

# Model-driven creation of Staged Participatory Multimedia Events on TV

Jan Van den Bergh<sup>1</sup>, Bert Bruynooghe<sup>2</sup>, Jan Moons<sup>2</sup>, Steven Huypens<sup>1</sup>, Koen Handekyn<sup>2</sup>, and Karin Coninx<sup>1</sup>

<sup>1</sup> Hasselt University – transnationale Universiteit Limburg  
Expertise Centre for Digital Media – Institute for BroadBand Technology  
Wetenschapspark 2, 3590 Diepenbeek, Belgium

{jan.vandenbergh, steven.huypens, karin.coninx}@uhasselt.be

<sup>2</sup> Alcatel-Lucent

Copernicuslaan 50, 2018 Antwerp, Belgium

{bert.bruynooghe, jan.moons, koen.handekyn}@alcatel-lucent.be

**Abstract.** Broadcasted television shows are becoming more interactive. Some shows even let home viewers without professional equipment be part of a broadcasted television show. Staged Participatory Multimedia Events on TV take this approach another step further. In this type of television shows, viewers can not only participate in the show through interaction or videostreams, but also direct the show. In this paper we discuss this type of participation TV into more detail and discuss the models allowing quick and graphical creation of the structure of a format. The models can be serialized to a set of XML-files, which can be interpreted by the ParticipationTV runtime. Working proof-of-concept implementations for creating the models, generating the XML-files and the runtime that interprets the XML-files have been realized.

## 1 Introduction

<sup>3</sup> The world of broadcast television is changing from analog to digital broadcasting and from a mostly passive towards a more interactive medium with increased viewer participation in the broadcasted shows. Many television shows also have related counterparts in other media such as websites offering additional capabilities.

Increasingly television streams are also transmitted over the Internet or traditional TV broadcasts are viewed on desktop computers, laptops or even mobile phones, offering new possibilities for interaction. Many of these devices are sold with cameras, built into the device or as accessories and have the possibility to send live video-streams over the Internet. At the same time the usage of set-top boxes, which have high-speed Internet access, built-in hard-disk and ample computing power increases.

---

<sup>3</sup> The original article is available at <http://www.springerlink.com/content/237242uu11275175/>

Staged Participatory Multimedia Events on TV (SPMEs) will exploit these additional interaction and participation capabilities. SPMEs are events whose director and participants can all be television viewers. In this paper, we only discuss the television-based viewers, although other devices such as PCs or smartphones will also be supported. Section 2 describes the structure and properties of such shows into more detail and introduces an example. Section 3 then discusses the ParticipationTV runtime infrastructure and the XML files used to configure the flow of the SPME. These XML files can be generated from graphical models. These models are discussed in section 4. The paper ends with a discussion of related work and the presentation of conclusions and future work.

## 2 Staged Participatory Multimedia Events

Staged Participatory Multimedia Events are broadcasted (for large audiences) or multicasted (for small communities) television events that actively engage TV viewers and turn them into true participants. They thus provide a stage for viewers to participate in interactive television applications that are not yet publicly or commercially available. The basic requirement for viewing a SPME is having a television. In order to participate, a viewer will have to have some means of communication with server such as a set-top box and appropriate remote control or a device that can send a SMS. For full participation, other devices such as webcams and microphones and a high-speed Internet connection will be necessary; live video and audio streams from selected participants of the show will be merged and integrated into the multimedia stream.

SPMEs belong to a *format* whose structure can be defined using the graphical modeling language discussed in section 4 or directly using the XML-language that is used as a blue print of the show (see section 3). Show participants can have different interaction capabilities based on their *role*. Some roles may have hardware requirements such as a webcam and microphone. Each format allows all participants to chat with one another.

A SPME is started when the first viewer activates a format. This first viewer will not only start the show but will also become its director, identified by the role *master*. All other viewers that join the active format will initially get the same role, *participant*. The show is driven by viewer interaction (and time-based events when desired). The format, however, determines the actions that can be performed by viewers based on their role and the actions that are already performed.

AuctionTV is an example of such a format. It is a format that allows one of the participants to offer an item for sale through an auction. The format is started by the auctioneer, who gets the role *master*. All other people that join the format afterwards, have the initial role *participant*. One of these participants can offer an item for sale and as such becomes the seller; the seller does no longer have the role *participant* but gets the role *seller* instead. Then the auctioneer initiates an interview with the seller, followed by the bidding process. Whenever a participant  $p$  bids, the auctioneer raises the price, confirming the bid. In doing



**Fig. 1.** AuctionTV application used with different interaction methods/devices (a) a TV remote control (b) a game controller

so the role *winner* is added to the roles of  $p$  and removed from the previous bidder (if there was one). When an acceptable bid has been made and confirmed, the bidding process is ended by the auctioneer. The auctioneer finally does an interview with the winner of the highest bid. As long as no bid is made, the auction can be cancelled by the auctioneer, just as all other viewer actions, using a remote control (see Fig. 1).

The AuctionTV example illustrates some important properties of SPME:

1. SPME can be completely driven by viewer actions, although time-based events can be used.
2. Each person watching a SPME has one or more roles.
3. Fig. 2 illustrates that viewers with different roles can be part of the show through live video streams provided by webcams.
4. SPMEs have a script; the consequences of viewer actions have predefined (predictable) results.
5. In some situations, it is necessary that one action is to be performed before any other actions can take place. For example, when a bid is made the auctioneer has to accept it and raise the amount for the next bid.
6. Highly different interaction methods or devices can be used (see Fig. 1).

All SPME formats have a similar screen structure, which is illustrated in Fig. 2. The *background* consists of non-interactive content and is shared by all viewers whatever their role is. All interactive content (including instructions) is shown in *interaction panels* or *popup dialogs*.

Each interaction panel allows the viewer to perform one task such as making a bid, setting the price of the item to be sold, start the interview with the winner (a transition to a new screen layout). The interaction panels are layered on top of each other in one designated area of the screen, the *interaction bar* (see Fig. 2). A *popup* allows its viewer to perform exactly one task, while temporarily disabling the performance of all other tasks (realizing property 5). A popup thus has behavior similar to that of a modal dialog box on a desktop computer.

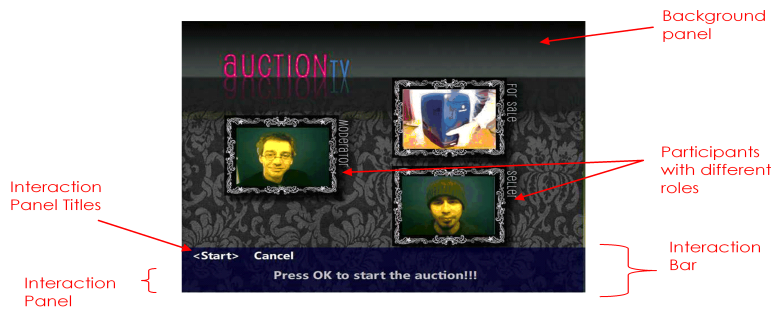


Fig. 2. Example structure of the screen contents of an SPME

### 3 The SPME runtime support

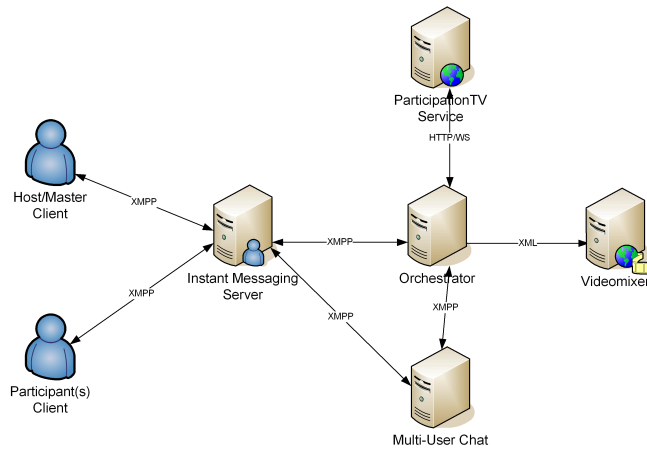
#### 3.1 The runtime infrastructure

The ParticipationTV infrastructure (see Fig. 3) is build around an instant messaging (IM) server that is based on the eXtensible Messaging and Presence Protocol, short XMPP [4, 5]. Although this is not a necessity it gives a lot of advantages. The protocol is based on XML, which means that it is easily extensible with ParticipationTVs own XML-based SPME language.

ParticipationTV clients or server-side components that need to send ParticipationTV specific messages to each other can simply do so by embedding them in an XMPP message and the IM server will make sure that they will arrive at the corresponding entity. Furthermore the presence part makes it easy to see which buddies are online and you can invite them to play a TV program that you made or host, although people can also join a ParticipationTV program if they know the format and host.

The actual core of ParticipationTV is the Orchestrator, a server-side XMPP component that plugs into the IM server. The Orchestrator has a variety of functions: keeping track of ParticipationTV format instances and sessions, making sure participants have the correct roles, handling workflow, sending the right interactive components to them and steering the videomixer.

When a user logs into ParticipationTV he will send an XMPP message to the Orchestrator component to ask for the available formats. The Orchestrator will make use of the ParticipationTV web service to retrieve the list of formats. After the user has selected a format and states that he wants to be the host, the Orchestrator will exercise the following steps: (1) Load the corresponding XML SPME files into its workflow engine. (2) Create the new session and informs the web service of the newly created session. (3) Attach the role *master* to this user. (4) Start an instance of the videomixer. (5) Create a room in the Multi-User chat component [2] of the IM server. (6) Send a message back to the user that everything is setup correctly and that he can join the Multi-User chat component.



**Fig. 3.** Runtime infrastructure

All participants of the same format instance will end up in the same Multi-User chat room.

All other users that subsequently log in will be able to join the format instance/session created by this first user. The workflow engine that is part of the Orchestrator will now start to process the XML SPME files and will act accordingly: (1) Steer the videomixer, which composes the correct video image for all users. (2) If there is interactivity, make sure the user with the correct role will receive it and can act upon it. (3) Go through the flow until the format has finished or stopped by the user who has the role *master*.

### 3.2 The runtime SPME description

The XML SPME description consists of two parts: the layout and the flow description. The layout consists of three major blocks: the *background* (which is the same for every user), the *interaction panels* that can be added to the interaction bar based on the actual roles of the user, and *popup* dialogs for modal dialogs to users, based on the roles. The description of the layout is based on MyXaml<sup>4</sup>, an open source language and library to describe GUIs of Windows applications. This XML language, together with some custom controls for ParticipationTV, allows to position labels (**Label**), pictures (**PictureBox**), synchronized video (**Video**), presence (**Presence**: webcam, microphone input, avatar, ...) on the screen. Some of the XML for such controls is shown in Listing 1.1. Furthermore, some specific interactions were developed to allow interaction based on a standard set-top box remote control.

The flow description is a list of named *events* and the according event descriptions, which consist of a set of *commands* such as those shown in Listing 1.2.

<sup>4</sup> <http://www.myxaml.org>

---

### Listing 1.1. SPME XML - control examples

---

```
<Label Location="75,400" Size="200,60" Text="Bid: _$RequestedBid$" />
<Presence Location="100,230" Size="160,_115" Source="#MASTER#"
  video="true" audio="true" />
<PictureBox Location="0,_0" Size="720,_576">
  <Image>
    <Bitmap URL="http://participationtv/auctiontv/sellerinterview.png" />
  </Image>
</PictureBox>
<Video Source="#MASTER#" Filename="ItemPresentation_SGI.avi"
  Location="400,150" Size="160,_115" />
```

---

---

### Listing 1.2. SPME XML - example commands

---

```
<AddToRole user="#INTERACTOR#" roleName="Seller" />
<RemoveFromRole user="#PARTICIPANT#" roleName="PARTICIPANT" />
<SetVariable name="RequestedBid" value="100" />
<SetVariable name="RequestedBid" value="$RequestedBid$+100" />

<SetBackground panelName="Intro" />
<AddInteraction user="#PARTICIPANT#" panelName="Sell"
  associatedEvent="BecomeSeller" />
<RemoveInteraction user="#MASTER#" panelName="Cancel" />
<ShowPopup user="#MASTER#" panelName="AckBid"
  associatedEvent="MasterAckBid" />
```

---

The first group of commands applies to runtime items known as *variables* and *roles*. Both variables and roles can be resolved to strings to be used in the layout files. Variables will typically be used for the dynamic behavior of the show as they support basic JScript <sup>5</sup> evaluation, while roles have to be considered as dynamic groups of users, which are typically used for role dependent content and interactions (popup dialogs and interaction panels).

The second group of commands defines the layout of the screen of the different users: a new background can be assigned, interaction panels can be added and removed, and popup dialog can be displayed. Interaction panels and popup dialogs also have an associated event, which the user will spawn when he presses OK when the interaction panel or popup is active. These graphical items are designated by a *panelName*, which maps to a description in the corresponding layout files.

## 4 SpIeLan: models for SPME

SpIeLan (*SPME interface language*) is a graphical modeling language created for the design of SPME. It consist of three models: the scenario, the scene stage and scene script. It is important to note that the names and the contents of these models have been slightly changed to better fit the updated requirements of the

---

<sup>5</sup> <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/jscript7/html/jsoriJScript.asp>

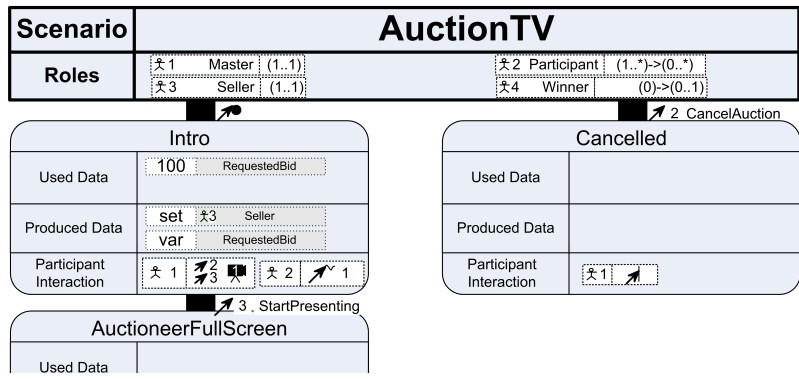


Fig. 4. Partial scenario of AuctionTV SPME

runtime infrastructure discussed in section 3 since earlier reports on SpIeLan [6, 7]. The visual syntax has also been slightly updated, based upon an informal test with both programmers and non-programmers, people familiar with participation tv and people that were not familiar with it.

The first model describes the *scenario* of the SPME; it provides a high-level overview of what the show is about. Fig. 4 shows a part of the AuctionTV scenario. It shows all roles that are involved in the show in the header, including the maximum amount of viewers that can take this role initially (i.e. before the end of the first scene) and during the rest of the show. The rounded rectangles represent the *scenes*, a part of the show continuously showing a single *background*. The roles that are actively involved in a scene (providing live streaming of video or sound or able to interact) are represented in the rounded rectangle, as well as the data that is used or changed in a scene. Each of the scenes is activated by a user-generated event.

The different types of events that are currently supported by SPME are shown in Fig. 5. The symbols for these actions are taken from the Canonical Abstract Prototypes notation [1]. The semantics of these symbols is kept the same, although it is more restricted and more specific to SPME. Depending on the actions performed by the viewers, alternative sequences of scenes may be specified. E.g. in Fig. 4 action 2 and 3 trigger to alternative sequences of which the sequence triggered by action 2 is the “normal” sequence. Should this not be sufficient, control structures can be used. Discussion of the control structures is however out of the scope of this paper. The symbols for these actions are abstract because highly different remote controls could be used to interact with the SPME, even when only the TV is considered as a visual medium, as can be seen in Fig. 1.

The second model specifies the *scene stage*; the arrangement of the user interface controls on the television screen. The model consists of two required parts and two optional parts. The first required part is the scene header (see

Icon	Semantics
	Starts the SPME
	Ends the SPME
	Performs a simple action not defined by the other types
	Selects a value from a predefined set (currently only strings)
	Edits a value (currently only numbers)

Fig. 5. Types of user-generated events in a SPME

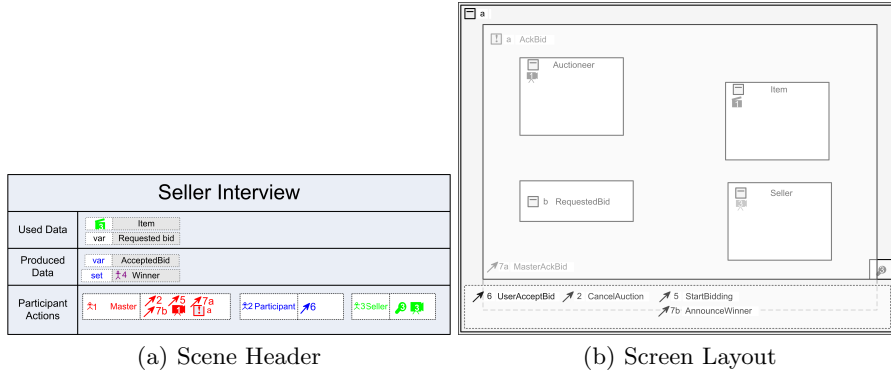


Fig. 6. Scene stage specification

Fig. 6(a)) which contains the name of the scene, the roles actively involved in the scene, and the data used and produced during the scene.

The second part describes the screen layout. It completely specifies the *background* and specifies the regions where the *interaction bar* and the *popups* will be placed. Each part of the screen is identified by a rectangle with one or more icons and an explanatory text in it. The rectangle always contains an icon that identifies the purpose of the part (see Fig. 7) and optionally a source (such as recorded video, live video or live audio).

The location of the interaction bar is indicated by a dashed rectangle. All the contained interaction panels are represented by the corresponding events. A similar approach is taken for popups; only the area where the popup will be placed is indicated in the screen layout. The contents of both the interaction panels and the popups is shown separately.

The third and last model specifies the scene script; it specifies which actions can be performed by the viewers in which order and what the consequences are. It has the same header as the scene stage (see Fig. 6(a)) but the rest of the model content differs. An example of this contents is shown in Fig. 8. It shows that in the scene *Seller Interview*, the *master* can initially see two interaction panels (with associated events 2 and 5). Event 2 cancels the auction and ends the scene while event 5 enables the *participants* to make a bid (event 6). When a bid is made, the auction cannot longer be canceled. The *participant* that makes the bid gets the additional role *winner*. In addition to this, the value for *AcceptedBid* is set



Icon	Semantics
<i>Passive user interface controls</i>	
	Displays media or text
	Displays media about a viewer
	Displays media about a group of viewers
<i>Interactive user interface controls</i>	
	Allows selection of a value
	Allows editing a value (currently only numbers)

Fig. 7. User interface controls for SPME

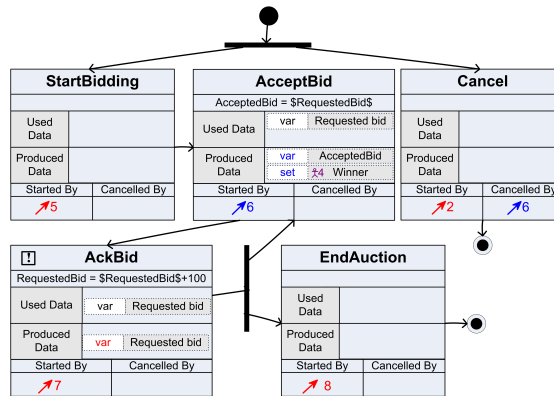


Fig. 8. Scene script corresponding to the header in Fig. 6(a)

and a popup with associated event 7 is shown. When the *master* acknowledges the bid, `RequestedBid` is incremented. The auction can be ended (event 8) or a *participant* can make a new bid.

We have implemented a tool that can generate the SPME XML from the models. The information for the layout-related SMPE XML is (almost) entirely generated from the scene stage specifications, while the flow description is generated by combining the scenario and scene scripts.

## 5 Related Work

There is a body of work regarding participation TV besides the effort discussed in this paper. Some participation shows have already been broadcasted such as CultTV in France and Mattina in Italy. The latter uses the Mycast system from Digital Magics<sup>6</sup> to integrate live feedback from viewers from videophones and webcams into the daily morning show by Rai Uno, a national TV station in Italy [3].

<sup>6</sup> <http://www.digitalmagics.com>

OpenTV offers a tool and framework that allows creation of Participation TV <sup>7</sup> without coding. As far as the authors know the framework is focussed on enhancing existing television shows and formats with interactivity and extended statistics and does not allow the advanced viewer-driven shows which integrate live video-feeds as we presented in this paper.

TVML <sup>8</sup> has a different focus and allows to easily create 3D non-interactive TV shows using a simple scripting language. SoapShow <sup>9</sup> allows users to create their own soap online using video, still images, text and sound clips. Currently it is limited to the web, but they are planning to show the best soaps on Dutch television.

## 6 Conclusions and future work

In this paper we described a new type of participation TV: Staged Participatory Multimedia Events on TV. It gives an unprecedented amount of control to television viewers: they are no longer just viewers but participants or even the host of a show. They can be part of the show through the use of currently available technology such as webcams, microphones and a remote control. Examples of SPME formats can include existing TV shows adapted for greater viewer participation, AuctionTV, adaptations of board games, shows for specific communities or even a headbangers competition.

The runtime infrastructure is flexible and uses open specifications and protocols such as MyXaml and XMPP. The creation of a format can be done using the SPME XML or through graphical models, from which the SPME XML can be generated. The creation and evaluation (possibly in cooperation with a TV production company) of an end-user tool that allows to quickly and graphically model, generate and preview SPME is a major focus of current and future work.

**Acknowledgements** This research was performed in the context of the IWT project Participate of Alcatel Bell. Part of the research at the Expertise Centre for Digital Media is funded by the ERDF (European Regional Development Fund), the Flemish Government and the Flemish Interdisciplinary institute for Broadband Technology (IBBT).

## References

1. Larry L. Constantine. Canonical abstract prototypes for abstract visual and interaction design. In *Proceedings of DSV-IS 2003*, number 2844 in LNCS, pages 1 – 15, Funchal, Madeira Island, Portugal, June 11-13 2003. Springer.
2. Toon Coppens, Lieven Trappeniers, and Marc Godon. Amigotv : towards a social tv experience. In *Proceedings of EuroITV 2004*, 2004.

<sup>7</sup> <http://www.opentvparticipate.com/>

<sup>8</sup> <http://www.nhk.or.jp/str1/tvml/>

<sup>9</sup> <http://www.soapshow.nl>

3. Monique Van Dusseldorp. Video-phone feeds getting into mainstream media. E-Media TidBits, <http://www.poynter.org/column.asp?id=31&aid=81683>, 2005.
4. P. Saint-Andre. Extensible messaging and presence protocol (xmpp): Core. <ftp://ftp.rfc-editor.org/in-notes/rfc3920.txt>, 2004. ©The Internet Society.
5. P. Saint-Andre. Extensible messaging and presence protocol (xmpp): Instant messaging and presence. <ftp://ftp.rfc-editor.org/in-notes/rfc3921.txt>, 2004. ©The Internet Society.
6. Jan Van den Bergh, Steven Huypens, and Karin Coninx. Towards Model-Driven Development of Staged Participatory Multimedia Events. In *Proceedings of DSV-IS 2006*, LNCS, Dublin, Ireland, July 2006. Springer. To be published.
7. Jan Van den Bergh, Kris Luyten, and Karin Coninx. High-Level Modeling of Multi-User Interactive applications. In *Proceedings of TaMoDia 2006*, Diepenbeek, Belgium, October 2006. To be published.