

# Modeling Web Site Structure in UriGraph

Hrvoje Simic

Department of Telecommunications, Faculty of Electrical Engineering and Computing, University of Zagreb  
Unska 3, 10 000 Zagreb, Croatia  
+385 (1) 6129-748

hrvoje.simic@fer.hr

## ABSTRACT

This paper introduces a Web site structure model called UriGraph and, using the model, describes several important patterns of site structure. Web site structure is defined as the collective information about the identity, identifier, position and composition of every resource constituting the Web site. UriGraph models the site's resource identifiers and through them the resource identity and composition, and indirectly the resource position. UriGraph is designed specifically for the Web and it is compatible with the current practice. It can be represented graphically and as an XML document.

## Categories and Subject Descriptors

D.2.2 [Software Engineering]: Design Tools and Techniques.

## General Terms

Design, Languages, Theory

## Keywords

UriGraph, URI, Web site structure, Web resource

## 1. INTRODUCTION

The term "Web site structure" is often used, but seldom defined. A novel definition of Web site structure is given here, discriminating between the structure of Web site content and the navigation schema on one hand and the Web site structure on the other, although all three concepts are mutually dependent. We present a new Web site structure model, used to analyze the Web resource identifier (the http-scheme URI) to provide clues about the resource identity. This identity is in effect used by the components of each resource to determine what content they should generate. Creating passages through the structure graph during the identifier analysis allows for inheritability of components, similar to propagating permissions in a file system.

## 2. WEB SITE STRUCTURE

It is useful to regard a Web application as a composition of two distinct systems: the target system and the Web adaptation system. The target system contains functionality and content independent of the technology used to access it. The Web adaptation system is used to interface between the target system and the Web, allowing the whole system to be perceived as a Web site. Web adaptation system defines the structure and presentation (the "look") of the site. The structure is the central ingredient catalyzing the other three, and defined as the collective information about the four qualities of every resource constituting the site: its identity (what the resource is about), identifier (how can it be referred to), composition (what kind of information it delivers), and position (how does it relate to other resources).

Every Web resource should represent a concept, typically to provide information about it. *Identity* of the resource corresponds to the intension of the represented concept, i.e. the set of all attributes necessary and sufficient for defining that concept.

The *identifier* should convey the information about resource's identity, nothing more and nothing less. Common identifier standard for the resources on the Internet is the Uniform Resource Identifier (URI). The URI is widely exposed to the user and therefore a vital part of the Web user interface [3][5].

In our approach, each Web resource is a *composition* of one or more separate, self-sufficient, encapsulated logical parts, called resource components (as in WebML [1]). The information about which components are included in a specific resource as well as the relationships between individual components is called the composition of the resource. Also, a component has its own identity, embedded in the identity context of the resource in which the component is placed.

Resource's *position* in the Web site defines its relationships with other resources on that site (as in [2]). The principle of relating resources to concepts ties the resource position tightly to its identity – relationship between resources is similar to relationships between represented concepts.

## 3. URIGRAPH

This section gives a brief overview of the model based on the definition of Web site structure. For a formal definition and a more detailed look at UriGraph, see [5].

The model is composed of three layers. The bottom layer is called the *topology layer*, defining the nodes and edges of the graph. The middle layer defines the rules for analyzing the resource identifier and is called the *request analysis layer*. The top, *response synthesis layer*, is used to define how the information is extracted from the identifier and incorporated in the response. Site structure in UriGraph can be clearly presented graphically and also in special XML grammar.

### 3.1 Topology Layer

UriGraph's topology layer is defined as a directed graph constituting of the set of nodes  $N$  and a set of edges  $E$  (a binary relation over  $N$ ). There are two types of nodes: *places* (collected in a set  $P$ ) and *transitions* (in set  $T$ ). There is one prominent place called the *root node* ( $r$ ). Any two nodes may be directly connected via at most one directed edge in each direction, unless the following cases apply: a node cannot be directly connected to itself ( $E$  is irreflexive); two places cannot be directly connected; and a transition can have at most one outgoing edge connecting it to a place.

Places (depicted as circles) represent classes of resources containing a single resource or several similar resources that differ in content, but not in the way they are represented on the site. Root node (marked with a symbol of a house) represents the

home page and transitions (wide shapes) mark the analysis of pieces of information.

### 3.2 Request Analysis Layer

The UriGraph request container holds two sequences, a sequence of path segments and a sequence of query segments. The analysis starts at the root node with container filled with all the segments from the HTTP request URI and follows the edges through the nodes, dropping one segment at each transition. Analysis regularly finishes at some place with an empty container, thus constructing a passage in the graph.

Each node in the constructed passage gets *processed*, starting at the root node. To process a node means to: (1) take out one path or query segment from the request container, and (2) find the next node, append it to the walk and continue the analysis by further processing it. Trimming (step 1) happens only in transitions. There are two types of transitions: *path transitions* (depicted as parallelograms) which trim path segments and *query transitions* (with curved sides) which trim query segments. Path segments are always trimmed in order they appear in the URI; query segments can be trimmed in any order. To find the next node in the passage (step 2) one has to establish which of current node's destination nodes are traversable. There should regularly be only one traversable node, which is taken to be the next node.

Every place is traversable. To determine if a transition is traversable, a special logical function called a *pass* is introduced. A pass evaluates to true (open) or false (closed) depending on the part of the request being tested and on the state of the analysis. Traversing a transition includes *activating* the open passes in the transition. Each transition has exactly one pass, which can be composite, i.e. contain other passes.

In some cases it is necessary to establish *priority relation* on the set of destination nodes for some node *n*, defining the order of testing traversability and selecting the next node in the analysis. Our approach uses the HNL model of priority markings when assigning priorities to edges. Each marking is a string of letters 'H' (representing high priority), 'N' (normal) and 'L' (low priority). the Priority is determined by the difference in the priority level of the first letter at which they differ. Extending the marking by appending priority letter 'N' to it does not change its priority level.

### 3.3 Response Synthesis Layer

The top layer of the UriGraph model provides mechanisms called the response synthesis, in which the information extracted by request analysis is used to determine the resource identity and composition.

To describe the identity of a resource, UriGraph defines *clues*, elementary pieces of information corresponding to the general attributes of the concept that the resource represents. Clues can be located in passes: each pass has its set of clues. Clues are collected from activated passes while traversing transitions during the analysis. The set of collected clues at the end of synthesis represents the resource identity.

The composition of a resource is a set of *components*. Components are also collected during the analysis, but they reside in the places. Each component can be *local* (included in the response if the passage ends in the place it resides) and/or *inheritable* (included if the place where it resides is in the passage, but not as the last node). Components have their identity which contains the resource's identity. The difference may be in

components' extra clues, which are assigned to the components themselves.

## 4. EXAMPLE GRAPH

To illustrate the model, a simple example of a site structure graph for displaying and manipulating messages is shown in Figure 1. The information in the response synthesis layer is usually not presented graphically, to avoid cluttering the picture with too much detail. Place number 2 (identified by "/create" URI) models a resource for entering new messages. Resource class 4 represents a specific message (e.g. "/12"), and 5 a selection of messages, either a range ("/?from=1&to=9") or an enumeration (like "/?id=2&id=5" – iterating messages is enabled by higher-priority edge). Resource classes 3, 7, and 8 perform transactions. Deleting a message number 3 can be identified as "/3/delete" or "delete?id=3".

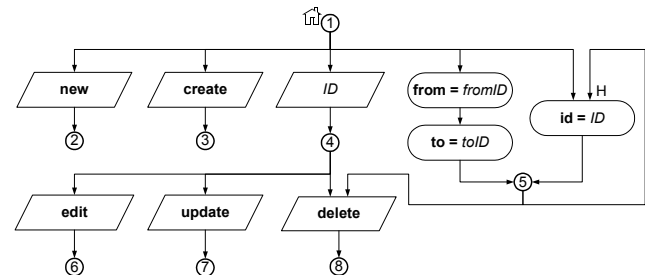


Figure 1: An example of UriGraph containing the topology and request analysis layer.

## 5. CONCLUSION

UriGraph can be used to describe the structure of any Web site, but it is especially intended to be used as a blueprint for larger, even enterprise-sized Web applications. It is a tool for software engineers and some of its features can only be exploited through programming. It is a foundation for a specific software engineering approach aimed at reducing inner redundancy, facilitating development and maintenance of larger sites. Specific features of UriGraph need a special Web server to deploy (like Wance [5]), but the basic graphic representations may be used on any platform.

## 6. REFERENCES

- [1] Ceri, S., Fraternali, P., and Bongio, A. Web Modeling Language (WebML): a modeling language for designing Web sites, 9th International World Wide Web Conference, May 2000, Amsterdam, The Netherlands
- [2] Liechti, O., Sifer, M. J., and Ichikawaielsen, T. Structured graph format: XML metadata for describing Web site structure. 7th International World Wide Web Conference, August 1998, Brisbane, Australia
- [3] Nielsen, J. Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Indianapolis, 2000
- [4] Simic, H. Application of UriGraph to Uniform Resource Identifier Design. CARNet User Conference 2002 Proceedings, Zagreb, Croatia, September 2002, <http://www.tel.fer.hr/users/hsimic/cuc2002>
- [5] Simic, H. Modelling of WWW site structure. Master's Thesis, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia, May 2002 (In Croatian), <http://www.tel.fer.hr/users/hsimic/magisterij.pdf>