Jl. of Educational Multimedia and Hypermedia (1994) 3(1), 3-19

An Abstract Model for Hypermedia Study Guides

JUDI JONES
Novell, Inc., 122 East 1700 South, M/S D-22-2
Provo, UT 84606-6194, USA

SUZANNE WAGNER DIETRICH
Department of Computer Science and Engineering
Arizona State University, Box 875406, Tempe, AZ 85287-5406, USA

To date, hypermedia development has been ad hoc, driven by user interface and implementation concerns, resulting in proprietary systems. As a first step toward portable systems, researchers have identified the development of hypermedia abstract data models. This paper presents an abstract model for hypermedia in the application domain of study guides. The model uses an object-oriented approach, where objects of the model correspond to hypermedia objects, and is built on the declarative foundation of first-order logic. This method produces a powerful model that is easy to construct and extends to cover new media technology. Built-in object names validate model integrity and provide cross-referencing in addition to explicit links. Other user-defined objects support a rich and flexible modeling environment. Incorporating deduction into the model enriches its expressiveness by capturing derivations, behavior, or actions.

INTRODUCTION

The beauty of hypermedia is its ability to reduce the mechanics of information search to a trivial task. One educational application that benefits from this feature is test guides. These guides are typically organized into sections of questions with answers, explanations, figures and a glossary of terms or abbreviations. In the standard paper version of a test guide, the reader must constantly flip between these sections to practice taking the examination. Subsequently, the reader is distracted from learning by the mechanics necessary to utilize the guide. A hypermedia implementation of a test guide overcomes this distraction and presents the information intuitively.

In contrast to traditional business databases, a hypertext database consists of units of text that are arbitrarily diverse in form and content (Tompa, 1989). As an example, instead of simple atomic values for data units such as strings, a unit may be a sentence, a paragraph, a page, or even an entire document. These units are tied together by logical association. Each unit is then traversed as directed by the user. That is, the user controls the navigation through the information. Consequently, only interesting or relevant information is reviewed and the user is not burdened with wading through useless information.

Hypertext promises to enhance existing learning strategies since it allows users to make their own decisions about which links to follow and in what order (Conklin, 1987; Marsh & Kumar, 1992; McAleese, 1989). Thus, hypertext eases restrictions on the user (Conklin, 1987).

Multimedia describes the phenomena of combining various media, such as text, graphics, scanned images, sound, slides, and animation to express concepts. Hypermedia extends hypertext to include multimedia (Nielsen, 1990; Yankelovich, Haan, Meyrowitz, & Drucker, 1988). Hypermedia supports the dynamic communication of concepts by illustrating ideas with audio, video and/or textual media (Abbott, 1987; Ambron, 1988). This allows the student to visualize and comprehend complex phenomena as well as to encourage further exploration. By creating new ways of communicating combined with flexibility, hypermedia presents an excellent data structure for applications organized in a question and answer format such as study guides.

Although hypermedia is a useful data organizer, there are limitations with current hypermedia authoring tools. The primary limitation is that it is virtually impossible to transfer the information from one hypermedia system to a different one. This is a serious obstacle since information that has been hypertexted is no longer available for alternative presentation. As technology advances, the data essentially becomes lost as it may need to be manually re-entered for the next generation of hypermedia systems. This problem of data interchange or transfer is not new to computerized information systems. Database technology has overcome related issues with the development of the relational data model.

Relational databases provide an abstraction mechanism that hypertext lacks. Conceptually, information based abstraction mechanisms are beneficial for describing the data relationships separate from the data itself. This is useful for discussion, data validation and standardizing data units associated with a problem (Garg, 1988). The benefits of abstraction suggest that the development of models for hypertext will encourage advancement of this technology (Garg, 1988; Halasz & Schwartz, 1990).

The need for development of an abstract model for hypertext (hypermedia) has been identified by many researchers (Campbell & Goodman, 1988; Garg, 1988; Tompa, 1989). The National Institute of Standards and Technology (NIST) conducted a Hypertext Standardization Workshop in January, 1990, to discuss a reference model and data interchange format (Nielsen, 1990). The NIST workshop attendees concluded that not only should discussions continue on model development, but also that a standards committee with official status in the American National Standards Institute (ANSI) should be established (Baronas, 1990). Clearly, attention from a national agency to the issue of model development signifies its importance and value.

The primary goal of this research is to model the salient features of a hypermedia database system as a basis for exploration of a data interchange format appropriate to these systems. A number of reference models have been proposed; however, a standard model is yet to been defined (Baronas, 1990). For discussions of related work on hypermedia models, refer to Campbell and Goodman (1988), Furuta and Stotts (1990), Garg (1988), Halasz and Schwartz (1990), Hardt-Kornzacki, Gomez, and Patterson (1990), Lange (1990), Oren (1990), Parunak (1990), and Thompson (1990). This paper introduces model components for hypermedia based on first order logic (Jones, 1991), extending the work of Garg (1988) and Lange (1990), and demonstrates concepts necessary to constructively represent a test guide hypermedia system regardless of the test guide subject matter or the hypermedia authoring tool. The paper concludes with a discussion of model usage in data exchange followed by a brief summary.

HYPERMEDIA TEST GUIDE ABSTRACT MODEL

A database reference model is useful for description, standardization, design and innovation (Parunak, 1990). Abstract models provide a means for authors to organize and describe the database to users. This concept is demonstrated by a review of the relational model, which is a logical model based on records (Korth & Silberschatz, 1991). The relational model depicts the database as a collection of tables with named columns. Tables are a simple and intuitive notion used to describe the logical structure of the database. After the relational model was developed, relational database theory prospered and resulted in a theory for the design of relational databases (Korth & Silberschatz, 1991) that assists database designers. In contrast to relational databases, hypermedia is not yet associated with any abstract model to assist users or authors (Baronas, 1990; Nielsen, 1990).

A reference model guides the designer in identifying the issues and broad functions that the systems must provide (Parunak, 1990). Abstraction mechanisms benefit readers and implementers in terms of authoring and information exchange (Hardt-Kornzacki et al., 1990). Combining diverse hardware and software, hypermedia systems demand the use of abstract models (Oren, 1990) in order to accommodate a broad spectrum of authors (Hardt-Kornzacki et al., 1990). Standards for hypermedia must emerge before hypermedia databases are fully useful (Crane, 1990). Until hypertext and multimedia standards provide a platform that supports different applications, it cannot play a major role in the publication or long-term archiving of information (Crane, 1990; Mann, 1992; Schroeder, 1992).

Although presentation, problem domains and functionality of hypermedia systems vary substantially, the underlying database structure remains very similar (Ressler & Stribling, 1990). The database structure consists of a directed graph whose vertices represent nodes and their contents, while edges are cross-references between the nodes (Conklin, 1987; Ressler & Stribling, 1990; Tompa, 1989). Fortunately, the consistent structure lends credence to the idea that a model for diverse systems can be developed. Standardizing a hypermedia model requires ignoring the dissimilarities between applications and developing a metaphor based on the similarities.

Abstraction Design

An effective abstraction provides a number of advantages, these include support of a logical structure and separation of information, detection of information relevancy, and consistent style (Cole & Brown, 1990; Hashim, 1990). Designing an effective abstraction is not a simple task as demonstrated by the history of the languages used to program a computer. Advancement in computer languages have continuously guided a programmer's worries away from the actual machine and towards the process steps to complete a task, thereby validating program expression in domain terms as a legitimate goal (Abbott, 1987). The model should allow the designer to define data types that correspond to categories of objects that are found in the problem domain (Abbott, 1987).

The selection of aspects to include in the model is a complex challenge (Hardt-Kornzacki et al., 1990). The challenge may be reduced to identification of reasonable modeling constructs based on the data structure (Bornstein & Riley, 1990). The standardization goal takes form by classifying critical themes and relationships in the domain (Parunak, 1990). The reference model should be as comprehensive as possible, able to embrace any implementation (Oren, 1990; Parunak, 1990), but must also be flexible enough to evolve with technological advances (Hardt-Kornzacki et al., 1990; Oren, 1990).

Analyzing hypermedia databases in terms of dependencies provides a clear direction for the areas best suited for standardization. Obviously, user interfaces and presentation styles may differ dramatically among applications. Furthermore, file and database management systems rely heavily upon the machine architecture in their implementations (Nielsen, 1990). Recognition of these factors suggests that the abstraction level should occur between the presentation and storage levels as shown in Figure 1, which illustrates the appropriate level for abstraction of hypermedia systems (Campbell & Goodman, 1988; Nielsen, 1990).

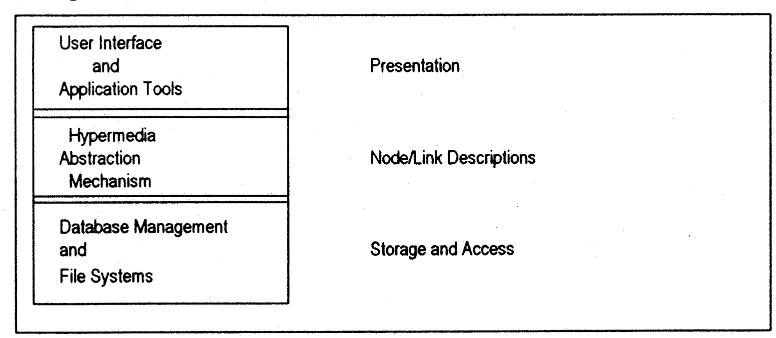

Figure 1. Location of hypermedia abstraction mechanism

The Hypermedia Abstraction Mechanism (HAM) determines the basic nature of objects and the relations between them (Nielsen, 1990). Issues such as screen layout are specified within the presentation level (Oren, 1990), while data retrieval is contained in the storage level.

The intent of this research is to identify a flexible, extendible and natural modeling tool for hypermedia systems between the presentation and storage levels. The model must be able to capture the essence of hypermedia, capturing the objects in that domain yet be able to provide a data interchange capability through a declarative representation. We chose a model based on first-order logic, which provides an intuitive format for structuring the hypermedia objects. The logic paradigm will be introduced by example as the components of the abstract model are introduced. This paper illustrates the hypermedia abstract model within the domain of study guides.

Test Guide Model

Test guides follow a generic format of questions, possible answers, a correct solution with an explanation and perhaps a source reference for further clarification. This format provides the framework for deciphering the model objects. Using these objects, the relationships can be identified. Questions, of course, are associated with a set of answers, only one of which is correct. Each question and correct solution has an explanation and reference. These objects and relationships are illustrated in Figure 2.

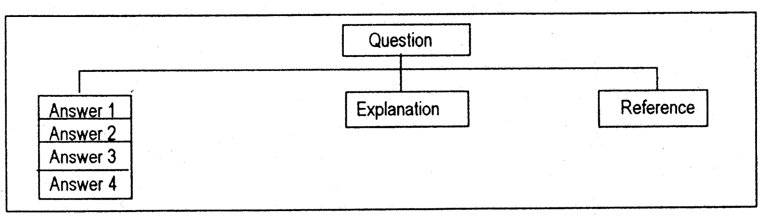

Figure 2. Test guide object relations

Hypertext study guides provide considerable advantage for the educator and the student. The educator can enable valuable learning strategies including the random presentation of questions and answers to questions, easy test construction, and easy test evaluation. The student benefits by learning the material rather than memorizing solutions based on positional identification. Hypermedia manages the material presentation of related information rather than forcing the student to manually organize information by page flipping through the guide. Also the student can emphasize critical information by selecting interesting links and avoiding unnecessary distraction by avoiding irrelevant links.

The test guide model is used to illustrate the abstract model for hypermedia objects based on a declarative logic paradigm. The integral components of the model are domain objects, in this case the objects associated with a study guide, and the nodes and links recognized as the foundation of a hypertext system. Other abstractions, such as buttons, hotwords, object properties, keywords, filtering and derived objects, are also allowed by the model.

Domain Objects. The first step in the model is to identify the fundamental information units to be modeled. For a typical paper version of a study guide, these units are questions, answers, explanations and references. Additional objects included in the domain are the types of media available for presentation such as text, figures, or graphics. In the model, information units are specified as: info_unit(unit_name). Figure 3 shows the information unit specification for a study guide, which declares questions, answers, explanations, references, texts and figures as domain objects relevant to the application domain.

```
info_unit( question )
info_unit( answer )
info_unit( explanation )
info_unit( reference )
info_unit( text )
info_unit( figure )
```

Figure 3. Study guide domain objects

The significance of identifying the information units is to ensure model integrity, that is, information objects must be derived from the defined domain objects (Garg, 1988). An obvious first step toward model validation would therefore consist of verification that object types are only those listed as information units. This verification can be automated easily.

Nodes. Using the objects identified as information units, the modeler enumerates the information objects instantiating the test guide. Each object must be identifiable to allow effortless reuse of the object. For instance, several questions might refer to the same figure, so the figure need only be identified once and its name reused. Accordingly, all objects are uniquely identifiable by a name and number. For test guides, most objects like answers, explanations and statistics emanate from questions. Consequently, the question number is a reasonable numbering convention for object identity. Sample identifications are question(1028) or answer(1028-1) where the 1028-1 refers to selection 1 as an answer for question 1028. The model's format of an object is: object-name(object-number, object-instantiation).

An instantiation of an object is itself an object. A question or answer is typically comprised of a text object, which is a sequence of words. Figure 4 displays an instantiated answer object composed of text with the usage of the construct [] to establish textual order.

```
answer (1028-1, [text (1028-1, [more efficiency due to less friction ] ) ] )
   |                          |
   object                     instantiation
   type                       with sequence
```

Figure 4. Sample answer domain object

One of the advantages of the first order logic model is illustrated by the ease of extending the model to incorporate other media. Another media type or class is added merely by defining a new object-type. Representation of additional media and their instantiations is shown in Figure 5. Figure 5 also demonstrates model flexibility by using either a draw object or a file specification (square.bmp) to delineate a figure containing a square box. The modeler's selection of the representation would depend on the most natural fit for the problem domain.

```
figure (draw-square, [ □ ] )

figure (bitmap-square, [square.bmp] )
```

Figure 5. Sample media objects

Of course, Figure 5 shows a trivial example of sequence containing a single piece of information within the [] brackets. But, it is easy to imagine adding more file names within the square brackets to represent an animation. For clarity, the modeler could choose to rename the object from figure to animation (e.g., animation (fig 50, [square.bmp, circle.bmp, ellipse.bmp])).

In Figure 6a, notice the composition of a more complex object of a question consisting of text and a figure. The sequence construct provides an implicit means of linking text with other media. This particular construction enables the implementer to imitate the published test guide format, thereby, providing initial problem framing in terms of the domain. Of

course, the model does not limit the implementer to using just implicit associations. Explicit linking, to be discussed later, could also be used to link the figure and text.

Another constructor, sets, is depicted with curly brackets. Sequence objects ([]) are a method of defining order or composition. In contrast, sets are an arbitrary grouping of items. For example, a reference may apply to several different questions and these questions constitute a set. The key distinction between sequence and sets is the imposed ordering of elements. In other words, when traversing a link from a reference to a question, it makes no difference which question in the set is seen first. However, the set of questions clearly indicates that several links are available for traversal as illustrated in Figure 6b.

```
a. Question with Text and Figure
   question (1889, [text (1889, [the maximum cloud top in Figure 58 is] ),
                    figure (58, [fig58.pcx] ) ].

b. Reference for Two Questions
   reference (18, {question (1889), question (2002) } ).
```

Figure 6. Composition using sequence or sets

Links. Links represent explicit relations between objects in the model. A link is specified as a link relation by the format: link_relation (link-id, link-source, link-destination). Link-ids must adhere to the previously listed object identity requirements. The link's source and destination are labeled anchor and destination, respectively. Any model object is eligible as a link anchor or destination.

The general form of an anchor or destination object is identical except for the name: anchor(anchor-id, anchor-object) or destination(dest-id, dest-object). Again, the id combines with the name to uniquely specify the object. The anchor-object or dest-object is constructed from any existing object within the model. Figure 7a illustrates a link anchor of question 1028 and Figure 7b illustrates a link destination of answer 1028-1. Once the anchors and destinations are defined, these objects are inserted into the link relation as the anchor and destination, respectively. Figure 7c uses the previously defined anchor and destination (Figures 7a and 7b) to illustrate the link between question number 1028 and its answer selection 1028-1.

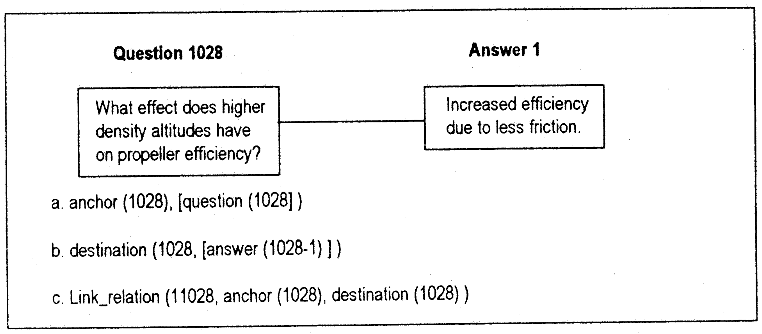

Figure 7. Anchor, destination, and link examples

One might consider whether the anchor and destination objects unnecessarily complicate matters. Another approach is to load the link relation with the question and answer identifiers directly. Unfortunately, that approach limits the expressive nature of the model. Using the anchor and destination representation allows succinct expression of multiple sources or destinations, perhaps portraying nodes that are meant to be traversed simultaneously. Figure 8 illustrates a link having multiple destinations from a single anchor. Figure 8 illustrates that there are three questions (12, 39 and 45) that have the same reference (28).

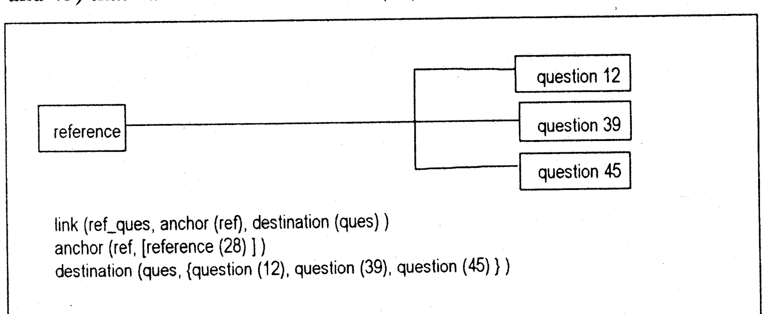

Figure 8. Example link with multiple destinations

To summarize, the link relation is a versatile and flexible modeling tool. It allows the modeler to gracefully represent links with multiple start and end points. Using model objects in the anchor and destination objects permit the modeling of links between nodes, or links between node contents and another node. Specification of a link connection as a set or sequence enables modeling a variety of connection semantics. Modelers are free to define their own relations, such as annotations, in order to better accommodate different domains. With all of these features, the model supports many link abstractions.

Other Abstractions. Additional abstractions are supported by the model including hotwords and buttons, properties and keywords, and derived objects. The remainder of this section summarizes the purpose and implementation of these additional abstractions provided by the model using, as a motivational example, a portion of a test guide database depicted in Figure 9, with its corresponding model components given in Table I. This test guide example was derived from an implementation that was used to validate model development.

Hotwords and buttons are modeled by a handle model object, which represents a connection point to other information. A hotword is usually a distinguished subset of text that is typically italicized or underlined, and the activation of the hotword results in navigation to a new location. For example, Question 1868 consists of text and a reference to a supporting figure in tif format (fig 54.tif) using a hotword link. The implicit link to the tif file is indicated by arrows heads (▲,▼). The hotword link is activated when the user selects the italicized text "Figure 54". Handles are also used to model buttons, which typically are icons having a textual label. In Figure 9, the explicit link from question 1868 to explanation 1868 is activated by a user selecting an explanation button.

The model also allows the assignment of properties or keywords to objects, which assists in the categorization of material. A property or keyword can be associated with an object to aid in navigation. For example, in a test guide, a reader trying to locate questions about a specific subtopic may be forced to sequentially scan the entire guide. An implicit cross-reference of questions by subtopic allows readers to search test material effectively. Properties and keywords then provide another means to navigate or search the model based on that property. In Figure 9, an aircraft property is associated with questions, indicating which question is applicable to which types of aircraft. Question 1868 is applicable to students of *all* types of aircraft as indicated by the double-ended property box labelled All in Figure 9.

A deductive capability enhances the model dramatically by providing a method to compute any kind of object based on evaluable characteristics. Deduction provides a succinct modeling capability. For example, in Table I, deduction is used to identify correct and incorrect answer selections. The validity of the answer selection is logically embedded in the question representation, which characterizes the behavior of answer selection 2 (answer(1868-2)) as the correct solution and all other answer selections asincorrect.

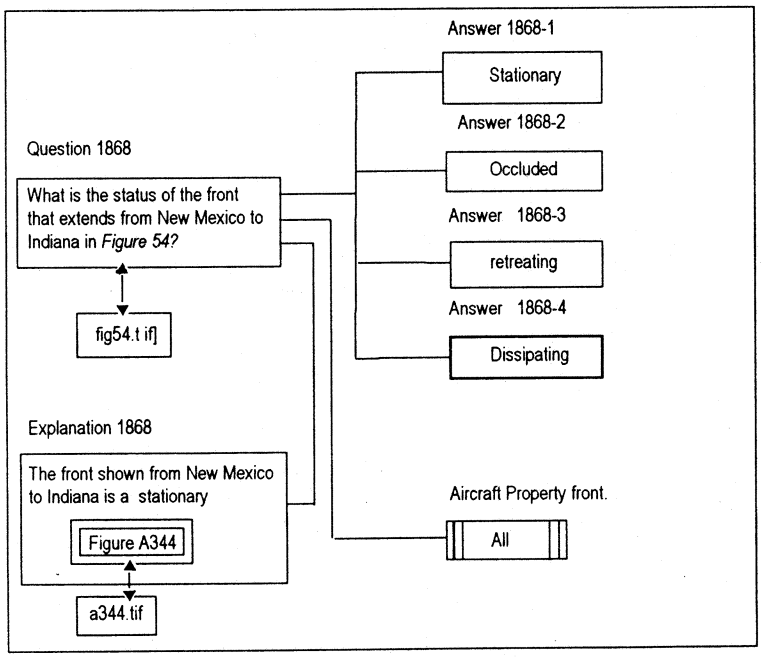

Figure 9. Sample test guide database

Model Summary

The model presented advances an object-oriented approach based in first-order logic to abstract hypermedia test guides. The combination of objects modeled in first order logic generalizes objects in the database, making them available for reasoning activities. This generalization permits the declarative specification of nodes and links to lead to any conceivable computer operation (Parunak, 1990). Treating nodes as objects composed of other objects overcomes representation problems associated with the variety of node contents (Cole & Brown, 1990; Parunak, 1990).

Table 1
Partial Test Guide Model

Components	Model Statements
Domain Objects	info_unit(question) info_unit(answer) info_unit(explanation) info_unit(reference)
Question & Hotword	question(1868, [text(q1868, [what is the status of the front that extends from New Mexico to Indiana in handle(Figure 54, [fig54.tif])])]) if select(answer(1868-2)) then (answer(right) else answer(wrong)
Property	aircraft(question(1868), {all}).
Answers	answer(1868-1, [text(a1868-1, [Stationary])]).
	answer(1868-2, [text(a1868-2, [Occluded])]).
	answer(1868-3, [text(a1868-3, [Retreating])]).
	answer(1868-4, [text(a1868-4, [Dissipating])]).
	answer(right, [text(t1, [Congratulations -- Correct])]).
	answer(wrong, [text(t2, [Try Again])]).
Relations	link relation(qa1868,anchor(q1868),destination(a1868)).
	link relation(qe1868,anchor(q1868),destination(e1868)).
Anchor	anchor(q1868, [question(1868)]).
Destinations	destination(a1868, {answer(1868-1), answer(1868-2), answer(1868-3), answer(1868-4)}).
	destination(e1868, [explanation(1868)]).
Explanation & Figure	explanation(1868, [text(e1868, [the front shown from New Mexico to Indiana is a stationary front]), figure(a344, [a344.tif])]).

Using object names to define object classes frees users from the rigors of manifesting the database in terms of restrictive modeling constructs. User definition of object names, not only allows modeling to be expressed in the terminology of the problem domain, but also permits incorporation of new media types as technology progresses. Deductive capabilities from first-order logic provide a mechanism to define the semantics of new technology as well as to communicate the behavior and properties of objects. Abstracting the hypermedia database through the use of an object-oriented strategy that is based in first-order logic produces a model that is flexible, extendible, and natural with respect to the problem domain.

The model structure incorporates a number of features. Table 2 summarizes the model components and abstractions described in this section. Nodes and their types are represented by domain objects (such as questions and answers), eliminating the need for a separate attribute for type. Node contents are built from media domain objects (such as text and figures), thereby freeing the contents from structural constraints. Objects may have order imposed on them by using the sequence construct ([]) or alternatively, objects may consist of an arbitrary group of elements denoted by a set

({}). Links are supported by relations using anchor and destination objects, permitting multiple sources or destinations or to simulate parallel traversal. Multimedia links are expressible directly using relations or implicitly through the grouping of domain objects.

Table 2

Model Components and Abstractions

Model Components	Abstractions Supported
Domain Objects	Nodes, Media Types, Composition, Information Units
Object Identity	Object Reuse
Instantiation	Actual text or other media displayed
Sequence Objects	Order and Composition
Sets	Arbitrary Grouping
Link-Relation	Connection
Anchors and Destinations	Multiple links or simultaneous traversal
Handles	Hotwords and Buttons
Keywords, Properties	Cross-Referencing or Classification
Deduction	Derived Objects, Links, Properties or Behavior

DATA TRANSFER

The purpose in cataloguing data into a generic model is to facilitate data transfer between applications. Building the model may appear a cumbersome task, but the benefit is enormous. The catalogue of information resulting from construction of the model encourages the usage of data to fit many different purposes by merely changing the presentation. In other words, a presentation of a chemistry experiment and chemistry lecture share common information. The modeler then can take advantage of the identical information and provide a general abstraction of the entire chemistry class material by extending information in the model.

A second benefit of the model is that it readily supports the situation where an application is developed in tandem by a domain expert and a hypermedia tool specialist. More importantly, the model assists the author responsible for both roles by logically separating these tasks. A better designed product will likely result since the concurrent focuses do not have to be maintained.

The reference model presented in Section 2 is primarily a tool to develop translation between hypermedia systems. The conversion from the

model to a particular hypermedia application requires merely the development of a translation scheme, which converts model components to specific hypermedia authoring tool constructs. Once these translation schemes has been derived, the generic model provides a mechanism to extract or import data from or to arbitrary applications regardless of internal protocols.

In contrast, porting between specific hypermedia applications requires extensive knowledge of the internal protocols associated with each system plus the actual information stored in them. Since the model is simple and contains a precise number of information units, it is likely that a mapping to complex objects with varying capabilities is possible. This mapping consists of a one-to-one correspondence between the model and hypermedia system's constructs. Mapping between dissimilar, complicated tools requires knowledge of not only both applications, but forces a complicated mapping between them where a one-to-one correspondence is less obvious without substantial analysis. The model language is expressively rich while being simple to learn and apply such that the information structure may be represented without concern for a specific application's internal workings.

SUMMARY AND CONCLUSION

This research developed an abstract model for hypermedia test guides. Using an object-oriented approach built on first order logic produced a powerful model that is easy to construct and readily upgradeable. Indicating the object names as information units can be used to validate model integrity. All object names except info_unit and link_relation are user-defined, supporting a rich and flexible modeling environment. Incorporating deduction into the model enriches its expressiveness by capturing behavior, actions, and quantification.

Standards cannot progress without applying them to bodies of data (Crane, 1990). As a consequence, a prototype system was built to ensure comprehensive model development (Jones, 1991). The research methodology ascribed a three part approach: proposing an initial model, building the hypermedia test guide, and then validating the model against the prototype. This process yielded invaluable insight as to the degree of complexity requiring model support. The implementation illustrated that the model is expressive enough to capture the essence of the prototype as well as features offered by the authoring tool.

In conclusion, an object-oriented reference model based on first order logic provides an effective representation of hypermedia test guides.

REFERENCES

Abbott, R.J. (1987). Knowledge abstraction. *Communications of the ACM, 3*(8), 664-671.

Ambron, S. (1988). Why is multimedia importation in education. *Interactive Multimedia.* Redmond, WA: Microsoft Press.

Baronas, J. (1990, May-June). Conference report: Hypertext standardization workshop, Gaithersburg, MD, January 16-18, 1990. *Journal of Research of the National Institute of Standards and Technology, 95*(3), 345-348.

Bornstein, J., & Riley, V. (1990). Hypertext interchange format. *Proceedings of the Hypertext Standardization Workshop* (pp. 39-48). Gaithersburg, MD.

Campbell, B., & Goodman, J.M. (1988). HAM: A general purpose hypertext abstract machine. *Communications of the ACM, 31*(7), 865-861.

Cole, F., & Brown, H. (1990). Standards: What can hypertext learn from paper documents? *Proceedings of the Hypertext Standardization Workshop* (pp. 59-70).

Conklin, J. (1987). Hypertext: An introduction and survey. *IEEE Computer, 20*(9), 17-41.

Crane, G. (1990). Standards for a hypermedia database: Diachronic vs. synchronic concerns. *Proceedings of the Hypertext Standardization Workshop*, (Gaithersburg, MD, January 1990). National Institute of Standards and Technology (pp. 71-81). Washington, D.C.

Furuta, R., & Stotts, P.D. (1990). The trellis hypertext reference Model. *Proceedings of the Hypertext Standardization Workshop* (pp. 83-94).

Garg, P. K. (1988). Abstraction mechanisms in hypertext. *Communications of the ACM, 31*(7), 862-870.

Halasz, F., & Schwartz, M. (1990). The dexter hypertext reference model. *Proceedings of the Hypertext Standardization Workshop* (pp. 95-134).

Hardt-Kornzacki, S., Gomez, L. M., & Patterson, J. F. (1990). Standardization of hypermedia: What's the point. *Proceedings of the Hypertext Standardization Workshop* (pp. 135-144).

Hashim, S. H. (1990). *Exploring hypertext programming: Writing knowledge representation and problem-solving programs.* Blue Ridge Summit, PA: Wincrest Books.

Jones, J. (1991). *An abstract model for hypermedia study guides.* Master's Thesis, Department of Computer Science and Engineering, Arizona State University, Tempe, AZ.

Korth, H.F., & Silberschatz, A. (1991). *Database system concepts* (2nd ed.). New York: McGraw Hill.

Lange, D.B. (1990). A formal model of hypertext. *Proceedings of the Hypertext Standardization Workshop* (pp. 145-166).

Mann, M. (1992). Multimedia must display Mmaturity to sway buyers. *PC Week, 9*(4), 79.

Marsh, J.E., & Kumar, D.D. (1992). Hypermedia: A conceptual framework for science education and review of recent findings. *Journal of Educational Multimedia and Hypermedia, 1*(1), 25-37.

McAleese, R. (1989). Navigation and browsing in hypertext. In Ray McAleese (Ed.), *Hypertext: Theory into practice* (pp. 6-44). Norwood, NJ: Ablex.

Nielsen, J. (1990). *Hypertext and hypermedia.* San Diego, CA: Academic Press.

Oren, T. (1990). Toward open hypertext: Requirements for distributed hypermedia standards. *Proceedings of the Hypertext Standardization Workshop* (pp. 189-196).

Parunak, H.V.D. (1990). Toward a reference model for hypermedia. *Proceedings of the Hypertext Standardization Workshop* (pp. 197-212).

Ressler, D., & Stribling, D. (1990). Designing and prototyping a portable hypertext application. *ACM SIGDOC, 14*(4), 87-94.

Schroeder, E. (1992). Multimedia applications slowly enter mainstream. *PC Week, 9*(11), 20.

Thompson, C.W. (1990). Strawman reference model for hypermedia systems. *Proceedings of the Hypertext Standardization Workshop* (pp. 223-246).

Tompa, F.W. (1989). A data model for flexible hypertext database systems. *ACM Transactions on Information Systems, 7*(1), 85-100.

Yankelovich, N., Haan, B.J., Meyrowitz, N.K., & Drucker, S.M. (1988). Intermedia: The concept and the construction of a seamless information environment. *IEEE Computer, 21*(1), 81-96.

Acknowledgement

This research was supported by a Minority Graduate Fellowship granted from Bull Worldwide Information Systems.