# PLUG-AND-PLAY UMLS KNOWLEDGE SOURCE SERVER USING WEB SERVICES AND PORTLETS

**Karen E. Thorn, Anantha Bangalore, Allen Browne**
**US National Library of Medicine, Bethesda, Maryland**

## Abstract

*The UMLS is large and complex and presents significant challenges in retrieving information in a comprehensive way. Typical users of the UMLSKS (UMLS Knowledge Source Server) run the gamut in terms of knowledge of the UMLS and retrieval needs from the UMLSKS. Even though we as developers of the UMLSKS have always strived to create a truly modular and flexible system, we felt that the available technology until recently was not mature enough to create a loosely coupled, plug-and-play UMLSKS. In the past year we became convinced that web services and portal technology had sufficiently matured to allow us to develop such a system. In this poster, we present the details of the new plug-and-play UMLSKS that has been in development for about a year.*

## Introduction

The centrally managed UMLSKS provides system developers with UMLS information remotely and on demand. Currently we provide the user community with a rich set of APIs that allows system developers to build applications to meet their needs and a web interface that provides a single comprehensive view of the data. As the UMLS continues to grow in size and complexity, it is reasonable to expect that a single view of the data may not be sufficient to meet the requirements of a majority of the users. Ideally these views of the data should be user driven and we as developers should provide a plug-and-play framework of reusable software components that will allow users to build these custom views.

## Current UMLSKS

The current implementation of the UMLSKS is based on JAVA/RMI (Remote Method Invocation) technology at the backend and a combination of servlets and XSLT stylesheets at the front end. Even though the current implementation is quite flexible, scalable and robust it is not very modular. Every time a change has to be made to the system, whether it involves adding a new feature, deleting a feature or temporarily disabling a feature to make a fix, the entire server needs to be taken off-line. Since the code for the current system is tightly coupled and based on low level mechanisms such as RMI, it is very hard to incorporate new functionality created by third party developers. The current system also does not provide a language independent way of integrating new functionality.

## Web Services

The term Web Services describes a framework in which services are described, published, and discovered. These services are deployed in a distributed computing environment and are available for dynamic invocation. The service provider implements a service, described as a collection of operations that perform a task, and publishes that service for use. These services are discovered dynamically through a service registry based on specified criteria and ultimately invoked by a service requester. The Web services architecture is based on a set of XML standards that allows the loose coupling of various services. The dynamic nature of web services allows us to add new services without having to shutdown the entire server. It also provides a platform and language independent way of integrating various services.

## Portlets

A portal is a web interface that can be customized by the end-user both in look and feel and in available content using applications provided by the portal. The portal functions as an aggregator of portlets. A portlet can be thought of as a miniature Web application that is running inside of a portal page along side any number of similar entities. A portlet is a reusable web component which is managed by a portal container and can process requests and generate dynamic content. Each of the portlets generates fragments of mark-up, which the portal container ultimately pieces together to create a complete page. There is a portlet registry similar to the web services registry containing a list of all the available portlets.

While web services allow users to reuse back-end services, portlets allow users to reuse web components. Used together, they provide a truly flexible plug-and-play framework.

## New UMLSKS

We are currently completing the development of the new UMLSKS. We are using Sun's Web services developer tool kit to build the back-end web services. uPortal an open source portal framework is being used to develop the portlets. On completion, the new system will have a rich set of web services and portlets that will allow end-users to create flexible and customizable applications.