"Vasistha" - An Instruction Delivery Framework for Online Learning

Philip S Tellis, Prem Sreenivasan Narayan, Suresh Dhamapurkar,
Sasikumar M, SP Mudur
National Centre for Software Technology, Juhu, Mumbai, India
{philip, prem, sureshd, sasi, mudur}@konark.ncst.ernet.in

Abstract

The Internet and the World Wide Web are becoming popular as media for delivering instruction, resulting in the concept of asynchronous teaching and learning. In order to make this model of instruction successful, one must consider all aspects of instruction delivery, to ensure that students find this new medium, comfortable and effective. Although there are a plethora of instruction delivery tools available worldwide, few of them have put in significant effort to look at the real issues pertaining to instruction delivery. In this paper, we discuss "Vasistha", an Instruction Delivery Framework for online learning, which we are developing at NCST. Vasistha provides a framework to organise course content for online courses. It has been designed keeping in mind the core issues pertinent to Instruction Delivery.

1 Background

The online learning environment has significant similarities and differences when compared to more traditional environments such as conventional classrooms or laboratories. We have to adjust our teaching methods, material and strategies to fit into this environment. The methods and strategies utilised in the classroom are very significant since they convey content and are considered the heart of the educational format. Of the many known instruction strategies, most have not been developed specifically for online instruction, but are based on strategies currently employed in traditional classrooms and can be successfully adapted for facilitating online learning [1]. However, "effective use of the computer as a resource in education necessitates changes in pedagogy" [2].

A very, perhaps the most, important component in any learning environment is the learner. The classroom environment, peer pressure, interaction, eye contact with the teacher, etc. are important elements of the traditional instruction model. Absence of these elements coupled with the constraints of a computer-based environment - such as limited screen area, and communication/interaction using mouse/keyboard-- makes an online student very different from a traditional classroom student. In face-to-face lectures, the teacher has the ability to motivate students to learn. He (For the remainder of this paper, the pronoun "he" will be used to mean "he/she".) plays a significant role in the entire learning process. The teacher employs various mechanisms to sustain the students' attention. Simulating this kind of an environment in the online learning model is a tough task. There are many learning management systems (LMS) available worldwide and some of them are quite popular also viz., WebCT, Blackboard, TopClass. None of them have addressed these issues in formulating instruction delivery models. Some of the major issues related to instruction delivery that need to be looked at, in the context of online learning, are described below.

1. **Flexibility:** The concept of asynchronous teaching and learning provides flexibility in terms of time and space. However, it is also essential to ensure that lessons, assignments and tests are taken in a meaningful order to enable students to build on already acquired skills. In an unrestricted and heavily hyper-linked world such as that of the web, it is easy for a student to get lost. At the same time, it is neither desirable nor required to force a strictly sequential order through the material for a course.
2. **Sustaining the student's interest:** Course material should be designed to motivate students to learn and hence should be engaging. This is very important because a student's attention span is shorter in the online mode of learning compared to face-to-face learning. Unless he is self disciplined and self-motivated, it is difficult to keep him attentive all the time. There are two orthogonal strategies for this. One is the presentation of the course material, for example, use of engaging and real-life examples, good visuals, etc. The second is good organisation of the content so that the student is aware of where

he is, his progress, his goals, etc at every point in the learning process.

3. **Assessment mechanism:** Remote administration and supervision of tests to gauge the depth of understanding of a particular topic is a Herculean task. Of equal concern is the fact that current online testing systems support only multiple-choice, true-false and fill-in-the-blank type of questions. Mechanisms to handle subjective type of questions or incorporate practical skills are virtually non-existent.
4. **Adaptive instruction delivery:** The online learning model has the capability to interact with each student individually and hence can measure and record the strengths and weaknesses of each student separately. It is a natural next step to visualise a course tailor-made to suit an individual's needs and abilities. A course should be capable of adapting its contents according to the level of understanding of the user.
5. **Audience diversity:** Because two learners participating in the same class may have different learning experiences [3], online educators must organise situations that address the various facets of learning in order to provide significant experiences for each class participant.

As mentioned earlier, effective organisation of course content is an important aspect of instruction delivery. Improper structuring of the content could confuse the learner. It could lead to disorientation and cognitive overload**. Disorientation** refers to the users not knowing where they are, where they have been, or how to get to where they want to go in hypermedia-space. **Cognitive overload** refers to users being overwhelmed by the options available to them in multi-path, multi-tool environments such as hypermedia documents. On the other hand, effective content sequencing may improve the learner's understanding. Hence, we discuss the various content sequencing strategies in the next section.

1.1 Content sequencing strategies

Content sequencing is the efficient ordering of content in such a way so as to facilitate the learning process. There are several general methods of sequencing content. One well-known method is the pre-requisite method [4], which is based on a learning hierarchy that identifies skills dependent on other skills. The basic idea is to teach the pre-requisite skills first. A second approach described by Posner and Strike [5], is a set of strategies for sequencing the instruction based on learning-related, world-related and concept-related content. The learning-related scheme suggests ways of sequencing the content based on learner characteristics identified during learner analysis. World-related and concept-related strategies recommend sequencing schemes based on the type of content treated in a unit. For example, world-related scheme suggests sequencing based on spatial, temporal and physical relationships identified in the content. Similarly, the concept related scheme suggests sequencing based on the relationships between concepts [6]. The following sections describe the notion of a concept and also enumerate some of the various kinds of relationships among concepts.

1.2 Notion of a concept

A definition of "concept" based on Bloom's model for Mastery Learning [7] is as follows: "A concept is a semantically meaningful unit of instruction with a specific intent type and a matching instruction and evaluation type." A concept must have clear learning objectives. This will make sure that the concepts are of a reasonable breadth and depth.

1.3 Relationships among concepts

Some of the ways in which concepts could be inter-linked in a domain model are as follows: [8]

1. Pre-requisite – Knowledge of concept A is essential to understand concept B.
2. Sibling – Concepts B and C are sub-concepts of a parent concept with identical pre-requisites.
3. Has-a – Concept B is a sub-concept of concept A. In a concept hierarchy, concept B is considered to be a child of concept A.
4. Is-a – Concept B is a specialization of concept A.
5. Co-requisite – Concept B should be learned if concept A is to be learned completely.

2 Instruction Delivery in Vasistha

In Vasistha, the initial content sequencing is based on a concept-related strategy and then on pre-requisites based on a learning-related scheme. The teacher identifies the various concepts, which constitute the course content. Vasistha provides the instructor with an interface to structure the course content thereby building a map of the course. This course map provides a systematic method for presenting the content that is likely to match the learner's expectations. Vasistha doesn't allow a student to access a particular concept unless he has demonstrated that he has cleared its pre-requisites by passing a test or by satisfying other criteria as laid down by the course designer.

A course designer is allowed to design the course in the form of topics, sub-topics, sub-subtopics, etc., analogous to chapters, sections and subsections and so on in a book.

Topics are similar to what are known as Learning Objects [9] in other tools and as such, each topic is stored internally as a separate object. In the remainder of this section, the terms 'topic' and 'object' will be used synonymously. The terms 'course' and 'module' will be used to mean a complete course - the aggregation of all related topics.

2.1 Managing Content

The course designer associates content with a topic, but stores the content separately. It may be modified independently of the rest of the object. In the current implementation, each topic's content may be stored as any web accessible content, with its URL being the associative key. Vasistha currently makes no attempt to understand the format of the content, or to standardise it in any way. Manipulating the content is the responsibility of the content developer. Future versions of Vasistha may address this issue in depth.

2.2 Relationships between Objects

Vasistha allows the course designer to create relationships between topics. These relationships decide how the entire course is organised, and in what order different topics may be covered. Vasistha provides two orthogonal relationships between objects. These could be defined as belongs-to and depends-on relationships. These relationships are similar to the has-a and pre-requisite relationships discussed earlier. No two topics may be related by more than one relationship.

2.3 Belongs-to relationships

A belongs-to relationship is similar to the book example mentioned earlier. Just like several sections belong to a chapter, and several chapters belong to a book, Vasistha allows a topic hierarchy to be built wherein a single topic may be broken up into several sub-topics, and each of those could be further decomposed into smaller sections. A belongs-to relationship is specified through a topic code attached to each topic. Similar to the sections in a book, topic codes follow a dotted notation. E.g.: Topic 1, Sub-topic 3, Section 2 would have topic code 1.3.2, and in the hierarchy, topic code 1 would contain a node with topic code 1.3, which would, in turn, contain a node with topic code 1.3.2. In other words, 1.3.2 belongs to 1.3, and 1.3 belongs to 1. Henceforth, any topic that belongs to another topic will be known as that topic's child, and the topic that the child belongs to will be known as the parent. So, in the example stated above, 1 is the parent of 1.3, which is the parent of 1.3.2, and 1.3.2 is a child of 1.3, which in turn is a child of 1. A parent may have multiple children, and through the use of aliases (covered in detail in section 2.6), a child may have multiple parents.

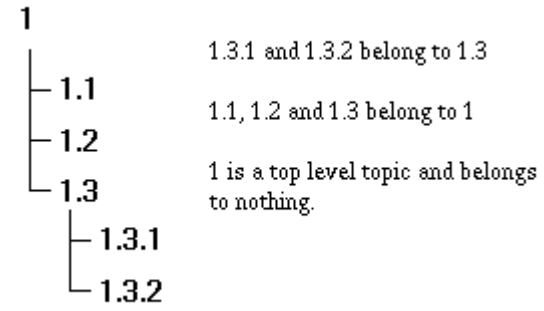

Figure 1. Belongs-to relationship

The course designer specifies belongs-to relationships while adding topics by specifying proper topic codes. On the face of it, this would pre-empt the use of any topic in more than one location; however, Vasistha provides a system of aliases, whereby a single object may be assigned multiple topic codes, thereby putting it into multiple hierarchies.

It goes without saying that belongs-to relationships are transitive.

2.4 Depends-on relationships

Vasistha allows dependencies to be specified for each topic. A dependency is defined as a topic that must be completed before the dependent topic can be attempted. In Vasistha, the course designer explicitly creates depends-on relationships by specifying a list of pre-requisites for every topic that is added to the course. This list may be empty.

Depends-on relationships are also transitive.

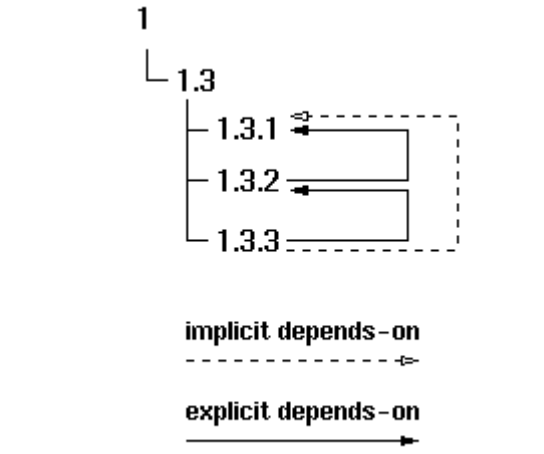

Figure 2. Depends-on relationships are transitive

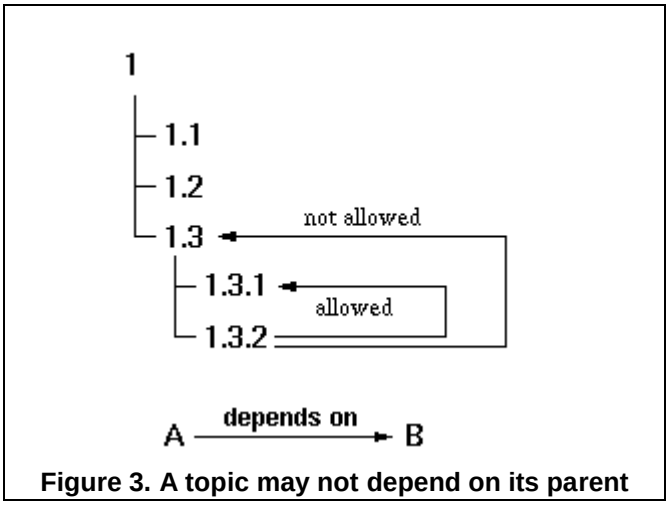

Figure 3. A topic may not depend on its parent

No topic may be dependent on any of its parents or on any topic that is its direct ancestor. It may however be dependent on its siblings or the siblings of its parents.

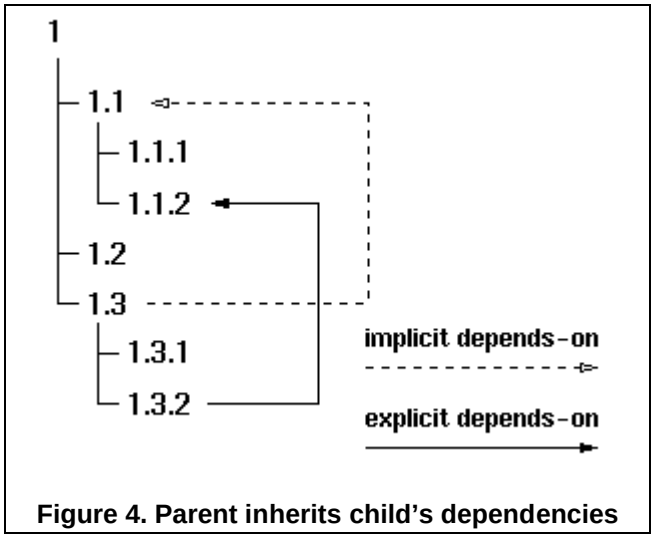

Figure 4. Parent inherits child's dependencies

The presence of a depends-on relationship between a child of a topic and a topic not in the same hierarchy automatically creates a depends-on relationship between all parents of the dependent topic up to but not including the level where they share the same ancestry. For example, if topic 1.4.5.2 is dependent on topic 1.4.3.1, then topic 1.4.5 is implicitly dependent on 1.4.3.1, but 1.4 is not, because at that level, 1.4.5.2 and 1.4.3.1 share the same ancestry - 1.4.

Conversely, all children inherit the dependencies of their parents.

Topic hierarchies created with belongs-to relationships inherit certain restrictions regarding the order in which topics are considered complete.

A parent topic can be read, but not tested until all of its children have been completed. In other words, a parent is not considered complete until all requirements for its completion are satisfied and all of its children are already complete.

2.5 Restrictions on creating dependencies

Two topics may not be cyclically dependent. This includes dependencies through transitivity.

A topic may not be dependent on its parents or its children. This may seem a bit restrictive at first, but when one considers the implicit dependencies created by belongs-to relationships, it becomes clear that they are troublesome.

2.6 Aliases

A topic may be placed in multiple hierarchies through the use of aliases. Although every topic may have one and only one primary topic code, it may have more than one alias. An alias is nothing but another topic code that the topic may be known by. The content and all other associated data is attached to the topic only once, but the topic itself may be placed in different positions within the course hierarchy. For example, a topic with code 1.5.2 may have aliases 1.6.4.2 and 1.6.5.6. In these cases, the same topic will appear at all of the above-mentioned places.

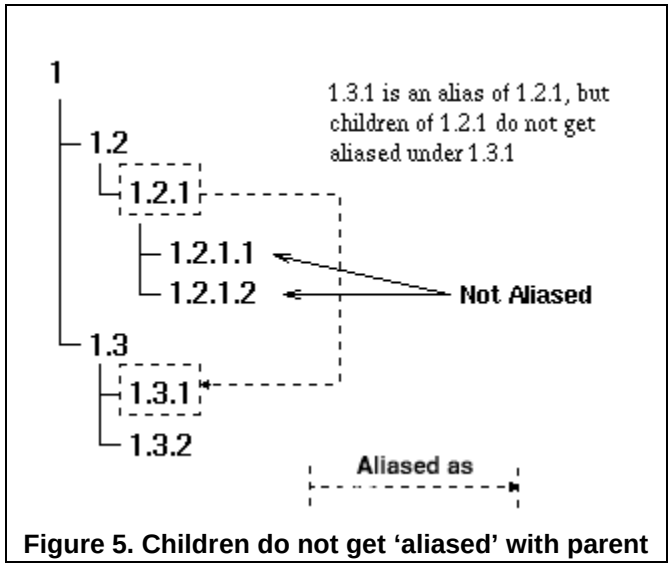

Figure 5. Children do not get 'aliased' with parent

When 'aliased', a topic retains its name, URL, dependencies and tests, but nothing else. As a result, when a topic is put into a different hierarchy through an alias, that topic, and only the topic are mirrored. None of its children move with it. The topic may have a new set of children at the new location.

This limitation makes it possible for a topic to be placed in its own ancestry, thereby creating a cyclic

belongs-to relationship. This usage is highly discouraged, and will be prohibited in future versions. The movement of all children along with a topic is being considered for a future version of Vasistha.

2.7 Completing a topic

While adding topics to a course, the course designer may state whether a topic has a test associated with it or not. If a test has been associated with the topic, then successful completion of the test is a necessary condition for completion of the topic. Since the test cannot be taken until all other requirements have been met, it is also a sufficient condition.

When no test has been associated with a topic, simply visiting the topic material is considered as the equivalent of the test on the node. Since topics may have other requirements (like the completion of child topics), this alone does not mark the topic as complete. In this situation, completion of all other requirements, which currently includes only completion of children, suffices for completion of topic.

2.8 Navigation

As mentioned earlier, good navigational support is necessary for the student to gauge his progress with respect to the requirements of the course. This would include a visual indication of where he is currently in the concept map of the course, an ability to visit other parts of the course, and non-obtrusive indications of what he can do next.

Vasistha's topic coding convention and the nested (indented) listing of the various topics provide an indication of the overall structure of the concept map relative to where he is. As the student expands the various topic nodes and completes the lessons, the display adjusts itself to show the overall topic-list and expanded view of his immediate whereabouts.

In addition, the use of a colour-coding scheme makes it easy for the student to visually identify the concepts that he has completed and the concepts that he is eligible/not eligible to take. This colour-coding scheme is based on the intuitive "**traffic light**" model. The colours used and their significance are as listed below:

1. Red: Indicates that the student is not eligible to take that particular concept.
2. Green: Indicates that the student is eligible to take that particular concept.
3. Blue: Indicates that the student has successfully completed that particular concept.

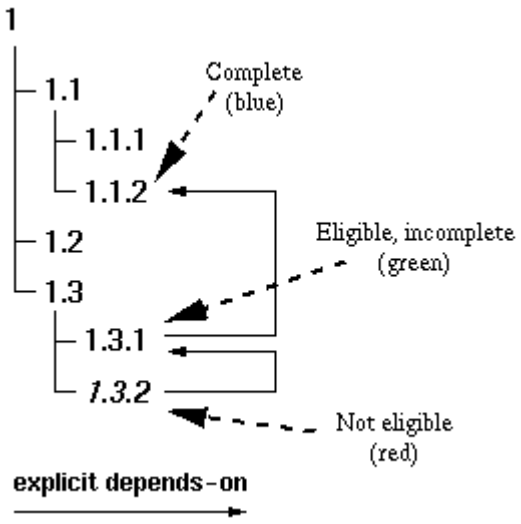

Figure 6. Traffic light approach

2.9 Features

Apart from being remotely accessible, an online course must also provide distance learners with the ability to participate in the course from varied locations. The system must be able to remember a course participant on a return visit, and must be able to provide him with a history of tasks already performed. Vasistha provides the student with three memory aids.

1. **Continuity of Sessions:** The most basic form is 'last topic remembrance'. Vasistha remembers a student on return visits and takes the student back to the last position where he left off. This facilitates a student to proceed with the course over several sessions, as is the case with a regular classroom course, instead of trying to complete it in one straight session.

   'Last topic memory' provides continuity to the course over a period of time without the student getting lost searching for what he was doing last. As last topic memory is automatic, it is kept completely transparent to the student without the need for his intervention at any point.

2. **Bookmarks**: Vasistha also provides the student with the ability to bookmark topics within a course. Unlike last topic memory, a student must explicitly state which topics he wishes to bookmark. He may also specify a short one-line description for the topic that he creates a bookmark for. The default description is the name of the topic. A student may jump to a bookmark at any time during the course, provided he is

allowed to do so based on other dependency rules. Since a bookmark may only be placed on a visited topic, it can be assumed that all bookmarks point to topics that the student is eligible to visit.

Only one bookmark may be assigned to each topic. Assigning a bookmark to a topic will overwrite any bookmarks previously set for that same topic. The only effect that this has is that of changing the description on the bookmark.

3. **Notes**: Since only one bookmark is allowed for each topic, and only a single line description can be added, this is often insufficient for a student to keep track of his thoughts while going through a topic. A note-keeping feature has been provided for this purpose. Just like a student would carry a notebook to a class and jot down points of interest, or things that he may want to look at later, Vasistha provides students with a notepad like interface for adding notes to a given topic.

   Any number of notes may be added to a topic, and all notes for a topic are listed sequentially when requested. Currently notes can only be added; the interface to delete and modify them is being developed.

Using the above-mentioned memory features, a student can move around the course in any order that he wishes, considering all pre-requisites along the way. A student may move back and forth between sections, in essence, creating his own hyperlinks in the content. These are private to the student and are not reflected in other students' profiles.

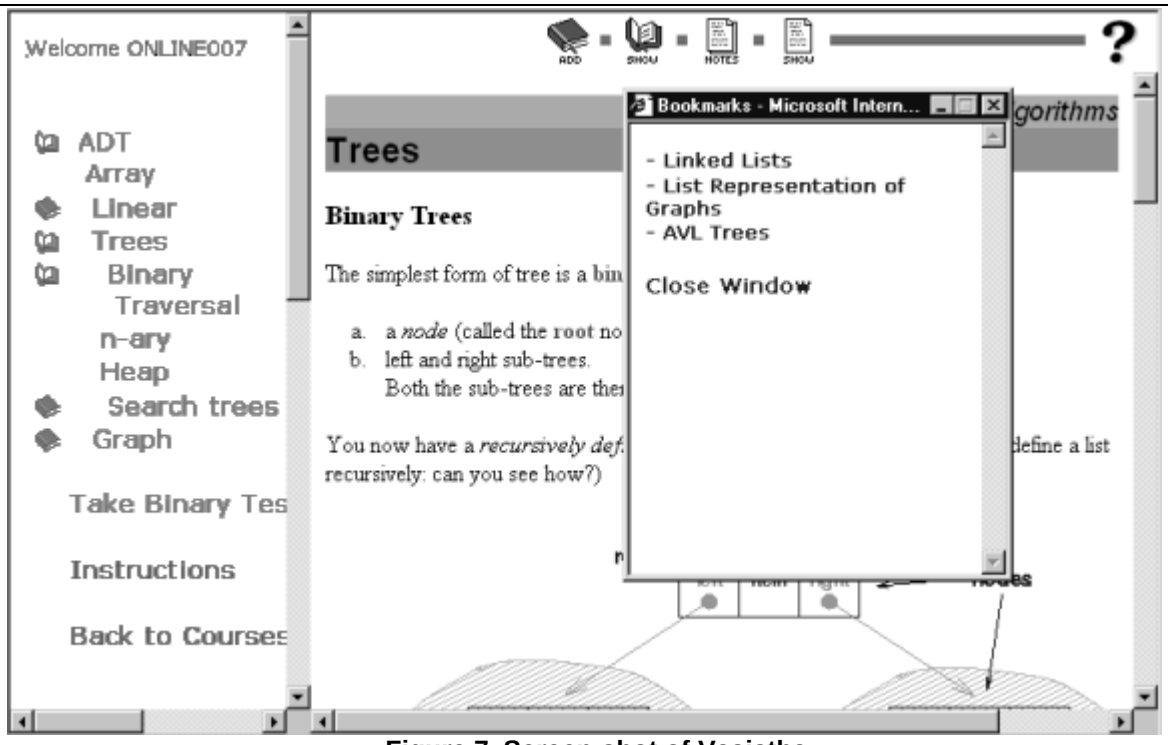

Figure 7. Screen-shot of Vasistha

3 Control/Data flow through Vasistha

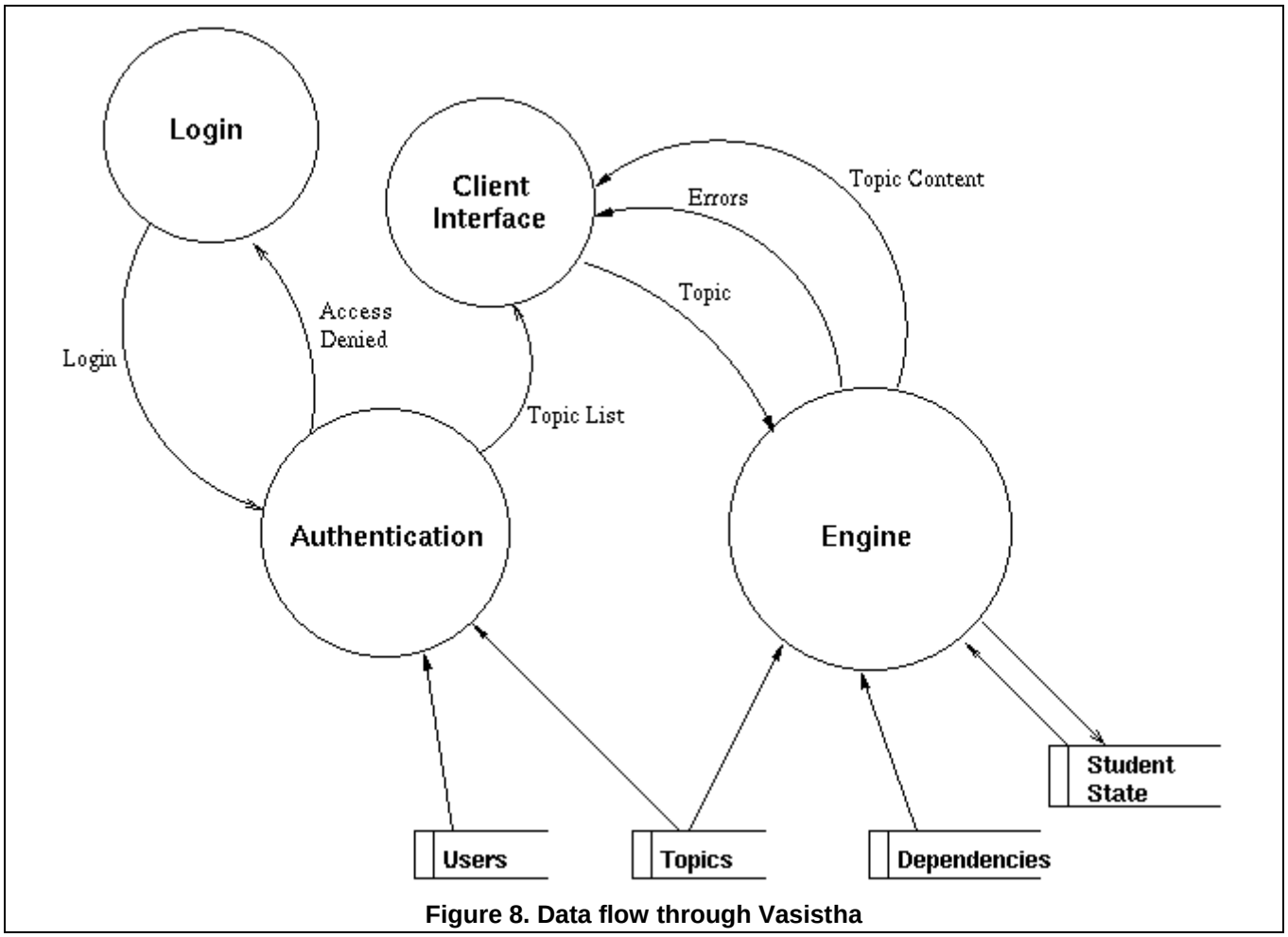

Figure 8. Data flow through Vasistha

Data and control flow through the system occurs as described in the figure above. When a user logs into the system, control is passed from the client interface to the authentication system. Users are authenticated against a user file, updateable only by the administrator of the system. An invalid user is rejected at this point.

If the user is successfully authenticated, the authentication system determines all the modules that he has been registered for, and provides him with this list. The user may choose any one of these modules, and is then presented with the contents of that module. The instruction delivery engine does this processing.

The student then interacts completely with the engine, selecting topics and receiving the topic content. If the student is not eligible to read a topic, the engine displays an error message stating the pre-requisites that need to be completed first.

When a student is sufficiently confident in a topic, he may take a test for that topic – if a test has been specified. Although Vasistha does not carry out testing and evaluation, it provides an abstract interface for testing. This enables any testing system to be plugged in to Vasistha with ease. Currently, Vasistha has an interface to the 'Veda' [10] testing system.

4 Future Work

Work is in progress to depict the course map in a visually intuitive manner. This will enable the student to easily identify his exact location with respect to the course.

Provision of a template for the instructional designer to enable him/her to create course content with the minimum possible effort is also being worked on.

5 Summary

We outlined some of the core issues related to instruction delivery and described an instruction delivery framework for online learning. The framework discussed enables the instruction designer to organise course content by specifying belongs-to and depends-on relationships. The traffic light approach is used to visually indicate the eligibility of a student to take a course. The learning experience is further enhanced through the 'bookmark' and 'notes' features within the framework itself.

6 References

[1] Tina Joy Pitt, Anne Clark, "Creating Powerful Online Courses Using Multiple Instructional Strategies", http://waltoncollege.uark.edu/disted/creating_powerful_online_courses.htm, 2001.

[2] Knupfer N. N. & R. Muffoletto, Teachers and Educational Computing: Changing Roles and Changing Pedagogy, in Robert Muffoletto and Nancy Nelson Knupfer (Eds.), Computers in Education: Social, Political, and Historical Perspectives, Hampton Press, Cresskill, NY, 1993, p. 173.

[3] Tyler R. W., Basic principles of curriculum and instruction, The University of Chicago Press, Chicago, 1949.

[4] Gagne R. M., Conditions of learning and theory of instruction (4th ed.), Holt, Rinehart & Winston, New York, 1985.

[5] Posner G. J., & Strike K. A., A categorization scheme for principles of sequencing content, Review of Educational Research, 46, pp. 665-690.

[6] Gary Morrison, Steven Ross and Jerrold Kemp, Designing Effective Instruction, John Wiley and Sons, Inc., 2001, pp.112-116.

[7] Bruce Joyce, Marsha Weil, Models of teaching (5th ed.), Prentice Hall of India, 1997.

[8] Prachi Gharpure et al., A Methodology for developing courseware for online learning, Working paper – Vidyakash Project, National Centre for Software Technology, Mumbai, 2001.

[9] Warren Longmire, Informania, "A Primer on Learning Objects", http://www.learningcircuits.org/mar2000/primer.html, 2000.

[10] National Centre for Software Technology, Veda – An Online Testing & Question Banking System, http://www.ncst.ernet.in/~veda/, 2001.