

Creating Non-decomposable Stream Bundles in Multimedia Web Services using uSMIL*

Work in Progress

Naren Kodali
Dept of Info. and Soft. Eng.
George Mason University,
Fairfax, VA 22030-4444
nkodali@gmu.edu

Csilla Farkas
Info. Security Laboratory
Dept of Comp. Sci. and Eng.
University of South Carolina,
Columbia, SC 29208
farkas@cse.sc.edu

Duminda Wijesekera
Center for Secure Info. Syst.,
George Mason University,
Fairfax, VA 22030-4444
dwijesek@gmu.edu

ABSTRACT

Technical advances in networks, operating systems and web based services are enabling new web-based services such as audiovisual surveillance, monitoring and recording of unfolding real world scenes. Leaving room to alter the chronology of such real-world recordings would facilitate the ability to fabricate audiovisual evidence, thereby seriously eroding emerging public trust on these up and coming services. One way of avoiding this problem is to be able to specify which chronology building composition operators - otherwise known as multimedia synchronization constructs - such as *before*, *after*, *simultaneously* etc., have more rigid failure semantics. Existing W3C standard for multimedia services, the *synchronized multimedia integration language (SMIL)* does not provide them. In order to address this deficiency, we propose *uSMIL* - an extension of SMIL that allows the creation of non-decomposable stream bundles with an *all or nothing* playout semantics. Here we present a report on our work in progress that defines a formal operational semantics for uSMIL and show some of its algebraic properties.

Categories and Subject Descriptors: F.3.2 [Semantics]: Algebraic approaches to semantics

General Terms: Security, Design, Theory.

Keywords: Synchronized Multimedia, Semantics, Integrity, Evidence Fabrication.

1. INTRODUCTION

As service provisioning is gaining popularity and multimedia is becoming a more convincing medium to convey more information on the web, many new applications such

*This work was partially supported by the National Science Foundation under grants CCS-0113515 and IIS-0112874.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM Workshop on Secure Web Services, October 29, 2004, Fairfax VA, USA.

Copyright 2004 ACM 1-58113-973-X/04/0010.... ..\$5.00.

as internet movies, digital music libraries, electronic surveillance over the web onto handheld devices, real-time traffic management, secure video conferencing are becoming available as web based services. In recognition of this trend the W3C proposed a language *synchronized multimedia integration language (SMIL)* to specify multimedia compositions to be distributed over the world-wide web. SMIL provides many constructs such as simultaneous and sequential playout of evolving scenarios etc., that can be used to mix and match many real-time streaming media sources to convey a well orchestrated unfolding of incidents. Among those there are many applications such as surveillance and monitoring where preserving the exact chronology of evolution is paramount to the comprehension of reality. If not done with care, recording of such histories can be used to manufacture or alter the chronology and thereby fabricate or tamper with audio-visual evidence. One way to preserve recorded audiovisual reality is to ensure that constructs used to define the synchrony and the chronology of evolution be given rigid, non-alterable semantics, so that any attempted alteration results in losing all of the constituent media streams. If possible, that would lead to the creation of non-decomposable bundles of multimedia streams, which is the subject matter of our current investigation - of which we present some preliminary results.

We provide such a capability by providing chronology creating operators of media streams - otherwise known as multimedia synchronization constructs that *cannot be broken*, thereby referred to as *unbreakable synchronization constructs*. Existing SMIL syntax do not provide such constructs. Consequently, we enhance SMIL to *SMIL with unbreakable synchronization constructs (uSMIL)* where parallel and sequential constructs require that all their components be played out or none of them are played. We provide formal operational semantics for our constructs and show some of their properties. Although this does not provide security in a traditional sense, our constructs provide an important aspect of security and integrity of surveillance services, thereby increasing their social credibility and reliability.

SMIL [1] is an XML-like language for authoring multimedia documents. Described in more detail in Section 3, it has media intervals as atomic entities (multimedia display units) and combine them using operators that enable parallel and sequential playout, conditional playing of multimedia

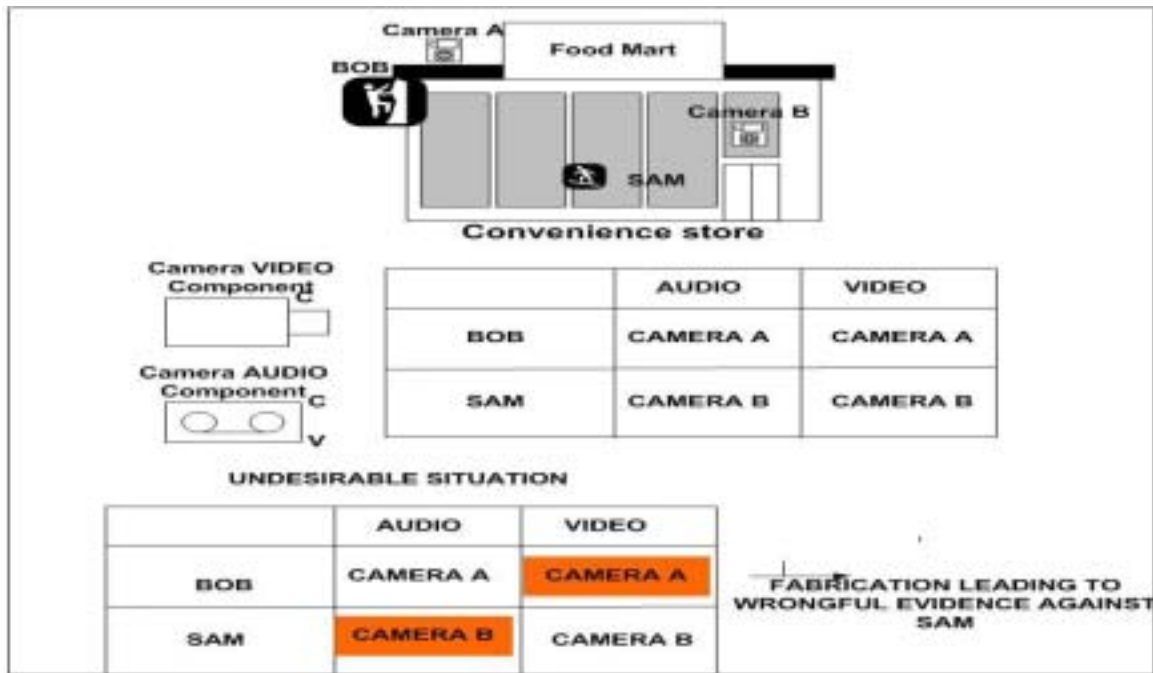


Figure 1: Motivating Example: Need for Unbreakability

streams and repeating media streams until some condition is satisfied, etc. Existing model and solutions for text data based web services do not apply to multimedia *as is* because multimedia has a sense of time and continuity that is absent in the former. Therefore, surveillance and similar multimedia services on the semantic web [2] require time and media continuity sensitive information flow in addition to traditional secure and trusted two-way communication. However using traditional SMIL allows improper recombining or dropping audio, video segments captured by strategically deployed devices that capture live and unfolding scenes. This is important in the context of surveillance as it does not prevent chronology re-engineering and fabricating histories, that can be exploited to produce incriminating evidence. As shown in the motivating example in figure 1, Bob is trying to break into a grocery store and his audio and video are captured by camera A. Sam is an customer in the store, ignorant of the burglary attempt and his audio and video are being captured by camera B. Existing SMIL constructs could allow malicious or inadvertent fabrication by mixing the video of Bob with the audio of Sam leading to a situation that incriminates both Bob and Sam together.

Therefore, there is a need to enhance the current SMIL syntax, and define constructs that disallow this ability as needed. Such *unbreakable synchronization constructs* when used in combination with existing SMIL constructs provide the flexibility needed to author secure audio-visual data gathering applications while respecting the intended meaning without permitting malicious mis-interpretation.

The rest of the paper is organized as follows. Section 2 describes related work and Section 3 briefly introduces SMIL. Section 4 describes the syntactic additions proposed and provides their semantics. Section 5 has some properties of proposed constructs. Section 6 concludes the paper.

2. RELATED WORK

A distributed architecture for multi-participant and interactive multimedia that enables multiple users to share media streams within a networked environment is presented in [17]. In this architecture, multimedia streams originating from multiple sources can be combined to provide media clips that accommodate look-around capabilities.

Regulating access to XML formatted text documents has been actively researched in the recent years. Bertino et al. [3] have developed Author-X, a Java based system to secure XML documents that enforces access control policies at various granularities and corresponding user credentials. Author-X encodes security policies for a set of XML documents in an XML file referred to as the policy base containing both permissions and prohibitions. Damiani et al. [6, 7] developed an access control model where the tree structure of XML documents is exploited using XPATH expressions. The smallest protection granularity is an XPATH node, and security policies specify permissions or prohibitions to all descendent objects of a node.

In the realm of multimedia, substantial amounts of contemporary research addresses real-time moving object detection and tracking them from stationary and moving camera platforms [18], object pose estimation with respect to a geospatial site model, human gait analysis [19], recognizing simple multi-agent activities, real-time data dissemination, data logging and dynamic scene visualization. Damiani et al. [5] discuss feature protection of SVG format images. Its primary focus is controlled dissemination of sensitive data within an image. They propose an access control model with complex filtering conditions. Bertino et al. [4] provides a security framework to model access control in video databases. They provide security for objects which are sequences of frames or particular objects within frames. Their access control model is based on the concepts of security ob-

jects, subjects, and permitted access modes, like viewing and editing.

Kodali et al. [10, 13] proposed an MLS application for secure surveillance of physical facilities, where guards with different security classification in charge of the physical security of the building are provided live feeds matching their level in the security hierarchy. This paper is an extended version of [9], in which multimedia surveillance is described with limited operational semantics. Kodali et al. propose models for multimedia access control for different security paradigms. A release control for SMIL formatted multimedia objects for pay-per-view movies on the Internet that enforces DAC is described in [14]. RDF [16] based security architectures for multimedia security [12] and authorization models for secure digital libraries [8] have also been proposed. A secure and progressively updatable SMIL document [15] is used to enforce RBAC and respond to traffic emergencies.

3. SMIL: SYNCHRONIZED MULTIMEDIA INTEGRATION LANGUAGE

SMIL [1] is an extension to XML developed by W3C to author multimedia presentations with audio, video, text and images to be integrated and synchronized. The distinguishing features of SMIL over XML are the syntactic constructs for timing and synchronizing live and stored media streams with qualitative requirements.

SMIL constructs for synchronizing media are `<seq>`, `<excl>` and `<par>`. They are used to hierarchically specify synchronized multimedia compositions. Media composed with `<seq>` element plays its children one after another in sequence. Media composed using `<excl>` plays all of its components, one at a time without imposing any order. Media composed with `<par>` plays all components parallelly. For example, the SMIL specification `<par>video,src=camera1 <audio, src = microphone1 </par>` plays camera1 and microphone in parallel.

In SMIL, the time period that a media clip is played out is referred to as its *active duration*. For parallel play out to be meaningful, both sources must have equal active durations. When clips do not have equal active durations, SMIL provides many constructs to equate them. Some examples are `begin` (allows to begin components after a given amount of time), `dur` (controls the duration), `end` (specifies the ending time of the component with respect to the whole construct), `repeatCount` (allows a media clip to be repeated a maximum number of times). In addition, attributes such as `syncTolerance` and `syncMaster` controls runtime synchronization, where the former specifies the tolerable mis-synchronization (such as tolerable lip-synchronization delays) and the latter specifies a master-slave relationship between synchronized streams. In this paper we assume that children of `<par>` have active durations.

4. FORMALIZING SMIL SEMANTICS

Consider the audio and video frame pairs A2, V2 played in parallel after audio video pairs A1 and A2 as shown in Figure 2. The lower right hand corner shows two possible SMIL trees specifying this layout. The constructs `<par>` and `<seq>` are used by SMIL to specify parallel and sequential layouts respectively. SMIL uses media intervals beginning and ending at specified times as basic playout elements. Although

specified using XML-like syntax, SMIL trees do not convey the informal runtime semantics intended of multimedia synchronization. In order to precisely specify them [11, 12] formalized a semantics for SMIL trees using media intervals. Naturally, in that semantics any media presentation can be specified as a parallel playout of sequence of media frames. This can also be achieved using SMIL syntax, referred to as the *normal form* of a SMIL specification. [11, 12] provides some basic algebraic identities that can be used to rewrite a SMIL tree into its normal form. We now recall these definitions. First, we formalize the semantics.

Definition 1 (Timed Display Instance and Sets). We say that a triple $(type, t_{begin}, t_{end})$ is a *timed display instance (TDI)* provided that:

1. *type* is the name of a media type such as *audio*, *video*, $t_{begin} \leq t_{end}$ are arithmetic expressions of a single real number variable t satisfying $t_{end} = t_{begin} + \delta$.
2. A set of timed display instances is said to be a *timed display set (TDS)* parameterized by t provided that at least one timed display element begins at t .
3. Taken as arithmetic expressions containing the variable t , the smallest t_{begin} value of a timed display set is said to be the *origin* of the timed display set. We use the notation $O(TDS)$ for the origin of the timed display set TDI. The largest t_{end} value of a timed display set is said to be the *end* of the timed display set. We use the notation $E(TDS)$ for the end of the timed display set TDS.

The following representations $tdi_1 = (video, t, t+7)$ and $tdi_2 = (audio, t+7, t+17)$ are examples of timed display instances.

Definition 2 (Time Shift Operator). Suppose TDS is a timed display set parameterized t and s . Then $TDS(s/t)$ is the timed display set obtained by syntactically substituting s for t .

For the example given before 2, $TDS(2t+7/t)$ consists of $\{tdi_1(2t+7/t), tdi_2(2t+7/t)\}$ where $tdi_1(2t+7/t) = (video, 2t+7, 2t+21)$ and $tdi_2(2t+7/t) = (audio, 2t+21, 2t+31)$. We now use these basic notations to formalize semantics mapping of SMIL specifications.

Definition 3 (Basis Mapping). Suppose M is the set of basic media elements of any SMIL specification. A mapping $[[]]$: $M \mapsto TDS$ is a *basis* iff all T -begin values of M have the same value t . Then we say that $[[]]$ is a *basis mapping* parameterized by t .

Lemma 1 (Existence of basis mappings). Suppose M is a set of basic media streams with time durations. Then M has a *basis mapping*.

The proof is given in [11, 12]. We now use a basis mapping to define operational semantics of any SMIL specification as follows.

Definition 4 (Operational Semantics for SMIL). Suppose S is a SMIL specification and $[[]]$ is a basis mapping for the atomic media intervals B of S with the formal parameter t . Then we inductively extend $[[]]$ to S as follows.

1. $[[Null]] = \Phi$

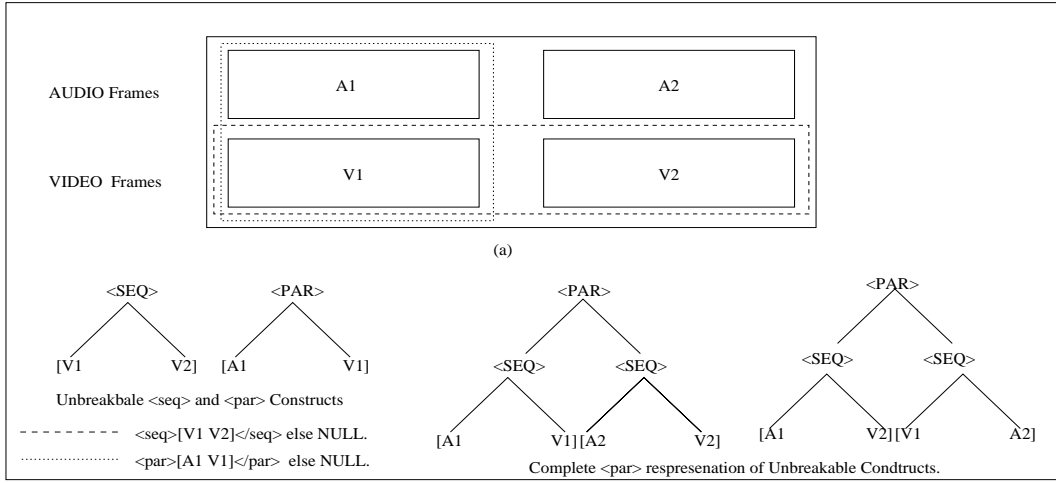


Figure 2: Unbreakable SMIL Constructs

2. $[[\langle seq \rangle S_1 S_2 \langle /seq \rangle]] = [[S_1]] \cup [[S_2]](end([[S_1]])/t)$
3. $[[\langle par \rangle S_1 S_2 \langle /par \rangle]] = [[S_1]] \cup [[S_2]]$.

We say that $[[\]]$ is a semantic mapping parameterized by t . The stated informal notion of meaning is captured by our operational semantics. Under this mapping, any SMIL tree is mapped to a time display set - namely to the set of atomic media intervals with precise offsets with respect to an origin t . The mapping $[[\]]$ lists the set of all possible media intervals, among which all, some or neither may be displayed. But for the unbreakable synchronization constructs the second possibility is forbidden - and therefore the set of all permissible possibilities have to be explicitly listed. Consequently, we need to replace all time display sequences with explicitly permissible sets of them. For example consider playing A1 and V1 of Figure 2 unbreakably using a $\langle par \rangle$ construct. Then the only two combinations that are allowed to be played are $\{\emptyset\}$, $\{(audio, t_{begin}(A1), t_{end}(A1)), (video, t_{begin}(V1), t_{end}(V1))\}$. The other two options of playing A1 without V1 or V1 without A2 are prohibited.

We use $\langle Upar \rangle$ and $\langle Useq \rangle$ as names for unbreakable parallel and sequential composition of media streams. In our new semantics, two streams are connected by a $\langle Upar \rangle$ if they begin and end playout at the same time and are always played as a inseparable group(enclosed by a dotted line in Figure 2). Similarly, two streams are connected by a $\langle Useq \rangle$ if the second mandatorily begins when the first ends and either both streams are played as an inseparable group(enclosed by a dashed line in Figure 2 in sequence or none of them are played. A representation using the proposed constructs shown in part c of Figure 2 is $\langle Useq \rangle \langle Upar \rangle [A_1, V_1] \langle /Upar \rangle \langle Upar \rangle [A_2, V_2] \langle /Upar \rangle \langle /Useq \rangle$. We now show the semantic enhancements necessary to enforce unbreakability.

4.1 Semantics of Unbreakability

Definition 5 (Unbreakable Time Display Instance).

We say that a quadruple $(type, t_{begin}, t_{end}, index)$ is an unbreakable timed display instance (UTDI) provided that:

1. $type$ is audio, video or a name of a basic media type and $t_{begin} \leq t_{end}$ are arithmetic expressions of a single real variable t .

2. $index$ is a positive integer.

3. We say that a set of unbreakable timed display instances is an unbreakable timed display set (UTDS) provided that there is at least one timed display element with t as its t_{begin} value, and the set of indices in the elements form an interval $[1, n]$ for some integer n .
4. Taken as arithmetic expressions containing the variable t , the smallest t_{begin} value of a timed display set is said to be the origin of the timed display set. We use the notation $O(UTDS)$ for the origin of the timed display set UTDS. The largest t_{end} value of a timed display set is said to be the end of the timed display set. We use the notation $E(UTDS)$ for the end of the unbreakable timed display set UTDS.

There are two major differences between timed display sequences of given in definitions 1 and 5. The first does not have an index as the fourth coordinate of a time display instance, but the latter does. It is used in the latter to identify two instances of the same media interval. For example $\langle par \rangle tdi_1, tdi_1 \langle /par \rangle$ constitutes playing the same media interval in parallel. By having two indices to separate these two instances we can distinguish tdi_1 from $\langle par \rangle tdi_1, tdi_1 \langle /par \rangle$ as the former maps to $\{\emptyset, \{(audio, t_{begin}, t_{end}, 1)\}\}$ and the latter maps to $\{\emptyset, \{(audio, t_{begin}, t_{end}, 1)\}, \{(audio, t_{begin}, t_{end}, 2)\}\}$. Notice that without distinct indices, both these sets become the same.

The second difference is that definition 1 has sets of time display sequences for timed display sets and definition 5 uses sets of sets of unbreakable timed display sequences for the same. The need for the extra level of set embedding is necessary to group the unbreakable combinations together in the latter as that was not a modelling requirement of the former.

The following representations $tdi_1=(video, t, t+7, 1)$ and $tdi_2=(audio, t+7, t+17, 2)$ are examples of unbreakable timed display instances. Consequently, $\{\emptyset, \{tdi_1\}, \{tdi_2\}, \{tdi_1, tdi_2\}\}$ is an unbreakable time display set denoting that one, both or neither of tdi_1, tdi_2 can be played. Similarly, $\{\emptyset, \{tdi_1, tdi_2\}\}$ is another time display set denoting that either both nor neither of tdi_1, tdi_2 are played. As explained, the difference is

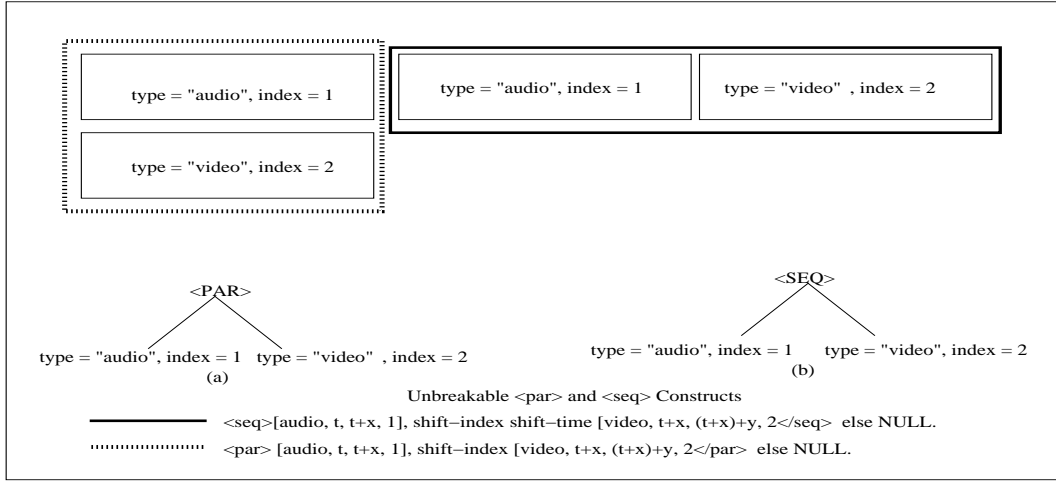


Figure 3: Unbreakable Synchronization Constructs

the semantic dichotomy between a breakable sequential construct and its unbreakable counterpart.

Now we describe an algebraic operation on timed display sets required to define our operational semantics of breakable and unbreakable SMIL constructs. The first is that of index shift operation on TDI sets defined as follows.

Definition 6 (Index Shift). Define

1. $Ishift(k, (type, t_{begin}, t_{end}, index)) = (type, t_{begin}, t_{end}, k + index)$
2. $Ishift(k, UTDS) = \{Ishift(k, UTDI) \mid UTDI \in UTDS\}$

The index shift operation is necessary because definition 5 requires the set of indices of a UTDI set to be a closed interval. Thus, preserving this property across synchronization constructs require reassigning indices of some timed display instances, as will be seen shortly. First we redefine a basis mapping.

Definition 7 (Parameterized u-Basis Mapping). Suppose $\mathcal{B}(\mathcal{C})$ is the set of atomic media intervals used in a collection \mathcal{C} of SMIL constructs. Any mapping $[] : \mathcal{B}(\mathcal{C}) \mapsto UTDI$ from $\mathcal{B}(\mathcal{C})$ to a set of timed display instances $UTDI$ is said to be a u-basis (unbreakable basis) mapping parameterized by t iff all t_{begin} elements of $\mathcal{B}(\mathcal{C})$ have the same value t for some real variable.

Again the following can be easily shown.

Lemma 2 (Existence of basis Mappings). Every set of atomic media streams $\mathcal{B}(\mathcal{C})$ with time duration has a basis mapping that preserves the beginning and end times.

The proof is the same as the previous case. We now use the basis mapping to formalize the meaning of the enhancements proposed for SMIL unbreakability. We now use these basis mappings to formally define an operational semantics of extended uSMIL.

Definition 8 (Semantics of uSMIL Constructs). Suppose \mathcal{C} is a collection of uSMIL constructs and $[]$ is a u-basis mapping from $\mathcal{B}(\mathcal{C})$ to some timed display set $UTDS$. Then we extend $[]$ to \mathcal{C} as follows.

1. $[\langle par \rangle S_1 S_2 \langle /par \rangle] = \{\emptyset\} \cup [S_1] \cup Ishift(size(S_1), [S_2]) \cup \{P \cup Ishift(size(S_1), Q) \mid P \in [S_1], Q \in [S_2]\}$.
2. $[\langle seq \rangle S_1 S_2 \langle /seq \rangle] = \{\emptyset\} \cup [S_1] \cup Ishift(size(S_1), [S_2](E(S_1/t))) \cup \{P \cup Ishift(size(S_1), Q(E(S_1/t))) \mid P \in [S_1], Q \in [S_2]\}$.
3. $[\langle Upar \rangle S_1 S_2 \langle /Upar \rangle] = \{\emptyset\} \cup [S_1] \cup Ishift(size(S_1), [S_2])$.
4. $[\langle Useq \rangle S_1 S_2 \langle /Useq \rangle] = \{\emptyset\} \cup [S_1] \cup Ishift(size(S_1), [S_2](E(S_1/t)))$.

As stated in definition 8, already known SMIL operators **seq** and **par** list all possible combinations of sets from both components that are conjoined by the construct. Conversely, the unbreakable constructs **Useq** and **Upar** either allow only the combination without being able to break them into constituents. As seen from definition 8, the only extra part in **par** that is absent from **Upar** is $\{P \cup Ishift(size(S_1), Q) \mid P \in [S_1], Q \in [S_2]\}$. Notice that P and Q are any UTDI's in $[S_1]$ and $[S_2]$ respectively, and this extra set shifts the index of Q and plays them in parallel. Consequently, if P is the null UTDI (indicating silence) then it plays out only Q and vice-versa. A similar contrast exists between **seq** and **Useq**.

As shown in part (a) of Figure 3 $\langle Upar \rangle (audio, t_{begin}, t_{end}, 1), (video, t_{begin}, t_{end}, 2) \langle /par \rangle$, either plays both audio and the video components together (i.e. begins and ends at the same time) or neither (i.e. resulting in silence). Similarly, as shown in part (b) of Figure 3 $\langle Useq \rangle (audio, t_{begin}, t_{end}, 1), (video, t_{begin}, t_{end}, 2) \langle /par \rangle$, either plays both audio and the video components in that order or plays neither of them resulting in a long silence in both streams.

As stated earlier, the only reason for having an index in the definition of UTDI's is (i.e. given in Definition 5 as $(type, t_{begin}, t_{end}, index)$) is to ensure that two copies are not identified as sets. Thus the particular value of the index does not add any extra information to the semantics. Consequently, we consider all semantic mappings that differ in the indexing to be equivalent, and take the equivalence class as the meaning, formalized as follows.

Definition 9 (Equivalence index reindexing). We say that $[A] \cong [B]$ iff there is a bijection $f : I_A \rightarrow I_B$ such that $[B] = \{(type, t_{begin}, t_{end}, f(i)) : i \in I_A\}, (type, t_{begin}, t_{end}, i) \in [A]\}$

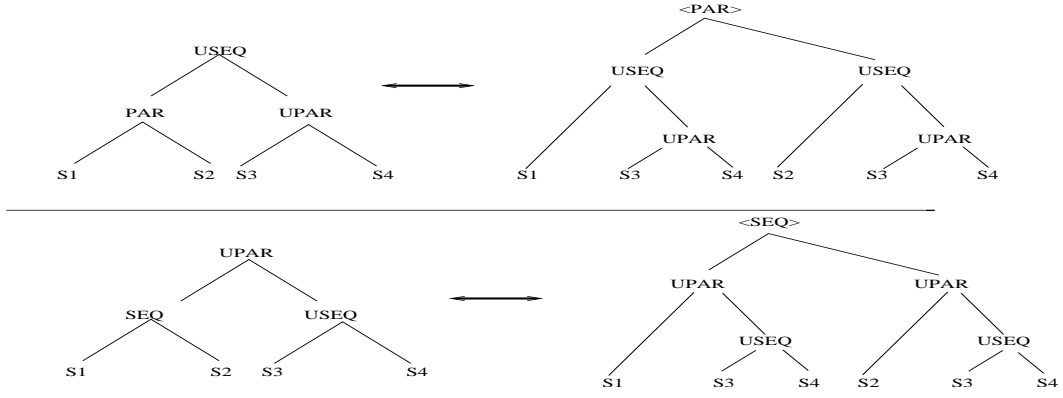


Figure 4: Parallel and Sequential Reducibility: Unbreakable Parent

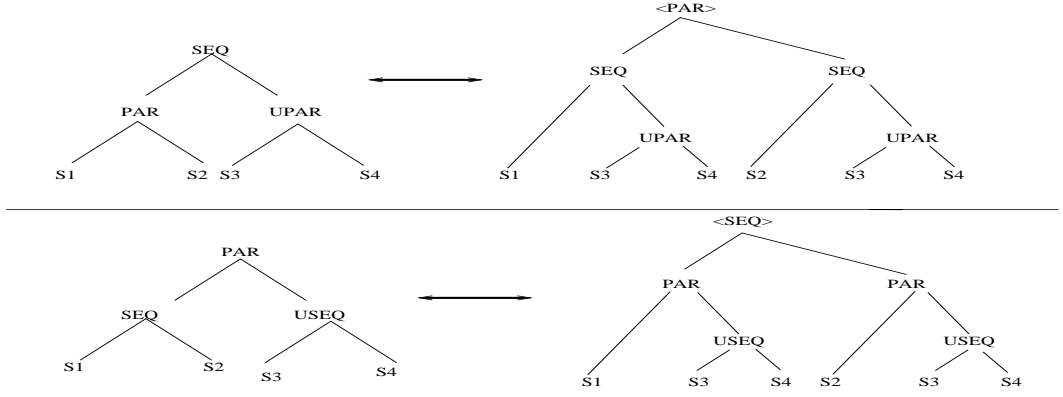


Figure 5: Parallel and Sequential Reducibility : Breakable Parent

Lemma 3 (Existence of an Equivalence Class). \cong is an equivalence relation on semantic maps $[\]$ from \mathcal{C} to $UTDI$.

It is trivial to verify that \cong is an equivalence class. Using this equivalence relation, we now define the semantic map as follows.

Definition 10 (Semantics of uSMIL). Let $[[A]] = [A]/\cong$. We say that for any uSMIL specification tree S , $[[S]]$ is its operational semantics.

5. ALGEBRA OF UNBREAKABILITY - SOME PRELIMINARY RESULTS

Synchronization constructs par , seq satisfy the following properties.

1. $[[\langle \text{par} \rangle S_1 S_2 \langle /\text{par} \rangle]] = [[\langle \text{par} \rangle S_2 S_1 \langle /\text{par} \rangle]]$
2. $[[\langle \langle \text{par} \rangle S_1 S_2 \langle /\text{par} \rangle \rangle \langle \langle \text{par} \rangle S_3 \langle /\text{par} \rangle \rangle]] = [[\langle \langle \text{par} \rangle S_1 \langle /\text{par} \rangle \rangle \langle \langle \text{par} \rangle S_2 S_3 \langle /\text{par} \rangle \rangle]]$
3. The identity $\langle \langle \text{seq} \rangle S_1 S_2 \langle /\text{seq} \rangle \rangle = \langle \langle \text{seq} \rangle S_2 S_1 \langle /\text{seq} \rangle \rangle$ does not always hold.
4. $[[\langle \langle \text{seq} \rangle S_1 S_2 \langle /\text{seq} \rangle \rangle \langle \langle \text{seq} \rangle S_3 \langle /\text{seq} \rangle \rangle]] = [[\langle \langle \text{seq} \rangle S_1 \langle /\text{seq} \rangle \rangle \langle \langle \text{seq} \rangle S_2 S_3 \langle /\text{seq} \rangle \rangle]]$
5. The identity $[[\langle \text{par} \rangle S_1 S_1 \langle /\text{par} \rangle]] = [S_1]$ does not always hold.

6. The identity $[[\langle \text{seq} \rangle S_1 S_1 \langle /\text{seq} \rangle]] = [S_1]$ does not always hold.

Synchronization constructs Upar and Useq satisfy the following properties.

1. $[[\langle \langle \text{Useq} \rangle \langle \langle \text{par} \rangle S_1 S_2 \langle /\text{par} \rangle \rangle \rangle \langle \langle \text{Upar} \rangle S_3 S_4 \langle /\text{Upar} \rangle \rangle \langle /\text{Useq} \rangle \rangle]] = [[\langle \langle \text{par} \rangle \langle \langle \text{Useq} \rangle S_1 \langle \langle \text{Upar} \rangle S_3 S_4 \langle /\text{Upar} \rangle \rangle \rangle \langle /\text{Useq} \rangle \rangle \langle \langle \text{Useq} \rangle S_2 \langle \langle \text{Upar} \rangle S_3 S_4 \langle /\text{Upar} \rangle \rangle \langle /\text{Useq} \rangle \rangle \langle /\text{par} \rangle \rangle]]$
2. $[[\langle \langle \text{Upar} \rangle \langle \langle \text{seq} \rangle S_1 S_2 \langle /\text{seq} \rangle \rangle \rangle \langle \langle \text{Useq} \rangle S_3 S_4 \langle /\text{Useq} \rangle \rangle \langle /\text{Upar} \rangle \rangle]] = [[\langle \langle \text{seq} \rangle \langle \langle \text{Upar} \rangle S_1 \langle \langle \text{Useq} \rangle S_3 S_4 \langle /\text{Useq} \rangle \rangle \rangle \langle /\text{Upar} \rangle \rangle \langle \langle \text{Upar} \rangle S_2 \langle \langle \text{Useq} \rangle S_3 S_4 \langle /\text{Useq} \rangle \rangle \langle /\text{Upar} \rangle \rangle \langle /\text{seq} \rangle \rangle]]$
3. $[[\langle \langle \text{seq} \rangle \langle \langle \text{par} \rangle S_1 S_2 \langle /\text{par} \rangle \rangle \rangle \langle \langle \text{Upar} \rangle S_3 S_4 \langle /\text{Upar} \rangle \rangle \langle /\text{seq} \rangle \rangle]] = [[\langle \langle \text{par} \rangle \langle \langle \text{seq} \rangle S_1 \langle \langle \text{Upar} \rangle S_3 S_4 \langle /\text{Upar} \rangle \rangle \rangle \langle /\text{seq} \rangle \rangle \langle \langle \text{seq} \rangle S_2 \langle \langle \text{Upar} \rangle S_3 S_4 \langle /\text{Upar} \rangle \rangle \langle /\text{seq} \rangle \rangle \langle /\text{par} \rangle \rangle]]$
4. $[[\langle \langle \text{par} \rangle \langle \langle \text{seq} \rangle S_1 S_2 \langle /\text{seq} \rangle \rangle \rangle \langle \langle \text{Useq} \rangle S_3 S_4 \langle /\text{Useq} \rangle \rangle \langle /\text{par} \rangle \rangle]] = [[\langle \langle \text{seq} \rangle \langle \langle \text{par} \rangle S_1 \langle \langle \text{Useq} \rangle S_3 S_4 \langle /\text{Useq} \rangle \rangle \rangle \langle /\text{par} \rangle \rangle \langle \langle \text{par} \rangle S_2 \langle \langle \text{Useq} \rangle S_3 S_4 \langle /\text{Useq} \rangle \rangle \langle /\text{par} \rangle \rangle \langle /\text{seq} \rangle \rangle]]$

Figures 4 and 5 show the breakable parallel construct $\langle \langle \text{par} \rangle S_1 S_2 \langle /\text{par} \rangle \rangle$ and an unbreakable parallel construct $\langle \langle \text{Upar} \rangle S_3 S_4 \langle /\text{Upar} \rangle \rangle$ where S_1, S_2, S_3, S_4 are timed display instances representable as quadruples $(\text{type}, t_{\text{begin}}, t_{\text{end}}, \text{index})$ and synchronized by either $\langle \text{seq} \rangle$ or $\langle \text{Useq} \rangle$, and the tree obtained as a result of the rewriting the the SMIL tree according to the stated properties. Additionally, the figures 4 and 5 also show the breakable sequential construct

$\langle\langle seq \rangle S_1 S_2 \langle /seq \rangle\rangle$ and the unbreakable sequential construct $\langle\langle Useq \rangle S_3 S_4 \langle /Useq \rangle\rangle$ synchronized by either $\langle par \rangle$ or $\langle Upar \rangle$, and the tree obtained as a result of the rewrite. The *intended meaning* is represented by all maintained during the rewrite. These properties would eventually enable us to normalize arbitrary SMIL trees containing both breakable and unbreakable timed display instances to an atomic form, which would be the *Unbreakable Normal form*. The proofs of the stated properties for breakable and unbreakable parents are given in the Appendix.

6. CONCLUSIONS

Due to the need to enforce the condition that some collections of media streams should be played in *all or nothing* fashion, we introduced unbreakable synchronization constructs to SMIL. We then showed how to extend the operational semantics of SMIL to that of uSMIL - SMIL with unbreakable synchronization constructs. We also showed that uSMIL satisfies some algebraic properties. Our objectives in exploring these are to find a syntactic representation of the uSMIL semantics in the form of a *normal form* and find a set of tree transformations that can be used to reduce any SMIL document to its normal form.

7. REFERENCES

- [1] J. Ayars. *Synchronized Multimedia Integration Language*. W3C Recommendation, 2001. <http://www.w3.org/TR/2001/REC-smil20-20010807>.
- [2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *The Scientific American Journal*, 2001.
- [3] E. Bertino, M. Braun, S. Castano, E. Ferrari, and M. Mesiti. Author-x: A java-based system for XML data protection. In *IFIP Workshop on Database Security*, pages 15–26, 2000.
- [4] E. Bertino, M. Hammad, W. Aref, and A. Elmagarmid. An access control model for video database systems. In *Conferece on Information and Knowledge Management*, 2002.
- [5] E. Damiani and S. D. C. di Vimercati. Securing xml based multimedia content. In *18th IFIP International Information Security Conference*, 2003.
- [6] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. Securing XML documents. *Lecture Notes in Computer Science*, 1777:121–122, 2000.
- [7] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. A fine grained access control system for xml documents. *ACM Transactions on Information and System Security*, 5, 2002.
- [8] N. Kodali, C. Farkas, and D. Wijesekera. An authorization model for multimedia digital libraries. In *Journal of Digital Libraries Special issue on Security*, 2003.
- [9] N. Kodali, C. Farkas, and D. Wijesekera. Enforcing integrity in multimedia surveillance. In *IFIP 11.5 Working Conference on Integrity and Internal Control in Information Systems*, 2003.
- [10] N. Kodali, C. Farkas, and D. Wijesekera. Enforcing semantic-aware security in multimedia surveillance. In *Journal of Data Semantics*, 2003.
- [11] N. Kodali, C. Farkas, and D. Wijesekera. Metadata for multimedia access control. In *International Journal of Computer Systems, Science, and Engineering*, 2003.
- [12] N. Kodali, C. Farkas, and D. Wijesekera. Multimedia access control using RDF metadata. In *Workshop on Metadata for Security, WMS 03*, 2003.
- [13] N. Kodali, C. Farkas, and D. Wijesekera. Secrets: A secure realtime emergency response system. In *NSF/NIJ 2nd Symposium on Intelligence and Security Informatics*, 2003.
- [14] N. Kodali and D. Wijesekera. Regulating access to SMIL formatted pay-per-view movies. In *2002 ACM Workshop on XML Security*, 2002.
- [15] N. Kodali, D. Wijesekera, and J.B.Michael. SPUTERS: a secure traffic surveillance and emergency response architecture. In *submission to the Journal of Intelligent Transportaion Systems*, 2003.
- [16] F. Manola and E. Miller. *RDF Primer*. W3C Working Draft, January 23 2003. <http://www.w3.org/TR/2003/WD-rdf-primer-20030123>.
- [17] B. K. Schmidt. An architecture for distributed, interactive, multi-stream, multi-participant audio and video. In *Technical Report No CSL-TR-99-781, Stanford Computer Science Department*, 1999.
- [18] M. VCMS. Field data collection system <http://www.acrcorp.com>.
- [19] VSAM. Video surveillance and monitoring webpage at <http://www-2.cs.cmu.edu/vsam/>.

APPENDIX

A. PROOFS OF THE PROPERTIES

A.1 Properties of Reducibility - Unbreakable Parent

Lemma 4 (Parallel Reducibility - Unbreakable Parent).

$$\begin{aligned} & [\langle\langle Useq \rangle (\langle par \rangle S_1 S_2 \langle /par \rangle) (\langle Upar \rangle S_3 S_4 \langle /Upar \rangle) \langle /Useq \rangle] \\ &= [\langle\langle /par \rangle (\langle Useq \rangle S_1 (\langle Upar \rangle S_3 S_4 \langle /Upar \rangle) \langle /Useq \rangle) (\langle Useq \rangle S_2 (\langle Upar \rangle S_3 S_4 \langle /Upar \rangle) \langle /Useq \rangle) \langle /par \rangle] \end{aligned}$$

Proof:

Rewriting the LHS

$$\begin{aligned} & [\langle\langle par \rangle S_1 S_2 \langle /par \rangle] = \{\emptyset\} \cup [S_1] \cup \text{Ishift}(\text{size}(S_1), [S_2]) \\ & \cup \{P \cup \text{Ishift}(\text{size}(S_1), Q) \mid P \in [S_1], Q \in [S_2]\} \\ &= \{\{\emptyset\}, \{(type, t_{begin}, t_{end}, \text{index}_{S_1}) \cup \text{Ishift}(\text{size}(S_1)), \\ & (type, t_{begin}, t_{end}, \text{index}_{S_2})\}, \{(type, t_{begin}, t_{end}, \text{index}_{S_1})\}, \\ & \{(type, t_{begin}, t_{end}, \text{index}_{S_2})\}\} \\ & [\langle\langle Upar \rangle S_3 S_4 \langle /Upar \rangle] = \{\emptyset\} \cup [S_3] \cup \text{Ishift}(\text{size}(S_3), [S_4]) \\ &= \{\{\emptyset\}, \{(type, t_{begin}, t_{end}, \text{index}_{S_3}) \cup \text{Ishift}(\text{size}(S_3)), \\ & (type, t_{begin}, t_{end}, \text{index}_{S_4})\}\} \end{aligned}$$

Therefore,

$$\begin{aligned} & [\langle\langle Useq \rangle (\langle par \rangle S_1 S_2 \langle /par \rangle) (\langle Upar \rangle S_3 S_4 \langle /Upar \rangle) \langle /Useq \rangle] \\ &= \{\{\emptyset\}, \{(type, t_{begin}, t_{end}, \text{index}_{S_1}), \text{Ishift}(\text{size}(S_1))(type, \\ & t_{begin}, t_{end}, \text{index}_{S_3}) \text{Ishift}(\text{size}(S_3))(type, t_{begin}, t_{end}, \\ & \text{index}_{S_4})\} E (type, t_{begin}, t_{end}, \text{index}_{S_1})/t\}, \{\{\emptyset\}, \{(type, \\ & t_{begin}, t_{end}, \text{index}_{S_2}), \text{Ishift}(\text{size}(S_2)) (type, t_{begin}, t_{end}, \end{aligned}$$

$$t_{end}, index_{S_4}) E((type, t_{begin}, t_{end}, index_{S_3})) \widehat{(E (type, t_{begin}, t_{end}, index_{S_2}))}$$

Now, we consider RHS

$$[(\langle /seq \rangle (\langle par \rangle S_1 (\langle Useq \rangle S_3 S_4 \langle /Useq \rangle) \langle /par \rangle) (\langle par \rangle S_2 (\langle Useq \rangle S_3 S_4 \langle /Useq \rangle) \langle /par \rangle) \langle /seq \rangle)]$$

We resolve each component

$$\langle /seq \rangle (\langle par \rangle S_1 (\langle Useq \rangle S_3 S_4 \langle /Useq \rangle) \langle /par \rangle) \text{ and } \langle /seq \rangle (\langle par \rangle S_2 (\langle Useq \rangle S_3 S_4 \langle /Useq \rangle) \langle /par \rangle)$$

separately and then $\langle seq \rangle$ the result sets.

By definition

$$[\langle Useq \rangle S_1 S_2 \langle /Useq \rangle] = \{\emptyset\} \cup [S_1] \cup \text{Ishift}(size(S_1), [S_2](E(S_1/t)))$$

Combining the above results we get,

$$\begin{aligned} & [(\langle /seq \rangle (\langle par \rangle S_1 (\langle Useq \rangle S_3 S_4 \langle /Useq \rangle) \langle /par \rangle) (\langle par \rangle S_2 (\langle Useq \rangle S_3 S_4 \langle /Useq \rangle) \langle /par \rangle) \langle /seq \rangle)] \\ &= \{ \{\emptyset\}, \{(type, t_{begin}, t_{end}, index_{S_1})\} \{ \text{Ishift}(size(S_1)) \\ & (type, t_{begin}, t_{end}, index_b) \} \{(type, t_{begin}, t_{end}, index_{S_1}) \\ & \text{Ishift}(size(S_1)) (type, t_{begin}, t_{end}, index_b) (E (type, t_{begin}, \\ & t_{end}, index_{S_1})) \} \{(type, t_{begin}, t_{end}, index_{S_3}) \cup \text{Ishift}(size(S_3)) \\ & (type, t_{begin}, t_{end}, index_{S_4}) (E (type, t_{begin}, t_{end}, index_{S_3})) \} \} \\ & \{(type, t_{begin}, t_{end}, index_{S_1}) \text{Ishift}(size(S_1)) (type, t_{begin}, \\ & t_{end}, index_{S_3}) \cup \text{Ishift}(size(S_3)) (type, t_{begin}, t_{end}, index_{S_4}) \\ & E((type, t_{begin}, t_{end}, index_{S_3})) (E (type, t_{begin}, t_{end}, index_{S_1} \\ &)) \} \{(type, t_{begin}, t_{end}, index_{S_2}) \text{Ishift}(size(S_2)) (type, \\ & t_{begin}, t_{end}, index_{S_3}) \cup \text{Ishift}(size(S_3)) (type, t_{begin}, t_{end}, \\ & index_{S_4}) E((type, t_{begin}, t_{end}, index_{S_3})) \widehat{(E (type, t_{begin}, \\ & t_{end}, index_{S_2})) \} \{(type, t_{begin}, t_{end}, index_{S_1}) \text{Ishift}(size(S_1)) \\ & (type, t_{begin}, t_{end}, index_{S_3}) \cup \text{Ishift}(size(S_3)) (type, t_{begin}, \\ & t_{end}, index_{S_4}) E((type, t_{begin}, t_{end}, index_{S_3})) \widehat{(E (type, t_{begin}, \\ & t_{end}, index_{S_1})) \cup (type, t_{begin}, t_{end}, index_{S_2}) \text{Ishift}(size(S_2)) \\ & (type, t_{begin}, t_{end}, index_{S_3}) \cup \text{Ishift}(size(S_3)) (type, t_{begin}, \\ & t_{end}, index_{S_4}) E((type, t_{begin}, t_{end}, index_{S_3})) \widehat{(E (type, t_{begin}, \\ & t_{end}, index_{S_2})) \} \end{aligned}$$

Hence proving

$$[(\langle par \rangle (\langle seq \rangle S_1 S_2 \langle /seq \rangle) (\langle Useq \rangle S_3 S_4 \langle /Useq \rangle) \langle /par \rangle)] = [(\langle /seq \rangle (\langle par \rangle S_1 (\langle Useq \rangle S_3 S_4 \langle /Useq \rangle) \langle /par \rangle) (\langle par \rangle S_2 (\langle Useq \rangle S_3 S_4 \langle /Useq \rangle) \langle /par \rangle) \langle /seq \rangle)]$$