Media Processing Workflow Design and Execution with ARIA *

Lina Peng, Gisik Kwon, K. Selçuk Candan, Kyung Ryu, Karam Chatha,
Hari Sundaram, and Yinpeng Chen
Computer Science and Engineering Dept.
Arizona State University, Tempe, AZ, 85287, USA.
{lina.peng, gisik.kwon, candan, kdryu, kchatha,
hari.sundaram, yinpeng.chen}@asu.edu

ABSTRACT

Recently, we introduced a novel ARchitecture for Interactive Arts (ARIA) middleware that processes, filters, and fuses sensory inputs and actuates responses in real-time while providing various Quality of Service (QoS) guarantees. The objective of ARIA is to incorporate realtime, sensed, and archived media and audience responses into live performances, on demand. An ARIA media workflow graph describes *how* the data sensed through media capture devices will be processed and *what* audio-visual responses will be actuated. Thus, each data object streamed between ARIA processing components is subject to transformations, as described by a media workflow graph. The media capture and processing components, such as media filters and fusion operators, are programmable and adaptable; i.e, the delay, size, frequency, and quality/precision characteristics of individual operators can be controlled via a number of parameters. In [1, 4, 5], we developed static and dynamic optimization algorithms which maximize the quality of the actuated responses, minimize the corresponding delay and the resource usage. In this demonstration, we present the ARIA GUI and the underlying kernel. More specifically, we describe how to design a media processing workflow, with adaptive operators, using the ARIA GUI and how to use the various optimization and adaptation alternatives provided by the ARIA kernel to execute media processing workflows.

Categories and Subject Descriptors: J.5 [Arts & Humanities]: Performing arts; C.3 [Special-purpose & Application-based Systems]: Real-time and embedded systems

General Terms: Design

Keywords: Tools for creating interactive, multimedia, multimodal art, realtime sensory/reactive environments, filtering, fusion

*This work has been supported by NSF grant # IIS-0308268, "Quality-adaptive Media-flow Architectures to Support Sensor Data Management."

MM'05, November 6–11, 2005, Singapore.
ACM 1-59593-044-2/05/0011.

1. INTRODUCTION

At ASU, we are building an Intelligent Stage to allow performers to have real-time control of the stage, by detecting and classifying the movement of the performers and responding by changing environmental elements to achieve spontaneous lighting and sound effects. This Intelligent Stage requires an innovative information architecture that processes, filters, and fuses sensory inputs and actuates audio-visual responses in realtime. Thus, we are developing an adaptive and programmable ARchitecture for Interactive Arts (ARIA) which will enable design, simulation, and execution of interactive performances.

ARIA is being designed as an *adaptive media processing workflow architecture* to capture, process, filter, fuse, and stream various types of audio, video, and motion data [2, 3]. For instance, an example scenario would execute as follows: Two performers, an adult and a child performers, are tracked by a 3D-motion tracking device and their locations on the stage are tracked using pressure sensors. ARIA continuously monitors the position of the body markers of performers in 3D space, the positions of performers and details of their gestures. The output of the 3D motion tracking is filtered through ARIA to obtain the shape and degree of confidences of the pose. For instance, the adult performer is scripted to draw shapes in the air with his arm. The shapes are recognized by ARIA with the degree-of-confidence and the recognized shape will drive the image from media repository to be projected on the stage. The degree of confidence of recognition and relative spatial locality of the two performers will be fused to generate the color pattern of images to be projected. The image and color information is transferred to the projection actuators.

2. ARIA GUI AND KERNEL

ARIA media processing workflows are modeled as object flow and transformation graphs where vertices (or nodes) represent sensors, filters, fusion operators, and actuators and edges represent connections that stream objects between components (Figure 1(a)). The basic information unit is a data object. Depending on the task, an object can be as simple as a numeric value (such as an integer denoting the pressure applied on a surface sensor) or as complex as an image component segmented out from frames in a video sequence. Each object streamed between two ARIA components contains *an object payload*, such as a string, a numeric value, or an image region, and *a meta-data header*, describ-

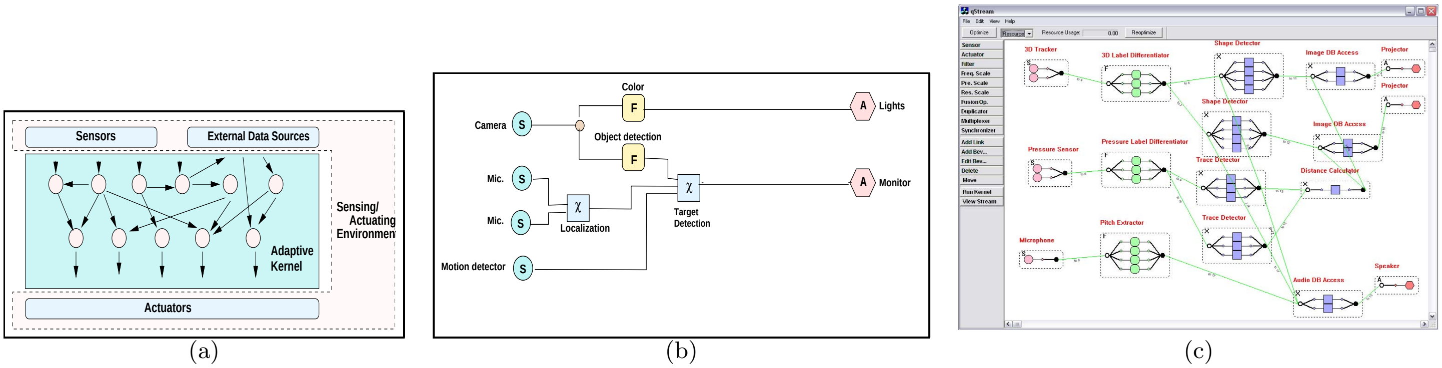

Figure 1: (a) ARIA media processing workflow architecture; (b)an example media processing workflow: "S" denotes sensors, "F" denotes filters, "χ" denotes fusion operators, and "A" denotes actuators; and (c) ARIA GUI for editing media workflows; most of the nodes in the workflow have multiple/alternative behaviors enabling them to adapt to runtime characteristics of the system

ing the object properties, such as the *size* of the object, the *precision* of the object, a *history* consisting of the *resource usage stamps* and *timestamps* acquired as the object goes through various operators.

Sensors act as stream sources. For example, a sequence of 2-byte surface pressure values, measured within 99.9% precision and generated every 10 milliseconds by a floor sensor, forms an object stream. While sensors generate object streams, actuators consume object streams and map them to appropriate outputs. A *filter* takes an object stream as input, processes and transforms the objects in the stream, and outputs a new stream consisting of the transformed objects. For example, consider a module that takes a stream of facial images as its input and returns a face signature vector as its output. This module is a transforming filter. Note that the precision of the result may depend on the number of consecutive faces considered or may depend on what type of a heuristic/neural-net is used. Consequently, filters are scalable and may provide multiple precisions, each with its own end-to-end delay and resource requirement. A *fusion* operator is similar to a filter, except that it takes as its input multiple streams and returns as its output multiple streams. For example, consider a module which receives object-tracking information from multiple redundant sensors and outputs *fused* highly-precise object-tracking information. Fusion operators are also scalable; the total fusion delay, as well as other resources, would depend on the degree of the required precision.

The quality-adaptive nature of ARIA is due to the components of the data flow architecture (i.e., the ARIA Kernel) that are scalable and programmable (Figures 1(b)) and adaptable: delay and quality characteristics of individual operators can be controlled via a number of parameter values. Operator specifications used by the ARIA scheduler and optimizer include (1) alternative characteristics of the sensors; (2) precisions and computational overheads of the filters (such as object feature extractors); (3) precisions and computational overheads of the fusion operators; and (4) characteristics of the actuators. QoS specifications include sensor-to-actuator delay, precision, resource, and frequency constraints. In general, due to various transformations possible during the processing of a media object, there are tradeoffs between desired characteristics, such as the delay and perceptual distortion. The trade-offs at the operator level result in trade-offs between the performance objectives. Since optimization problems with trade-offs between objectives are generally intractable, we developed efficient heuristics that provide very-close-optimal routes [1, 4, 5] applicable to different scenarios.

3. CONCLUSION

In this demo, we present the GUI and the adaptive kernel of the novel media processing flow architecture, ARIA, to support interactive sensing/actuating environments. ARIA workflows extract features from streamed data, fuse and map streams onto output devices, and satisfy the QoS requirements using quality-adaptive operators. ARIA offers a new medium which will allow artists to integrate novel sensing, interaction, content, and response mechanisms in stage performances, thus enabling digital media and art researchers to expand the expressive and interactive possibilities in multimedia environments.

4. REFERENCES

[1] K.S. Candan, L. Peng, K.D. Ryu, K.S. Chatha and C. Mayer. *Efficient Stream Routing in Quality- and Resource-Adaptive Flow Architectures.* Multimedia Information Systems (MIS), 2004.

[2] L. Peng, K.S. Candan, K.D. Ryu, K.S. Chatha and H. Sundaram. *ARIA: An Adaptive and Programmable Media-flow Architecture for Interactive Arts.* ACM MM Arts Program, 2004.

[3] L. Peng and K. S. Candan, *Adaptive Multi-Sensor, Multi-Actuator Media Workflow System for Interactive Arts.* Workshop on Managing Data for Emerging Multimedia Applications (EMMA), 2005.

[4] L. Peng, K.S. Candan, C. Mayer, K.D. Ryu and K.S. Chatha. *Optimization of Media Processing Workflows with Adaptive Operator Behaviors.* to appear in Multimedia Tools and Applications, Kluwer.

[5] L. Peng and K. S. Candan. Confidence-driven Early Object Elimination in Quality-Aware Sensor Workflows. 2nd International VLDB Workshop on Data Management for Sensor Networks (DMSN), 2005.