC download application.
- A Magazine Browser application. The magazine browser application comprises

a cyclic video broadcast of images with end user navigation via on–screen buttons.

- A Quiz application. The quiz application is preferably synchronised with a broadcast quiz programme. As an example, multiple choice questions are displayed on the screen of the television 2022, and the user can select an answer using the remote controller 2026. The quiz application can inform the user whether the answer is correct or not, and can keep count of the user's score.
- A Teleshopping application. In one example of the teleshopping application, offers of goods for sale are transmitted to the receiver/decoder 2020 and displayed on the television 2022. Using the remote controller, the user can select a particular item to buy. The order for the item is sent via the modemmed back channel 4002 to the application and data server 4006 or to a separate sales system the telephone number of which has been downloaded to the receiver/decoder, possibly with an order to debit the account for a credit card which has been inserted into one of the card readers 4036 of the receiver/decoder 2020.
- A Telebanking application. In one example of the telebanking application, the user inserts a bank card into one of the card readers 4036 of the receiver/decoder 2020. The receiver/decoder 2020 dials up the user's bank, using a telephone number stored in the bank card or stored in the receiver/decoder, and then the application provides a number of facilities which can be selected using the remote controller 2026, for example for downloading via the telephone line a statement of account, transferring funds between accounts, requesting a cheque book, etc.
- An Internet Browser application. In one example of the Internet browser application, instructions from the user, such as a request to view a web page having a particular URL, are entered using the remote controller 2026, and these are sent by the modemmed back channel 4002 to the application and data server 4006. The appropriate web page is then included in the transmissions from the broadcast centre, received by the receiver/decoder 2020 via the uplink 2012, transponder 2014 and downlink 2016, and displayed on the television

2022.

Applications are stored in memory locations in the receiver/decoder 2020 and represented as resource files. The resource files comprise graphic object description unit files, variables block unit files, instruction sequence files, application files and data files.

The graphic object description unit files describe the screens, the man-machine interface of the application. The variables block unit files describe the data structures handled by the application. The instruction sequence files describe the processing operations of the applications. The application files provide the entry points for the applications.

The applications constituted in this way can use data files, such as icon library files, image files, character font files, colour table files and ASCII text files. An interactive application can also obtain on-line data by effecting inputs and/or outputs.

The engine 4008 only loads into its memory those resource files it needs at a given time. These resource files are read from the graphic object description unit, instruction sequence and application files; variables block unit files are stored in memory following a call to procedure for loading modules and remain locked there until a specific call to a procedure for unloading modules is made.

With reference to Figure 3, a module 4010, such as a tele-shopping module, is a set of resource files and data comprising the following:

a single application file 4012;

an undetermined number of graphic object description unit files 4014;

an undetermined number of variables block unit files 4016;

an undetermined number of instruction sequence files 4018; and

where appropriate, data files 4020 such as icon library files, image files, character font files, colour table files and ASCII text files.

In the MPEG data stream, each module comprises a group of MPEG tables. Each MPEG table may be formatted as a number of sections. In the MPEG data stream, each section has a "size" of up to 4 kbytes. For data transfer via the serial and parallel port, for example, modules similarly are split into tables and sections, the size of the section varying with the transport medium.

Modules are transported in the MPEG data stream in the form of data packets of typically 188 bytes within respective types of data stream, for example, video data streams, audio data streams and teletext data streams. Each packet is preceded by a Packet Identifier (PID) of 13 bits, one PID for every packet transported in the MPEG data stream. A programme map table (PMT table) contains a list of the different data streams and defines the contents of each data stream according to the respective PID. A PID may alert a device to the presence of applications in the data stream, the PID being identified using the PMT table.

With reference to Figure 4, the receiver/decoder 2020 includes several interfaces; specifically, a tuner 4028 for the MPEG signal flow, a serial interface 4030, a parallel interface 4032, and two card readers 4036, one for a smartcard forming part of the system and one for bank cards (used for making payments, home banking, etc). The receiver/decoder also includes an interface 4034 to a modemmed back channel 4002 to the television signal producer, so that the user can indicate preferences, etc back to the television signal (programme) producer.

A memory volume is a storage space for modules 4010. Such storage spaces are located in the memory 2024 of the receiver/decoder 2020. With reference to Figure 5, the memory 2024 is divided into a RAM volume 4022, FLASH volume 4024, and ROM volume 4026, but this physical organization is distinct from the logical organization. The memory may further be divided into memory volumes associated with the various interfaces through which modules are downloaded into the receiver/decoder 2020, for example an MPEG volume for storing modules downloaded from the MPEG bitstream and a serial volume for storing modules received via a serial interface. From one point of view, the memory can be regarded as part of the

hardware; from another point of view, the memory can be regarded as supporting or containing the whole of the system shown apart from the hardware.

It is expected that the receiver/decoder may be designed and manufactured by various different manufacturers. It may therefore have various different hardware designs, though they will of course all conform to the same functional specification. It is therefore important that a given application behaves in the same way on every receiver/decoder, and that a receiver/decoder should execute all applications in the same, correct manner.

The system can be regarded a centred on a run time engine 4008 forming part of a virtual machine 4007. This is coupled to applications on one side (the "high level" side), and, on the other side (the "low level" side), via various intermediate logical units discussed below, to the receiver/decoder hardware 4061. The receiver/decoder hardware can be regarded as including the various ports or interfaces as discussed above (the interface 2030 for the handset 2026, the MPEG stream interface 4028, the serial interface 4030, the parallel interface 4032, the interfaces to the card readers 4036, and the interface 4034 to the modemmed back channel 4002).

With reference to Figure 6, various applications 4057 are coupled to the unit 4007; some of the more commonly used applications may be more or less permanently resident in the system, as indicated at 4057, while others will be downloaded into the system, eg from the MPEG data stream or from other ports as required.

The unit 4007 includes, in addition to the run time engine 4008, some resident library functions 4006 which include a toolbox 4058. The library contains miscellaneous functions in C language used by the engine 4008. These include data manipulation such as compression, expansion or comparison of data structures, line drawing, etc. The library 4006 also includes information about firmware 4060 in the receiver/decoder 2020, such as hardware and software version numbers and available RAM space, and a function used when downloading a new device 4062. Functions can be downloaded into the library, being stored in Flash or RAM memory.

The run time engine 4008 is coupled to a device manager 4068 which is coupled to a set of devices 4064 which are coupled to device drivers 4060 which are in turn coupled to the ports or interfaces. In broad terms, a device driver can be regarded as defining a logical interface, so that two different device drivers may be coupled to a common physical port. A device driver will normally be coupled to more than one device driver; if a device is coupled to a single device driver, the device will normally be designed to incorporate the full functionality required for communication, so that the need for a separate device driver is obviated. Certain devices may communicate among themselves.

As will be described below, there are 3 forms of communication from the devices 4064 up to the run time engine: by means of variables, buffers, and events which are passed to a set of event queues.

In terms of the designers or providers of these various functions, the applications will be generated by various service (programme) providers. The run time engine will be designed and provided by the system authority or designer. The device manager, the devices, and the device drivers will be provided by the receiver/decoder manufacturer (hardware provider). It will however be realized that this correspondence between the various levels of the receiver/decoder and the three levels of providers will not normally be precise. For example, the system authority will also in practice normally provide some of the applications, the receiver/decoder manufacturer may be involved in the logical device design, and so on.

Each function of the receiver/decoder 2020 is represented as a device 4062. Devices can be either local or remote. Local devices 4064 include smartcards, SCART connector signals, modems, serial and parallel interfaces, a MPEG video and audio player and an MPEG section and table extractor. Remote devices 4066, executed in a remote location, differ from local devices in that a port and procedure must be defined by the system authority or designer, rather than by a device and device driver provided and designed by the receiver/decoder manufacturer.

When a new device 4062 is created, it can be installed in existing receiver/decoders 2020 by downloading the relevant application 4056 from the broadcast centre. This downloading is performed in the receiver/decoder 2020 by an application 4056 which checks the hardware and software versions and, if correct, loads the software module representing the new device 4062 and asks a procedure of the library 4006 to install the new device code within the firmware (in Flash memory). This can provide a flexible and secure installation of new functions within the receiver/decoder 2020 without affecting the rest of the software.

The device manager 4068 is a common software interface between the application 4056 and the specific functions of the receiver/decoder 2020. The device manager 4068 controls access to devices 4062, declares receipt of an unexpected event, and manages shared memory.

The run time engine 4008 runs under the control of the microprocessor and a common application programming interface. They are installed in every receiver/decoder 2020 so that all receiver/decoders 2020 are identical from the application point of view.

The engine 4008 runs applications 4056 on the receiver/decoder 2020. It executes interactive applications 4056 and receives events from outside the receiver/decoder 2020, displays graphics and text, calls devices for services and uses functions of the library 4006 connected to the engine 4008 for specific computation.

The run time engine 4008 is an executable code installed in each receiver/decoder 2020, and includes an interpreter for interpreting and running applications. The engine 4008 is adaptable to any operating system, including a single task operating system (such as MS-DOS). The engine 4008 is based on process sequencer units (which take various events such as a key press, to carry out various actions), and contains its own scheduler to manage event queues from the different hardware interfaces. It also handles the display of graphics and text. A process sequencer unit comprises a set of action-groups. Each event causes the process sequencer unit to move from its current action-group to another action-group in dependence on the

character of the event, and to execute the actions of the new action–group.

As noted above, a logical channel is distinct from a physical port, so that there can for example be two different kinds of logical channel using the same physical port. Events on those two logical channels will be passed to respective different process sequencer units. In addition, there can be two distinct logical channels of the same type using a physical port. In that case, each of the two channels will of course require its own process sequencer unit, so two instances (instantiations) of the same process sequencer unit will be created.

The engine 4008 comprises a code loader to load and download applications 4056 into the receiver/decoder memory 2028. Only the necessary code is loaded into the RAM or Flash memory, in order to ensure optimal use. The downloaded data is verified by an authentication mechanism to prevent any modification of an application 4056 or the execution of any unauthorized application. The engine 4008 further comprises a decompressor. As the application code (a form of intermediate code) is compressed for space saving and fast downloading from the MPEG–2 transport stream or via a built–in receiver/decoder mode, the code must be decompressed before loading it into the RAM. The engine 4008 also comprises an interpreter to interpret the application code to update various variable values and determine status changes, and an error checker.

The main loop of the engine 4008 extracts events from the queue means and actuates the associated process sequencer units accordingly. At every traverse of the main loop, a procedure is called up to receive external events (such as pressure on one of the remote control keys, reception of an MPEG–2 section or a message on a serial port) from the event interface. All events, whether input to the presentation function (discussed below) or received through an interface, pass through an event interface before being processed by the engine 4008. For each detected event, the procedure comprises a message, called an "event", which is put into one of 5 queues of the engine 4008, with respective priority levels 0 to 4. After the call to the event interface, which fills the queues with events, the engine 4008 searches the queues in

order of decreasing priority (from level 4 to level 0) for an event. The event so found is removed from the queue and used to actuate the process sequencer unit for which it is intended.

Before using the services of any device 4062, a program (such as an application instruction sequence) has to be declared as a "client", that is, a logical access–way to the device 4066 or the device manager 4068. The manager gives the client a client number which is referred to in all accesses to the device. A device 4066 can have several clients, the number of clients for each device 4066 being specified depending on the type of device 4066. A client is introduced to the device 4066 by a procedure "Device: Open Channel". This procedure assigns a client number to the client. A client can be taken out of the device manager 4068 client list by a procedure "Device: Close Channel".

The access to devices 4062 provided by the device manager 4068 can be either synchronous or asynchronous. For synchronous access, a procedure "Device: Call" is used. This is a means of accessing data which is immediately available or a functionality which does not involve waiting for the desired response. For asynchronous access, a procedure "Device: I/O" is used. This is a means of accessing data which involves waiting for a response, for example scanning tuner frequencies to find a multiplex or getting back a table from the MPEG stream. When the requested result is available, an event is put in the queue of the engine to signal its arrival. A further procedure "Device: Event" provides a means of managing unexpected events.

As noted above, the main loop of the run time engine is coupled to a variety of process sequencer units, and when the main loop encounters an appropriate event, control is temporarily transferred to one of the process sequencer units.

Communication between the run time engine and the applications is also performed largely via two types of process sequencer units, graphic object description unit process sequencer units and an instruction sequence process sequencer unit.

The essentials of the operation of the process sequencer units can best be described by considering the operation of the graphic object description unit and instruction sequence process sequencer units; the operation of the other process sequencer units is similar. The graphic object description unit process sequencer units manage the man–machine interface, and the instruction sequence process sequencer unit executes instruction sequences in response to instruction sequence execution requests received from the graphic object description unit process sequencer unit. (An instruction sequence is a sequence of decompressed intermediate code commands.)

A presentation function communicating with the engine 4008 administers the presentation of text and graphics to the end user, and the presentation of end user actions to the engine 4008. The text and graphics are overlaid on the display on the television set 2022, and the user may interact with the application 4056 by means of a keyboard (using that term to include the remote controller 2026).

A graphic object description unit process sequencer unit receives query events from the television screen of the end user, and processes such a query by starting to read the corresponding interface file 4014. It then uses the graphic functions of the interface to trace the graphic object on the screen of the television set 2022. The end user can use the four arrow keys of the remote control 2026 to move around the graphic object. Every time that a key is pressed, an event is processed by the graphic object description unit process sequencer unit. When the user validates a choice using the VALID key on the remote control 2026, the code of the event is processed by the graphic object description unit process sequencer unit, which generates events requesting instruction sequences to be run and sends them to the instruction sequence process sequencer unit.

When the instruction sequence process sequencer unit receives a instruction sequence execution request from the graphic object description unit process sequencer unit, it reads the corresponding instruction sequence file 4018 and loads it entirely into memory. It then starts running the instruction sequence and continues until the instruction sequence is finished or until it encounters a function called rerouting.

When the instruction sequence is finished, the system returns to cycle through the main loop for acquiring events and rescheduling.

There are also several downloading process sequencer units whose function is to process the various protocols linked to the different volumes containing applications 4056.

Referring to Fig. 7, the device manager includes a queue 100 into which events from the devices are passed for temporary storage. At suitable intervals, the virtual machine sends a signal to this queue to extract the first item from it. This event item is moved to a queue structure 101 in the virtual machine. Depending on the priority level of the event item, it is inserted into the appropriate one of the 5 queues 0 to 4. Event items are extracted from the queue structure 101 by a queue selector unit 102 under the control of the run time engine.

When an event is selected from the queue structure 101, it is passed to a process sequencer unit engine 104, which consists of a process sequencer unit driver 105 and a set of process sequencer units 106. Each process sequencer unit is a set of action-groups linked together, so that each step from one action-group to the next action-group is, in general, dependent on the current action-group and the nature of the event. Different process sequencer units have different sizes and complexities, including one in which the "next" action-group, ie the action-group to which the system steps on in response to an event, is dependent solely on the nature of the event but is independent of the current action-group. Also, as is shown at the right-hand side of the process sequencer units block, there may be several copies of a process sequencer unit, ie several identical process sequencer units, to deal eg with several separate data streams using identical protocols through a single port.

When an event is selected, it is passed to the appropriate process sequencer unit. This selects the appropriate outlet from the current action-group on the process sequencer unit. This results in the appropriate next action-group being selected and the actions in that action-group being performed, involving eg the sending of a

message to the device manager or the execution of a instruction sequence. Action-groups in the process sequencer unit can also send event messages to other process sequencer units.

If a instruction sequence is selected, the identification of the instruction sequence is sent to a instruction sequence selector 107. This obtains the desired instruction sequence from a instruction sequence memory 108 and passes it to a instruction sequence interpreter 109, which executes the instruction sequence.

The system also includes a filter 110, which is loaded with event types eg from the process sequencer units 106. When an event item is passed from the queue 100 in the device manager to the queue structure 101 in the virtual machine, its type or character is matched against the list in the filter 110, and if it is of a type which is not recognized, it is rejected. This ensures that if say the device manager or the keyboard generates events of a type which the virtual machine cannot deal with, those events are not passed to the queue structure 101. (If events of this kind were passed to the queue structure 101, either they would accumulate in that queue structure or they might cause malfunctioning of the process sequencer unit engine 104.)

It will be understood that the present invention has been described above purely by way of example, and modifications of detail can be made within the scope of the invention.

Each feature disclosed in the description, and (where appropriate) the claims and drawings may be provided independently or in any appropriate combination.

In the aforementioned preferred embodiments, certain features of the present invention have been implemented using computer software. However, it will of course be clear to the skilled man that any of these features may be implemented using hardware. Furthermore, it will be readily understood that the functions performed by the hardware, the computer software, and such like are performed on or using electrical and like signals.

Cross reference is made to our co-pending applications, all bearing the same filing date, and entitled Signal Generation and Broadcasting (Attorney Reference no. PC/ASB/19707), Smartcard for use with a Receiver of Encrypted Broadcast Signals, and Receiver (Attorney Reference No. PC/ASB/19708), Broadcast and Reception System and Conditional Access System therefor (Attorney Reference No. PC/ASB/19710), Downloading a Computer File from a Transmitter via a Receiver/Decoder to a Computer (Attorney Reference No. PC/ASB/19711), Transmission and Reception of Television Programmes and Other Data (Attorney Reference No. PC/ASB/19712), Downloading Data (Attorney Reference No. PC/ASB/19713), Computer Memory Organisation (Attorney Reference No. PC/ASB/19714), Television or Radio Control System Development (Attorney Reference No. PC/ASB/19715), Extracting Data Sections from a Transmitted Data Stream (Attorney Reference No. PC/ASB/19716), Access Control System (Attorney reference No. PC/ASB/19717), Data Processing System (Attorney Reference No. PC/ASB/19718), and Broadcast and Reception System, and Receiver/Decoder and Remote Controller therefor (Attorney Reference No. PC/ASB/19720). The disclosures of these documents are incorporated herein by reference. The list of applications includes the present application.

THIS SHEET HAS BEEN REPLACED

## CLAIMS

1. A receiver/decoder for a digital television system, comprising:

means for receiving a compressed MPEG-type signal;

means for decoding the received signal to provide a television signal; means for supplying the television signal to a television;

a user input interface; and

a computer system for controlling the receiver/decoder in accordance with input signals received through the user input interface, the computer system comprising: device manager means for receiving signals from a plurality of ports and providing data to said ports; and

virtual machine means, coupled to the device manager means, comprising means for processing data received from the device manager means and returning data thereto.

2. A receiver/decoder according to claim 1 wherein the virtual machine means comprise an operating engine, a library of routines, an interpreter, storage means for storing a plurality of instruction sequences, queue management means, and buffer management means.

3. A receiver/decoder according to claim 1 or 2, further comprising:

a plurality of interfaces for coupling to external units;

said device manager means comprising at least one device means associated with either at least one interface or with at least one function of the receiver/decoder.

4. A receiver/decoder according to claim 3 wherein at least one of said device means coupled to at least one device driver.

5. A receiver/decoder according to claim 3 or 4 including a plurality of application sources each for providing an application for controlling the receiver/decoder and/or the television, and wherein the operating engine is located

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

between the application sources and the device means, and includes queue means for receiving events generated through the device manager means and passing them to the engine.

6. A receiver/decoder according to claim 5 wherein the operating engine comprises a plurality of process sequencer units associated with the device means, and means for extracting events from the queue means and actuating the associated process sequencer units accordingly.

7. A receiver/decoder according to either claim 5 or 6 including filter means for matching an event against a list of event types before entering the event in the queue means.

8. A receiver/decoder according to any previous claim, further including means for storing at least one application.

9. A receiver/decoder for receiving broadcast signals, said receiver/decoder comprising:

means for controlling the receiver/decoder in accordance with received signals, said controlling means comprising:

means for receiving signals from a plurality of ports and providing data to said ports; and

virtual machine means for processing data received from the receiving means and returning data thereto.

10. A system comprising a plurality of receiver/decoders each according to any of claims 1 to 8, wherein the different receiver/decoders have a common virtual machine means and respective different device manager means coupled to respective different circuitry.

11. A receiver/decoder substantially as herein described.

This Sheet Has Been Replace

THIS SHEET HAS BEEN REPLACED

12. A system substantially as herein described.

This Sheet Has Been Replaced