noc.system

Truman State University, Missouri
Alexander Horn, main developer
Diane Tobin Johnson, faculty member
January, 2005

LOR Application

We began this project with the assumption that improved student access and use of learning objects would enhance student learning. Dr. Johnson is interested in taking advantage of technology in form of a Learning Object Repository in her upper level communication classes. She believes, for example, that immersion in case studies will help her students engage higher-order thinking skills and thereby learn more effectively. For more information about pedagogic strategy, see the description of the pilot project in Appendix A: Pedagogical Description of Comm 353.

A learning object repository (LOR) can be simplistically conceptualized as a electronic storage space for objects used in teaching, a grown-up and more accessible version of a multimedia library. The initial challenge posed by the LOR is the selection/creation of the objects. As the number of objects and the numbers of uses to which they are put increase, the *management* of the LOR becomes the primary challenge. It is important to try to note the difference between the management of learning objects and management of a classroom.

Tools such as Moodle or Blackboard create the means for electronic discussions, assignment submission or the access to grades. A tool to administer a large LOR, on the other hand, is specifically designed to manage multimedia files such as audio, text, and visual resources and to provide access to students in the most user-friendly ways possible.

The challenge of managing the LOR became the focus in developing a pilot project because the developer and the professor shared two concerns. The first was ease of student access. Tools that are not easily accessible will not be used. The second was freedom to use materials in emerging as well as anticipated ways. Much learning at the college level is not a linear activity and any learning system needs to respond to the synergy in that learning. That is to say, we had to move away from the assumption that data structures for the organization of learning resources have different properties than the actual content of those structures. In fact, the distinction between both, a group versus its content, is at best inaccurate because an object can be a group and a group may be an object depending on the context. Noteworthy, content systems that try to encapsulate the idea of a hierarchical structure often lack sufficient flexibility, forcing the user, for instance, into a “standard” organizational pattern of fixed categories in a tree structure. This privileges the category over the content. The instructor is forced to make learning objects “fit” into preconceived units. Even when categories are numerous and customizable to a certain extent, this approach is too narrow and static to respond to changing learning situations. To

the fullest extent possible, the user of the system needs to be able to decide how the content will be used.

Noc Architecture

The noc.system is a data storage system that we developed to create effective ways to categorize, find and distribute content to students without assuming anything about the nature of the information or its “position” within an organizational structure. Noc is a *metadata-file management system* that is designed from ground up to provide *functionality* in the form of building blocks. The software design follows one fundamental principle: provide the tools for a powerful storage system as opposed to a narrow and static end-product. That means that the noc.system composes only the storage engine while each building block, in turn, implements specific functionalities that are performed on content assets. Thus, the Noc allows the (re)definition of the capabilities of the content system creating a powerful architecture. (For more information about the noc.system and in particular its application program interface (API) that allows customized extensions to bind natively into already existing source code, refer to the documentation in Appendix B: noc.system v2.3.)

There are a number of distinct design features in Noc that make this kind of flexibility possible in a large repository. First, similar to UNIX-based operating systems, everything that is stored in Noc is an object (even a “directory” or “group” is an object). Yet, the noc.system goes a step further and establishes a

tight integration between metadata and the actual data (file) that is to be stored. In fact, every object is metadata while a binary file is only associated with it.

Second, because the nature of data often cannot be described with a single attribute, Noc allows multiple-inheritance, that is, it allows objects that can have more than one parent group. For example, a “ball” object could be categorized by its physical attributes and/or by its use (play, exercise, sport). Within a course, an object may have several uses, so different approaches are necessary. Relevant to most users, hierarchic structured data (single-inheritance, or having one parent group) is maintained throughout the system by default; however, data may also be referenced in a realm which bundles identical objects that are scattered in mutually exclusive “groups,” thus supporting multiple inheritance. As a result, the instructor administers in all cases only one instance of the object which reduces workload considerably.

From the very beginning, the development of Noc has always been inspired by UNIX technology. Hence, Noc strictly implements the clean separation between program logic and the web design: the entire browser-interface, is purely written in HTML/CSS using a template system; the web interface may be designed from scratch for the needs of today's instructors.

The noc.system may not offer all the functionalities needed in a specific LOR, but it does offer an effective mechanism to create building blocks that extend functionality and that will natively bind with already existing source code. We believe that it is worthy of continued experimentation and development. We

have made tremendous progress on this project within the last eight months. Our hope is to be able to implement a modest LOR with the Comm 353 class at a beta-testing stage by Fall 2005 and in full operation with that class by Spring 2006. So far, the project has been privately funded. It is likely that we will need additional financial and technical support in order to achieve our goals.