

WebWisdom 2000— Database Backend for Distance Learning Systems

M. Podgorny, WebWisdom.com

1. Introduction

One of the important factors that are currently limiting wide access to high-quality education is a high cost associated with the teaching process. As the result, we observe growing disparities in access to quality education - a highly undesirable social phenomenon. One of the possible methods of cost containment is reduction of the costs associated with creation of courseware and delivery of courses. These costs can be reduced by introducing better, more efficient methods of preparing courses, with special focus on courseware re-use and team authoring, by increasing the average number of students participating in a class, and by elimination of travel costs. These goals are difficult to achieve in traditional educational environments.

During the last couple of years, we have witnessed a significant progress in computing and networking technologies. Computer systems became able to record, store, efficiently manage, and reliably transmit all kinds of multimedia data. This includes text, graphics, audio, and video. Furthermore, the Internet revolution brought broad social acceptance of computer technologies. All these changes lead to a situation where many traditionally non-computer based processes can be significantly improved by the use of computer and Internet technologies.

Application of advanced computer and network technologies to the education process resulted in development of a new domain known as distance education. Since the very beginning, the Web has been embraced by educators as a new data repository, effectively replacing traditional libraries. Today, at the US colleges and universities, even students of humanities expect a Web site to support almost every class. E-mail is quickly replacing traditional visits to professor's office, and discussion groups promote interaction between students. This set of Internet technologies, almost unheard of just five years ago, has become a mandatory part of the services that a college must provide - or risk students' criticism.

This spontaneous process of technology adoption by education is a true revolution. The idea of using computers to improve education is not new at all. There were innumerable programs trying to increase use of computers in teaching process, at all levels - from kindergarten to college. In the past, the results were mixed, at best. Teaching computer skills for their own sake was definitely useful, but a general thesis that use of computers improved overall quality of education in other subjects is almost indefensible. Advent of the Web changed this situation completely. We attribute this to the fact that while earlier attempts were almost always mandated by various "visionaries" with their own agendas, either research or administrative, the technology adoption sparked by the Web was essentially a *grass roots* movement, initiated by active educators, and enthusiastically embraced by the students. This revolution fed to real needs of the community of students and educators alike.

What is the future of this revolution? We observe that the set of technologies already adopted is a relatively simple one, and, in most cases, it is undifferentiated. This means, the educators prevalently use generic technologies, or, in the lingo of technology managers, the COTS (Commercial Of The Shelf) components. This set of technologies, including the most basic fabric of the Web, has proven sufficient to support the basic functionality useful for the community: publishing of the lecture slides, lists of recommended readings, frequent students' questions, or answering students' questions via e-mail. While of high value, this development per se quickly led to explosive growth of the amount of educational material on the Web. This mountain of data shares all common characteristics of other Internet repositories: disorganization, transient nature, incoherency, poor (or non-existent) searchability, lack of metadata, format incompatibility, etc.

In this paper, we argue that it is time now to provide educators community with advanced tools to organize and manage their Web-accessible courseware repositories, and we present general architectural design and implementation of a system that meets critical needs of educators wanting to move from the current infant phase to professional level. We believe that the solution we present is a logical and necessary continuation of the ongoing process of integrating the Web into the fabric of our educational systems.

The system we describe in this paper has not been designed as an academic, computer science research project with lofty goals of little practical relevance, or as an industrial project laid out by programmers with limited knowledge of business process they intend to support. Instead, the project was built as a tool to support educational process being a part of the professional lives of the system creators. Webwisdom project is as much "grass roots" as the technology adoption process we have described. The only difference is that the educators involved in this project happen to be computer science and Internet technology experts and/or professional software developers. It is a system designed, built, and deployed by its end-users. As such, it meets the most important requirement for a successful product: know the business problems of your customers!

1.1. Technical Requirements

Insertion of new technology can potentially improve the entirety of the educational process, starting from preparation of curricula, and ending with gathering and evaluation of test results. The use of properly designed technology may have a significant impact on the reduction of costs associated with the teaching process. The cost reduction is caused by three factors. First, the process of preparation of the educational material is more efficient. It can involve use of specialized content-preparation tools, content reuse, and sharing of content between courses and among teachers. Secondly, large groups of students can simultaneously participate in a course provided by the same teacher at the same time using the same facilities. Finally, the cost of traveling – for both students and teachers – can be reduced or eliminated.

Development of a general system supporting both preparation and delivery of courses is a challenging task. The system should support efficient and reliable storage of hundreds of presentations that are usually produced by colleges and corporate training departments. Convenient management of the educational contents and utilization of commonly used authoring tools (e.g., MS PowerPoint) is a prerequisite. Frequently, a team of instructors must work together on a set of complex, interwoven presentations. They should have an instantaneous access to the work of other team members while preserving all security requirements. The system should support complex presentations that encompass various media types, including digital audio and video, add-ons, references, and glossaries, and keep all the material in a database for easy management, update, and extension. Moreover, separation of the presentation layer from the actual educational material is highly desirable. Various authors and contributors should be able to manage the contents while system administrators can control visual impact independently. Furthermore, such a system should ensure scalable performance and adequate response time that does not depend on the number of concurrent users or courseware material gathered in the database. The system should enable working in the off-line mode when no satisfactory Internet

connection is available. Finally, the system should support widely accepted standards (e.g., IMS, XML).

In this paper, we describe the WebWisdom 2000 system – an integrated solution for creating, managing, and delivering educational material by the use of computer technologies. The WebWisdom 2000 system benefits from both advanced database technologies for storing and managing data, and from Internet technologies for sending the data to end-users.

2. The WebWisdom 2000 System

2.1. Overview

The WebWisdom 2000 system is an integrated set of tools supporting academic and corporate courseware storage, management, and delivery. The system can be used as a deployment engine for any network-based teaching and training system, as well as a database backend for educational portals. Special high-performance features of the system make it particularly useful as a back-end for interactive synchronous and asynchronous distance learning systems.

The WebWisdom system is built on top of a database management system. This solution was dictated by the volume and complexity of data usually involved in the education process. Use of advanced data management techniques available in contemporary database systems like indexing, integrity enforcement, binary object management, and full-text search greatly improves the overall system performance and usability.

Since the modern educational process can benefit from various types of multimedia contents, the WebWisdom system provides support for wide range of data types. These types include text, graphics, audio, and video. The system provides tools for importing collections of multimedia data directly from the file system.

The WebWisdom system enables storing additional meta-data describing the actual educational contents.

Examples are titles, authors, dates, and events associated with the presentations. Availability of these data in the database significantly enhances the system search capabilities.

The system seamlessly integrates with Microsoft PowerPoint. Support for this de-facto industry standard greatly enhances the usability of the system. All existing PowerPoint presentations can be directly imported into the database. PowerPoint can be also used for preparing new contents. Operations on slides, like editing and printing, whenever possible, are performed by the use of PowerPoint in-database. In addition, any PowerPoint-originated presentation, or combination thereof, can be exported to a PowerPoint-compatible file.

Presentations stored in the database are automatically made available on the web by the dynamic web interface and can be accessed by a standard web browser. No explicit exporting is required. The dynamic web interface uses user/password security system to enforce user access privileges.

An important feature of the WebWisdom system is separation of contents and output formatting. Authors can prepare educational material without worrying about the visual design of the prepared contents. The visual formatting can be modified without modification of the actual contents. Moreover, contents prepared by different authoring tools can be presented in a uniform way.

The output generated by the WebWisdom web interface is dynamically created from templates. This approach enables different presentation of the contents depending on the user preferences, privileges, or computer system capabilities. An interesting example of different output formatting is a system using separate templates for a teacher and for students in a synchronous distant teaching environment.

The WebWisdom system allows importing and exporting presentations in XML format, which is an emerging industry standard for data interchange [10]. The XML presentation files can be used for moving or copying contents between different databases and for off-line content delivery.

2.2. Architecture of the WebWisdom System

The architecture of the WebWisdom system is presented in Figure 1.

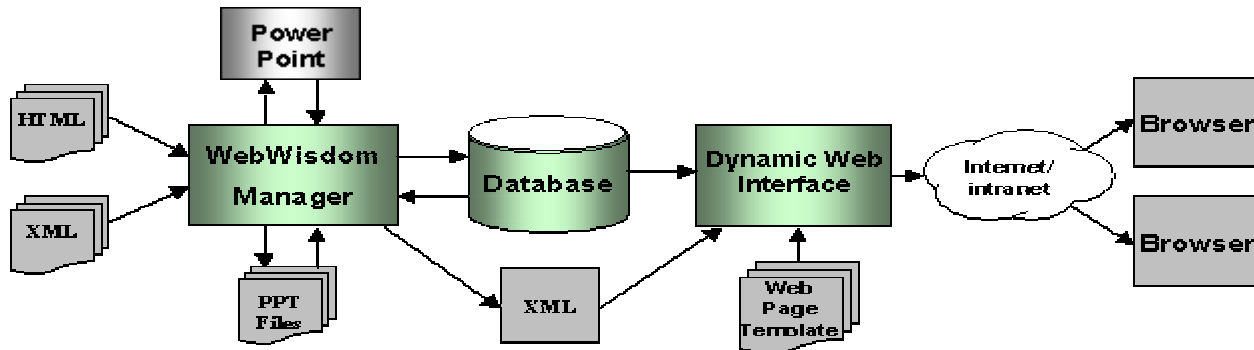


Figure 1. Architecture of the WebWisdom system

The system is composed of three main parts:

- WebWisdom Database,
- WebWisdom Manager,
- Dynamic Web Interface.

The WebWisdom Database is the core of the system. It is responsible for storing all kinds of data used in the educational process. This includes:

- Slides with associated educational objects: images, text, sound, video, external links, etc.,
- Presentations in form of lists of slides and/or other presentations,
- The data describing hierarchical structure of presentations,
- Information about various events like conferences or lectures related to presentations,

- Visual formatting parameters (preferred screen size, colors, font sizes, location, etc.),
- Information about users (name, identifier, password, access privileges, etc.),
- Images used by the dynamic web interface,
- Metadata describing the educational contents.

The WebWisdom Manager is the main contents-preparation tool. It enables management of the database contents, importing data to the database (in form of HTML, PPT, or XML files), and exporting data from the database to the file system. The WebWisdom Manager cooperates with PowerPoint for importing, editing, and printing of PowerPoint slides and presentations.

The Dynamic Web Interface is a set of tools enabling instantaneous access to data stored in the database through the WWW interface. An important feature of the system is independence of information and page formatting. A special set of XML-based templates is used to dynamically create web pages delivered to the client browsers.

2.3. Organization of the Educational Material

The WebWisdom system assumes certain organization of the educational content. The basic building blocks of the educational material are called *educational objects*. Different types of educational objects may be used in the system. This includes: text, graphics, audio, video, and external links. Educational objects may be used in two ways – either to present the same information in different forms, e.g., textual and graphical versions, or to enrich the presentation by introducing some additional elements, e.g., associated audio or external links.

Educational objects are grouped into *slides*. Each slide has also a list of attributes (e.g., title, preferred method of presentation, time of use). A list of slides ordered by their intended sequence of appearance forms a *presentation*. Usually a presentation contains the amount of educational material used during a single lecture.

Presentation has its own list of attributes (e.g., name, title, type, and owner).

Presentations are logically grouped into *educational domains*. The same presentation can be in more than one educational domain at the same time. Educational domains can be nested and form a tree-like structure. The depth of the tree is not limited. Educational domains are the units of user access control. A user can have no access to an educational domain, read-only access, or full access. The structure of the tree may correspond to the structure of the educational institution, e.g., departments, programs, and courses.

In real educational environments, often the same material is used more than once in different contexts. For example, the same slide is used in two presentations, the same lecture is presented in two courses, or the same course belongs to two different programs. To avoid physical copying and to ensure integrity of the data, a complex system of logical links is used. It allows accessing the same educational material from different presentations and educational domains.

3. The WebWisdom Manager

The WebWisdom Manager offers an intuitive in use, integrated environment for developing and managing educational contents. The manager is equipped with a graphical interface providing intuitive method of visualization of the data stored in the underlying database. It smoothly integrates with office applications linking the potential of the PowerPoint with the power of the relational database model. The tools constituting the WebWisdom Manager address all the most important issues in managing educational data.

The user may import a presentation in PowerPoint, HTML, or XML formats, or just as a collection of files. The presentations may later undergo editing of slides, editing of data associated with slides, reorganizing the order in which the slides will be presented, etc. A user may create a new foil within the selected presentation, attach images, sound, or video data, annotate the selected foil,

or import a selected HTML page and store it in the database.

Since for the majority of slides, the main educational objects are images, the system is equipped with a number of tools for editing images. These tools allow cropping, resizing, rotating, and flipping images. Images may be imported and exported in GIF or JPG formats.

The WebWisdom Manager provides a set of tools for managing visual properties used in formatting of web pages. The properties are hierarchically defined attributes, which allow defining fonts, colors, backgrounds, bullet images, and all other visual elements of web pages generated by the Dynamic Web Interface. Each property has a system-wide default value. This value can be redefined at the user-, presentation-, or foil-level. This makes the system very flexible and adaptable for different requirements or preferences.

One of the important features of the WebWisdom system, which significantly enhances the comfort of use, is a tight integration with office applications such as Microsoft PowerPoint during the process of preparation or modification of presentations.

The PowerPoint is controlled by a specially designed add-in. By the use of this add-in, the WebWisdom system may invoke all standard functions of the PowerPoint. The same add-in implements a sophisticated exporter, which enables exporting of presentations in form of web pages while retaining all text formatting (fonts, sizes, positions, colors, bullets, etc.). To achieve high conformance with the original formatting the exporter uses Dynamic HTML technology. The WebWisdom exporter differs from the product offered in PowerPoint 2000 in two aspects:(1) is available also for earlier Office versions (97); (2) the output from the exporter can be viewed by both Netscape and Internet Explorer.

As an example of integration with PowerPoint, let us consider the process of importing of a PowerPoint presentation to the database. A user may load such presentation by selecting a function from the system menu. To import the presentation the WebWisdom Manager launches in background an instance of

PowerPoint and passes a request to load the desired presentation. Then, the presentation is exported to a temporary location in the file system in three forms: D-HTML, images, and PPT sources. At least three files, corresponding to these three forms, are generated for each slide. Then, all these files are imported to the database. Each file corresponds to one educational object. Educational objects are grouped into slides, and slides are grouped into a presentation. From now on, the presentation may be accessed from both the WebWisdom Manager and Dynamic Web Interface.

The WebWisdom system can use compressed XML files as an alternative method of storing data. Such files allow transferring data between instances of the WebWisdom database. Each presentation in the WebWisdom system may be exported to an XML file and later imported to the same or another database. The XML files contain all information about the exported presentations including attributes, formatting, sources, etc.

A very important feature of the WebWisdom system is the ability to run without the database. The Dynamic Web Interface can use XML files instead of the database system. This feature is very important when a part of the whole repository has to be taken, e.g., on a laptop to a remote location where Internet connectivity would not permit using the main repository.

4. Dynamic Web Interface

The Dynamic Web Interface allows retrieval of educational data from the database and displaying them in the form of web pages. The architecture of the Dynamic Web Interface is presented in Figure 2. It consists of the TDL Subsystem and the Data Delivery Subsystem.

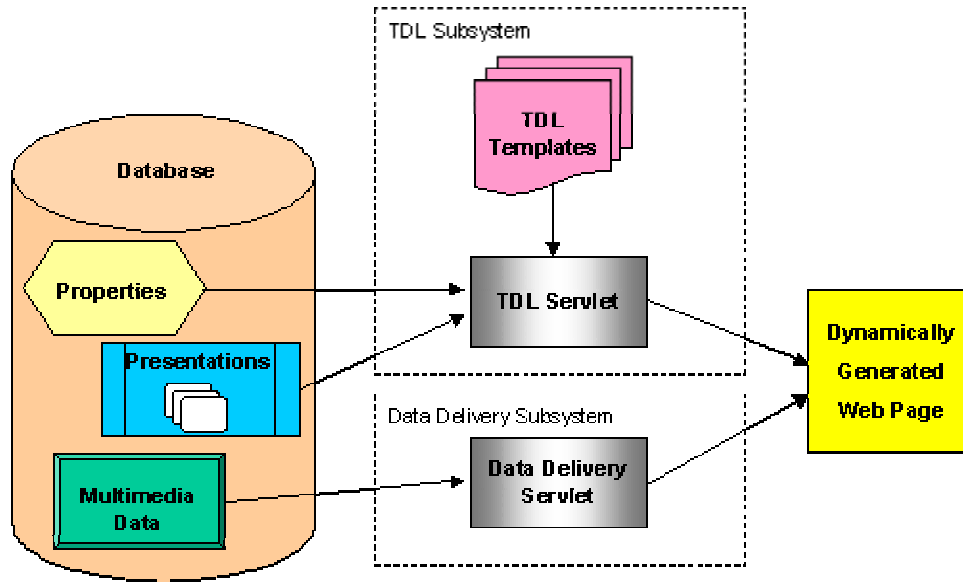


Figure 2. Dynamic composition of WebWisdom web pages

The TDL Subsystem is responsible for dynamic generation of web pages from specially prepared templates. During the generation process the templates are filled with educational data and formatted in accordance with visual properties read from the database. By the use of these properties, a user can control the appearance of the generated web pages.

The Data Delivery Subsystem is responsible for delivering all kinds of multimedia data stored in the database (e.g., images, sound files, and video sequences) to the user browser.

4.1. The TDL Subsystem

The architecture of the TDL subsystem is presented in Figure 3. It consists of a TDL servlet, a set of TDL templates, and implementations of the TDL tags.

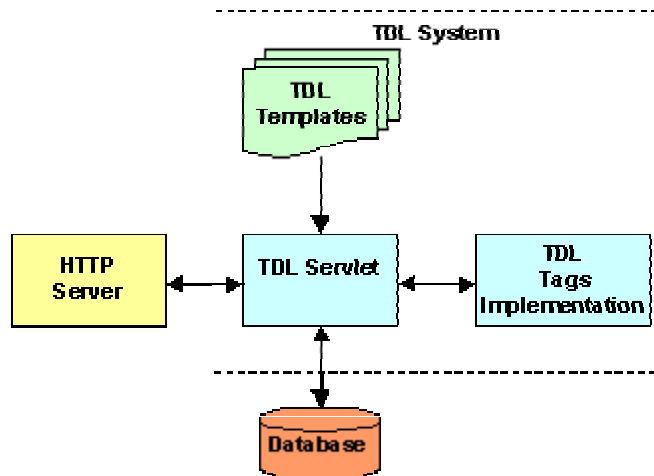


Figure 3. Architecture of the TDL system

The TDL servlet is a Java module conforming to Java Servlet Specification [9] and thus can be used with any servlet-compatible HTTP server. In the currently fashionable language, WebWisdom 2000 is an educational *applications server*.

TDL templates are files written in the TDL language. The TDL language is a user-extensible set of XML tags that can be used inside HTML pages as active elements. Each tag used in the template file has a Java method associated with it. When the template file is processed, the corresponding method is invoked and the result of its execution replaces the tag in the resulting page. Generation of web pages by the TDL servlet may involve algorithmic processing, accessing databases, and can be parameterized. The parameters can be provided by the user, set in configuration files, retrieved from a database, or computed during the web page generation. The overall architecture is an example of *server-side XML application*.

Use of templates and dynamically generated pages instead of static HTML files has several advantages. The same educational data can be presented in many different ways, depending on the user needs, hardware limitations, or educational process requirements. For instance, when a lecture is presented in a big lecture room using a high-resolution screen projector, an instructor can use a full color template with advanced JavaScript functions. The same lecture delivered through an ISDN line to a classroom with several black-and-white workstations would be presented using a simple, non-

resource-consuming template. Moreover, because template files can be easily changed – even by the use of a text editor – a lecturer can always introduce necessary corrections. Furthermore, the same template can be used to deliver in a uniform way presentations prepared by different people, often using different authoring tools. An example of the same presentation displayed by the use of different templates is shown in Figure 4.

4.2. Template Definition Language

The Template Definition Language (TDL) is an application of XML, specialized for defining templates for use with the TDL system. The TDL language is a user-extensible set of XML tags. The TDL tags have names and attributes. Each attribute is specified in the following form: *attribute_name*="value". In TDL language, the value may be a constant or an expression that must be evaluated prior to tag interpretation.

The TDL tags are embedded by '<' and '>' parenthesis and cannot be nested. There are three types of tags: empty-tags, start-tags, and end-tags. Empty-tags are used to denote empty elements, i.e., elements that do not have any contents. Such tags are indicated by using the slash "/" character at the end of the tag. Examples of empty-tags in TDL are **<CONNECT/>** and **<INSERT/>**.

Start-tags and end-tags are used to denote non-empty elements. Examples of start-tags in TDL are **<DOCUMENT>** and **<FOR>**. The end-tags have the same name as the corresponding start-tags but begin with a slash '/' character in front of the tag name, e.g., **</DOCUMENT>** and **</FOR>**. A pair of start- and end-tags defines a non-empty TDL element that may contain other TDL elements or text. The meaning of the element contents depends on the element type. Valid nesting of elements is defined by the Document Type Declaration (DTD).

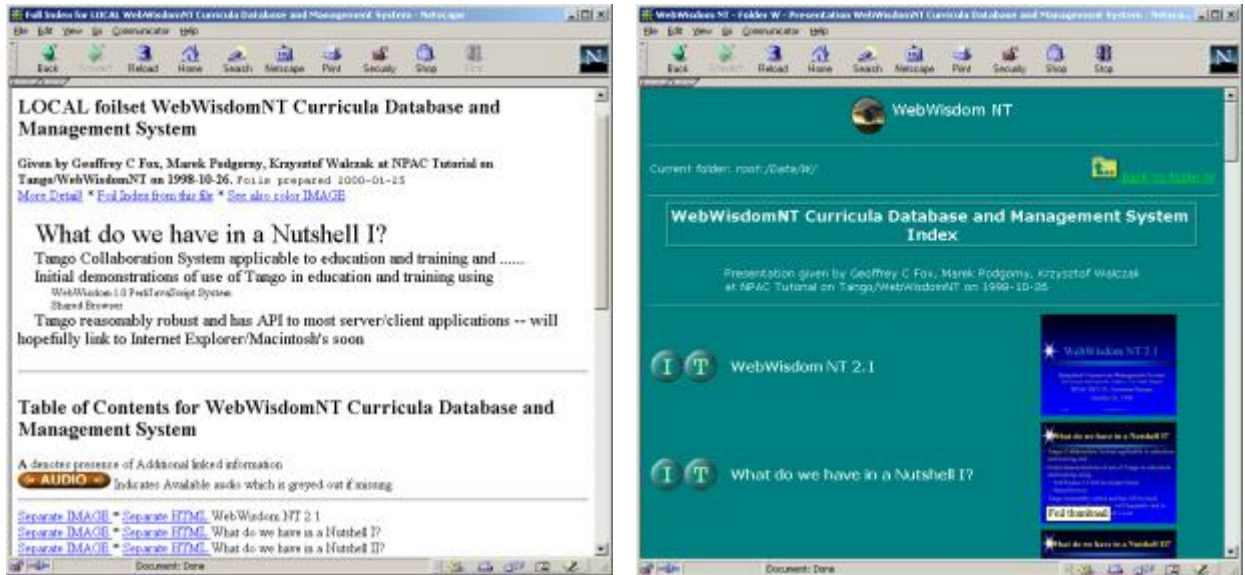


Figure 4. The same presentation displayed using two different templates

There are a number of common, general-purpose tags defined in the TDL language:

- The root tag **<DOCUMENT>**,
- Conditional statement tags **<IF>**, **<THEN>**, and **<ELSE>**,
- Loop control tags **<FOR>**, **<WHILE>**, and **<ITERATION>**,
- Database access tags **<CONNECT/>** and **<QUERY>**, and
- Variable manipulation tags **<INSERT/>** and **<SET/>**

A very important feature of the TDL language is its extensibility. New tags extending the TDL language can be easily added to the system. Adding a new tag requires writing Java code that is responsible for handling of this tag and adding this information to a configuration file. A powerful API available with the system makes the process of writing new tags quick and easy.

4.3. Operation of the TDL System

The cycle of servicing a user request by the TDL system is presented in Figure 5. The process is started by a client browser sending a request to the TDL system. The

request has a form of a URL containing the TDL host name, template identifier, and additional parameters. After the request is received by a TDL system, the TDL servlet is invoked. The TDL servlet uses sophisticated caching and prediction algorithms that prevent it from processing repeated requests and shorten significantly the average response time.

If the result is not available in the cache, the process of web page generation is started. First, the template file is loaded either from its original location or from the template cache. Then, the file is parsed by the XML parser and all TDL tags are identified. Next, each tag is interpreted, starting from the document root. For each tag, the associated Java method is invoked.

The method can access a database through the database access module, which also stores a copy of the query results in the query cache. If the SQL query is repeated, the result from the cache will be used. The results of tag execution are collected by the output module. After processing all tags in the template file, the resulting HTML page is sent to the client and saved in the output cache.

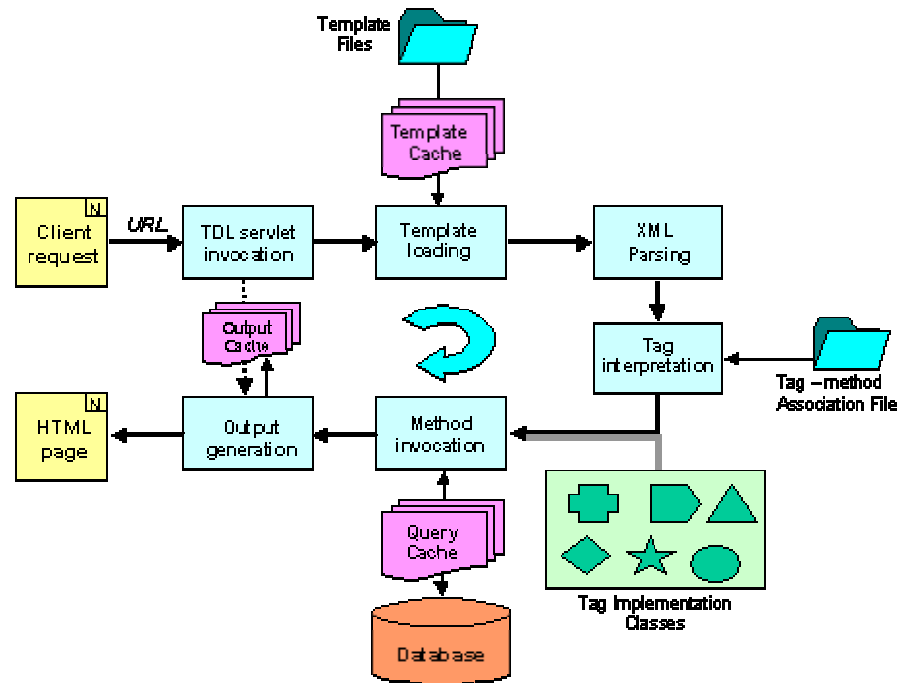


Figure 5. Operation of the TDL system

4.4. Features of the TDL System

The TDL system has been designed in a generic way that allows its use in different applications and environments. The following list summarizes most important features of the TDL system:

- **Generic execution engine.** The TDL system has a generic, application-independent execution engine. The system provides a framework for interpretation of templates and execution of application-dependent TDL tags, and it can be used as a general application server.
- **Tag API.** The TDL system provides a powerful set of methods that can be used in tag implementations.
- **XML compliance.** The TDL language is build on XML – the emerging web standard developed by the WWW Consortium.
- **Caching and prediction.** The TDL system is equipped with multilevel cache and sophisticated prediction system. All results of processing are stored in the cache for later use. Moreover, the

system pre-computes the pages, which are most likely going to be requested by the user.

- **Database access module.** The TDL system uses an advanced database connection interface. The interface allows serving multiple simultaneous requests to the database in an efficient way.
- **Customization.** Behavior of the TDL system can be easily controlled by a set of parameters. This allows adaptation of the system to different environments.

5. Deployment of the WebWisdom 2000 System

Current version of the software uses Oracle 8i as data repository. The application server requires a servlet-compliant HTTP server. We currently support Apache and Netscape Enterprise servers. Some of our customers implemented sizeable data repositories. For instance, NPAC at Syracuse University, uses our system as a universal repository for all presentations and lectures that have been given there since 1994. The system works with approximately 8 GB of data consisting of approximately 1500 presentations with a total of about 40,000 foils. This particular implementation is running on the dual Pentium II Xeon 450MHz server. A two-CPU servers are recommended for our system and ensure excellent performance. Dual CPU machines enable TDL servlet and the database engine use separate CPUs. This leads to very significant performance boost as the system design allows for high level of concurrency. Syracuse University system is being used in production mode during the ISDL classes being delivered to other universities.