Equuleus: Presentation from Legacy Documents*

Susan W. McRoy, Syed S. Ali, and Nipat Nalamlieng
Natural Language and Knowledge Representation Research Group
Department of Electrical Engineering and Computer Science
University of Wisconsin-Milwaukee

Abstract

We are investigating computational models for human-computer interaction based on a uniform, declarative, knowledge representation. We describe a new implementation of our work called Equuleus, a system for interactive, adaptive, document presentation. Equuleus achieves this by using annotated descriptions of a document to be presented and a planner to plan the document's presentation. All presentation occurs as a result of planning and re-planning presentation goals.

1. Introduction

There is a large quantity of pre-existing text for training or education. The presentation of these documents would be enhanced if they could be presented adaptively, incrementally, and dynamically. To our knowledge, most presentation systems currently available use hand-constructed knowledge representation. As a result, such systems are capable of sophisticated presentations, for example, building graphics from scratch to fit the text presentation. However, re-targeting such systems to other applications or domains requires a great deal of effort.

We propose a plan-based approach for providing adaptive presentation of existing documents with minimal additional markup. This plan-based approach allows interactions between the adaptive presentation system and users. In addition, the system can monitor the responses from users by observing the overall planning process. The result is that we have an adaptive presentation system that dynamically and incrementally customizes the presentation to suit the user most effectively.

By plan-based, we mean that the presentation system can reason about its possible action(s) to achieve goal(s), can infer new knowledge related to the current environment, and adapts its action(s) according to the new knowledge. Our implementation of this approach is called Equuleus. Equuleus utilizes the Semantic Network Processing System (SNePS) [20] as a planner/reasoner to produce presentation plans. SNePS provides a STRIPS-style planner [5] (each plan has a precondition, a postcondition, acts, and effects) as a general purpose planning tool. Additionally, SNePS performs truth maintenance [8] and allows partitioning the knowledge to make planning more efficient.

Besides the planner/reasoner, Equuleus makes use of: efficient techniques, which require minimal expertise, for re-authoring existing documents to work adaptively; knowledge representation(s) for document content and pedagogy, domain-independent knowledge about pedagogy, and user model(s); and user feedback about the state of the interaction.

In Equuleus, the knowledge representation includes information about, for example, how the document is structured, how subsections in the document are semantically linked together, what the user is presumed to know, and rules for deciding when to modify information believed about the user. Equuleus automatically generates STRIPS-style plans from a pre-authored document (the *master document*) which is authored in a dialect of the eXtended Markup Language (XML). The master document used in Equuleus requires only minimal markup to create subsections of the document, add semantic relations among those subsections, put tests into the document, and organize subsections into a document in the desired manner. This approach gives Equuleus the flexibility to be re-targeted to other documents easily. (We present a more detailed discussion of the authored document in Section 3.2.)

Knowledge concerning the master document, the so-called document specific knowledge, is comprised of:

- Document specific rules about how the document is structured.
- Document specific knowledge about semantic relations [7] among document constituents

* We acknowledge the financial support of the National Science Foundation (under grants IRI-9701617 and DUE-9952703), Wright State University, and Intel Corporation.

For example, the rules about document structure might specify that the document is composed of five lessons, each of which is composed of several objectives or units. So, a plan for presenting this document is to present these five lessons. The plan for presenting each lesson will activate the act(s) of presenting objective(s) or unit(s) according to the structure of each lesson. The effects of presentations are new knowledge about which subsections have been presented. Semantic knowledge about a document might include information such as one objective being an introduction of a unit. This information facilitates the planning and reasoning when the document independent knowledge is applied to customize a presentation.

The document independent knowledge is comprised of:

- Generic rules for creating a presentation that determine what to present and what not to present. Currently, this is tied to the type of user: Novice, Intermediate (which has three sub-levels), and Expert.
- Generic rules for adjusting the user level.

For example, the presentation rules allow the presentation or omission of semantically marked constituents such as elaborations, examples, and figures, based on the user type. The plan for presenting any document constituent has to check for the type of semantic relation and the current type of user (these are two preconditions of the current plan). The user adjustment rules allow changes in user types based upon the result(s) of prior test(s). Although the current set of generic rules is limited, Equuleus provides a generic architecture for more complex types of user models and rules for adjusting presentations on the basis of these user models.

The combination of document independent and document specific knowledge is used for planning the presentation.

Finally, Equuleus includes a simple graphical user interface that allows a user to navigate the document and to respond to a presentation. Currently, the response forms are tests of understanding; either a self-assessment or an automatically generated comprehension test.

2. Background

We are specifying and building intelligent systems that collaborate with users for health education and training. As part of this work we have previously specified and built systems for: giving medical students an opportunity to practice their decision making skills in English (B2); assessing, educating, and evaluating user's knowledge of health states (PEAS); detecting and rebutting arguments (ARGUER); and performing template-based natural language generation (YAG) [3, 10, 11, 12, 13, 14, 15].

The general goal of our work is to investigate computational models of dialog or presentation that can support effective interaction between people and computer systems. We are doing so by providing a uniform specification and representation of the knowledge that determines how people understand and respond in an ongoing interaction. Our research also aims to provide computational methods for integrating and using this information to produce relevant output and to identify and resolve communication problems as they arise. Our work makes use of standard first-order reasoning and planning.

Here, we are concerned with the effective presentation of pre-existing documents, in particular, training manuals. To be effective, computer systems for training must facilitate users' understanding by incrementally presenting information, by evaluating users' understanding, and by adapting the interaction to address communication problems as they arise.

Techniques for adaptive interaction and presentation can be categorized into two main categories: **adaptive navigation** and **adaptive text planning**. Adaptive navigation is concerned with how to provide useful links to help users focus on relevant information, i.e. reducing the search space. They assume that each link will lead to a complete, self-contained text fragment. Adaptive navigation systems include IDAS [19], ILEX [17], and PEBA-II [16]. By contrast, adaptive text planning is concerned with how to select fragments of text and possibly re-establish coherence among them to form a single fragment targeted to the background or goals of the user. Adaptive text planning systems include Health Doc [4], MATCH [21], MIGRAINE [2], and PPP [1]. A common feature of all these systems is that they use a fine-grained, hand-constructed knowledge representation appropriate to the particular task. As a result, while such systems are capable of sophisticated presentations, re-targeting such systems to other applications or domains requires significant effort. By contrast, because Equuleus makes use of a coarse, hierarchical document structure it requires minimal additional knowledge representation (task-specific or otherwise) for documentation presentation.

3. Dynamic Plan-Based Presentation

The architecture of our system, Equuleus, is shown in Figure 1. The main components of Equuleus are the following:

- *Document-Specific Knowledge:* the authored document (in XML) which is mapped into a series of plans and actions that achieve an incremental presentation of a text;
- *Document-Independent Knowledge:* a set of document-independent plans that modify a presentation to take into account the user model, user input, and modality;

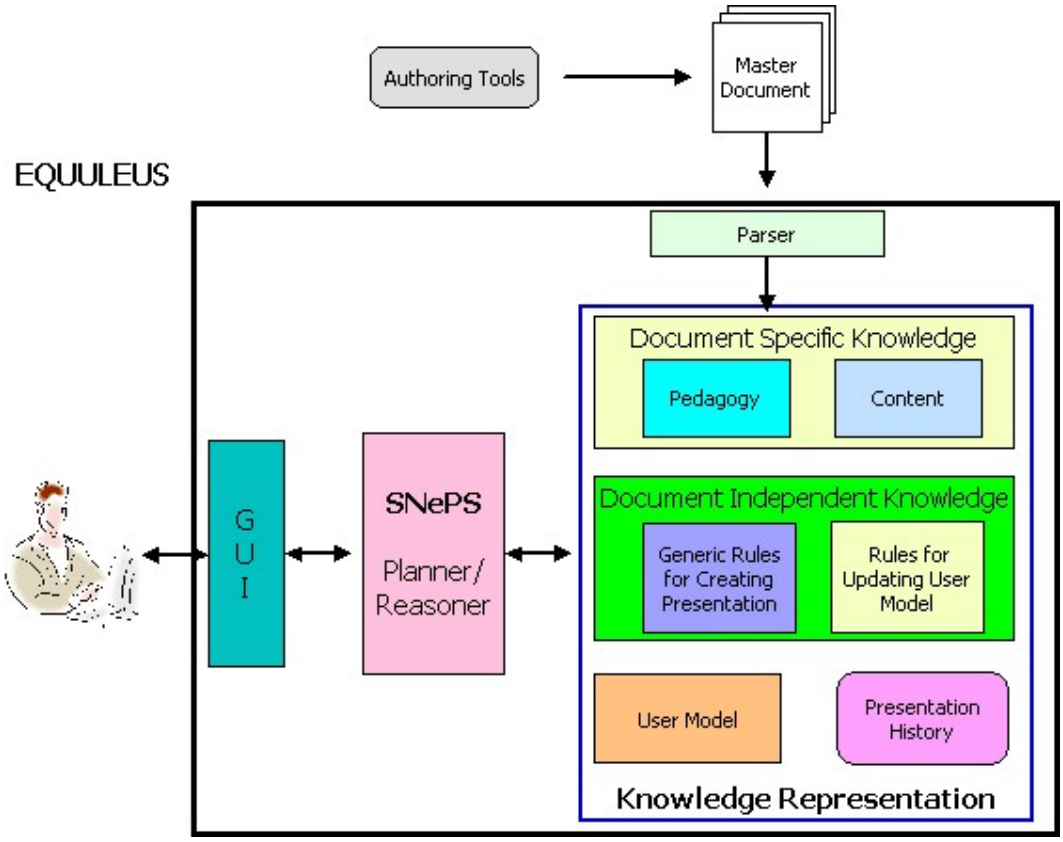

Figure 1. Architecture of Equuleus

- *SNePS:* a general-purpose planner and reasoner that plans content;
- *GUI:* and an interface between the user and system that displays content and solicits feedback from the user.

At runtime, Equuleus considers only the necessary knowledge to plan the presentation of a specified document constituent, which optimizes the speed of the planning process and the amount of memory used.

The diagram in Figure 1 illustrates the architecture of the Equuleus system. The Graphical User Interface(GUI) deals with input and output between the user and the Planner/Reasoner. The users interact (explore document structure, select document constituents to present, see the presentation, and give feedback) with the GUI. The input modality is mouse clicks. The output modalities include text and graphics. The Planner/Reasoner (SNePS [20]) receives commands and responses from the GUI and uses the Knowledge Representation to produce the presentation. The Document Independent Knowledge includes the Generic Rules for Creating Presentations and the Rules for Updating the User Model. The Authoring Tools semi-automatically convert a legacy document into an XML master document (these tools automatically segment the text and provide a GUI for marking-up the resulting segmented document.) The Document Specific Knowledge is produced by parsing the master document (which is authored from a text document: *Blood Pressure Measurement Education Program Instructor Manual* produced by the American Heart Association) and is comprised of the Pedagogy (how to present the document as well as semantic relations between document constituents) and the Content (paragraph-level text including headings).

3.1. SNePS: Planning and Reasoning

SNePS is a system for building, using, and retrieving information from propositional semantic networks. SNePS includes a planning and acting engine, SNeRE[20], which is the package used in Equuleus for planning and reasoning about presentation plans. SNePS also provides truth maintenance and knowledge partitioning using *contexts* which allow reasoning and planning using only relevant portions of the knowledge base. Contexts are created dynamically by Equuleus using only the necessary knowledge for planning the presentation of the current goal and will not affect other knowledge or planning. This makes planning and rea-

soning faster because the complexity of planning and reasoning does not depend on the size of the document—it depends on the current presentation goal. The document constituents that have already been presented are logged in the Presentation History.

A presentation goal (*i.e.* a goal to present a document object, such as a goal to present the first lesson) is triggered by the user explicitly requesting (by selecting and mouse clicking in the GUI) the presentation of a document constituent. The planner finds a suitable plan to satisfy the goal (which will have been previously generated from the authored document), which in turn will trigger additional presentation subgoals. When the top-level presentation goal is done, the output is the presentation requested.

3.2. Document Specific Knowledge: The Authored Document

An original machine-readable document (e.g. in MSWord or HTML) is semi-automatically processed into a sequence of segments (corresponding, roughly, to paragraphs, figures, tables, and headings). These segments must then be authored into a coarse, hierarchical, syntactic structure, which includes some semantic information about relationships between document objects. The resulting document is a tree (see Figure 2). The **document** is the root, which is composed of one or more lessons. A **lesson** is composed of segments, objectives, and units. A **unit** (a group of objectives or segments that have a related purpose in a presentation) is composed of objectives or segments. An **objective** (a group of segments that have a single purpose e.g. educate the users about the meaning of high blood pressure) is composed of one or more segments. A **segment** is the smallest constituent in the document. It is a paragraph-level piece of text, a heading, or a graphic from the document. This authored document is automatically converted into planning rules at run-time. This makes Equuleus very flexible since changes to the master document can be reflected immediately. All of these components that form a document are called document objects. The knowledge representation for each document object is stored in its own context in a partitioned semantic network—reasoning and planning to present each document object will only use relevant knowledge.

3.2.1. Semantic Links Links represent semantic relations among two or more document objects. Our usage is based on RST [6]. A link indicates the function between two objects within a document, *i.e.* between a segment and another segment, a segment and an objective, a segment and a unit, or an objective to another objective. Links are not limited to any scope, but it can be extended to link objects from different parts of the document or even objects from different documents. We make use of fifteen link types in Equuleus: TITLE, BACKGROUND, ELABORATE, EXAMPLE, PROCEDURE, REAL-RESULT, POSSIBLE-RESULT, SOLUTION, CONCESSION, NUCLEUS, FIGURE, SUMMARY, RESTATEMENT, COMPARISON, and WARNING.

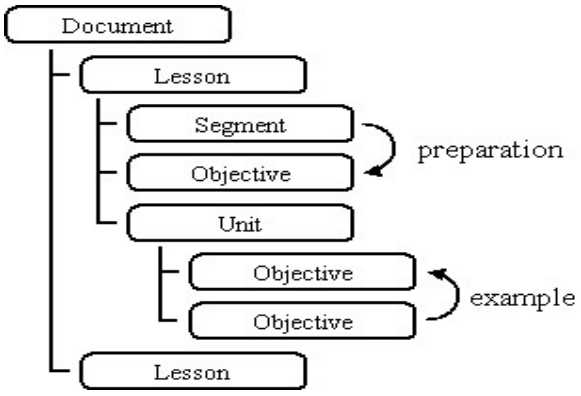

Figure 2. Examples of subsections with semantic relations

Document objects (lessons, units, objectives, and segments) are semantically linked together to create a meaningful document. Figure 2 is an example of document objects and links that can be interpreted as there is a segment that is the PREPARATION of its following objective, and there is a unit that is an EXAMPLE of the preceding segment. Links can be made between arbitrary document objects (this is useful, for example, for the linking the definition of a technical term, or the restatement of a prior document object.) The knowledge representation for link information is stored in a separate link context.

The master document is represented in a dialect of XML. For example, Figure 3 shows the pedagogy for presenting Objective O25, which consists of presenting Segment135 (the TITLE of Segment136), Segment136 (the NUCLEUS of this objective), Segment137 (a PROCEDURE for Segment136), Segment138 (an ELABORATION of Segment137), and Segment139 (a FIGURE for Segment138). After the presentation plan is performed, the necessary text or graphic in each segment will be combined to create a presentation.

3.2.2. The XML representation The XML example in Figure 4 shows the pedagogy of a sample lesson, Lesson L01. In this example, to present Lesson L01 is to present

- Segment133, which is the TITLE of Unit U14,
- Segment134, which is the BACKGROUND of Unit U14, and
- Unit U14, which is composed of Objective O25 and O26, to define blood pressure and hypertension, and
- Objective O27.

```
<Objective O_id = "O25" Title="What is blood pressure">
    <Segment S_id = "Segment135">
<Link rel="Title" from="Segment135" to="Segment136"/>
    </Segment>
    <Segment S_id = "Segment136"
<Link rel="Nucleus" from="Segment136" to="O25"/>
    </Segment>
    <Segment S_id = "Segment137"
<Link rel="Procedure" from="Segment137" to="Segment136"/>
    </Segment>
    <Segment S_id = "Segment138"
<Link rel="Elaborate" from="Segment138" to="Segment137"/>
    </Segment>
    <Segment S_id = "Segment139"
<Link rel="Figure" from="Segment139" to="Segment138"/>
    </Segment>
</Objective>
```

Figure 3. Pedagogy of Objective O25

```
<Lesson L_id = "L01" Title="Introduction to blood pressure">
   <Segment S_id = "Segment133">
       <Link rel="Title" from="Segment133" to="U14"/>
   </Segment>
   <Segment S_id = "Segment134">
       <Link rel="Background" from="Segment134" to = "U14"/>
   </Segment>
   <Unit U_id = "U14"
         Title="Blood pressure and Hypertension definition"
         test_option="Yes">
       <Objective O_id = "O25"
                 Title="What is blood pressure"/>
       <Objective O_id = "O26"
                 Title="What is hypertension"/>
   </Unit>
   <Objective O_id = "O27"
              Title="How can high blood pressure hurt you"/>
</Lesson>
```

Figure 4. Pedagogy of Lesson L01

To present Unit U14, is to present Objective O25 and Objective O26 sequentially. There is a test for the users after Unit U14 indicated by the parameter `test_option="Yes"`. Only after a successful test can the users proceed to Objective O27 to finish Lesson L01.

Figure 3 shows that to present Objective O25 is to present

- Segment135, which is the TITLE of Segment136,
- Segment136, which is the NUCLEUS of this objective,
- Segment137, which is the PROCEDURE for Segment136,
- Segment138, which is the ELABORATION of Segment137, and
- Segment139, which is the PICTURE for Segment138.

Figures 4 and 3 comprise the pedagogy for presenting Lesson L01. This pedagogy is the core content of Lesson L01. The actual content presented to a user is further constrained by the document independent knowledge, which will include or omit portions of this basic pedagogy.

3.2.3. An Example In the authored unit shown in Figure 5, there are two categories of markup shown, document structure and semantic relations (based on [7]). The document structure is represented by groupings of text at the paragraph level. The semantic relations are represented by dashed arrows with relation names. This markup requires minimal expertise and effort for converting existing documents.

The Unit shown in Figure 5 consists of a title and some background segments (`S133` and `S134`, respectively) and two objectives (`O25` and `O26`). Not shown in Figure 5 is the test option (which requires a test after this unit is presented).

3.3. Document Independent Knowledge

The Document Independent Knowledge is composed of a) the generic rules for adapting presentations, b) the knowledge about user models, and c) the rules for updating the user model. This knowledge is predefined, extensible, and domain-independent.

3.3.1. Rules for Adapting Presentation This knowledge is represented using STRIPS-style [5] rules which have preconditions, postconditions, acts, and effects for achieving goals. All this information is represented, in SNePS, as a propositional semantic network, that is, as a graph composed of nodes and labeled arcs, where the propositions are represented by the nodes. For clarity, the example rule in Figure 6 is shown using a simple STRIPS-style format[5] (rather than its SNePS representation). So, the rule in Figure 6 is interpreted as: the *act* of presenting a segment in a scope has two *preconditions*. If the preconditions are satisfied, the *plan* is to do the conditional act given by the plan.

Generic rules for adapting presentation work in concert with the user model knowledge to decide whether a document constituent will be included in the presentation. They modify the default action of presenting a document object by suppressing it, presenting it, or augmenting it with other

```
act:      Present(?segment, ?scope)
preconds: Link(?relation,
               ?segment,
               ?segment2)
          In_Scope(?segment2, ?scope)
plan:   IF ( Suppress(?relation)
           do_nothing()
         ELSE IF( Express(?relation) )
           Display(?segment)
           Check_other_segment(?segment,
                               ?scope)
```

Figure 6. Rule for presenting a segment

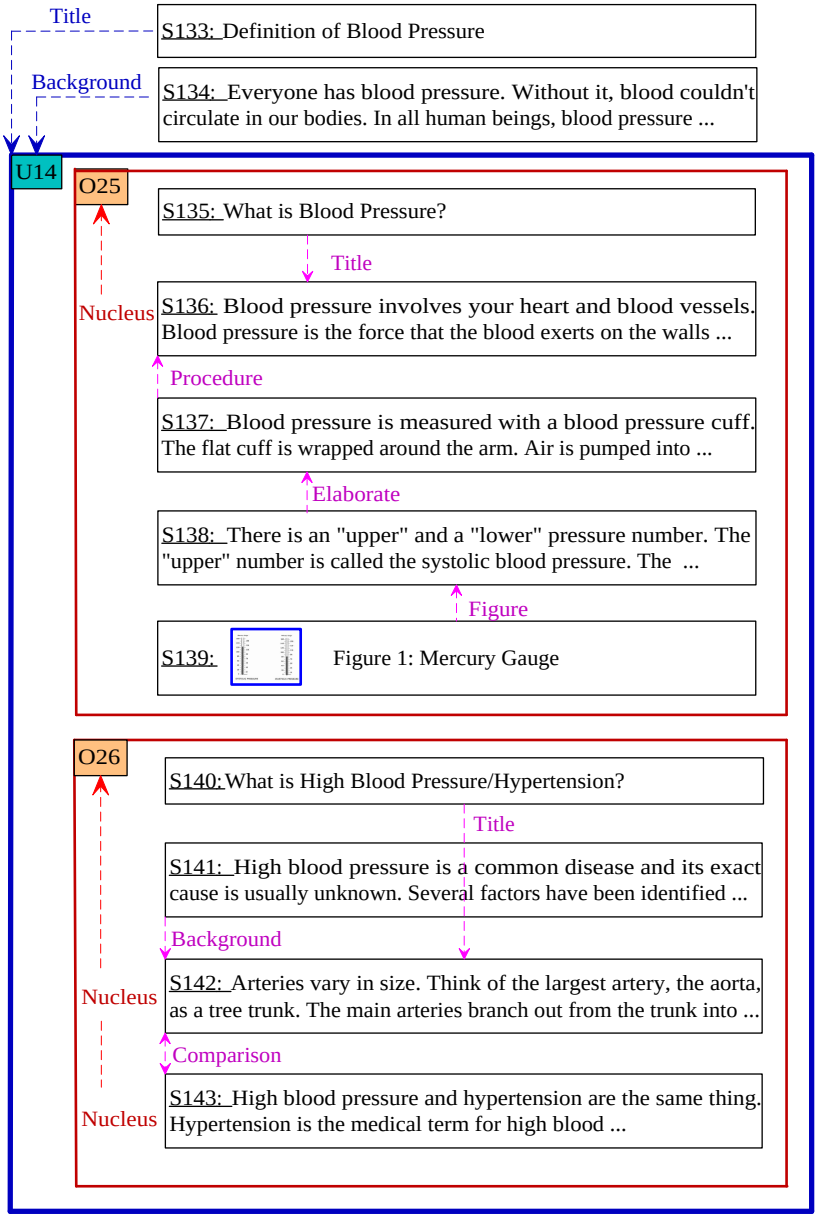

Figure 5. Unit from Blood Pressure Manual Master Document

document objects. Figure 6 is an example of a generic rule that adapts the presentation of the pedagogy. This rule will express or omit a portion of the pedagogy of a document constituent based on the user model. This rule checks with the user model to see whether the type of relation associated with that portion of the document should be included in the presentation. If the relation is to be suppressed, then that document object is suppressed. If the relation is to be expressed, then it will display this portion of the document and check if further augmentations are necessary.

For example, suppose an objective to be presented has TITLE, NUCLEUS, PROCEDURE, ELABORATE, and FIGURE segments in it. The Present rule in Figure 6 will be applied to every segment. The PROCEDURE segment is processed by the rule in Figure 6. It extracts the relations that emanate from the segment and verifies that the segment (?segment2) that the main segment modifies is in the current context (scope). After these two preconditions are satisfied, the procedure segment is processed. If the current user level is expert, the user will not see this segment. If the current user level is novice or intermediate 3, the user will see this segment (refer to Table 1 to see which user levels see which

Type of link	Novice	Intermediate			Expert
		level1	level2	level3	
background	yes	yes			
comparison	yes	yes	yes	yes	yes
concession	yes				
elaborate	yes				
example	yes	yes	yes		
figure	yes	yes	yes		
possible-result	yes	yes	yes	yes	yes
procedure	yes			yes	
real-result	yes	yes	yes	yes	yes
restatement	yes	yes		yes	
solution	yes				
summary	yes	yes	yes	yes	yes
title	yes	yes	yes	yes	yes
warning	yes	yes	yes	yes	yes

Table 1. The relationship of the user levels and the link types

types of links). As a result, an expert will see only the Title and Nucleus segment, while the novice user will see all segments. This knowledge is pre-defined by the document author and loaded upfront.

3.3.2. User Model Knowledge At the beginning of an interaction, a user must self-classify as either novice, intermediate, or expert. This classification can change based on the user's experience with the system, and includes finer grained distinctions than this basic three-level model. The user model is a set of semantic relation types that will be suppressed or expressed for a particular user. Table1 shows the default set of choices for selecting content. For example, we might decide that a novice should see every type of relation, so that all segments will be presented, whereas an expert should see only the title and nucleus of each document object.

While this user model is simplistic, more complex models are possible. We are investigating skills-based models, for example. However, the goal of minimal authoring of the master document constrains the complexity of the user model.

3.3.3. Rules for Updating the User Model In addition to rules that adapt the presentation, we have rules that change the classification of the user, upgrading or downgrading her knowledge level on the basis of the feedback provided to the system as the result of tests. Document objects can specify that testing should occur after presentation (there can be a test field in the XML tag). Currently, there are two types of tests: one test simply asks the user if they understand or do not understand the, just presented, portion of the document; the other performs an automatically generated Cloze test [9, 18]) of the, just presented, portion of the document (a Cloze test is a test of reading comprehension that asks the user to fill in missing words from the, just presented, text.) Figure 7 is a testing rule that upgrades users' classification if they understand. If the user indicates a lack of understanding, the rules will temporarily downgrade the user level, the current document object is re-presented and another testing rule that can permanently downgrade them is invoked.

We can illustrate the operation of these rules by considering the presentation (and re-presentation) of unit `U14` (as shown in Figure 5). This unit includes a request for feedback after presentation (indicated by a field in the XML tag). If the user is an expert, she would see, in order, `S133`, `S135`, `S136`, `S140`, `S142`, and `S143` with the other segments being suppressed. Testing feedback is requested after presentation (invoking the rule in Figure 7); not understanding can result in the user being (temporarily) re-classified as an intermediate and she would then see, in order, `S133`, `S135`, `S136`, `S137`, `S140`, `S142`, and `S143` (other adaptations are possible—based on the level of intermediate.) A novice would see all the segments `S133`-`S140`.

3.4. Adaptive Presentation of an Authored Text

When the document is input by Equuleus, the XML-based representation of the authored document is parsed into planning rules in the knowledge base. For example, unit `U14` in Figure 5 would map into a rule whose plan for presenting `U14` is to present the unit title and preparation (segments `S133` and `S134`), followed by objectives `O25` and `O26` (the latter objectives also have plans). Presentation of segments just outputs their associated text or graphics; Lessons, Units, and Objectives involve further planning. These document-specific rules are a minimal plan for presentation of each document constituent.

Most simply, the entire document can be presented by

```
act:       TestUser(Primary)
preconds:  Created(?user)
plan:      GetFeedback(?user)
           IF ( Understand(?user) )
              MoveUserUp(?user)
           ELSE IF( NotUnderstand(?user) )
              MoveUserDown(?user)
              Re_Present()
              MoveUserUp(?user)
              TestUser(Secondary)
```

Figure 7. A Testing Rule

doing a depth-first traversal of the document. This satisfies our goal of always being able to produce reasonable output. To achieve a more targeted presentation, Equuleus also includes a pre-authored (but extensible) set of document-independent rules for adapting the presentation by expressing or suppressing content based on a user model and rules for updating the user model. These rules make use of the semantic links that the document author has provided. For example, in our implementation, the user model includes a set of stereotype classifications along with the semantic relation types that will be suppressed or expressed for each class.

Adapting the presentation of the document-specific rules to take into account the user model is accomplished by generic rules that change the effect of presenting a document constituent. As previously mentioned, Figure 6 is a rule that suppresses the presentation of a segment if it has a relation (e. g., elaboration) to another constituent in the current scope (based on the user model). If the relation is to be expressed, then present this segment and check if further elaborations are necessary.

4. Summary

Summarizing, the authored document is mapped into rules in the knowledge base that reflect the document structure. The effect of these rules is mediated by the rules for content expression or suppression based on the user model. The user model is updated based on feedback provided by the user. All this reasoning, planning, and presentation occurs in a uniform, declarative, partitioned knowledge base. Our approach requires minimal authoring expertise to convert legacy training documents into dynamic, interactive, presentations. Currently, we have implemented a complete prototype for a portion of a legacy document.

References

[1] E. André and T. Rist. Coping with temporal constraints in multimedia presentation planning. In *Proceedings of AAAI-96*, pages 142–147, 1996.

[2] B. G. Buchanan, J. D. Moore, D. E. Forsythe, G. Carenini, S. Ohlsson, and G. Banks. An intelligent interactive system for delivering individualized information to patients. *Artificial Intelligence in Medicine*, 7:117–154, 1995.

[3] S. Channarukul, S. McRoy, and S. Ali. YAG: A Template-Based Text Realization System for Dialog. *Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, 9(6):649–659, 2001.

[4] C. DiMarco, G. Hirst, and E. H. Hovy. Generation by selection and repair as a method for adapting text for the individual reader. In *Proceedings of the Workshop on Flexible Hypertext, 8th ACM International Hypertext Conference*, Southampton, 1997.

[5] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[6] W. Mann, C. Matthiessen, and S. Thompson. Rhetorical structure theory and text analysis. Technical Report ISIIRR-89-242, USC/ISI, 1989.

[7] W. W. Mann and S. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. text 8 (3):243-281, 1988.

[8] J. Martins and S. C. Shapiro. A model for belief revision. *Artificial Intelligence*, 35(1):25–79, 1988.

[9] M. McKenna and R. Robinson. *An Introduction to the Cloze Procedure: An Annotated Bibliography*. International Reading Association, 1980.

[10] S. McRoy. Achieving Robust Human-Computer Communication. *International Journal of Human-Computer Studies*, 48(5):681–704, 1998.

[11] S. McRoy, S. Haller, and S. S. Ali. Uniform Knowledge Representation for NLP in the B2 System. *Natural Language Engineering*, 3(2):123–145, 1997.

[12] S. W. McRoy, S. S. Ali, A. Restificar, and S. Channarukul. Building intelligent dialog systems. *intelligence: New Visions of AI in Practice*, 10(1):14–23, 1999.

[13] S. W. McRoy and S. Channarukul. Creating Natural Language Output for Real-Time Applications. *intelligence: New Visions of AI in Practice*, 12(2):20–34, Summer 2001.

[14] S. W. McRoy, S. Channarukul, and S. S. Ali. An Augmented Template-Based Approach to Text Realization. *Natural Language Engineering*, 2003. To Appear.

[15] S. W. McRoy, S. M. Haller, and S. S. Ali. Mixed-Depth Representations for Dialog Processing. In *Proceedings of Cognitive Science '98*, pages 687–692. Lawrence Erlbaum Associates, 1998.

[16] M. Milosavljevic, A. Tulloch, and R. Dale. Text generation in a dynamic hypertext environment. In *Proceedings of the 19th Australasian Computer Science Conference (ACSC '96)*, pages 417–426, 1996.

[17] M. O'Donnell, C. Mellish, J. Oberlander, and A. Knott. ILEX: An architecture for a dynamic hypertext generation system. *Journal of Natural Language Engineering*, 2003. (to appear).

[18] J. Oller and J. Jonz, editors. *Cloze and Coherence*. Associated University Presses, 1994.

[19] E. Reiter, C. Mellish, and J. Levine. Automatic generation of on-line documentation in the IDAS project. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 64 – 71, 1996.

[20] S. C. Shapiro and W. J. Rapaport. The SNePS family. *Computers & Mathematics with Applications*, 23(2–5), 1992.

[21] A. Stent, M. Walker, S. Whittaker, and P. Maloor. User-tailored generation for spoken dialogue: An experiment. In *Proceedings of ICSLP*, 2002.