Modeling and Verification of Interactive Flexible Multimedia Presentations Using PROMELA/SPIN

Ramazan Savaş Aygün and Aidong Zhang

Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo NY 14260-2000, USA
{aygun,azhang}@cse.buffalo.edu

Abstract. The modeling and verification of flexible and interactive multimedia presentations are important for consistent presentations over networks. There has been querying tools proposed whether the specification of a multimedia presentation satisfy inter-stream relationships. Since these tools are based on the interval-based relationships, they cannot guarantee the verification in real-life presentations. Moreover, the irregular user interactions which change the course of the presentation like backward and skip are not considered in the specification. Using PROMELA/SPIN, it is possible to verify the temporal relationships between streams using our model allowing irregular user interactions. Since the model considers the delay of data, the author is assured that the requirements are really satisfied.

1 Introduction

There have been models proposed for the management of multimedia presentations. The synchronization specification languages like SMIL [10] have been introduced to properly specify the synchronization requirements. Multimedia query languages have been developed to check the relationships defined in the specification [5]. These tools check the correctness of the specification. However, the synchronization tools do not satisfy all the requirements in the specification or put further limitations. Moreover, the specification does not include user interactions. The previous query-based verification techniques cannot verify whether the system remains in a consistent state after a user interaction.

There are also verification tools to check the integrity of multimedia presentations [7]. The user interactions are limited and interactions like backward and skip are ignored. This kind of interactions is hard to model. The Petri-Nets are also used to verify the specification of multimedia presentations [9]. But Petri-Net modeling requires complex Petri-Net modeling for each interaction possible. Authors usually do not have much information about Petri-Nets.

⋆ This research is supported by NSF grant IIS-9733730 and NSF grant IIS-9905603.
PROMELA/SPIN is a powerful tool for modeling and verification of software systems [6]. Since PROMELA/SPIN traces all possible executions among parallel running processes, it provides a way of managing delay in the presentation of streams. In this paper, we discuss the properties that should be satisfied for a multimedia presentation. We report on the complexity introduced by user interactions. The experiments are conducted for parallel, sequential, and synchronized presentations.

This paper is organized as follows. The synchronization model and PROMELA are discussed in Section 2. Section 3 explains the properties that should be satisfied for a multimedia presentation. Section 4 reports the experiments. The last section concludes our paper.

2 Modeling of a Multimedia Presentation

The synchronization model is based on synchronization rules [2]. Synchronization rules form the basis of the management of relationships among the multimedia streams. Each synchronization rule is based on the Event-Condition-Action (ECA) paradigm. Our synchronization model has receivers, controllers and actors to handle events, condition expression and actor expression, respectively. Timelines are kept for receivers, controllers, actors and actions to keep track when events are signaled, when conditions are satisfied, and when actions start and end. This synchronization model is favored over the others since it allows interactions that change the course of the presentation.

2.1 Presentation

The presentation can be in idle, initial, play, forward, backward, paused, and end states (Figure 1(a)). The presentation is initially in the idle state. The user interface is based on the model presented at [3]. When the user clicks START button, the presentation enters play state. The presentation enters end state when the presentation ends in the forward presentation. The presentation enters the initial state when it reaches its beginning in the backward presentation. The user may quit the presentation at any state. Skip can be performed in play, forward, backward, initial, and end states. If the skip is clicked in play, forward, and backward states, it will return to the same state unless skip to initial or end state is not performed. If the presentation state is in end or initial states, skip interaction will put into the previous state before reaching these states.

2.2 Containers and Streams

A container or a stream may enter 4 states. A container is in IdlePoint state initially. Once started, a container is at InitPoint state in which it starts the containers and streams that it contains. After the InitPoint state, a container enters its RunPoint state. In RunPoint state, a container has some streams that are being played. When all the streams it contains reach their end or when the container is notified to end, it stops execution of the streams and signals its end and then enter idle state again. In the backward presentation, the reverse path is followed (Figure 1(b)). If a stream has to signal an event, a new state is added