ABSTRACT
The Production Scripting Engine (PSE) is a Virtual Director software automatically selecting and framing camera shots in an interactive live broadcast production system. It is informed through various channels, some of which send a low-level event stream. The decision making process is distributed along a chain over the production and delivery networks, and the concrete amount of PSE instances depends on the actual production.

Categories and Subject Descriptors
C.2.4 [Distributed Systems]: Distributed applications; H.4 [Information Systems Applications]: Miscellaneous

Keywords
Virtual director, event processing, rule engine, distributed decision making

1. INTRODUCTION
The FascinatE system is an interactive event broadcast system that reasons for a range of viewers in parallel. It allows viewers to both watch what’s automatically directed by its Virtual Director or to interactively navigate around an ultra-high resolution video panorama. The output is adapted to the user’s viewing device, covering anything from a mobile handset to an immersive panoramic display. FascinatE uses the OmniCam, a collection of 6 HD cameras sharing a single optical center for obtaining a 180° panoramic video sequence with 6984 x 1920 pixels stitched together in real-time. The 180° field of view allows to capture a whole scene like a sports field with one camera and its resolution allows to capture even distant objects in good quality so that e.g. persons at the other end of a sports field can be recognized. We additionally use a range of broadcast cameras.

The Virtual Director of the system is a distributed component that takes inputs from various channels into account to finally come up with live decisions which camera shots/views to show for each viewer group, when to cut, and how to follow moving objects. It is called the Production Scripting Engine, PSE, and is reasoning in real-time in a distributed fashion over the broadcast and delivery networks. Several sources inform the PSE via event streams that describe actions in the scene on a low semantic level, for example the results of a person tracking module. This knowledge is fused and interpreted as it is required to take meaningful decisions.

FascinatE’s approach can be compared to less automated work in the LIVE production support system [4] that was successfully evaluated during the 2008 Olympic Games. Recent automated camera selection results from the research project APIDIS were impressive but were achieved within different constraints [2]. For a problem scope comparable to the one at hand, approaches using the advantages of event processing [7, 3] and rule-based processing [1, 8] have been proposed.

2. DISTRIBUTED DECISION MAKING
The PSE’s architecture and the role of its subcomponents are described in [6]. Relevant interfaces are shown in Figure 1. It performs distributed decision making in parallel for different viewers. The process starts at the production end where a set of global shot candidates is computed and assigned with pragmatics-based priorities.

To be able to take reasonable decisions, the engine needs knowledge about the production domain as well as information about what is happening in the scene. This can be categorized into two types: (a) Static knowledge describes the domain knowledge including production rules and visual grammar, privacy rules, content rights and user profiles. (b) Dynamic events describe what is happening in the scene. The content analysis modules produce a real-time event stream of low-level descriptions of the scene such as person tracks [5] or a saliency measure for regions. Live annotations of high-level semantic concepts are entered by the professional production team. Main tasks are to identify important actions and interesting persons.

The PSE is a distributed component with at least 2 instances in the chain. The first, the primary PSE processes a high volume of real-time low-level events based on a set of rules that express the production grammar. It determines suitable candidates of shots described by a bounding box and metadata. The reasoning component outputs both fixed shots and such smoothly framing moving objects.
Based on a static shot ontology model and several dynamic factors, shot candidates are assigned a priority value per viewer group. Each PSE sends scripts to the next PSE instance in the chain. Scripts contain prioritized shot candidate metadata, rendering instructions and user interface options. Scripts further inform the Delivery Scripting Engine (DSE) to optimize bandwidth usage and transmit certain areas of the panorama in higher resolution. Each PSE along the cascade refines the decision based on a dedicated aspect, for example content rights (see Figure 1). The terminal PSE has to take the final decision and instruct the renderer. It has the special task of deciding when to cut.

The first experimental implementation led us to use event-condition-action (ECA) rules and a rule engine enabled for event processing. We chose JBoss Drools\(^2\), an Open Source business rule management system that provides comprehensive functionality for hybrid chaining rules and integrates event processing capabilities. A rule base implemented with forward chaining rules allows us to easily plug in new functionality. A challenge we encountered is that contradicting rules within one component have to be resolved explicitly with additional rules, e.g. using a layered approach. Testing and finding contradicting behaviour, especially with temporal logic involved, is a challenge when dealing with a large rule base. Layering the decision making process helps maintaining the logic but it is not clear how scalable that approach is.

3. ACKNOWLEDGMENTS

The research leading to this paper has been supported by the European Commission under the contract FP7-248138, “FascinatE – Format-Agnostic SChript-based INterAcTive Experience”.

4. REFERENCES


\(^2\)http://www.jboss.org/drools