CHAPTER 1

INTRODUCTION

Authoring multimedia presentations that support interaction and adaptation is a challenging issue due to the inherent dynamism in these features. The creation of an effective presentation requires imagination and skill. The primary role of the authoring system would be to model the specifications accurately and to verify them. The presentations are represented by spatial and temporal relations among media objects, which are specified in the form of constraints while authoring. However, some of these constraints could contradict each other and lead to inconsistencies. The verification of these constraints need to be performed during authoring in order to ensure consistency during the playout of the presentation. The presentation system on the other hand, needs to ensure synchronized playouts, and should also be capable of handling interactions and requests for adaption of the content efficiently. The research work carried out deals with the modeling of presentations, consistency checking, synchronization and content adaptation. Efficient approaches that provide design support for these issues have been proposed in the thesis.

1.1 MULTIMEDIA AUTHORING

The term authoring paradigm refers to the way the underlying structure of the multimedia document is presented by the authoring tool. The existing authoring tools have addressed and given solutions to several complex
problems that relate to the authoring process. Several other systems limit the presentation features in order to reduce complexity. The temporal dimension that dominates the multimedia authoring process and several other issues that need special focus have been identified and documented by Bulterman and Hardman (2005). They also provide an overview of the various authoring environments and classify them based on four paradigms: \textit{structured}, \textit{timeline}, \textit{graph} and \textit{scripting}. These paradigms will be briefly presented below along with their merits and demerits.

1. A \textit{structure} based paradigm uses an abstract representation to define the presentation. The most common representation used is that of a document hierarchy or a document tree, with intermediate nodes containing composition elements and root nodes containing pointers to media objects. The paradigm supports explicit representation and manipulation of the structure of a presentation. Multiple instances of the same objects are treated as logically separate entities. \textit{Madeus} (Jourdan et al 1998b, Tran-Thuong and Roisin 2002), is an example of the \textit{structure} based paradigm and is one of the first systems to provide a uniform constraint based specification approach for temporal and spatial relations. It does not support abstract media specification, non-temporal composition or adaptive content. \textit{Madeus} eventually evolved into the \textit{LimSee2} (Deltour et al 2005) authoring system which is an example of the \textit{timeline} based paradigm. The advantage of \textit{structure} based systems is that the specification and manipulation can be done explicitly. Thus, the timing can be visualized and edited in \textit{structure} based paradigms. However, the drawback is the extra authoring effort that is required to create the structure.

2. \textit{Timeline} based paradigms use an explicit time scale as a reference to schedule the media objects involved in the presentation. A single
track is usually assigned for every participating media object and their relative orderings can be emphasized effectively. This paradigm is widely used in commercial authoring systems. The LimSee2 follows the timeline paradigm and supports both direct and constraint based temporal scheduling. The advantage in this paradigm is that the temporal synchronization points can be specified and manipulated explicitly, but it can be effectively used only for non-iterative and non-adaptive presentations.

3. Graph based paradigms model the presentations using directed graphs. A graph provides the author with a visual representation of the logical sequence of the components but time is not explicitly modeled in this approach. Various forms of Timed petri nets (Little and Ghafoor 1990) have also been successfully used by graph based approaches to model the presentations. The Firefly authoring system (Buchanan and Zellweger 1993, Buchanan and Zellweger 2005) is an example of a graph based paradigm. This system integrates a constraint language with a general timeline approach and maps the specifications onto a graph based language. The main advantage of using a graph based paradigm is its ability to support non-temporal composition i.e permitting unpredictable presentation timing among media objects. However, the disadvantage is that modeling presentations involving a significant number of media objects is complex and restricted.

4. A script based approach permits unlimited control over synchronous and asynchronous behavior of the media objects. These systems specify the behavior of a presentation using a procedural approach. The Nsync authoring system (Bailey et al 1998) is an example of script based authoring tool. This system uses a scripting language integrated with a run time constraint manager, bringing in considerable flexibility. However,
the limitation is that presentations that need to be dynamically altered to support adaptive content are usually impossible to create using the *script* based approach.

Most of the existing tools use a combination of the approaches and present themselves as a hybrid system with advanced authoring and presentation features. *Structure* based systems are good for presenting views, editing and navigating. *Timeline* based systems have no direct means of editing the structure of the presentation. It is, nevertheless, the best way of showing the synchronization relationships among media objects. The *graph* based and *script* based systems are, however, more expressive when compared to *structure* and *timeline* paradigms. Editing documents are cumbersome with *scripts*, and viewing interactions remain a problem with *graph* and *script* specifications. *GRiNs* (Bulterman et al 1998, Bulterman et al 1999) is a system based on *structure* and *timeline* paradigms and provides multiple document views. The structured timeline view allows editing and viewing of the presentation’s hierarchical structure, from which high level timing constraints can be derived. In addition, a *graph* model is also used internally in *GRiNs* to determine presentation scheduling and performance analysis.

The approaches presented in the thesis focus on the effective use of *graphs* and *timeline* paradigms, to model consistent multimedia presentations. These concepts could be incorporated into existing authoring systems to augment their internal functionalities. As *structure* based systems providing multiview interfaces have been successfully implemented, the focus here is on the functions of the internal *graph* based representations in authoring systems. In one of the proposed approaches, a *minimum spanning tree* (*graph* based) is initially used to identify and remove inconsistency, and also to deal with incompleteness. A temporal layout (*timeline* based) is then automatically generated from the *tree* and is used to guide a synchronized presentation
playout. Another approach presented here further overcomes limitations in the use of trees by translating them into dynamic petri nets (Tan and Guan 2005) which can model and verify complex workflow patterns and present a graphical view of the presentation structure to the user.

1.2 CONSISTENCY CHECKING

Multimedia presentations are created by specifying a set of constraints that represent the relationships between media objects. When a user specifies such a set of constraints while authoring, there is a possibility of having inconsistencies. Several research efforts have been directed towards the problem of identifying and removing these inconsistencies, at compile time as well as at run time. Several consistency models have been proposed for distributed interactive multimedia applications. An overview of the existing consistency models is presented by Bouillot and Soudan (2004) and it aids in choosing the appropriate consistency models while designing distributed multimedia applications.

A formal approach towards consistency checking of interactive multimedia documents has been presented by Sampaio and Courtiat (2000) and Santos et al (1999), using the formal description technique RT-LOTOS. Here the high level specifications of the multimedia documents are automatically translated into the RT-LOTOS specification and from this, a minimal reachability graph is generated. Consistency analysis is then performed on this graph. If the document is found to be consistent, the scheduling graph is generated. Mirbel et al (1999), deal with an approach for temporal consistency checking, based on the transformation of scenario specification into networks, called constraint network fragments. The rules for building these constraint networks corresponding to the fundamental functional units of the scenario are
defined, and along with temporal constraint verification techniques, provide the basis for checking the consistency of the scenario. Sadat and Ghorbani (2005), have proposed a new language for presentation description which is suitable for use with adaptive web systems. Here, a parser is used to generate an abstract syntax tree which is then processed to generate relation graphs. Inconsistencies in relations are identified using these relation graphs and valid relations are encoded into a file to be used by a constraint solver and renderer. Gaggi and Celentano (2002) have presented an authoring system whose main component is a graph based visual editor. The system also maintains a tree structure internally, which represents parallel and sequential composition, and consistency checking is performed using this generated tree. However, all the above approaches are not dynamic consistency checking mechanisms as they do not provided immediate feedback to the author of the presentation when an invalid constraint is specified. Further, authoring tools equipped with a dynamic consistency checking mechanism would be able to present a truly interactive authoring environment. The framework presented by Buchanan and Zellweger (1993) also expresses the need for such an approach stating that “Running a compile time formatter incrementally while an author is creating a temporal specification may help authors to locate errors by providing immediate feedback when an editing operation introduces a temporal mismatch”.

The graph based approach for consistency checking presented by Ma and Shin (2004) proposes the use of a minimum spanning tree to represent consistent presentations. Here, consistency checking commences only after all the constraints have been specified and hence it is more of a static approach. The dynamic consistency checking approach presented in this thesis is an extension of the approach proposed by Ma and Shin (2004). The proposed approach responds to inconsistency as each constraint is specified by generating a spanning tree and detecting cycles as a possible inconsistency. Since
cycles in the tree representing the presentation would be treated as inconsistencies, they will not be permitted to exist during the creation of the minimum spanning tree i.e. during consistency checking. The limitation of using this approach is that presentations with repetitive components and media objects cannot be modeled, as cycles (loops) are not permitted in the tree. In such cases, repetitive components have to be represented as new instances of the existing objects, making the graph representing the presentation too elaborate. In order to extend the dynamic consistency checking feature to multimedia presentations that involve repetitions, the use of the Dynamic Petri Net (DPN) (Tan and Guan 2005), that has powerful iterative capabilities, has been proposed here to model the presentation specifications. An algorithm that helps to create the DPN automatically from the specified set of constraints has also been presented. The rest of this subsection briefly highlights a few existing consistency checking approaches which are static and have polynomial time complexities. A graph based model to visually represent specifications based on directions is presented by Kong and Mang (2003). It deals with consistency checking of cardinal directions in qualitative spatial reasoning. The proposed algorithm converts a set of definitive direction specifications into a directed graph. The graph is refined and a Breadth First Search (Cormen et al 1990) for connectivity is then performed. The algorithm analyzes the connectivity of the participating nodes and deduces whether the set of direction specifications is consistent but it does it in polynomial time $O(n^4)$.

The framework for the temporal specification and synchronization of multimedia documents presented by Jourdan et al (1997), deals with the representation of time in multimedia documents. The time representation must capture the different temporal characteristics of static and dynamic objects effectively. Dynamic objects have an associated duration that can be distinguished based on their temporal behaviour as controllable and uncontrollable
objects (Layaida et al 2002). The range of possible durations is known during authoring for controllable objects and objects whose durations are determined only when they end are referred to as uncontrollable objects. Consistency checking here is achieved by three different methods depending on the kind of inconsistency. A topological sort of constraint networks is maintained in order to avoid qualitative inconsistencies by detecting cycles. Polynomial time algorithms based on constraint propagation techniques are used to check quantitative inconsistencies. Indeterministic inconsistencies are checked statically to see if it is possible to recover the indeterministic behaviour of uncontrolled intervals dynamically at the presentation stage. The experimental authoring environment Madeus (Jourdan et al 1998a) is based on an extension of the temporal constraint network and deals efficiently with uncontrollable objects. The consistency checking approach discussed by Jourdan et al (1997) has been incorporated in the Madeus system.

Methodologies for automatic assembly of presentations from multimedia databases have been presented by Hakkoomaz et al (1999). Inclusion and exclusion constraints are used to define presentation specifications by exploiting the semantic coherence property of multimedia presentations. Interesting results and theorems relating to consistency and satisfiability of inclusion and exclusion constraints are also presented by Hakkoomaz et al (1999). It is possible that a set of presentation inclusion/exclusion constraints contradict each other in the sense that a segment $x$ must be both absent and present in the presentation in order to satisfy the constraints. Hakkoomaz et al (1999) present a polynomial time algorithm that checks whether a given set constraints are consistent and also whether a given presentation request is satisfied. SMILY, an authoring environment that provides several views of the document, like the timeline view and the hierarchical view has been introduced by Jourdan et al (1999a) and Jourdan et al (1999b). The main feature
of SMILY presented here is the integration of the presentation views during the editing process and the display of an execution report through a timeline view. Spatial layout specification will strongly benefit by such an integration. Temporal consistency checking is done during the editing process, and not at the end of an editing session. But the temporal check here is restricted to the verification of just two temporal rules that feature in the SMIL specifications (W3C 2007). The first rule focuses on the begin attribute, while the second rule ensures that the absolute value of the duration is lower than the actual duration of the media object.

Bes and Roisin (2002), use a quantitative consistency checking initially to filter out inconsistency before applying qualitative consistency checking. But, this approach is limited because all aspects of a multimedia presentation cannot be described easily with qualitative relations. In constraint based authoring tools, one of the major issues is to provide efficient constraint solvers in terms of time performances and solution quality. The solver has a central position in this kind of authoring environments. Numerical solvers used to check quantitative consistency are now more efficient and can manage the qualitative relations. Liu et al (2005), deal with a formalized notion of consistency of XML DTDs. This work identifies the necessary and sufficient condition for a DTD to be consistent and proposes a linear algorithm for checking the consistency of DTDs.

The issues in modeling temporal relations proposed by Allen (1983) are discussed in several research papers and spatio-temporal composition issues are discussed by Vazirgiannis et al (1998). An approach for the generation of the temporal layout that requires precise and efficient handling is presented by Buchanan and Zellweger (1993). Several other interesting features are also being incorporated in the new versions of existing authoring prototypes. For
instance, the *LimSee3* authoring system (Deltour and Roisin 2006), follows a *template* based approach. In this authoring approach, a document is progressively instantiated from a *template* by providing content to template nodes. A template document is a kind of reusable document skeleton that provides a starting point to create document instances. The *MPEG-Pro* (Boughoufalah et al 2003) is a prototype of an *MPEG-4* authoring system which focuses on the development aspects of a user interface for the specification phase. This system also uses a *tree* based structural view internally, focusing primarily on template based authoring and editing. The *Madeus* authoring system has been extended with two new elements: *sub-intervals* for the time dimension and *sub-region* for the spatial dimension. This authoring tool will thus be capable of making fine-grained synchronization specifications while most of the existing authoring tools deal with coarse-grained temporal specifications. As new and innovative authoring tools are continuously evolving, the existing tools are also being upgraded with enhanced features to cater to the consistently demanding area of distributed multimedia applications.

1.3 MULTIMEDIA PRESENTATION

Multimedia document systems should provide mechanisms for automatically producing the temporal layout of the documents. Several automatic temporal formatters exist, and the framework presented by Bertino and Ferrari (1998) and Buchanan and Zellweger (2005) help to compare and understand the potential of each one of them. Buchanan and Zellweger (2005) also describe the multimedia document system *FireFly*, that has been incorporated with a very efficient automatic temporal formatter. The temporal formatters are normally part of an authoring or a presentation system. The authoring and presentation systems themselves may be integrated to form
what is known as a *multimedia document system*. The temporal formatters normally adopt an appropriate approach to capture the spatio-temporal composition of the document. During layout generation most of the formatters largely focus on the temporal composition that are based on *Allen’s relations* (Allen 1983). However, no standard exists for the temporal and spatial composition, and Vazirgiannis et al (1999) present a survey of the approaches used by the existing authoring tools, for spatio-temporal compositions. Jourdan et al (2001), have categorized the approaches used for specifying temporal compositions into *absolute placements*, *event-based approaches*, *hierarchy of temporal operators* and *constraint based definitions*. They also highlight the advantages of using *constraints* and discuss its application in the *Madeus* system. The most important advantage in using *constraints* is the support it provides for *dynamic consistency checking* and also for the effective modeling of *adaptive multimedia documents*. Under the open issues, Jourdan et al (2001) present the need for high performance algorithms for *consistency checking* and for *automatic temporal formatting* for which the *constraint based* specification approach is most suitable.

### 1.3.1 Temporal Formatters

*Multimedia document systems that provide an effective automatic temporal formatter must satisfy three major requirements: they must support media components that have a rich set of capabilities, they must have a method for explicitly representing temporal relationships among media objects and they must provide a powerful temporal formatting algorithm.* The framework presented by Buchanan and Zellweger (2005) and the comparative study of multimedia document systems are briefly stated:

1. Media Components: For the purpose of effective temporal formatting
media components have four primary attributes - *granularity*, *duration*, *flexibility* and *flexibility metrics*. *Granularity* depends on the amount of internal structure the media component makes accessible during authoring. Defining temporal relations based on starting and ending points alone is referred to as *coarse granularity* while *fine granularity* permits reference to internal points within a media object. *Events* are represented by points in time and they guide the presentation playouts. These can be *predictable* or *unpredictable* in nature. *Durations* provide information on the lengths of time required to prepare and present media objects. Preparation time includes functions such as retrieval, compression, transmission and decompression. Presentation time refers to the time interval between the start event and the end event of the media object. This could also be predictable or unpredictable in nature. A media object’s flexibility specifies parameters for manipulating its duration and media type. A media component may have no flexibility, or it may be *continuously adjustable*, *discretely* adjustable or both. The media objects that provide flexibility may also optionally provide *flexibility metrics* that will help the formatter to choose the best representation in adaptive environments.

2. Temporal Relationships: It describes how the media components should be combined temporally to produce a multimedia document. The temporal relationships serve both as input to a temporal formatter and as an explicit record of the author’s intentions. *Temporal relation* can be grouped into three major classes depending on whether they control temporal ordering of media components (referred to as *ordering relations*), duration of media components (*duration relations*) or more complex relationships such as grouping, iteration and conditionals. *Iteration relations* specify that an interval be presented repeatedly a specific
number of times or until (or while) a certain condition holds. The complex relations, however, require a runtime component to maintain, monitor and react to the states upon which they depend. Two forms of flexibility are considered here - *priority* assignment to temporal relations and *range* specifications for temporal relations. *Flexibility metrics* help to measure the degradation in presentation quality that results from situations where temporal relations are ignored or varied within the specified range.

3. Automatic Temporal Formatters: Temporal formatters process the temporal specifications of multimedia documents. A temporal layout comprises of a sequence of entries, each of which contains a time and an event. A partial layout is one in which incomplete entries exist for time and events. A temporal layout is said to be *consistent* if it is free of *temporal mismatches* and a temporal layout is considered to be presentable if it is *consistent*. This presentation layout guides the presentation playout. The actual presentation could deviate from the intended layout due to media delays which may occur for a variety of reasons, including queuing for resources or bandwidth limitations. Thus automatic temporal formatting begins by attempting to position media components without considering the flexibility of their durations and temporal relationships. If the initial temporal layout is not consistent, the formatter attempts to resolve mismatches with the help of the flexibility option specified. The formatter continues to exploit a document’s flexibility until either a consistent temporal layout is produced or a formatter has exhausted the document’s flexibility.

The primary characteristics of an automatic temporal formatter are its flexibility in assigning time, the temporal behaviour it handles, its mismatch detection capabilities and run time capabilities. Secondary characteristics are its
efficiency and the quality of its error messages. Temporal formatting can be performed either before the document is presented (compile time), while the document is presented (run time) or at both times (hybrid). A compile time formatter has two advantages

1. It can identify temporal mismatches prior to playout and use the flexibility options to resolve them unlike the runtime formatter.

2. A compile time formatter can identify errors that normally (if present) would not be detected until runtime. These errors include inconsistencies in temporal specifications, such as unsatisfiable temporal relationships and inconsistencies arising from the environment. This error checking is also handled by the algorithm that produces the temporal layout.

The major advantage of a run time formatter is its ability to handle unpredictable behaviour and hence a hybrid formatter that comprises of both components will be suitable. A compile time formatter would separate the temporal specifications into a predictable portion and a set of unpredictable portions and compute a partial layout. The run time formatter would then resolve the unpredictable behaviour to create a complete temporal layout. Determining which algorithm should be used by the compile time and run time formatters is still an open issue. Some of the existing compile time algorithms are based on topological sort, all-pairs-shortest-paths, linear programming, dynamic programming, constraint hierarchies and incremental algorithms. Running a compile time formatter incrementally while an author is creating a temporal specification may help the author locate errors by providing immediate feedback when an editing operation introduces a temporal mismatch. The existing run time formatters are based on interesting algorithms that include a graph, using temporal data flow, merging precomputed temporal layouts to handle unpredictable behaviour, prefetching data to avoid media delays, using runtime signaling to detect and correct media delays and incremen-
tally running a formatter to handle documents generated on the fly. In the simplest case, temporal specifications are converted into a graph or into a temporal scripting language which is then processed by an interpreter that serves as a run time formatter. Another approach uses temporal data flow representation such as petri nets or temporal dependency graphs in which the temporal specification is converted into a marked graph that drives a document’s presentation. Executing a runtime formatter presents a promising approach to handle documents generated on-the-fly. Xavier (Hamakawa and Rekimoto 1994) is an example of a compile time formatter while Trellis (Scotts and Furuta 1990), Geneva (Fiume et al 1987) and CMIFed (Rossum et al 1993) are examples of run time formatters. The MODE (Blakowski et al 1992) and the Firefly are hybrid systems. The dynamic consistency checking algorithm presented in the thesis is also functionally an automatic temporal formatter. It performs consistency checking dynamically (prior to compilation) and during compilation it checks for incompleteness in the set of specifications. Incompleteness occurs when the graph representing the presentation has disconnected components. On successful compilation, the temporal layout is automatically generated as the output. This temporal layout can be used as a guideline by any presentation system to schedule the participating media objects.

1.3.2 Presentation Models

Run time temporal formatters are popularly called players or synchronizers. They deal with a variety of issues ranging from ensuring a synchronized playout to dealing with heterogeneous environments and unpredictable temporal behaviour due to user interactions. The runtime formatter Trellis uses a Timed Petri Net representation internally to model the presen-
tation prior to its playout. The execution semantics of the *Timed Petri Net* helps *Trellis* to control the playout and handle user interactions efficiently.

*Petri nets* (Balbo et al. 2000) have been used extensively to model concurrency in multimedia systems. As multimedia presentations require synchronized playouts and support for user interactions during playout, several extensions of the basic *petri nets* have been proposed. These include *OCPN* (Little and Ghafoor 1990), *Extended OCPN (XOPCN)* (Woo et al. 1994), *Dynamic Timed Petri Nets* (Prabhakaran and Raghavan 1993), and *Dynamic Fuzzy Timing Petri Nets* (Zhou and Murata 2001). *Petri nets* have also been designed to support multimedia presentations that adapt themselves based on user profiles (Athuri et al. 2003, Gomaa et al. 1993) and network profiles (Guan and Liu 2003). The work proposed by Liu and Hao (2005) focuses on capturing the temporal and spatial relations of moving objects. For this, two new petri nets referred to as *Moving Object Petri Net (MOPN)* and *Spatial Object Petri Net (SCPN)* have been proposed to model the temporal and spatial relations respectively. Dealing with temporal and spatial relations separately and the lack of *iterative* capability make the models inefficient for use in an interactive authoring environment.

Chung and Pereira (2003), deal with a mechanism to convert a *synchronized Multimedia Integration Language (SMIL)* document into a *Timed Petri Net*. Mazouz et al (2006), have enhanced this work further by presenting a formal approach to deal with incoherence. An author of *SMIL* documents may incrementally modify the initial specifications by adding or removing constraints, or replacing objects that have different durations. A scheme for translating *SMIL* documents into petri nets and coherence verification using *Timed Petri Nets* is presented by Mazouz et al (2006). The limitation here is mainly attributed to the limitation of the *Timed Petri Net* in modeling
particular cases in the presentation. *Dynamic Petri Nets (DPN)* (Tan and Guan 2005) have been chosen in the approach proposed here to perform the functions of a *run time* formatter. The programmable feature of the *DPN* makes it superior to all existing petri nets in representing iterations and interactions. Attempts have been made by Chung and Pereira (2003) to translate specifications in *SMIL* automatically into petri nets. Here, the conversion is performed under the assumption that there is no contradiction in the specified presentation schedule, and is thus limited in its consistency checking capabilities. A similar attempt has been made here to translate specifications directly into *DPNs*. This approach is augmented with the *dynamic consistency checking* and *completeness checking* performed prior to run time.

### 1.3.3 Synchronized Presentations

Another primary function of a *run time* formatter is to ensure a synchronized playout. *MODE (Multimedia Objects in a Distributed Environment)* is a *hybrid* formatter with a compile time component called the *optimizer* and a runtime component called the *synchronizer*. In *MODE*, the *synchronizer* uses a *runtime* signaling mechanism that handles media delays at synchronization points. Synchronization points are locations identified on the timeline and are used as reference points to ensure synchronized playouts. When an object reaches a synchronization point, it signals all other objects sharing that synchronization point and waits until it has heard from all the other objects. A concept called *restricted blocking* is employed here, which helps an object to perform alternative *filling actions* while waiting at the synchronization point, instead of just *pausing* the presentation. Meanwhile, the objects that have not yet reached the synchronization point may be forced to perform continuous adjustments to reach the synchronization point faster.
Distributed applications face greater challenges because they deal with the presentation of media objects over a network. The issues related to distributed multimedia presentations are:

- the overall design of the system
- scheduling the object in a network - retrieving
- dealing with the temporal relationship - synchronization
- support for user interaction

Thus, in distributed applications, the common objective is to provide a smooth playback for jitter free viewing. However, since the media objects reside in remote servers, the critical issue is to ensure synchronization between the media objects while simultaneously permitting user interactions. In order to deal with the second issue effectively, Yang and Yang (2003) and Yang and Yang (2004) emphasise the need for an object retrieval engine to be integrated into the presentation systems. The proposed object retrieval engine adopts a just-in-time policy, having better buffer utilization and network bandwidth efficiency compared to the existing pre-loading and passive loading policies. In order to deal with the issue of synchronization effectively, Yang and Yang (2004) employ a Real-Time Synchronization Model (RTSM). RTSMs are petri based models that are suitable for modeling real-time network applications as well as user interactions. The object retrieval engine accepts SMIL scripts as input and converts the synchronization relationships to RTSM. The object retrieval engine uses the RTSM to compute the estimated time that is appropriate for the retrieval of each object. However, the RTSM has limitations while modeling loops in presentations. These are handled efficiently by the Dynamic Petri Net (DPN) because it is programmable.

In order to deal with the issue of synchronization in distributed presentations Finite State Machines have been used effectively in several exist-
ing works. The approach adopted in the thesis is an extension of the DEFSTM model presented by Huang and Wang (1998) and Huang et al (2000) and is based on the concept of communicating finite state machines (Brand and Zafiropulo 1983). The DEFSTM model is an extension of the EFSM model presented by Huang and Lo (1996). In the presentation layout, synchronization points are identified as points in the layout where one or more media objects begin or end their presentation. The limitation of the DEFSTM model is that each synchronization point in the presentation layout is modeled as a state in the DEFSTM. Thus, the DEFSMs representing fast changing multimedia presentations will have several states and transitions. The synchronization model called the Communicating Adaptive Finite State Machine CAFSM proposed in the thesis models distributed multimedia presentation systems efficiently. In this model, the number of states in the FSM is independent of the number of synchronization points and it can model any multimedia presentation using just 3 states and 18 transitions.

Layaida et al (2002) deal with the synchronization of media objects having uncertain durations. The paper presents scenario based scheduling of multimedia presentations under indeterministic durations. The algorithms presented are adaptive and executed dynamically each time a desynchronization occurs. The algorithms are based on two complementary phases the reformatting which attempts to modify the duration of the objects while maintaining the author’s specification, and the reparation which aims at minimizing the period of desynchronization resulting from the indeterminism. This indeterminism or uncertainty can be due to external factors such as the access delay over the network or due to user interactions. An uncertain duration is often followed by a period of desynchronization during which the presentation deviates from the desired scenarios. Kurkovsky and Loganathanaraj (2000), have presented an approach where time is incorporated into the framework
of petri nets. Layaida et al (2002) have extended the concepts of Timed and Fuzzy Petri Nets by using the probabilistic duration of a task. This model also deals with media objects with uncertain durations.

A few other existing work that deals with other related issues are briefly described here. The Multimedia Document Versatile Architecture (MAVA) (Hauser and Tian 2002) is a presentation system and authoring tool based on Java and JMF. The MAVA approach uses an operator based language on a high abstraction level. The support of abstractions in authoring systems increases the efficiency of the specification of multimedia documents. The existing authoring systems that support high abstractions are domain specific. Hence, this paper presents an extensible approach that also supports generalized abstractions by using templates to reuse parts of existing documents.

King et al (1998) present a taxonomy of possible synchronization relationships between a pair of media items. 72 possible relations have been identified and formalized in the author’s temporal logic notation - Mexitl. King et al (1998) focus on overcoming the drawbacks of Allen’s relations, where there is no direct provision for synchronizing one object upon events or conditions occurring dynamically in the second object. Mexitl is a formal notation developed by the authors for use in specifying multimedia documents. It is based on an interval temporal logic and provides a platform to perform consistency verification, modeling, prototyping and specification refinements as well.

Wahl and Rothermel (1999) have presented an interesting comparative study showing that none of the existing interval based models exceed the relation set that is expressible in a point based framework. To overcome the existing limitations, a set of interval based operators were developed. This was shown to be a high expressive set providing fine grained temporal relations.
The interval based framework reduces the number of possible inconsistencies, thus simplifying consistency checking.

A scheme for the continuous and synchronous delivery of distributed stored multimedia streams across a communication network has been presented by Biersack et al (1996). They propose a protocol for synchronized playback and deal with buffer management and clock drifts efficiently. The proposed scheme comprises of three models that assure synchronization in an environment with different delays, jitter, server drop-outs and clock drifts. This is suitable for streams that are striped across multiple server nodes as well as for a single server approach.

1.4 CONTENT ADAPTATION

Multimedia as a service and an information framework has gained wide attention due to advancements in mobile multimedia communication technologies. The combination of various elements in a single multimedia file makes it necessary to add descriptive information on how these elements interact with each other. In a heterogeneous environment, there is also a need for the multimedia framework to be flexible enough to operate over a range of networks ensuring high Quality of Service (QoS) (Davies et al (1994). The use of distributed systems for the delivery of multimedia objects over the network preserves the flexibility and transparency of these objects.

When creating a service for distributed systems, the service content model should be planned carefully to support different terminals. This ensures that the service is available to different types of devices, allowing it to be accessed universally. Whenever the content is not presentable in a given context, various adaptation techniques are used as illustrated by Berhe et al
(2005), Bertolotti et al (2006), Lei and Georganas (2001), Lum and Lau (2002) and Mohan et al (1999). These techniques are used to format the content as per the requirements and deliver the same information in accordance with the context. These requirements are modeled from either the user or system profiles. For context aware applications, matching of profile semantics (Chevalier et al 2005) needs to be addressed to enable reusability of profile models and interoperability between applications.

A formal framework was put forth by Bertolotti et al (2006) making use of an automaton to convert the multimedia document into the required format. It identifies the undeliverable formats and converts them into semantically equivalent formats. This technique can be used to deliver multimedia content to a large number of mobile users as discussed by Wee et al (2003). The network conditions should also be considered to achieve acceptable and predictable performance. The various research studies in this field are explained by Cao et al (2004).

The emergence of various standards such as MPEG-21 presented by Vetro and Timmerer (2005), Keukelaere et al (2005), Burnett et al (2005) and Suttera et al (2003) have paved the way to simplify dynamic user interactions with a multimedia document. For example, it is possible for the user to give his preferences dynamically in order to filter out unnecessary data. This lets the user decide the optimal QoS level.

An architectural framework which involves the use of, on one hand, a proxy server at the client-end to process requests from clients, and on the other hand, a proxy server at the server-end to perform content adaptation has been proposed in the thesis. The proxy at the client-end also plays a role in the content adaptation by generating the adaptation plan in the form of
an adaptation tree for each client request, taking into consideration the device
and user profiles. This framework has been designed according to the require-
ments of the MPEG-21 standard (Vetro and Timmerer 2005), with a built-in
layer that performs the core adaptation function. A new graph-based content
adaptation algorithm that helps to effectively generate the adaptation tree is
proposed in the thesis. This work is a variation of the algorithm proposed by
Berhe et al (2005) for content adaptation. The proposed algorithm runs in
linear time hence making it more efficient than the existing algorithm.

One of the first complete systems on content adaptation was developed by IBM (Mohan et al 1999). The system had two main components: a
data structure known as the InfoPyramid and a customiser. The customiser
selects the best version of the content from the InfoPyramid to meet client
resources. The content and adaptation information were described using
XML. The emerging standard MPEG-21 (Huang and Lo 1996) focuses a lot
on content adaptation. It aims at defining an open framework for multi-
media delivery. The DIA (Vetro and Timmerer 2005) specifications of the
MPEG-21 standard deals with content adaptation. Timmerer and Hellwagner
(2005), present an approach where the user preferences are captured inside
standardized DIA descriptors. It is the DIA engine that is responsible for
finding an optimal adaptation decisions. Universal Multimedia Access (UMA)
(Kasutani and Ebrahimi 2004) refers to the ability of any type of terminal
to access and consume a rich set of multimedia content. Working towards
this goal of universal accessibility and user-based adaptability, techniques for
scalable coding and transcoding presented by Vetro and Timmerer (2005) are
less computation-intensive. Thus, an XML-based standard called MPEG-21
(Keukelaere et al 2005) has emerged to be an open framework for multime-
dia delivery and consumption across various networks and terminals. Unlike
other MPEG standards that describe compression coding methods, it includes
several components to describe the content and also processes for accessing, adapting, searching, storing and protecting the multimedia content.

1.5 SCOPE AND CONTRIBUTIONS

The framework presented by Buchanan and Zellweger (2005) helps to evaluate the features and capabilities of the existing temporal formatters and also to indicate the potential areas for further research. This framework was developed and used in 1993, to analyse the strength and weaknesses of the then existing systems. This framework was used again after a decade in 2005, to evaluate how the framework itself stood the test of time and also compare the features of nine new systems developed since 1993. The need for the framework itself to be refined in order to represent the characteristics of the systems more accurately has also been highlighted by Buchanan and Zellweger (2005). The discussions by Buchanan and Zellweger (2005) on the potential refinements to the framework, helps to understand the scope and goal of this thesis. The automatic temporal formatters were evaluated based on several criteria. These have been categorized into support for media components having rich capabilities, explicitly representing temporal relations and automatic temporal formatting features as shown in Figure 1.1.

The discussion on potential refinements to the framework mainly pertain to the specification of temporal relations. The existing framework hides important characteristics of temporal specifications. For instance, support for hierarchical composition is not explicitly represented. The framework emphasises the need for representing composite events using either the temporal operator, the spatial operator or a combination referred to as spatio-temporal operator. The work presented in the thesis, introduces a functionally complete temporal operator, a spatial operator as well as a spatio-temporal operator.
The thesis also describes an approach which uses a hierarchical composition (a minimum spanning tree) of the above operators to model a presentation. This representation helped to perform dynamic consistency checking effectively and efficiently. Two approaches for dynamic consistency checking have been presented in the thesis. The first approach uses temporal and spatial operators separately, performing temporal consistency checking followed by spatial consistency checking, while the second uses the integrated spatio-temporal operator using just one spanning tree to represent the entire presentation. The operators proposed in the thesis can be easily extended to accommodate indefinite temporal relations which is likely to be an important criteria in the revised framework. Indefinite temporal relations are those that permit a range of relations to be specified eg. A starts before or after or at the same time as B.

An important feature that was used by the framework to evaluate compile time formatters is mismatch detection as indicated in Figure 1.1 while
run time formatters were evaluated based on their capability in dealing with unpredictable events like user interactions and request for content adaptation. The components of the framework that require further refinements are the criteria considered by the formatter, mismatch detection, adaptation and algorithm characteristics. The criteria considered by the formatter indicates what type of information the formatter takes into account when producing temporal layouts. This aspect relates to the use of an efficient retrieval policy employed by the formatter. For instance, environmental factors are used to compute the retrieval time for each media in distributed applications. The formatter presented in this thesis does not use a retrieval policy, but it could be easily extended and incorporated with an effective retrieval policy. The environmental as well as spatial specifications will be better emphasised in the revised framework. The algorithms that perform mismatch detection fall into two categories validators and resolvers. Validators indicate the location of inconsistencies and offer suggestion in some cases. However, resolvers are those that automatically correct temporal, spatial and environmental mismatches. These categories will be integrated in the revised framework. The dynamic consistency checking approach presented in the thesis performs mismatch detection prior to compilation. This can be compared to a text editor which indicates spelling and grammatical errors as soon as they appear. The compile time formatter proposed in this thesis performs only a completeness check and automatically generates the temporal layout at the compile time. This approach helps to create a truly interactive authoring environment. In order to identify the true dynamism that is supported by authoring systems, it is proposed here that the framework incorporates this feature in its next revised version. The framework also discusses the need to include the concept of query engines which would permit the authors to ask questions regarding specific characteristic in the generated layout. Systems
may differ depending on the type and complexity of the queries they support. Querying the layout is yet another interesting feature that can easily be incorporated into the formatter proposed in this thesis.

A synchronization model has also been proposed and presented in the thesis. This model uses a signaling approach to ensure synchronized playout. The layout generated by the compile time formatter serves as a guideline. A suitable retrieval policy could be integrated here to ensure jitter free playout for distributed application but it is not been considered currently. The synchronization model is based on an extended Finite State Machines referred to as Communicating Adaptive Finite State Machine (CAFSM). The details of this model and the reduction achieved in time and space complexities are also presented in the thesis.

The original framework also describes the support for adaptation that enables the temporal formatter to alter a document’s presentation in a way that satisfies a set of run time conditions. The framework needs to be extended to characterize static adaptation and dynamic adaptation. In static adaptations, the author need to specify all the possible alternative media components when the document is created. The framework helps to understand that almost all the existing systems support static adaptation. However, if the set of possible alternatives is large, the approach becomes complex and inefficient. In cases where the identification of alternative media components is done at runtime by the formatter, the author needs to only specify the set of criteria that the alternate media objects must satisfy, instead of specifying the set of alternate media. This is referred to as dynamic adaptation and it requires the support of formatters with efficient response time. The runtime conditions in the form of profiles that are taken into consideration for the adaptation process, also needs to be integrated in the framework. These may be user profiles, devices
profiles and/or networks profiles. A very efficient dynamic content adaptation algorithm has been proposed and presented in the thesis. Formatters integrated with this algorithm will be capable of supporting dynamic adaptation taking into account all the three profiles mentioned above. The support for this feature in the emerging MPEG-21 standard has been discussed.

The thesis also presents a novel algorithm which automatically generates a dynamic petri net. The dynamic petri net can model iterations user interactions efficiently as it is a programmable petri net. The DPN representation of the layout would therefore help a runtime formatter to handle unpredictable behaviour efficiently. The use of dynamic petri nets to model complex workflow patterns has also been proposed and illustrated in the thesis. The distributed programming model Orc can very efficiently represent complex work patterns using a procedural approach. The thesis also presents a novel approach for automatic Orc code generation. The code generation algorithm takes as input a DPN representation of the workflow, and generate its Orc representation effectively.

The revised framework also needs to consider three characteristics of algorithms: computational complexity, fairness and whether the algorithm executes incrementally. All the algorithms presented in this thesis have a linear complexity and have achieved considerable reductions in running time compared to the existing closely related approaches. The consistency checking algorithm is incremental while the other algorithms are non-incremental because it does not help functionally in being incremental in those cases.
1.6 OUTLINE OF THE THESIS

The rest of the thesis is organized as follows. A brief introduction to a few terms is presented in chapter 2. The research work carried out is presented in the subsequent chapters 3, 4, 5 and 6. In each of these chapters, the limitation of the existing approaches and the problem definitions are clearly highlighted. These are followed by the presentation of the proposed approaches that deal with the algorithms in detail with appropriate illustrations. The simulation results and comparative study are also presented in each chapter.

Chapter 2: A few terms that are of relevance to the research work and used in the thesis, are defined and illustrated in this chapter.

Chapter 3: Approaches for detecting contradictions in specifications that arise during authoring, are presented in this chapter. The first approach uses temporal and spatial information independently, while the second integrates them using a single spatio-temporal operator. An interesting comparative study presents the advantages of these approaches over the existing ones, as well as the advantage of the second over the first approach.

Chapter 4: This chapter presents a synchronization model that is based on finite state machines. The model uses an innovative signaling mechanism that enables communication between the finite state machines. The existing approach is not described in detail here but its limitations are clearly explained. The detailed specification of each component of the synchronization model is presented in the form of states and transitions. A popular model checker (SPIN) has been used here to verify the effectiveness and efficiency of the model.

Chapter 5: An efficient approach for dynamic content adaptation has been presented in this chapter. The chapter also deals with the emerging MPEG-21 standard and its compatibility with the proposed approach.
Interesting simulation result and analysis are also presented here.

**Chapter 6:** This chapter deals with two novel approaches proposed in the thesis, to model multimedia presentations. The first approach uses *dynamic petri nets* while the second focuses on modeling distributed multimedia presentations using a new programming model *Orc*. The motivation of the first approach is to be able to model consistent presentations having loops or repetitive components. An algorithm that automatically generates a dynamic petri net at the backend, during the authoring process has been presented here. Simulation results and performance analysis of this algorithm are also presented. The second section focuses on modeling distributed multimedia presentations using a new programming model *Orc*. This is also an innovative approach which presents an algorithm for automatic *Orc* code generation.

**Chapter 7:** The concluding remarks and scope for further research have been presented in this chapter.