

(12) UK Patent Application (19) GB (11) 2 375 674 (13) A

(43) Date of A Publication 20.11.2002

(21) Application No 0111850.4

(22) Date of Filing 15.05.2001

(71) Applicant(s)
British Broadcasting Corporation
(Incorporated in the United Kingdom)
Broadcasting House, LONDON, W1A 1AA,
United Kingdom

(72) Inventor(s)
Andrew Kevin McParland
Timothy John Sargeant
Guy Quentin Winter
John Christopher William Newell
Hywel James Williams
Nigel C Earnshaw

(74) Agent and/or Address for Service
Mathys & Squire
100 Grays Inn Road, LONDON, WC1X 8AL,
United Kingdom

(51) INT CL⁷
H04N 7/088 5/455

(52) UK CL (Edition T)
H4F FBBG

(56) Documents Cited
GB 2325537 A **WO 2000/040021 A1**
WO 2000/016548 A1 **US 6005597 A**

(58) Field of Search
UK CL (Edition S) **H4F FBB**
INT CL⁷ **H04N 5/455 7/088**
Online: EPODOC, WPI, PAJ, INSPEC

(54) Abstract Title
Virtual Media Channel

(57) A method and apparatus for creating a viewing schedule for a user. Broadcast programs from a number of sources, such as terrestrial television, the internet or DVDs can be compiled into a broadcast schedule which is 'tailored' to a user, or a group of users. By allowing programmes to be played live, pre-recorded, or time delayed, a virtual channel can be created based upon user preference information. The schedule may indicate the source, timing and other information such as the cost of a programme, and may allow a user to modify the proposed schedule, and the stored user preference information.

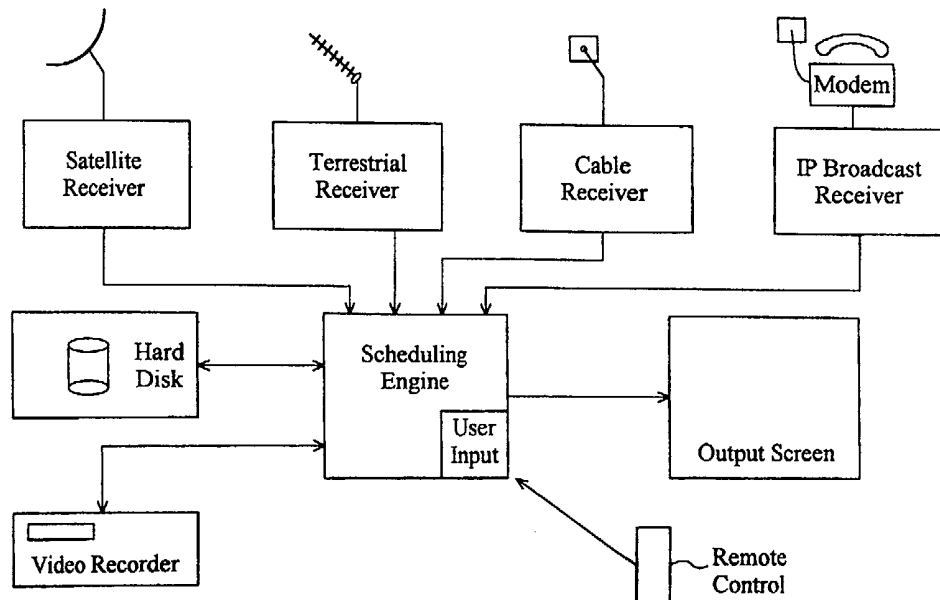


Fig. 2

The specification as filed includes a computer program which is not reproduced here; it may be inspected in accordance with Section 118 of the Patents Act 1977.

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

GB 2 375 674 A

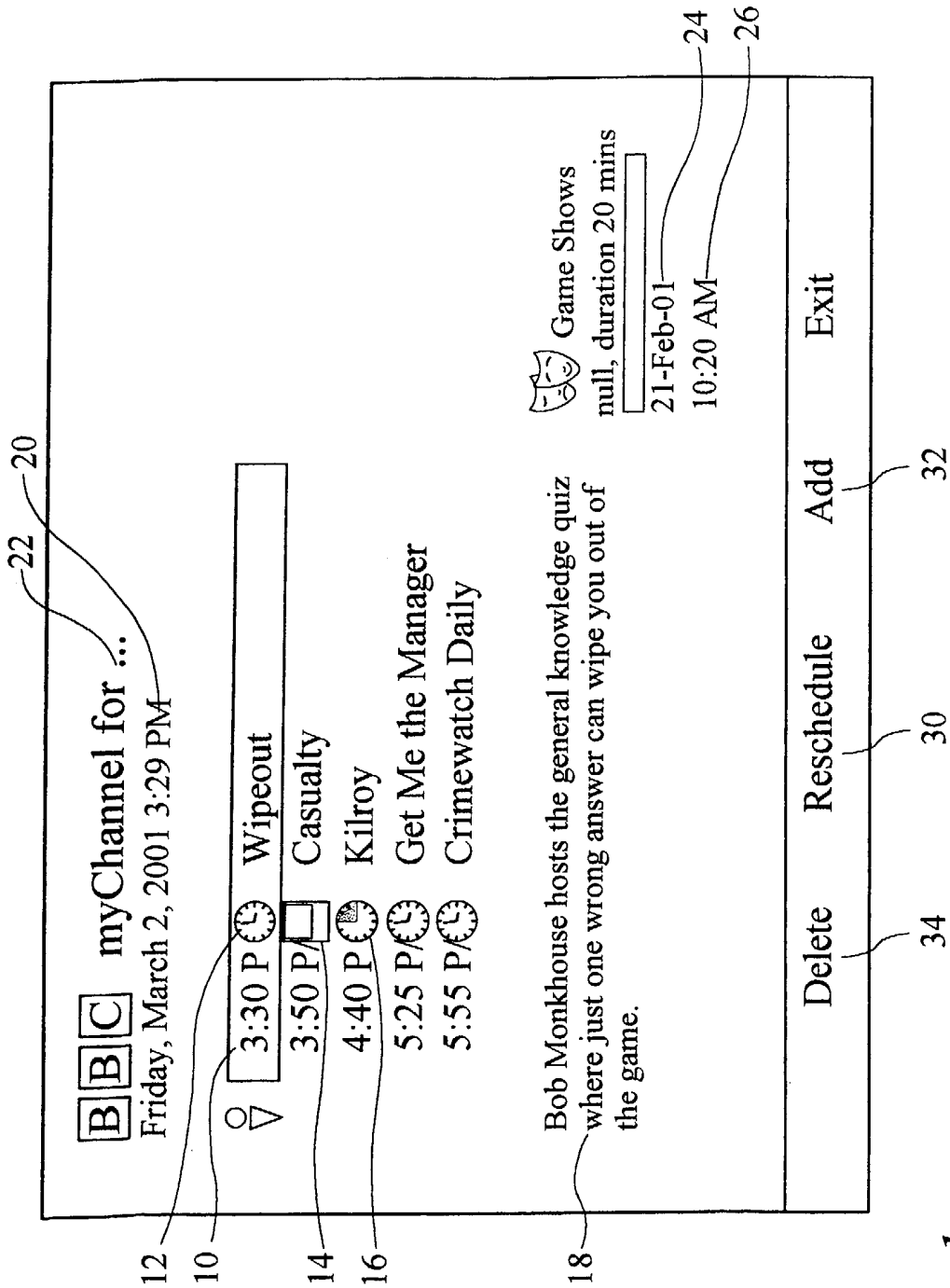


Fig. 1

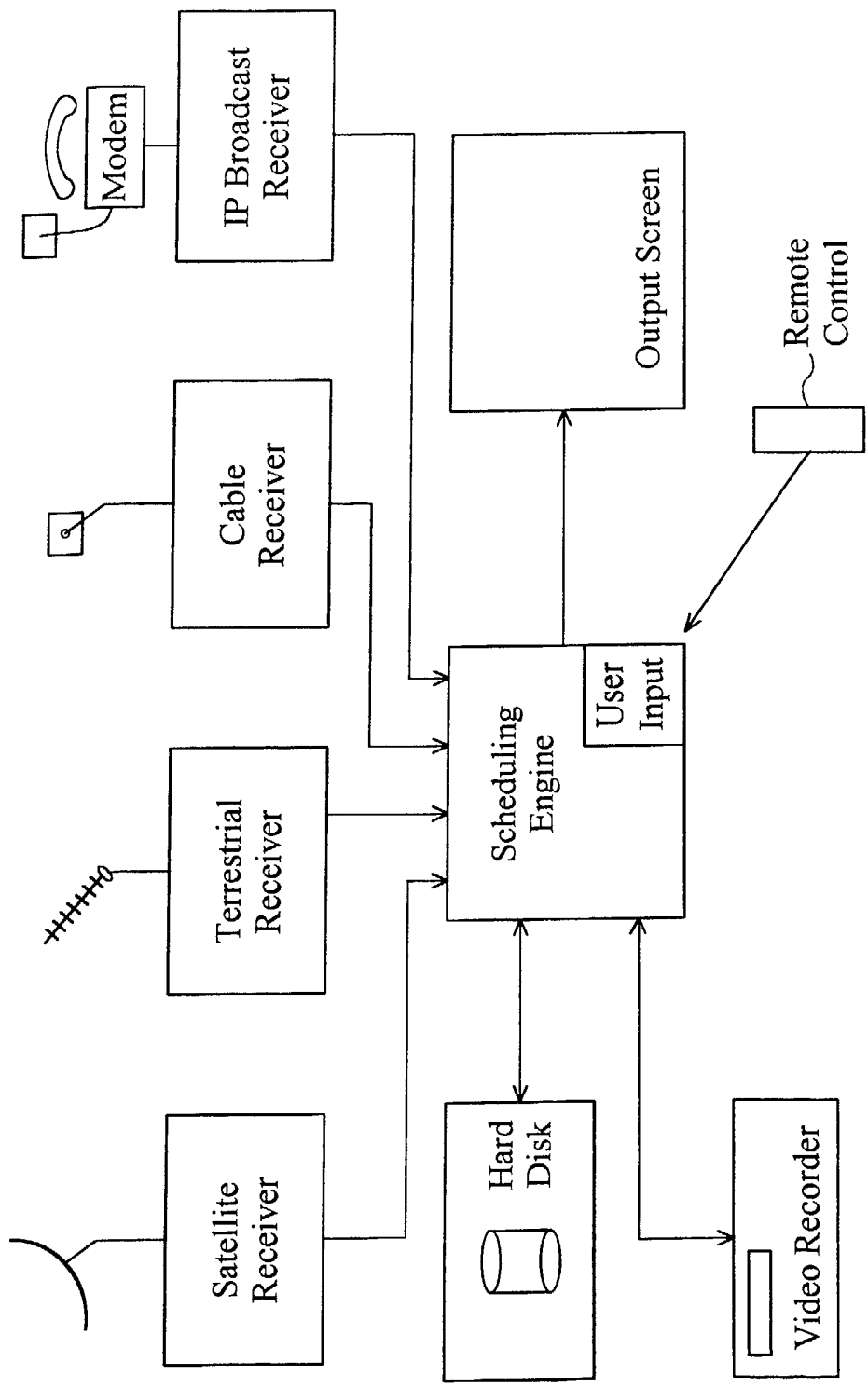


Fig. 2

Virtual Media Channel

The present invention relates to delivery of a schedule of media programmes to a user. The invention is particularly, but not exclusively, concerned with delivery
5 of television programmes but may be applied to delivery of audio only (for example radio programmes) or multi-media presentations or other interactive entertainment and the term media programme as used herein is intended to encompass any form of audio and/or visual experience delivered to a user.

10 With the advent in recent years of cable, satellite and digital television, the number of channels available to a user has greatly increased. Whereas some users may be prepared to watch whatever is available or may be prepared to organise their personal schedules round the timing of television programmes, others may be more selective and/or may wish to view television only at certain limited times and
15 may find that, despite the large number of channels available, there may be nothing of interest to them at the time they wish to view it.

Various methods and devices have been proposed to alleviate this problem. For many years, video recorders have been used to record programmes for later
20 viewing at a more convenient time. More recently, digital storage devices have been produced which can continuously record many hours of programme transmission and enable time-shifting of programmes without advance programming. Such devices may also contain menus and may store user preferences in order to facilitate navigation through stored programmes or to select
25 what is recorded.

US-A-5534911 discloses a device which evaluates a number of broadcast programmes and selects the "best" programme from the available for viewing and may also record the "best" programme of the day or week.

30 With tools such as the above and appropriate planning, a user can select from available programmes and decide what to watch.

However, the present invention stems from the realisation that, whilst such tools can be convenient for selecting a particular programme which the user wants to watch, in order to provide viewing for a given time period, a user may have to make numerous decisions and calculations and may end up compiling a sub-optimal
5 collection of programmes to view in that period. Historically, efforts have been put into creating schedules of programmes for general viewing on a particular channel and all the benefits of the efforts of scheduling programmes are lost when tools such as the above are employed to time-shift programmes.

10 In a first aspect, the invention provides a method for creating a viewing schedule for a user comprising:

receiving a schedule of broadcast programmes;

receiving a list of stored programmes;

storing user preference information;

15 compiling a schedule of programmes for the user selected from the schedule of broadcast programmes and the list of stored programmes, based on the user preferences.

Pursuant to the present invention it has been appreciated that conventional
20 apparatus may assist in providing a greater variety of programmes from which to choose and tools for doing so but cannot effectively provide a coherent schedule of programmes for viewing which can include both "live" and stored programmes. Furthermore, the invention stems from the realisation that whilst certain programmes may readily be recorded and time-shifted, a user may derive greater benefit from
25 viewing certain other programmes "live" that is at (or close to) the time at which they were intended to be viewed and conventional time-shifting devices, by their very nature which is intended to make, are not suited to this.

In contrast, with the invention, a personalised schedule can be created which
30 integrates both stored programmes and broadcast (which term is intended to encompass any distribution format, including, but not limited to, digital or analogue broadcasting, satellite or cable broadcasting and internet distribution) programmes

and can take into account the time of broadcast of programmes.

The method may further comprise playing out media corresponding to the schedule.

5

In a closely related method aspect, the invention provides a method of providing a media channel which presents a sequence of programmes, the method comprising:

receiving media programmes from a plurality of sources including at least one
10 source of recorded programmes and at least one source of broadcast programmes
and presenting a sequence of programmes selected from the received programmes
based on user preferences.

Preferably, the method includes selecting at least one programme from the
15 schedule of broadcast programmes to play back at a time other than the time of
broadcast of said programme. The method may include controlling a time-shifting
recording device to play back the selected programme at the desired time. The time
shifting device may be separate or integrated with the schedule generator.

20 Preferably, the method includes indicating on the schedule programmes
which are live. Preferably, the method further includes indicating programmes which
are pre-recorded. Preferably, the method further includes indicating programmes
which are time-shifted.

25 Video on demand programmes may be treated as pre-recorded programmes.
References to stored or pre-recorded programmes are intended to connote
programmes which can be accessed on demand by a user; these may be locally
stored (for example on a hard disk or video recorder) or may be available from a
remote video on demand server. The term is intended to distinguish from broadcast
30 programmes (i.e. to exclude programmes whose playback time cannot be selected
by the user).

Preferably, the method includes presenting the schedule to the user and modifying the schedule in response to user input. This may allow a personalised schedule to be produced and then tailored to the user's requirements; this may allow the user flexibility but present the user with a reasonable schedule without requiring direct user input for a particular schedule. Modifying may include deleting programmes. This allows programmes to be permanently deleted from the schedule. Modifying may include changing the order of programmes in the schedule. Modifying may include playing an available programme immediately in response to user selection and modifying the time of play of subsequent programmes.

Most preferably, information concerning user preferences is updated in response to user modification of a prepared schedule. In this way, the system can "learn" from the user's reaction to a given schedule.

The schedule may be prepared in advance of the start of the schedule. Alternatively, the schedule may be prepared starting immediately or imminently from the current time.

The schedule may be set to run for a predetermined period, for example an evening or a specified user time.

The schedule may include a planned break for the user to undertake other activity.

The schedule may include programmes which are not directly accessible by the scheduling device. For example cable, satellite or radio programmes or an internet live video broadcast may be included in the schedule, even if the user has to operate separate equipment to view the programmes. Thus the scheduler may be separated from the receiving equipment and may only receive a schedule but not the programme itself for certain media or programme sources. Similarly, the scheduler may include details of stored programmes (for example on video tape or

DVD) and may insert these in the schedule, without necessarily being able to play the programmes directly. Preferably, however, the scheduler is integrated with receiving apparatus for receiving at least one broadcast media channel and preferably also has means for time-shifting a received programme.

5

The method may include storing multiple user preferences and may include providing multiple user schedules. Alternatively the method may include providing a composite schedule in which programmes are selected according to different user preferences at different times. For example, a daytime schedule may be provided based on one user preference and an evening schedule may be provided based on another user preference and a yet further weekend schedule may be provided. Different user preferences may be merged; for example if two (or more) users who have individual preferences wish to view a schedule, programmes may be selected according to both (or all) users' preference information, in order to provide a selection suitable for both (or all) users.

In a greatly preferred implementation, the method comprises at least a step of ordering programmes according to time of broadcast or availability and a step of ordering the programmes according to closeness of match to user criteria. The ordering according to time is preferably performed as a first step. The ordering step is preferably performed so that pre-recorded programmes can be inserted at any convenient point, after the broadcast programmes have been ordered.

The method may be configured so that one or other source is favoured over another; this may be based on user preferences and/or time of day or week. Preferences may be expressed between broadcast channels and as between broadcast and pre-recorded programmes. For example, a hierarchy of preference may be specified as "BBC1, then other terrestrial, then pre-recorded, then satellite then cable" and this may be used to weight other parameters used in ranking programmes.

The invention extends to apparatus for performing any of the above methods.

The invention further extends to a computer program or computer program product arranged to perform any of the above method aspects.

5 The invention further extends to a schedule or sequence of commands to effect playback of a schedule produced by any of the above methods.

An embodiment of the invention will now be described by way of example, with reference to the accompanying drawings in which:-

10 Fig. 1 shows a screen shot of a schedule produced by the invention; and Fig. 2 is a schematic overview of the apparatus.

Overview of the invention

The embodiment provides a user with a personalised schedule of available
15 programmes. This personalised schedule can be presented to the user as a listing of programmes in time order, similar to what would be on offer from a traditional broadcast channel. The schedule is arranged to contain the best content for that user. On initialisation, when no programmes are stored, this would be a choice of the live broadcasts. With the availability of storage (such as a hard disk in a set-top
20 box or a video-on-demand server) the schedule can incorporate both live programmes as they are broadcast, time-shifted programmes to fit or tidy up the schedule, and those available on-demand (e.g. pre-recorded). An important feature is that the schedule can contain both live and recorded content as it is expected the user will still want to watch certain events live or near-live (e.g. a news or current
25 affairs programme, or an episode of a series; such programmes can be identified and flagged as programmes to be included at or as close to the "normal" broadcast time as possible).

Advantageous features of the embodiment include:-

Personalised service

An agent or filtering tool chooses the most appropriate content for the selected user. This preference could be based on a combination of past choices by the user and related to the genre and type of content that the user watches. It may also be based
5 on their preferences for watching particular types of programme at different times of day, e.g. a movie or drama between 8pm and 10pm. The aim of the personalisation is to give the user a choice amongst the best that is on offer.

Virtual channel

The schedule would be virtual in the sense that it is not based on an existing
10 broadcast schedule of programmes, but instead contains content from many different sources that was made available earlier. The schedule has been created in the box for the user. This means that it need not be stored or exist elsewhere, reducing distribution and storage resource requirements.

Adaptive

15 The schedule presented to the user can adapt to the time of day, in addition to personalisation to the user's preferences. As time moves on the schedule will change.

Planned

20 There is an element of planning to the virtual schedule. The user can add or delete programmes from the proposed schedule. The user can also look ahead and plan their evening or weekend viewing. In addition, in contrast, the user can watch a schedule without prior planning of what to record or any effort in selecting programmes.

Content agnostic

The schedule does not just have to be conventional TV material (i.e. Audio/Video), it may be audio-only material. It may also be pointers to material that the box itself cannot acquire but has information on. For example, the box may have programme
5 information for a radio station that it cannot tune to. The box notices that there is a concert by the user's favourite band and flags the event in the user's schedule. The delivery mechanism may be via traditional broadcast channels (e.g. via a teletext or Ceefax service) or via a broadband connection or phone-line or other internet connection. The mechanism for "delivering" a list of pre-recorded material
10 may include storing details of what has been recorded under the control of the scheduler and may also include inputting, either from a user or from a separate device, details of pre-recorded material available to view.

Specific Implementation

15 Referring to Fig. 1, a screen shot from an embodiment is shown.

The schedule displayed shows the time of a programme 10, an icon indicating whether the programme is live 12, pre-recorded 14 or (marginally) time-shifted 16 and gives a summary 18 of the highlighted programme. The display also
20 shows the date and time for which the schedule is prepared and has space to display the name of the user for whom it has been prepared. The display further shows the date on which the schedule has been prepared (the current date) 24 and time 26. The schedule includes inputs for rescheduling the currently highlighted event 30, adding an event 32 and deleting an event 34.

25

To summarise, the key advantages of this arrangement are:-

Simple layout – uncluttered and direct.

Informative – the "liveness" of a programme is indicated by icons, including straight
30 from disk, time shifted and live.

Personalised – the person or group that the screen is intended for is indicated directly.

The following sets out preferred criteria and constraints for implementing the scheduler.

5 Virtual Channel Behaviour

The virtual channel schedule is a short, ordered list of programmes that are time/availability ordered, then ordered in terms of a rating that reflects the quality of the match between the programme and the user preference information.

10

When the virtual channel is selected the programmes are played out sequentially.

The schedule will scroll upwards as each programme is replayed and the order of programmes will not change.

15

If the virtual channel is not selected then the order of the programmes may be revised as highly rated programmes become more immediately available and displace programmes earlier in the schedule.

20 The rating of programmes that remain in the schedule for some time but remain unwatched is progressively reduced so that ultimately they will be displaced by more recently broadcast content.

25 The first programme in the schedule preferably cannot be displaced once replay has been started.

User modifications

The user can preferably modify the schedule in one or more of the following ways:

30 programmes may be removed. In this case they will preferably never reappear in the schedule.

- programmes may be moved about in the schedule, subject to time/availability constraints.

- programmes can be added to the schedule, subject to time/availability
5 constraints.

- the user may choose to replay a programme further down the schedule. In this case it is immediately moved to the top of the schedule.

10 When a programme is added or removed from the schedule this is taken to indicate a user preference and can be used to refine the user profile information.

If a virtual channel with a modified schedule is continuously selected the order of programmes will not change and the behaviour is identical to the case when the
15 schedule is unmodified.

However, if the schedule is not played out the dynamic behaviour is preferably different to an unmodified schedule in at least the following aspects:

- added programmes will not be displaced from the schedule until watched or
20 removed by the user.

- the ordering of added programmes is preserved as they move forward in the
25 schedule.

- the ordering of moved programmes is preserved but they may be displaced
by higher rated programmes as they become available.

Schedule revisions

30 The virtual channel schedule may be revised in the following circumstances:

- the virtual channel is playing and the current programme finishes.

- the user chooses to replay a programme that is later in the schedule.
- the user removes, adds or moves a programme.
- 5 · a new programme becomes available (a recording starts).
- the tuner is not available when required so that the capture of a programme fails.

10 It will be apparent that numerous modifications to the above may be made and the above embodiment is illustrative only. By way of example only, Java code implementing an embodiment of the invention is appended hereto. Other than the right to make facsimile copies of the printed version of this patent application, all copyright in the code is reserved by the applicant.

15

Appendix - sample Java source code

```
package bbc.rd.navigator;

5  import java.util.*;
   import java.io.*;
   import mytv.resolution.*;

   /**
10  * This Class represents a schedule for a Virtual Channel.
   *
   * <p> All virtual channel schedule objects should be obtained using the following method:
   * <br>&nbsp; public static VirtualChannelSchedule BBCNavigatorXlet.getVirtualChannelSchedule();
   *
15  * <p>Created: 29/1/01
   * <br>Last edited: 23/2/01
   *
   * <p>Outstanding issues: None
   *
20  * @version 1.0
   * @author Chris Newell - BBC Research & Development
   */

   public class VirtualChannelSchedule implements Runnable {
25
       private static final int scheduleLength = 5;
       private VirtualChannelProgram[] schedule;
       private VirtualChannelProgram[] contentPool;
       private UserProfile userProfile;
30  private Thread builderThread;
       private Date buildDate;

       //-----
       /**
35  * Constructor.
       */
```

```
public VirtualChannelSchedule() {

    userProfile = BBCNavigatorXlet.getUserProfile();
    // Load the last content pool and virtual channel schedule from persistent storage.
5    loadSchedule();

    // Start the content pool revision thread.
    builderThread = new Thread(this, "VirtualChannelBuilder");
    builderThread.start();
10 }

//-----
/**
 * Create a dummy schedule for test purposes.
 */
15 public synchronized void dummySchedule() {

    contentPool = new VirtualChannelProgram[8];
    Date date;

    schedule = new VirtualChannelProgram[scheduleLength];
20    userProfile = BBCNavigatorXlet.getUserProfile();

    date = new Date(101, 1, 21, 9, 0);

    contentPool[0] = new VirtualChannelProgram("Kilroy", "Robert Kilroy-Silk hosts topical debate and
discussion with his studio audience.", "News and Documentaries:Features", "BBC1", 45, date, 2, 17);
25

    date = new Date(101, 1, 21, 10, 20);

    contentPool[2] = new VirtualChannelProgram("Wipeout", "Bob Monkhouse hosts the general knowledge
quiz where just one wrong answer can wipe you out of the game.", "Entertainment:Game Shows", "BBC2", 20,
date, 3, 24);
30

    date = new Date(101, 1, 21, 11, 30);

    contentPool[3] = new VirtualChannelProgram("Passport to the Sun", "A look at Mallorca during the
summer months, presented by Nadia Sawalha. Nadia visits the animal refuge centre and there's a happy event
on board the Emerald as one couple have their wedding blessed.", "Entertainment:Travel", "BBC1", 45, date, 1,
35 9);
```



```
date = new Date(101, 1, 21, 12, 0);
```

```
contentPool[4] = new VirtualChannelProgram("BBC News", "The latest national and international news  
from the BBC this lunchtime. Followed by Weather.", "News and Documentaries:News", "BBC1", 20, date, 1, 7);
```

```
5 date = new Date(101, 1, 21, 12, 0);
```

```
contentPool[6] = new VirtualChannelProgram("Get Me the Manager", "Transport: Tonight we meet the  
ladies who have taken on Heathrow Airport. We also go behind the scenes at the Oxford Bus Company, and visit  
the Isle of Skye's expensive toll bridge.", "Entertainment:Factual", "BBC2", 30, date, 2, 17);
```

```
10 date = new Date(101, 1, 21, 12, 30);
```

```
contentPool[5] = new VirtualChannelProgram("Neighbours", "Flick decides to pursue Joel whether he likes  
it or not, and Susan questions Darcy about the seriousness of his problems with Alice. Will Lou really evict the  
boys at Number 30?", "Entertainment:Soaps", "BBC1", 30, date, 0, 0);
```

```
15 date = new Date(101, 1, 21, 15, 15);
```

```
contentPool[1] = new VirtualChannelProgram("Crimewatch Daily", "The live daily crime prevention show  
presented by Phil Gayle and Jane Moore behind the scenes at New Scotland Yard.", "News and  
Documentaries:Features", "BBC1", 30, date, 2, 16);
```

```
20 date = new Date(101, 1, 21, 17, 10);
```

```
contentPool[7] = new VirtualChannelProgram("Casualty", "Something from the Heart: After a major hotel  
explosion, Charlie finds himself treating a high profile patient with controversial views on the NHS.",  
"Entertainment:Drama", "BBC1", 50, date, 2, 19);
```

```
25 // Build a dummy schedule to prevent null pointer exceptions
```

```
for (int i=0; i<scheduleLength; i++) {
```

```
    schedule[i] = contentPool[i];
```

```
}
```

```
}
```

```
30 //-----
```

```
/**
```

```
 * Save the Virtual Channel Schedule to persistent storage.
```

```
*/
```

```
public synchronized void saveSchedule() {
```

```
35
```

```
    String root = System.getProperty("persistent.root");
```

```
// Use the user profile name as the first part of file name.
String fileName = userProfile.getUserName() + "_vcs";

try {
5   FileOutputStream out = new FileOutputStream(new File(root, fileName));
   ObjectOutputStream s = new ObjectOutputStream(out);
   s.writeObject((Object)buildDate);
   s.writeObject((Object)schedule);
   s.writeObject((Object)contentPool);
10  s.flush();
   out.flush();
   s.close();
   out.close();

   System.out.println("Content pool and virtual channel schedule saved successfully to local disk.");
15  } catch (Exception e) {
   e.printStackTrace();
}

for (int i=0; i<schedule.length; i++) {
20  System.out.println(i + " " + schedule[i].title + " locked = " + schedule[i].locked);
}
}

//-----
/**
25  * Load the Virtual Channel Schedule from persistent storage.
*/
public synchronized void loadSchedule() {

30  String root = System.getProperty("persistent.root");

   // Try to load the contentPool and schedule for the current user from a persistent file.
   String fileName = userProfile.getUserName() + "_vcs";

try {
35  FileInputStream in = new FileInputStream(new File(root, fileName));
   ObjectInputStream stream = new ObjectInputStream(in);
```

```
    buildDate = (Date)stream.readObject();
    schedule = (VirtualChannelProgram[])stream.readObject();
    contentPool = (VirtualChannelProgram[])stream.readObject();
    stream.close();
5    in.close();

    System.out.println("Content pool and virtual channel schedule read successfully from local disk");
    // If file loads correctly then call reviseSchedule.
    reviseSchedule();
    } catch (Exception e) {
10    e.printStackTrace();
    // If loading fails.
    }
    }
//-----
15    /**
    * Get the number of programs in this virtual channel schedule.
    */
    public synchronized int getScheduleLength() {
        return schedule.length;
20    }
//-----
    /**
    * Get the nth program in this virtual channel schedule.
    */
25    public synchronized VirtualChannelProgram getProgram(int n) {
        return schedule[n];
    }
//-----
    /**
30    * Add a program at the nth position in this virtual channel schedule.
    * <br> The new programme will be "locked" at this position. This means that it will not be removed until it has
    been viewed or deleted by the user.
    * <br> The program currently at this position and those that follow are shifted upwards and may be replaced
    by higher rated programmes if not locked.
35    */
    public synchronized int add(ContentReference crid, int position) {
```

```
// Ignore invalid position numbers.
if(position < 0 || position >= scheduleLength) {return -1;}

// Get new virtual channel programme. ***** May need to deal with failure at this point.
5 VirtualChannelProgram prog = new VirtualChannelProgram(crid);

// Check that this programme will be available at this position/time.
if(prog.expectedStartTime.after(new Date(getStartTime(position)))) {return -1;}

10 // Ensure the programme at the top of the schedule is unlocked before it is discarded.
    schedule[scheduleLength-1].locked = false;

// Shift back the programmes after the inserted programme.
for (int i=scheduleLength-1; i>position; i--) {
15     schedule[i] = schedule[i-1];
}

// Add the new programme and lock its position;
schedule[position] = prog;
20 schedule[position].locked = true;

// Optimise re-ordering (locked programmes will remain in the schedule, unlocked programmes may be
replaced by higher rated programmes).
    reviseSchedule();
25

// Assume this user action indicates approval and feed the information back to the user profile builder.

BBCNavigatorXlet.userAction(schedule[position].crid, true);

30 return 0;
}

//-----
// /**
// * Replace a program at the nth position in this virtual channel schedule.
35 // * <br>This overwrites the program currently at this position.
// */
```

```
// public synchronized void replace(ContentReference crid, int n) {  
//  
// }  
//-----  
5  /**  
   * Move the program at a specific position in the virtual channel schedule one position earlier.  
   * Note that calling this method will fail (and return -1) if it attempts to make a programme available before  
transmission starts.  
   */  
10 public synchronized int moveUp(int position) {  
  
    // ignore request if invalid.  
    if (position == 0 || position >= scheduleLength) {return -1;}  
    if(schedule[position].expectedStartTime.after(new Date(getStartTime(position-1)))) {  
15      System.out.println("moveUp failed");  
      return -1;  
    }  
  
    // Exchange programs.  
20    VirtualChannelProgram prog = schedule[position-1];  
    schedule[position-1] = schedule[position];  
    schedule[position-1].locked = true;  
    schedule[position] = prog;  
    return 0;  
25 }  
//-----  
   /**  
   * Move the program at a specific position in the virtual channel schedule one position later.  
   * Note that calling this method will fail (and return -1) if it attempts to make a programme available before  
30 transmission starts.  
   */  
    public synchronized int moveDown(int position) {  
  
    // ignore request if invalid.  
35    if (position < 0 || position > (scheduleLength-2)) {return -1;}  
    if(schedule[position+1].expectedStartTime.after(new Date(getStartTime(position)))) {
```

```
        System.out.println("moveDown failed");
        return -1;
    }

5    // Exchange programs.
    VirtualChannelProgram prog = schedule[position+1];
    schedule[position+1] = schedule[position];
    schedule[position+1].locked = true;
    schedule[position] = prog;

10   return 0;
    }

//-----
    /**
    * Remove the program at the nth position in this virtual channel schedule.
15   */
    public synchronized void remove(int position) {
        // To remove a programme permanently from the schedule it is given an arbitrary low absolute rating
        within the content pool.
        schedule[position].absoluteRating = -999;
20   schedule[position].locked = false;
        // Assume this user action indicates disapproval and feed the information back to the user profile builder.

        BBCNavigatorXlet.userAction(schedule[position].crid, false);
        reviseSchedule();
25   }

//-----
    /**
    * Find the highest rated programme available by a specific time.
    * <p> Programmes that are already entered in the schedule, or locked, or watched are ignored.
30   */
    private synchronized VirtualChannelProgram findBestProgram(long time) {

        int bestRating = -999;
        int bestProgIndex = 0;
35   for (int i=0; i<contentPool.length; i++) {
```

```
// ignore programmes that are currently in the schedule to avoid double entries.
if (contentPool[i].entered) {continue;}

// ignore locked programmes.
5   if (contentPool[i].locked) {continue;}

// ignore programmes that have already been watched.
if (userProfile.hasBeenWatched(contentPool[i].crid)) {continue;}

10  // ignore programmes that are not yet available.
if (contentPool[i].expectedStartTime.after(new Date(time))) {continue;}

// Compare ratings of programmes.
if(contentPool[i].absoluteRating > bestRating) {
15  bestRating = contentPool[i].absoluteRating;
    bestProgIndex = i;
}
}
return contentPool[bestProgIndex];
20 }
//-----
/**
 * Get the planned start time of programme at a specific position in the virtual channel schedule.
 * This assumes that all the programmes in the schedule are viewed in the planned order and takes into
25 account that part of the programmes may have been watched.
 */
public synchronized long getStartTime(int position) {
    // Get the current time.
    long time = System.currentTimeMillis();
30
    for (int i=0; i<position; i++) {
        time += (long)((schedule[i].expectedDuration - schedule[i].lastViewingPosition)*60000);
    }
    return time;
35 }
//-----
```

```
/**
 * Revise the virtual channel schedule.
 * <p> This method is called when a new programme becomes available from the content pool or when first
building a schedule.
5  * <p> Locked programmes in the schedule maintain their position relative to each other - but they may move
forward as they become available if they exceed the rating of preceding unlocked programmes.
 * <br> Unlocked programmes may be replaced by higher rated programmes as they become available - the
highest rated programmes are added to the schedule as early as possible.
 * <br> Programmes that have been watched are removed and excluded.
10 */
public synchronized void reviseSchedule() {

    // Create a new schedule array.
    VirtualChannelProgram[] newSchedule = new VirtualChannelProgram[scheduleLength];
15

    // Clear the "entered" flag so that it can be re-used during the building of the new schedule.
    for(int i=0; i<contentPool.length; i++) {
        contentPool[i].entered = false;
    }
20

    // progNumber is an index number relating to the position within the new schedule.
    int progNumber=0;

    // Get current time.
25     long time = System.currentTimeMillis();
    System.out.println("ReviseSchedule called at: " + (new Date(time)));

    // Prog loop iterates through the schedule selecting the best programme for each position, advancing time
by the duration of each consecutive programme.
30     ProgLoop:
        while (progNumber < scheduleLength) {
            // If there is a programme locked in this slot that has not already been entered elsewhere in the new
schedule then keep it there.
            if (schedule[progNumber].locked && !schedule[progNumber].entered) {
35                 newSchedule[progNumber] = schedule[progNumber];
```


// Otherwise find the highest rated programme that is available. Exclude locked programmes unless they are the next locked programme in the schedule (to maintain the order of locked programmes).

`} else {`

// Find the highest rating available programme.

5 `VirtualChannelProgram bestProg = findBestProgram(time);`

// Find the next locked program in the old schedule that has not already been entered.

`VirtualChannelProgram nextLockedProg = null;`

`for (int i=progNumber+1; i<scheduleLength; i++) {`

10 `if(schedule[i].locked && !schedule[i].entered) {`

// Check the programme will be available at this time.

`if(schedule[i].expectedStartTime.after(new Date(time))) {`

`nextLockedProg = schedule[i];`

`}`

15 `continue;`

`}`

`}`

// Compare and add the best rated programme to the schedule.

20 `if (nextLockedProg != null && nextLockedProg.absoluteRating >= bestProg.absoluteRating) {`

`newSchedule[progNumber] = nextLockedProg;`

`} else {`

`newSchedule[progNumber] = bestProg;`

`}`

25 `}`

// Set the entered flag on this programme to avoid multiple entries.

`newSchedule[progNumber].entered = true;`

30 *// Shift time to the end of the programme, taking into account some of it may already have been viewed.*

`time += (long)((newSchedule[progNumber].expectedDuration -
newSchedule[progNumber].lastViewingPosition)*60000);`

`progNumber++;`

`}`

35 *// Replace the old schedule with the new schedule.*

`schedule = newSchedule;`

```
    for (int i=0; i<schedule.length; i++) {
        System.out.println(i + " " + schedule[i].title + " locked = " + schedule[i].locked);
    }
}
5 //-----
  /**
   * Runnable interface for the Virtual Channel Builder thread. This runs in the background and updates the
   content pool when necessary.
   * <p> If the content pool could not be loaded from persistent storage or is older than 36 hours it is rebuilt
10 with a high thread priority.
   * <p> If the content pool is less than 36 hours old it is rebuilt with a low thread priority.
   * <p> Following the rebuild the thread should sleep for a long period before another rebuild. However, for
simplicity it is allowed to terminate.
   */
15 public void run() {

    System.out.println("VirtualChannelBuilder thread started");
    // Set the thread priority low unless the content pool is out of date.
    if(contentPool != null && buildDate.after(new Date(System.currentTimeMillis() - (long)(36*60*1000))))
20 {
        builderThread.setPriority(Thread.MIN_PRIORITY);
        System.out.println("VirtualChannelBuilder thread has low priority");
    }

25 // Rebuild the content pool
// N.B. Any changes to contentPool must be done in a synchronised method which must not block.

// If loadSchedule has failed then create a dummy content pool & schedule.
    if (contentPool == null) {
30         dummySchedule();
        reviseSchedule();
    }

// Run reviseSchedule when stored programmes have been analysed.
35 // Run reviseSchedule when programme schedule has been analysed.
```

```
// Set the build date.
```

```
    buildDate = new Date(System.currentTimeMillis());
```

```
// Save the new VirtualChannelSchedule.
```

```
5    saveSchedule();
```

```
// This thread should in theory wait until the next update is required. However, for simplicity it is allowed  
to terminate.
```

```
    System.out.println("VirtualChannelBuilder thread terminating");
```

```
10 }
```

```
//-----
```

```
// End of class VirtualChannelSchedule.
```

```
}
```

```
//=====
```

```
15 =====
```

Claims

1. A method for creating a viewing schedule for a user comprising:
receiving a schedule of broadcast programmes;
5 receiving a list of stored programmes;
storing user preference information;
compiling a schedule of programmes for the user selected from the schedule
of broadcast programmes and the list of stored programmes, based on the user
preferences.
10
2. A method according to Claim 1, further comprising playing out media
corresponding to the schedule.
3. A method of providing a media channel which presents a sequence of
15 programmes, the method comprising:
receiving media programmes from a plurality of sources including at least one
source of stored programmes and at least one source of broadcast programmes
and presenting a sequence of programmes selected from the stored programmes
and the broadcast programmes based on user preferences.
20
4. A method according to any preceding claim including selecting at least one
programme from the schedule of broadcast programmes to play back at a time other
than the time of broadcast of said programme.
- 25 5. A method according to Claim 4 including controlling a time-shifting recording
device to play back the selected programme at a desired time.
6. A method according to Claim 5 wherein the time shifting device is separate
from the schedule compiler
30
7. A method according to any preceding claim including indicating on the
schedule programmes which are live.

8. A method according to any preceding claim including indicating programmes which are pre-recorded.

9. A method according to any preceding claim including indicating programmes
5 which are time-shifted.

10. A method according to any preceding claim including presenting the schedule to the user and modifying the schedule in response to user input.

10 11. A method according to Claim 10 wherein modifying includes deleting programmes.

12. A method according to Claim 10 or 11 wherein modifying includes changing the order of programmes in the schedule.

15

13. A method according to Claim 10, 11 or 12 wherein modifying includes playing an available programme immediately in response to user selection.

14. A method according to any preceding claim wherein information concerning
20 user preferences is updated in response to user modification of a prepared schedule.

15. A method according to any preceding claim wherein the schedule is prepared in advance of the start of the schedule.

25

16. A method according to any preceding claim wherein the schedule is set to run for a predetermined period.

17. A method according to any preceding claim wherein the schedule includes
30 a planned break for the user to undertake other activity.

18. A method according to any preceding claim wherein the schedule includes

programmes which are not directly accessible by the scheduling device.

19. A method according to any preceding claim wherein multiple user preferences are stored.

5

20. A method according to Claim 19 including providing multiple user schedules.

21. A method according to any preceding claim comprising at least a step of ordering programmes according to time of broadcast or availability and a step of
10 ordering the programmes according to closeness of match to user criteria.

22. Apparatus configured to perform a method according to any preceding claim.

23. Apparatus according to Claim 22 comprising:

15

means for receiving a schedule of broadcast programmes;

means for storing a list of stored programmes;

means for storing user preference information;

means for compiling a user schedule based on on the received schedule of broadcast programmes, the stored list of stored programmes and the user
20 preference information;

means for outputting the user schedule.

24. Apparatus according to Claim 23, further comprising means for playing out media corresponding to the user schedule.

25

25. Apparatus according to Claim 23 or 24, further comprising means for controlling media storage apparatus.

26. Apparatus according to Claim 23 or 24, further comprising media storage
30 apparatus.

27. A computer program or computer program product comprising instructions for performing a method according to any of Claims 1 to 21.



INVESTOR IN PEOPLE

Application No: GB 0111850.4
Claims searched: 1-27

-29-

Examiner: Natasha Chick
Date of search: 15 November 2001

Patents Act 1977 Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:
UK Cl (Ed.S): H4F FBB
Int Cl (Ed.7): H04N 5/455 7/088
Other: Online: EPODOC, PAJ, WPI, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	GB 2325537 A MICROSOFT CORP	
X	WO 00/40021 A1 KONINKLIJKE PHILIPS ELECTRONICS. See page 1, line 22 - page 2, line 10, page 2 lines 24 - 32	1,2,3,4, at least
X	WO 00/16548 A1 UNITED VIDEO PROPERTIES INC. See page 2 lines 6 - 17 & lines 22 - 28, page 3, line 25 - page 4, line 10	1,2,3,4, at least
X	US 6005597 A BARRETT ET AL. See all, especially column 14, lines 13-23.	1,2,3,4, at least

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.