INTELLIGENT VIDEO SYNTHESIS USING VIRTUAL VIDEO PRESCRIPTIONS

C. A. LINDLEY
CSIRO Mathematical and Information Sciences
E6B, Macquarie University Campus, North Ryde, NSW, Australia 2113
E-mail: craig.lindley@cmis.csiro.au

A.-M. VERCOUSTRE
INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt,
B. P. 105 - 78153 Le Chesnay Cedex, France
E-mail: vercous@margaux.inria.fr

To facilitate maximum reuse of video data, well-specified and systematic methods are required for the incorporation of digital video components into multiple delivery products. This paper presents an approach to the intelligent synthesis of digital video products by the representation of virtual videos using SGML. Virtual videos are defined using a Virtual Video Prescription to specify a generic model of a video production that includes embedded queries for the dynamic retrieval of video content from underlying databases. A detailed example is presented to demonstrate the prescription language syntax and its underlying concepts.

1 Introduction

This paper presents an approach to the intelligent synthesis of digital video products by the representation of virtual videos using Standard Generalised Markup Language (SGML[3]). SGML has the advantages of being highly standardised, portable, and suitable for the integration of video data streams with other media types, such as text, audio, and images. Virtual videos are here regarded as an extension of the concept of virtual documents[7]; a Virtual Video Prescription is used to specify the model of a video production using embedded queries for the dynamic retrieval of video content from underlying databases. The use of a prescription separates the specification and design of the virtual video from the potentially complex processing tasks required to find and/or generate specific video components for the resulting production.

2 Background and Related Work

Previous research on video database systems has largely been concerned with content-based retrieval of video data[1], and automatic analysis of video content[10]. Some research has been concerned with the development of tools suporting manual video production[4]. Research within the Research Data Network Cooperative Research Centre (RDN-CRC), in which CSIRO is a participant, has resulted in the successful implementation of the Film/TV Researchers Archival Navigation Kit (FRANK) prototype[5,9]. FRANK users can search and navigate through remote video content using representations including transcripts, shotlists, and key frames, all with associated synchronised video data.

The CSIRO RIO (Reuse of Information Objects) project is developing techniques for the reuse of structured information from heterogeneous sources, including object-oriented or relational databases, SGML and HTML documents[8]. Reuse is achieved by incorporating information into virtual documents according to a virtual document prescription. Document components are integrated using data structures that consist of ordered labelled trees, providing a generic and minimal approach independent of the source document models. The RIO prescription language has been designed for writing document prescriptions and querying the tree structures (a description of this language, and comparisons with Hytime, OQL, and SgmlQL, can be found in Vercoustre and Paradis[7]).

CSIRO research in video representation and synthesis is building upon this background to develop more advanced interfaces for access to video archives. In particular, it is desirable for users of video databases to see the results of data searches in the form of coherent video presentations, rather than receiving ad hoc sequences from arbitrary queries. To facilitate this, specific virtual video models may be predefined as Virtual Video Prescriptions to suit particular user tasks and purposes.

3 Virtual Video Prescription and Interpretation

Video information can be specified within SGML tags using a Continuous Media Description (CMD) language that includes direct reference to time and frame offsets within a video file specified by a Uniform Resource Locator (URL)[6]. As video archives increase in size and dynamism, it is attractive to extend this simple approach to allow the definition of virtual videos by methods that are robust, extensible, and support specification of the desired content at a high level. Prescription of virtual video sequences should not require knowledge of any existing sequential relationships between stored video components, and the final virtual video may include components that are complex manipulations and/or transformations of stored video components.

These requirements can be supported by the definition of a Virtual Video Architecture based upon the concept of virtual documents. In this Architecture, a virtual video production is defined by a Virtual Video Prescription expressed in SGML (or the eXtensible Markup Language, XML[2]). A prescription includes format and assembly information specifying how a Virtual Video Production is to be synthesised from textual and video components according to corresponding Document Type Definitions (DTDs), and queries specifying how components are to be generated from underlying knowledge bases and databases.

An interpreter executes embedded queries in a Virtual Video Prescription by routing them to appropriate processors for the search, retrieval and synthesis operations required. The interpreter also has a role in synthesising results, generating SGML structures representing the resulting Virtual Video Production. The interpreter may perform manipulations to increase the compatibility of retrieved segments (eg. by shifting colour balances to match consecutive shots), or inserting bridge material between segments or over NULL returns. The interpreter may send the results of queries, in the form of references to video components, directly to a video player. Alternatively, the interpreter may compile results into a script containing detailed references to video components (eg. specified in the CMD language) that can be executed at any time to generate a real-time video stream. The stream of video data constituting a virtual video can also be stored as a stand-alone production.

4 Example

As an example of virtual video generation using an SGML prescription and the RIO syntax[7], a hypothetical, customised holiday program will be considered. The user wishes to see video information about a destination that reflects their particular tastes and interests. Suppose they desire a fishing holiday in Sydney. We assume that a video database (tv_station_X_db) contains a set of generic city introductions and closing sequences (CITY_INTRODUCTIONS), and an archive of fishing magazine program episodes (GONE_FISHING). A Virtual Video Prescription can then have the following form:

```
<holiday>
<?parameters  $city default "Paris">
<introduction>
<? define $intro as http://video_info.cmis.csiro.au/tv_station_X/
        ?select intro_ sequence from CITY_INTRODUCTIONS where
                city = $city; define $intro as $intro[0]>
        <clip> <? attributes  $intro..#ATTR>
        <start> <? $intro.start> <end>  <? $intro.end>
        </clip>
</introduction>
<leisure>
<?map $rec_fishing in http://video_info.cmis.csiro.au/tv_station_X/
        ?select sequence from GONE_FISHING where  location = $city;>
        <clip> <? attributes  $rec_fishing..#ATTR>
        <start> <? $rec_fishing.start> <end>  <? $rec_fishing.end>
        </clip>
 </leisure>
<closing>
<? define $close as http://video_info.cmis.csiro.au/tv_station_X/
        ?select end_ sequence  from CITY_INTRODUCTIONS where
                city = $city; define $close as $close[0]>
        <clip> <? attributes  $close..#ATTR>
        <start> <? $close.start> <end>  <? $close.end>
        </clip>
</closing>
</holiday>
```

This example illustrates several powerful features of the RIO language:

• the Virtual Video Prescription is a generic and reusable template for videos of this type.
• the markup language supports search and querying of Virtual Video Prescriptions and scripts for components of different logical types.
• the specification contains rich information about the content that is being sought.
• parameters can be used to specify user- and task-specific information. In this example, a city parameter is used. The default city of "Paris" will result in the fishing query returning items on fishing in the Seine. Calling the prescription with the city set to "Sydney" will override the default.

• if there are several introductions for the city in the database, they will be all returned in $intros. The example selects the first one as $intro[0]. Replacing $intro[0] will generate another sequence. The number could be a parameter that could be changed dynamically, allowing the user to select the preferred one.
• $intro..#ATTR selects all attributes (ie. "name" and "source") of the first intro; the instruction <?attributes $intro..#ATTR> adds these attributes to the clip tag (ie. the tag preceding the process instruction).
• the map instruction will return all of the fishing clips about fishing in Sydney that the database contains.

In a more complete example, various forms of virtual document component (eg. scripts, transcripts and summaries) can be included in the Virtual Video Prescription in order to document the virtual video.

The definition of specific forms of Virtual Video Prescription for particular application domains, users, and tasks, is an important area of ongoing research. Adaptive linking to suitable databases based upon query and parameter contents is a promising strategy for increasing the generality of prescriptions.

5. Conclusion and Further Work

This paper has described a Virtual Video Prescription language for the synthesis of virtual video productions. Ongoing research at CSIRO is addressing issues including the articulation of conceptual and logical models of videos of different types, extensions to the query language to support a broader range of user- and task-specific queries, development of adaptive synthesis strategies, and the development of intelligent systems to manage and control these strategies. The synthesis of more dynamic and/or complex videos will require conditional branching within specifications, complex video models, and extensive rule sets specifying principles of composition for videos in various genres. Knowledge representation tools can be fully integrated with extensive text editing and presentation tools supported by SGML to provide a system that can inference over appropriate text structures within a document, while the presentation and high-level organisation of the document is based upon the specification of its logical structure using SGML.

6. References

1. Aigrain P., Zhang H., and Petkovic D. 1996 "Content-Based Representation and Retrieval of Visual Media: A State-of-the-Art Review", *Multimedia Tools and Applications* **3**:179-202.

2 2. T. Bray, and C.M. 1996 Sperberg-McQueen, *Extensible Markup Language (XML)*, W3C Working Draft Nov 96.

3. *ISO 8879. Information Processing-Text and office systems-Standard Generalized Markup Language (SGML)*, 1986.

4. Steinmetz A., DiVidEd - A distributed Video Production System, *Proceedings of VISUAL'96 Information Systems Conference*, Melbourne.

5. Simpson-Young B. and Yap, K. 1996 "FRANK: Trialing a system for remote navigation of film archives", *SPIE International Symposium on Voice, Video and Data Communications*, Boston, 18-22 November.

6. Simpson-Young B. and Yap K. 1997 "An open continuous media environment on the web", *AusWeb-97*.

7. Vercoustre A-M. and Paradis F. 1997 "A Descriptive Language for Information Object Reuse through Virtual Documents", submitted for publication.

8. Vercoustre A-M., Dell'Oro J. and Hills B. 1997 "Reuse of Information through Virtual Documents", *2d. ADCS Symposium*, Melbourne, Australia, April.

9. Yap, K., Simpson-Young B. and Srinivasan, U. 1996 "Enhancing Video Navigation with Existng Alternate Representations", *First International Workshop on Image Databases and Multimedia Search*, Amsterdam.

10. Zhang H. J., Tan S. Y., Smoliar S. W., and Yihong G. 1995 "Automated parsing and indexing of news video", *Multimedia Systems*, ACM, **2**:256-266.