The Dynamic Generation of User-Customized Multimedia Presentations[1]

Hyun Woong Shin[†], *Dennis McLeod*[†], *Larry Pryor*[‡]

[†]Computer Science Department
Integrated Media Systems Center
[‡]Annenberg School for Communication
University of Southern California
Phone: +1-213-740-4504 Fax: +1-213-740-5807
{hws, mcleod, lpryor}@usc.edu

1. INTRODUCTION

The rapid growth of online digital information over the last decade has made it difficult for a typical user to find and read information [13]. One way to address the problem of information overload is to tailor that information to specific user interests, needs and knowledge base. If there is an approach that responds to individual information requests with an original, dynamically built story, several problems are solved.

First, in today's Web service industry, information presentations and collections of data are static and having limited multi-modal presentations. Critically, there is no capability to dynamically adapt an integrated presentation of information to a user. Second, most of current web search engines deliver a huge amount of hyperlinks. Although this helps improve accuracy (recall), an end user has a trouble deciding which results are what he/she wants. Finally, most web services do not instantiate and customized (for an individual user) "generic" stories. Even though the accuracy of results in terms of precision and recall may be acceptable, the results might not be relevant to a user's intention(s).

The proposed system will create story structures that can be dynamically instantiated for different user requests from various multi-modal elements. In addition, the proposed system focuses on quality of the results not quantity of results. Furthermore, the system leverages information so that a user will read an appropriate level of story depending upon the user's intention level ranging from general to specific. For example, a user might impress the USC football game, but the user has very little knowledge of the USC football team. The user then wants to read very general information instead of specific information regarding the team. When the user requests a general level of the USC football team, the system will deliver a customized (in this case, a general story) dynamically generated multi-modal story.

In order to convey the nature of the information presentation, we propose to define and develop the precise nature of the dynamic generation of user-customized multimedia presentations that will draw upon visual techniques [3, 5], presentation constraints [1, 16], a content query formulation, a story assembly and a structured rule-based decision process.

To determine a user's intention and goal, a general knowledge-based process will be used. A key to the successful use of story types is the ability to relate and connect the user requests to the Content Database. An ontology [4, 7, 9] is essential for capturing the key concepts and relationships in an application domain. For our purposes, we are interested in sports domain dependent ontologies [10,11]. Metadata descriptions will connect a modified user request (by using domain ontology) to the Content Database for retrieving proper content elements.

2. STORY TYPES AND METADATA STRUCTURE

2.1. Story Types

The proposed story model defines four story types that lay out an appropriate presentation style depending on a user's intention and goal. In order to provide an efficient presentation, the story model employs visual techniques [3, 5] that solve layout problems such as combining and presenting different types of information. Moreover, the proposed model adapts presentation constraint specifications [1, 15] to abstract higher levels of presentation so that lower levels of presentation can automatically generate a story that meets those specifications. The component of the proposed story types consist of title, elements and elements description.

2.1.1. Summary Story Type

A summary story type presents a textual summary as a base story and some multi-modal elements based on element description. Since automatic text summarization has been studied in several research projects, we decide to adapt one of the well-known automatic text summarization systems called SUMMARIST [8].

2.1.2. Text-Based Story Type

A text-based story type will be used when a text element is in the top three of the relevant elements of a user's request. The text element will be used as a base story for a complementary secondary query. In this story type, multiple elements – including video, audio, images and text – are integrated into a multi-level story with a primary narrative.

In the text-based story type, multiple elements will be located alternately to the right or left side along the length of the base story container. After that, the rest of the elements will be presented at the bottom of the story type. To make a concrete story, the result of the secondary query will be put near a related concept in the base story. A related concept is a word in the base text that is in the domain dependent ontology. Furthermore, the organization style of this story type is to display relatively general elements first so that the story presents information along a general to specific spectrum.

2.1.3. Non-Textual Based Story Type

A non-text based story type delivers a story without a primary narrative. This story type will be chosen when there is no text element in the top three relevant elements or a user wants a presentation without a textual element.

The base element is the most general information. After the base element, relatively specific information will be presented so that the story keeps an organization style that is general to specific. For example, if a user wants to watch Kobe Bryant, the base element will be a picture of Kobe Bryant. After the picture, there will be video clips of slam-dunks and/or images of 3-point shots.

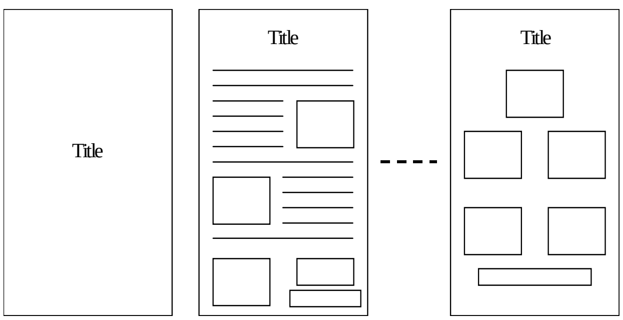

Figure 1 A sample structured collection story type

2.1.4. Structured Collection

Unlike other story types, a structured collection presents a series of text-based story types and/or non-text based story types. This story type consists of sets of pages associated with a collection cover that has a collection title. This story type is useful when results are a series of independent stories. Furthermore, one structured collection story type might also have another structured collection story type as an element. For example, if a user wants to know the members of the Los Angeles Lakers, the structured collection story type presents each member in a story in one page utilizing text-based story types and/or non-text based story types. Each page will provide the element description of chosen appropriate media types. Figure 1 illustrates a sample structured collection story type.

In this story type, the first page is a collection cover that includes the title of the whole collection. After the collection cover, each page is either a text-based story type and/or a non-text based story type. The generation of each page is the same as its own story assembly process.

2.2. Metadata Structure

In this section, we will discuss the structure and description of metadata. Moreover, we will discuss a mapping mechanism for seamless integration between metadata and a domain dependent ontology. We designed a metadata structure to effectively delineate content characteristics.

The TABLE Concept holds all ontology nodes, but this table does not present relationships between nodes. The possible relationships between nodes are "IS-A, PART-OF and INSTANCE-OF." The characteristics of these relationships are described in [11]. The purpose of this table is to map the ontology nodes and content objects.

The TABLE Content stores the metadata of multimedia objects including text, images, audio and video. The values of attributes are generated from content objects by an agent that automatically extracts proper values. For this table, the acquisition process is fully automated.

The TABLE Mapping consists of four attributes that will be used to retrieve appropriate objects. The values of attributes are also generated from content objects by an agent except the attribute, “generality” that will be assigned by a human intelligent. A concept can be mapped into different contents with different weights and generality. The weight represents relevance in scales of one to 100. More relevant objects have higher values. On the other hand, the generality is indicated through a spectrum of one to 10. The lesser value means that the content of objects is more general. For example, if there is an image object with a weight of 100 and a generality of 1 this image is the most relevant and most general object for a chosen concept.

One of our goals is to support an easy, simple and highly efficient way to integrate heterogeneous resources. In order to achieve that goal, our approach is to separate contents and general knowledge-based procedures. There is another way of integrating heterogeneous resources that is a unification of contents and a general knowledge-base structure utilizing ontology language such as DAML [12] and OIL [2]. To integrate a new resource, the unified approach has to reconstruct the whole structure to accommodate necessary features of the new source. This procedure might be very error prone and require a lot of time to re-design the structure. However, our approach simply extracts metadata from the new resource and maps it into relevant concepts with weight and generality. Moreover, the unified approach generally employs an ontology language that does not support powerful functionality such as indexing, concurrency control and recovery.

3. STORY ASSEMBLY AND CONTENT QUERY FORMULATION

3.1. Overall Functional Architecture

The overall functional architecture of a story model is illustrated in figure 2. The model has two key phases: story assembly and content query formulation. In the story assembly phase, a novel structured rule-based decision process is introduced to determine a proper story type and to invoke a primary search and a secondary search in the content query formulation phase. At the beginning, the story assembly module receives a modified user's request from a query processing procedure, which consists of related concepts, a level of generality spectrum, media types that a user prefers and so on. Note that our architecture does not request a particular query processing procedure because our intention is to devise and implement a Plug-and-Play (P&P) architecture. These inputs then invoke a primary search to retrieve multi-modal content objects, along with a constraint-based k-nearest neighbor search. These results are sent to the story type decision module to determine a proper story type and then fill in the chosen story type with multi-modal elements (content objects). If it is necessary, this decision module also invokes a secondary search to get extra elements.

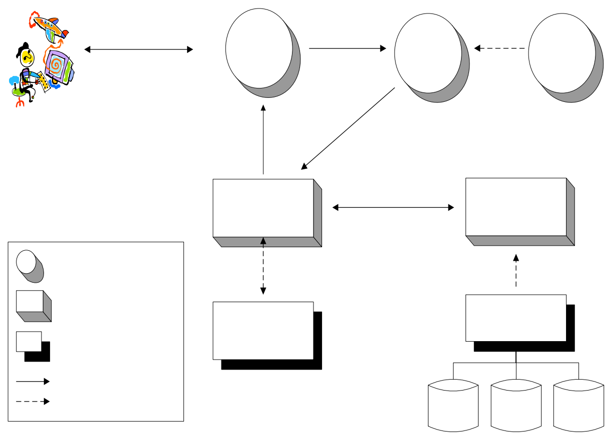

Figure 2 *Story model functional architecture*

3.2. Story Assembly

The story assembly module automatically generates a multi-modal story, in a proper organizational style. The concept of multi-modal stories is to integrate relevant content objects that are related to the chosen concepts. Those multi-modal content objects are instantiated to fill in elements of a story type that is designed to be a template of a story. This story type is chosen by a new structured rule-based decision process. Moreover, this story assembly instantiates a content query formulation process to retrieve relevant content objects in response to a user's intention.

3.2.1. Structured Rule-Based Decision Process

The following process is devised to generate a dynamic multi-modal presentation. Inputs of this algorithm are from an assumed component, called a query processing procedure. These inputs consist of related ontology concepts, including child nodes, a level of generality spectrum that is chosen by a user, and user preferred media types that are also selected by the user.

Input: Inputs consist of three elements: related concepts, generality level, and exclusive media types. These are separated by a comma and surrounded by parentheses. For related concepts, child nodes are located inside of parent parentheses. For example, if a user wants to know about NFL teams with a generality level of medium and without seeing video clips, this request will be presented as follows:
((NFL teams* (Packers 49ers … Patriots)), (5), (video))

Process:
If there is a asterisk symbol (*) in the parent concept,
    Go to Case: Structured Collection story type
Else if generality level is zero,
    Go to Case: Summary story type
Else invoke a primary search in a content query formulation process with inputs
    If there is a text content object in the top three of the ranked results,
        Go to Case: Text-based story type
    Else
        Go to Case: Non-text based story type

Case: Summary story type
Invoke a primary search in a content query formulation process with inputs
If there is a text story in the top three ranked results,
    Put a title of the base story on the top of the story type
    Put a summary into a story container
    Put two top-ranked image or video elements into big boxes, if applicable
    Put next two top-ranked image or video elements into small boxes, if applicable
    Put audio element into an audio box, if applicable
Else Return NULL

Case: Text-based story type
Invoke a secondary search in a content query formulation process utilizing a vector space model
Put a title of the base story on the top of the story type
Put a text story into a story container
Put secondary search results into paragraphs that contain the concepts for the secondary search
Put the top three general results of the primary search into either side of story container in turn
Put the rest of the results in order of generality level at the bottom of the story type

Case: Non-text based story type
Put a title of the most general element on the top of the story type
Put the most general element at the top
Put the rest of the results in order of generality level

Case: Structured collection story type
Put a title that is a parent concept on a cover page
For each child node concept in the related concepts
    Invoke a structured rule-based decision process with a modified input that substitutes the concept of the child node for the parent concept
    Put the result into the next page

User

3.3. Content Query Formulation

A content query formulation is used to instantiate proper content objects to fill out an appropriate story type that is determined by a structured rule-based decision process. The content query formulation consists of a primary search and a secondary search. The primary search employs a constraint-based k-nearest neighbor search, relevance calculation and generality calculation to retrieve relevant content objects that are truly meaningful to the user. The secondary search is only invoked when a text-based story type is selected. The purpose of this search is to generate a sufficient, comprehensive and effective story. This process adopts a Vector Space Model (VSM) [14] to determine subtopics of the narrative. Once these sub-topics are identified, the process retrieves relevant content objects utilizing a constraint-based k-nearest neighbor search.

3.3.1. Primary Search

The purpose of a primary search is to instantiate relevant content objects that fill out a chosen story type. In addition to this purpose, this process ranks the results based on a distance from an ideal point. In order to select relevant objects, we propose a new constraint-based k-nearest neighbor search. This function finds the k-nearest neighbor content objects from an ideal point that is determined by two variables – weight and generality. Notice that these variables are associated with content objects and a concept in the domain ontology. Thus, these variables are not general values, but specific values of a certain set of the content objects and the concept. These weights are initially assigned by a semi-automated metadata acquisition procedure based on frequency of term usage, while the generality is determined by an expert.

These weight and generality values are used to feed a constraint-based k-nearest neighbor search. The key of this search is a distance function. The use of effective distance functions has been explored for data domains of information retrieval [14]. Since our approach is represented by low-dimensional feature vectors, the similarity distance of two objects has typically been defined by an appropriate distance function of the points in the Euclidean space.

Although this basic k-nearest neighbor search [6] easily and simply finds k-nearest contents that are related to a concept, this search might not provide the kind of results the user really wants. For example, if the user wants to see a medium level of presentation for a certain topic, the generality point should be around five. The task of this search is to set borders for a generality axis and a relevance axis and to exclude content objects that are outside of those borders. A geometric representation of the above example is depicted in Figure 3.

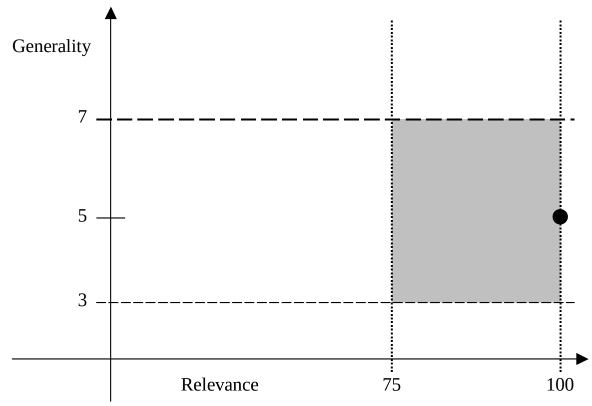

Figure 3 *Geometric representation of constraint-based k-nearest neighbor search*

3.3.2. Secondary Search

The purpose of a secondary search is to generate a complete and comprehensive story in response to a user's request. The secondary search is only initiated when a text-based story type is chosen. The objective of the secondary search is to retrieve content objects that are related to a base story that is determined by a primary search. Even though the elements of text-based story types are integrated by a set of results from the primary search, the base story may include other topics. Because of those topics, the story needs to add other appropriate content objects to deliver a sufficient and effective story. In other words, the final story will include not only multi-modal elements that are related to user's request, but also multimedia content objects that are related to the base story. This process employs a Vector Space Model (VSM) [14] to determine related concepts from the base story. Once related concepts are determined utilizing the VSM and the domain dependent ontology, the secondary search retrieves relevant content objects with a constraint-based k-nearest neighbor search. The secondary search will retrieve the best-matched content object for each concept that is used in the base story and defined in a domain dependent ontology. Moreover, each result will be placed near the concept that is the best match for the result.

4. CONCLUSIONS

We have proposed a new dynamic generation of user-customized multimedia presentations. The proposed system, currently under development[2], defines four domain independent story types to generate a dynamic multimedia presentation in response to a user's intension. – a summary story type, a text-based story type, a non-text based story type and a structured collection story type. In this paper we also focused on improving the quality of results utilizing story types, a constraint-based k-nearest neighbor search, content query formulation and a structured rule-based decision process.

In this research we anticipate basic engineering contributions as well as social science contributions: a new story model can be dynamically instantiated for different user requests using various multi-modal elements; a story model is domain-independent so that this work can easily be adapted to other information domains; a story construction will become independent of a single creator or editor, increasing objectivity in storytelling; and integration of multiple media – audio, video, animation, text, etc. – will increase user understanding, accessibility and interest in obtaining news information.

[1] This research was supported in part by the USC Integrated Media Systems Center, a National Science Foundation Engineering Center, cooperative agreement No. EEC-9529152.

[2] We have constructed several previous proof-of-concept prototypes to lead us to our current system.

5. REFERENCES

[1] Bertino E., Catania B., Ferrari E., and Trombetta A. Presentation Constraints for Multimedia Data. In International Workshop on Multimedia Information Systems, 1996; 26-28.

[2] Bechhofer S and et al. An Informational Description of OIL-Core and Standard OIL: A Layered Proposal for DAML-O. http://www.ontoknowledge.org/oil/down1/dialects.pdf, 2002.

[3] Bodart F, and Vanderdonckt J. Visual Layout Techniques in ultimedia Applications. In CHI Companion'94, 1994; 121-122.

[4] Bunge M. Treatise on basic Philosophy, Ontology I: The Furniture of the World. Reidel Publishing Co., vol.3, 1977.

[5] Cruz IF., and Lucas WT. A visual Approach to Multimedia Querying and Presentation. ACM Multimedia, 1997.

[6] Fukunaga K. and Narendra PM. A branch and bound algorithm for computing k-nearst neighbors. IEEE Transactions on Computers (TOC), 1975: C-24(7); 750-753.

[7] Gruber TR. Toward Principles for the design of Ontologies used for Knowledge Sharing. In Proceedings of the International Workshop on Formal Ontology, 1993.

[8] Hovy E and Lin C. Automated Text Summarization in SUMMARIST. In Advances in Automatic Text Summarization, Mani I. and Maybury M. (eds), 1999.

[9] Kahng, J., and McLeod, D. Dynamic Classificational Ontologies. Computing the Brain: A Guide to Neuroinformatics (editors Arbib, M., and Grethe, J.), Academic Press, 2001, Pages 241-254.

[10] Khan L. and McLeod D. Effective Retrieval of Audio Information from Annotated Text Using Ontologies. In Proceedings of ACM SIGKDD Workshop on Multimedia Data Mining, 2000; 37-45.

[11] Khan L. and McLeod D. Audio Structuring and Personalized Retrieval Using Ontologies. In Proceedings of IEEE Advances in Digital Libraries, Library of Congress, 2000; 116-126.

[12] McGuinness DL and et al. DAML-ONT: An Ontology Language for the Semantic Web. The Semantic Web, MIT Press, 2002.

[13] NetCraft. Web Server Survey. http://news.netcraft.com/, 2003.

[14] Salton G. and McGill MJ. Introduction to Modern Information Retrieval. Mcgraw Hill, 1983.

[15] Shneiderman B and Kang H. Direct Animation: A Drag and Drop Strategy for Labeling Photos. Proceedings of International Conference on Information Visualization (VI2000), 2000; 88-95.

[16] Zhouh MX. Visual Planning: A Practical Approach to Automated Presentation Design. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1999; 634-641.