1. Introduction

This document describes the module decomposition for the URL classification system. In order to understand this document, you should be familiar with “A URL Classification System: Requirement Specification.” The decomposition presented here was constructed using Object-oriented design techniques. Figure 1 shows the Is-composed-of hierarchy for the system. The URLClassifier module deals
with the application's main user interface and as such serves as the “root” of the Is-composed of hierarchy. There are five other main modules: URLManager, DocAnalyzer, Archiver, URLRepPrefs, and Utils. URLManager hides the details of how the information about URLs is created, manipulated, and retrieved. DocAnalyzer hides the algorithms used to fetch HTML documents and analyze them for identifying information. Archiver provides an interface to the data store where the collected information about a URL is archived. URLRepPrefs manages user preferences for the system. Utils is a module that collects various utility classes.

2. URLClassifier

URLClassifier is a class which sets up the user interface and manages the operation of the application. It acts as a container or driver for all of the other classes.

Requirements references: 3.1.1 User input; 4.1 URL entry; 4.2 Automatic classification; 4.3 URL information storage; 5.2 Usability.

3. URLManager

URLManager is a module which collects several related classes that together hide the interface to storage and retrieval of URL information.

Requirements references: 3.1.1 User input; 3.1.2 Remotely retrieved data; 3.2 Data output; 4.1 URL entry; 4.2.1 Retrieval of previously classified URLs; 4.3 URL information storage.

3.1. URLEntry

URLEntry is a class which hides the structure of a URL entry and provides mechanisms for manipulating them.

Requirements references: 3.1.1 User input; 4.1 URL entry; 4.3 URL information storage.

3.2. URLEntryArchiver

URLEntryArchiver hides the mechanism for storing a URLEntry object in the data store and for retrieving a URLEntry object from the data store.

Requirements references: 3.1.2 Remotely retrieved data; 3.2 Data output; 4.2.1 Retrieval of previously classified URLs; 4.3 URL information storage.

3.3. URLArchiverResult

URLArchiverResult is a class which hides a set of keywords; it is used to add new keywords to an existing URL description.

Requirements references: 3.1.2 Remotely retrieved data; 4.3 URL information storage.

4. DocAnalyzer

DocAnalyzer is a module which collects several classes that together hide the mechanism used to auto-
matically classify a web document given its URL.

4.1. HtmlDocHead
HtmlDocHead is a class which encapsulates the mechanism used to retrieve an HTML document and parse out its head portion which contains any metadata supplied by the author. Other classes can retrieve various pieces of the document head from an initialized HtmlDocHead object. Because it is likely that additional metadata formats will be invented and used, this class provides a flexible mechanism for adding and using multiple techniques for extracting information from the document head.

**Requirements references:** 4.2 Automatic classification; 4.2.2 Use of metadata in classifying URLs; 5.3 Intelligence; 6.2 Internet communication failure; 6.3 Invalid URL supplied; 7.1 Subsets; 7.3 Potential changes.

4.2. AttributeAnalyzer
AttributeAnalyzer is a module consisting of classes that analyze a document head for information related to a particular attribute. Examples of attributes that document heads may contain are keywords and descriptions. These classes all present a common interface to make it easy to analyze attributes that are parts of new metadata formats as they are invented and used.

**Requirements references:** 4.2 Automatic classification; 4.2.2 Use of metadata in classifying URLs; 5.3 Intelligence; 7.1 Subsets; 7.3 Potential changes.

4.2.1. Keyword
Keyword is a class which hides the algorithms that are used to extract keywords from HTML document heads.

**Requirements references:** 4.2 Automatic classification; 4.2.2 Use of metadata in classifying URLs; 5.3 Intelligence; 7.1 Subsets; 7.3 Potential changes.

4.2.2. Description
Description is a class which hides the algorithms that are used to extract description or summary information from HTML document heads.

**Requirements references:** 4.2 Automatic classification; 4.2.2 Use of metadata in classifying URLs; 5.3 Intelligence; 7.1 Subsets; 7.3 Potential changes.

5. Archiver
Archiver is a module which collects several classes that implement the way the system interacts with the data store. This module was designed in anticipation of using this architecture to store types of information other than URL classifications.

**Requirements references:** 2.3 Other Software; 3.2.1 Remotely retrieved data; 4.2.1 Retrieval of previously classified URLs; 4.3 URL information storage; 6.1 Inaccessible data store; 7.3 Potential changes.

5.1. ArchiverException
A special Java exception used to indicate difficulties accessing the data store.
5.2. **ArchiverUtils**  
ArchiverUtils is a class representing a convenience abstraction for some utility methods used by Archivers.

**Requirements references:** 2.3 Other Software; 3.2.1 Remotely retrieved data; 4.2.1 Retrieval of previously classified URLs; 4.3 URL information storage; 6.1 Inaccessible data store; 7.3 Potential changes.

6. **URLRepPrefs**  
URLRepPrefs is a class which hides the mechanisms used to manage user-specified preferences and other run-time configuration information.

**Requirements references:** This class does not directly address any of the stated requirements. It is, however, intended to support a typical feature of Graphical User Interface-style programs.

7. **Utils**  
Utils is a module which is composed of various utility classes which provide services to other modules.

7.1. **Set**  
Set is a class which provides an abstraction of the mathematical notion of a set of objects. It supports the manipulation of collections of keywords.

**Requirements references:** 4.2 Automatic classification.

7.2. **StringUtils**  
StringUtils is a convenience abstraction containing a number of methods which perform transformations on strings of characters.

**Requirements references:** The methods provided by StringUtils are used throughout the system.

7.3. **DBUtils**  
DBUtils is a module containing classes that interact with the database management system (DBMS) if one is used for the data store. It could be argued that this module belongs under the Archiver module, but the classes belonging to the Archiver module are intended to mask the presence of any particular mechanism for accessing the data store. DBUtils is therefore at a lower level of abstraction, since it deals with a particular mechanism. While it is not necessary for a DBMS to be used for the data store, it is a reasonable choice and this class exists in support of it.

**Requirements references:** 2.3 Other Software; 3.2.1 Remotely retrieved data; 4.2.1 Retrieval of previously classified URLs; 4.3 URL information storage; 6.1 Inaccessible data store; 7.3 Potential changes.
7.3.1. DBServerInfo
DBServerInfo hides the details of establishing communicating with a database management system server. It supports using a database management system as the data store.

**Requirements references:** 2.3 Other Software; 3.2.1 Remotely retrieved data; 4.2.1 Retrieval of previously classified URLs; 4.3 URL information storage; 6.1 Inaccessible data store; 7.3 Potential changes.

7.3.2. DBPipe
DBPipe hides the data structures used to maintain a connection to a DBMS server.

**Requirements references:** 2.3 Other Software; 3.2.1 Remotely retrieved data; 4.2.1 Retrieval of previously classified URLs; 4.3 URL information storage; 6.1 Inaccessible data store; 7.3 Potential changes.

7.3.3. TableCreator
TableCreator hides the structure of the DBMS tables used by the system and the mechanism that must be used to create the tables in a new database.

**Requirements references:** 2.3 Other Software; 3.2.1 Remotely retrieved data; 4.2.1 Retrieval of previously classified URLs; 4.3 URL information storage; 6.1 Inaccessible data store; 7.3 Potential changes.