

CASE STUDY: AUTOMATION OF THE BBC'S INTERACTIVE TV PLAYOUT SYSTEM

P. Molander & S. Wolf

itv|world, Sweden
www.itvworld.com

ABSTRACT

The BBC has deployed a sophisticated scheduling and automation system to support an increasing range of interactive television services while minimising operational costs. The British Broadcasting Corporation offers interactive services on eight public television channels on digital satellite, terrestrial and cable television in the UK, with national variations for the main channels in England, Scotland, Wales and Northern Ireland. This presents a complex challenge for the scheduling and automation of enhanced and interactive services. This has been addressed by a novel system developed specifically to meet the demanding requirements of cross-platform, multi-format, and multi-channel operations.

INTRODUCTION

This paper describes how the BBC, one of the world's leading producers of interactive television, has introduced a new system for scheduling and automating interactive services. It outlines the requirements and the technical solution, describes the software architecture and identifies the operational advantages and business benefits of the system.

BACKGROUND

Enhanced and interactive television services are becoming more widely adopted, with over 100 broadcast network deployments to date. With the growth in the take-up and scale of such services, it is now recognised that interactive operations require improved processes and tools to achieve greater scalability and availability.

Playout Operations

The operational challenge for delivering scheduled enhanced and interactive services is to ensure that the correct data is associated with the appropriate channel on the relevant platform at the right time.

Many networks and broadcasters are struggling with managing the "what", "where", "when", and "how" of delivering interactive services using largely manual systems. As the size of these operations grows, the management of these manual processes becomes an increasing overhead with the risk of operational errors impinging upon quality of service.

Many of these redundant and repetitive manual processes can be addressed by scheduling and automation solutions. These can result in greater productivity, improved accuracy and a reduction in the overall cost of service delivery.

Requirements

In response to the requirement to deliver increasingly frequent and sophisticated interactive services, BBC Broadcast identified the need for a flexible control system specifically designed to manage the operation of multiple interactive services in a complex, multi-channel, multi-platform environment.

The system needed to be able to allow multiple interactive services to be controlled according to a predetermined schedule, driven by an external scheduling or automation system, or manually by an operator. These modes would need to be combined according to operational requirements, with manual operation possible at all times.

Furthermore, the system needed to support multiple users working simultaneously in a distributed environment, with appropriate permissions assigned to individual users according to their operational role. It required an intuitive graphical user interface to enable users to operate the system with the minimum specialist training.

Needless to say the solution required a high level of reliability and availability to support continuous live operation in the broadcast environment.

TECHNICAL SOLUTION

BBC Broadcast commissioned itv|world to develop a custom system to meet these requirements, based upon their existing range of In|Orbit control products. The In|Orbit Broadcast system was designed to interface with existing linear scheduling and automation systems and downstream platform specific data playout systems.

The design of the system abstracts the conceptual elements of a service from their specific implementation. The system is specified through generic concepts such as Service, Channel and Platform.

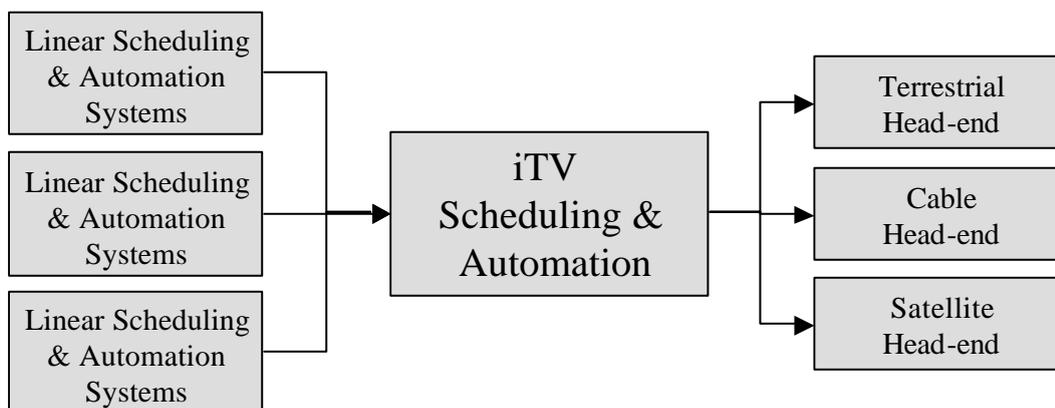


Figure 1 System Environment

The system accepts external inputs, applies predetermined rules, and generates appropriate output cues. These cues are received by plug-in adapters that translate the abstract cues into implementation-specific commands or functions.

Employing a rule-based decision engine, the generic behaviour of a service need only be specified once – when the service is first defined – rather than for each instance that the service will run.

It is designed as a role-based system, so separate operational functions can be assigned to different individual users or groups of users.

This allows a functional separation between those preparation activities that can be performed in advance of transmission, and those operational activities that take place at the time of transmission. It is even possible to delegate responsibility for individual services directly to editorial or production staff so that they can be managed remotely.

This is possible because the system has a robust and highly configurable security model that allows fine-grained control over the privileges and permissions of individual users. The system has been designed from the outset to support multiple users operating concurrently in multiple locations with updates delivered in real-time.

The intuitive graphical user interface is very similar in look and feel to an email client and allows relevant information to be filtered according to operator requirements.

Audit logs are maintained to record user operations, enabling administrator users to filter user and system operations to track errors or events.

System Usage

Users employ a simple point and click graphical user interface to establish business rules associated with an interactive service. These can specify multiple sub-events associated with the start or end of a programme in the linear television schedule. A sub-event might be a trigger or other type of platform specific event containing some information to be sent on a specific platform and channel. Figure 2 shows an example of how such sub-events are defined. These sub-events can be activated manually by an operator, a calendar schedule pattern, or by a filtering mechanism listening to incoming data from linear automation systems. The filtering mechanism is controlled by the user through filtering rules, specified as logical expressions relating to items in the linear programme schedule as supplied by conventional scheduling and automation systems. Figure 3 shows an example of a rule for filtering automation system messages. So, if a filter condition matches a data received from the automation system the system will send the corresponding sub-events further down the broadcast chain to relevant interactive television head end systems.

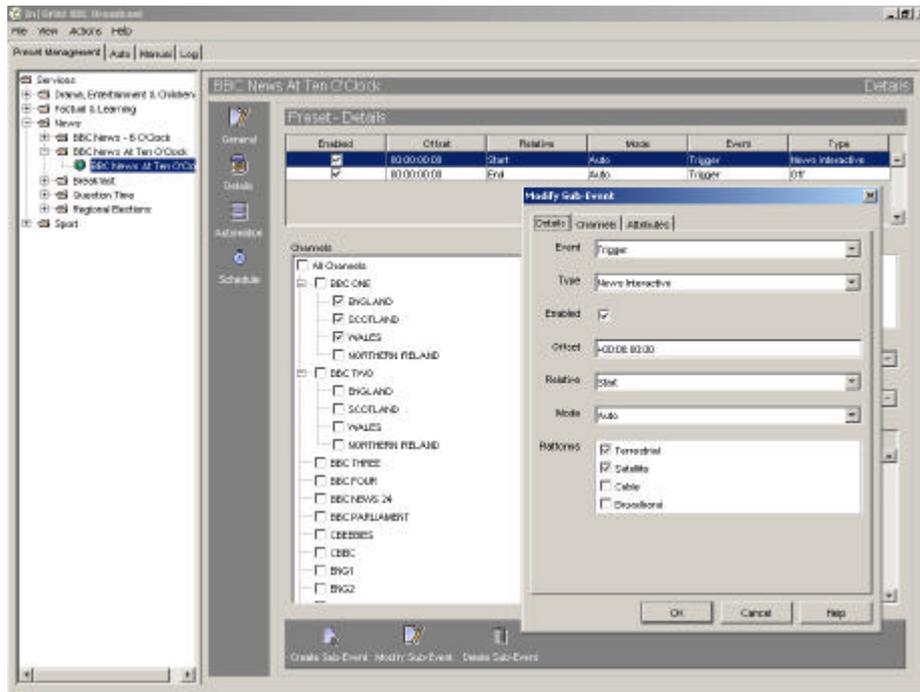


Figure 2 Configuring interactive events

Solution Approach

The system has a client-server architecture based upon widely adopted open standards and is implemented using XML, Java and Java 2 Enterprise Edition (J2EE) technologies. The

scalable nature of the J2EE platform and the choice of either file-based persistence or a SQL database enable the

implementation to be sized to performance and capacity requirements. The software has been designed as a real-time system. To achieve virtual frame accurate latency, some of the most intensive processing, such as filtering and scheduling, is performed outside the J2EE container in two multithreaded servers.

The software architecture is based on the use of components. The code base has been divided into layers, which have been further partitioned into components. The layering and component partitioning is based on the “type” of software not on the functional area.

The four layers are:



Figure 3 Automation Filtering Rule

The **presentation** layer is responsible for providing the client user interface. This layer represents components used to build the domain-specific user interface.

The **services** layer is responsible for providing domain-specific business logic. This includes mainly server side business logic, but also some stub classes for the client side process.

The **core** layer is responsible for providing domain-independent functionality that is used to construct the service and presentation layer. This layer contains components that are of a general nature and provides functionality that is reusable. Components of the core layer include a scheduling engine, decision engine, security framework and components providing more general functionality for persistence, internationalisation, fault handling, database connectivity etc. The core also includes the user interface framework, which is a set of building blocks for constructing user interfaces.

The **technology** layer represents the third party products employed. This includes the JDK, J2EE and Application Server.

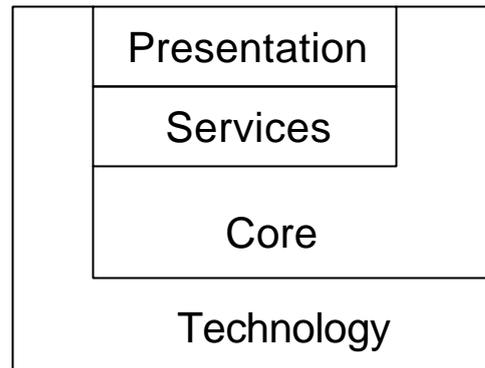


Figure 4 Software Layering

System Architecture

As mentioned above, the system has a distributed client server architecture based around a J2EE multi-tier model, where the client can run on any appropriate personal computer or workstation connected via a Local Area Network. All graphical user elements of the client are developed using Java Swing. The system is currently deployed on a JBOSS Application

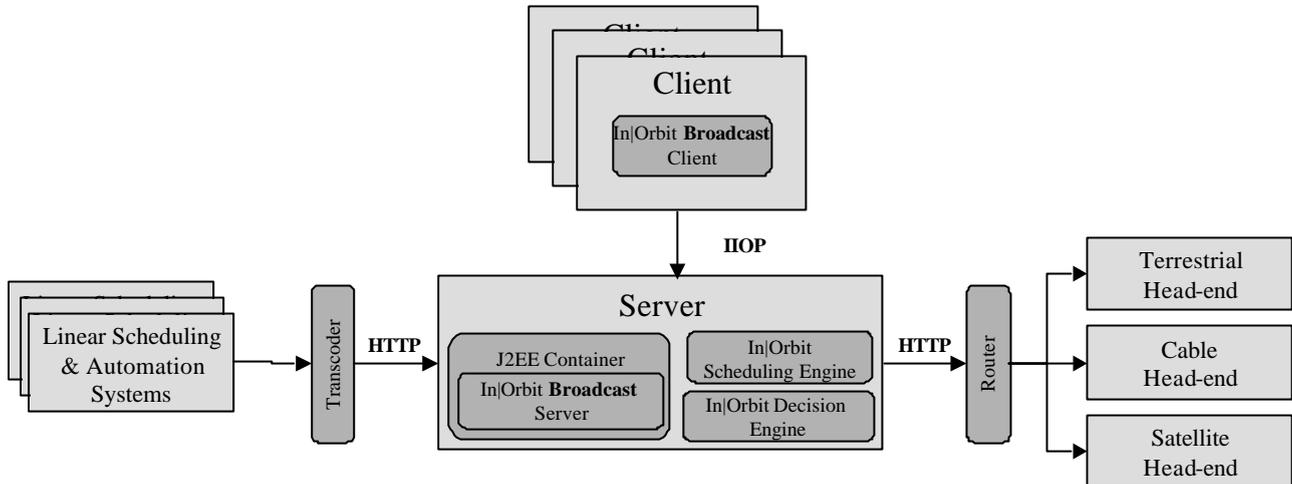


Figure 5 System Architecture

Server on Linux, with Windows 2000 clients, but can be ported to any other operating system or J2EE implementation. The server application runs within the J2EE container, communicating through the Java Messaging Service (JMS) with two multi-threaded Java VM processes providing functionality for scheduling, filtering and decision-making. The system interface is defined in XML and is managed by a Java Servlet deployed on a Tomcat Servlet Engine. The system configuration parameters, security roles, user data, and rules for filtering are all stored in a fault tolerant repository for back-up and redundancy purposes. To achieve greater availability and reliability in the broadcast operations environment, two chains of the system are deployed in parallel, each supported by RAID disk storage and dual power supplies. Therefore, it is feasible to recover any severe hardware or software failures by simply switching to the redundant chain.

System Dynamics

The system receives input messages in XML format over HTTP. These messages may be derived from one or more automation systems and provide information about the current and subsequent programme item for each linear television channel. An example input message is given below.

```
<?xml version="1.0"?>
<event>
  <messagetime>2002-09-05T11:30:43+00:00</messagetime>
  <now>
    <service>BBC One</service>
    <serviceid>4223</serviceid>
    <eventid>61530</eventid>
    <time>2002-09-05T11:30:46+00:00</time>
    <duration>00:28:00</duration>
    <publishedtime>2002-09-05T11:30:00+00:00</publishedtime>
    <publishedduration>00:30:00</publishedduration>
  </now>
</event>
```

```
<title>Judgemental</title>
<genre mainGenre="Entertainment" subGenre="Game Shows"></genre>
</now>
<next>
  <service>BBC One</service>
  <serviceid>4223</serviceid>
  <eventid>61531</eventid>
  <time>2002-09-05T12:00:00+00:00</time>
  <duration>00:30:30</duration>
  <publishedtime>2002-09-05T12:00:00+00:00</publishedtime>
  <publishedduration>00:30:00</publishedduration>
  <title>BBC News</title>
  <genre mainGenre="News" subGenre="News & Weather"></genre>
</next>
</event>
```

This message indicates that the current programme item is a game show, which will be followed by the midday News & Weather. This information is used to cue up any events and sub-events that will be associated with the news on this channel. These are defined by business rules within the system. As such, they need only be defined once as a general rule, rather than for each specific instance of a particular programme. This clearly delivers significant benefits for regular programmes, such as the news, which may be transmitted many times a day on each channel.

The receiving Servlet marshals the XML message and forwards the data for filtering. The system filters the message data based on predetermined criteria or business rules and decides whether to generate an output or do nothing. Output messages can also be created by the internal scheduling mechanism or directly by an operator. The system produces output messages combining metadata derived from the business rules and the incoming data. The outgoing message is also defined in XML format and is sent to a downstream adapter through HTTP. The custom adapter software then propagates the message received through to the relevant satellite, terrestrial or cable data playout systems.

ADVANTAGES

The interactive scheduling and automation system offers a number of operational advantages relating to flexibility, efficiency, accuracy, and scalability, over the previous largely manual mechanisms employed for controlling services. Concrete quantification of these advantages is currently under assessment by BBC Broadcast and itv|world as part of a formal post implementation review.

Flexibility

The solution was designed from the outset to be ultimately flexible. The system can be rapidly configured through the user interface and simple XML configuration files. This makes it very easy to change or add services, channels, platforms and classes of event, which reduces the time and resources required to accommodate new requirements in a rapidly changing environment. Any changes can be made without requiring modification to the main linear playout automation systems. Since the system is inherently platform agnostic, it can be used to support multiple interactive playout systems, such as OpenTV, Mediahighway, Liberate, MHEG-5. The BBC deployment supports OpenTV, Liberate, and MHEG-5 for the BSkyB, Telewest and ntl, and Freeview platforms respectively.

Efficiency

Automation reduces the need for redundant, repetitive manual operations involved in managing the lifecycle of interactive applications in response to reactive programme schedules. The details of the interactive events can be defined once in advance and then applied automatically in response to events from the linear automation system. This increases productivity and allows a single operator to control several complex interactive services across multiple channels through a single unified user interface. Repetitive, redundant manual operations are also significantly reduced, allowing the operator to concentrate on monitoring the services. This has enabled the BBC to support an increasing number of interactive services while containing the operational cost base.

Accuracy

Configuring the events associated with interactive services in advance reduces the risk of mistakes during live operation, improving accuracy and minimising on air errors. Event timing may be controlled either according to time of day or in response to the output of the linear automation system, improving the precision of presentation. Sub-events can even be pre-rolled by a precise number of frames to cater for any platform-specific latency considerations.

Scalability

Automation enables the number, frequency and complexity of services to be considerably increased while minimising operational overheads thus reducing overall costs. This was an essential factor for the BBC, which currently supports interactive services across eight channels of digital television over three distinct platforms. Automation of core functions was a key requirement to support the BBC's recently re-launched 24/7 services. These provide context sensitive menus that change dynamically across each channel according to the current programme.

SIGNIFICANCE

While developed specifically to meet the demanding requirements of the scale and scope of the BBC public service playout operation, the system is sufficiently generic to be applicable to varying levels of complexity, from a single channel to a network operator.

CONCLUSIONS

This paper describes how the development and deployment of a novel scheduling and automation system has been applied in a challenging multi-channel, multi-platform, and multi-format environment. The implementation has allowed the range of interactive services offered to be increased, increasing flexibility and efficiency while minimizing operational overheads.

REFERENCES

1. Wolf S., 2002, Scheduling and automation of interactive television, Broadcast Engineering, World Edition
2. Wolf S., 2001, Running iTV services on a live broadcast system, Digital Television 2001 Conference, Darling Park, Sydney

ACKNOWLEDGEMENTS

The authors would like to thank their colleagues at itv|world. They would also like to extend a special thank you to Dr William Cooper and Dr Sue Farrell of BBC Broadcast for their contributions to this work.