Data Modeling of Web Site Structure: An Oracle Prototype

Rajeev Kaula
Computer Information Systems Department, Missouri State University, Springfield, MO 65897 (USA)
E-Mail: RajeevKaula@missouristate.edu

***Abstract*— Web page design is essentially an intuitive approach. However, as Web pages become more complex in their structure and content, it is not easy to create and maintain their layout structure consistently. This paper outlines an approach to create an entity-relationship data model of Web site and Web page constructs for storage in a relational database. The approach also outlines the generation of Web pages from database repository. The approach details are illustrated by a prototype in Oracle to show the efficacy of the concepts.**

***Index Terms*— Data Modeling, Web Page, Web Site, Entity-Relationship Diagram, Relational model, Oracle.**

I. Introduction

Internet or Web is a worldwide phenomenon that involves creation of Web sites that go beyond simple sharing and exchange of information to applications that can perform business tasks. One of the basic element of a Web site is a Web page. Each Web page contains information ranging from simple text to database data including complex multimedia or analytics. Web site development involves the creation of one or more Web pages that are written in HTML language. HTML structures the Web page layout content through various markers called tags [11, 16].

Web page design is essentially an intuitive approach. However, as Web pages become more complex in their structure and content, it is not easy to create and maintain their layout structure consistently [1, 2, 10, 12, 13, 14, 15]. Web development tools like Adobe Dreamweaver ease the complexity of Web page design [8], but still such tools are not easy to apply.

One alternative to ease Web page design complexity is to facilitate Web page layout and content through a relational database. Even though databases are often utilized to store Web page contents in the form of Active Server Pages [17], Java Server Pages [6], and so on, the utilization of database to also facilitate Web page layout is not common. There have been attempts to utilize UML modeling to develop Web applications [3, 4, 18], but these attempts again focus on the entire Web architecture. Such attempts perceive a Web page to be one of the objects in its entirety.

This paper outlines an approach to model a Web site and Web page constructs or tags in a relational database as an alternative for Web design. To facilitate the modeling process, the approach utilizes the entity-relationship (ER) model for modeling the HTML tags initially, and, then the entity-relationship model is transformed into a relational database for storage as a database repository. The notion of using database concepts [5, 7, 9] for modeling Web pages enables (i) sharing of HTML tags across various Web pages; (ii) sharing of database content in HTML tags, (iii) adoption of database concepts towards Web site maintenance; and (iv) provide modularity in Web page design. The paper now describes the entity-relationship model of a Web page, followed by a prototype in Oracle of a sample Web site to illustrate the efficacy of the concepts.

II. Web Page Data Model

The Web page layout entity-relationship model schema consists of entity types representing various HTML tags and their relationship with each other from the perspective of tag usage during page layout. Every HTML tag can be modeled as an entity type. Since all Web pages always have two tags <html> and <body>, these tags need not be modeled separately. Also some HTML tags that appear by themselves, like <hr> or
 also need not be modeled as separate entity types, but instead included directly in other entity types based on their usage. The Web page modeling schema will consist of three entity types: individual tag entity, group tag entity, and site layout tag entity. Each of these entity type is now described.

A. Individual Tag Entity

Individual tag entity is an entity type that represents individual tags. The structure of the entity type is shown in Figure 1. In the Figure, the entity name “Tag” will be name of the specific HTML tag. TagNo attribute is the primary key of the entity type. Type attribute is the tag specification value. TagID attribute represents the HTML tag ID identifier value. Content attribute represents the HTML tag content value. The content value can be some fixed value or transactional database derived value updated during Web page generation. Attribute1, Attribute2, ... represent other HTML tag attributes.

doi:10.4304/jsw.9.6.1453-1462

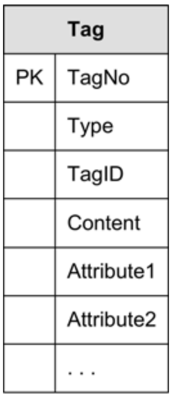

Figure 1. Individual Tag Entity Type

For example, `<p>This page shows items available.</p>` tag will be represented as an individual tag entity type “pTag” as shown below in Figure 2. In Figure 2, the entity type structure is shown in (a), while the entity instance for the tag statement is shown in (b).

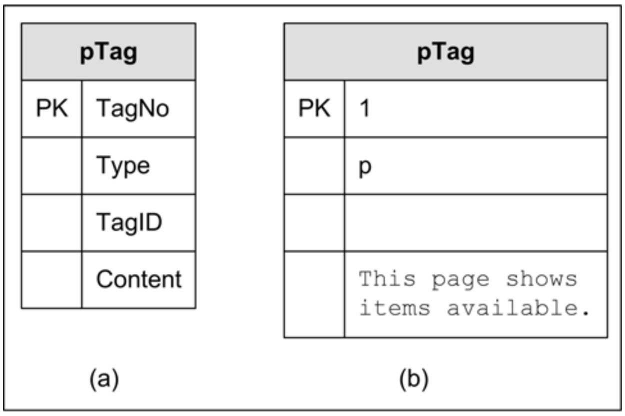

Figure 2. Individual Tag Entity Instance

B. Group Tag Entity

Often times, during Web page layout, a HTML tag includes other HTML tags as a group. Group tag entity is an entity type that represents a collection of HTML tags within some HTML tag. In such situations, the HTML tag that groups other tags for reference can be referred as the outer tag, while the tags that appear inside the outer tag can be referred as inner tags. The outer tag will now be represented as an individual tag entity type. However, the HTML tags that comprise the inner tags are represented through database relationships with a separate entity type “Tag Details” as shown in Figure 3. Tag Details entity type is an associative (weak) entity type that has (i) a 1:N relationship with the individual tag entity type that defines the outer tag for the group, and (ii) develops separate database relationships with entity types that represent the inner tags.

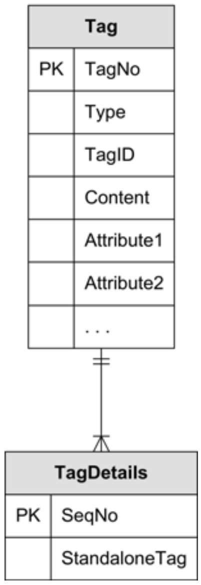

Figure 3. Group Tag Entity Representation

In Figure 3, the entity name “Tag” represents the entity type for the outer tag that holds the inner tags. The entity name “TagDetails” represents the entity type that will provide the relationship with inner tags that appear within the outer tag. The SeqNo attribute values will sequence the individual tags as they appear in the page layout within the group. The Standalone Tag attribute represents any standalone tag that may appear in the group. It is also possible for the TagDetails entity type to become the outer tag to some other group of inner tags. In such case of nested group tags, the TagDetails entity type will have another weak entity type to show additional details of its inner tags.

For example, the following HTML statements contain a <div> tag that contains other individual tags <p>, <a>, and <hr> in a Web page layout. In Figure 4, the entity type structure for such statements is shown.

```
<div align="center"><p>Web Design ER
ER</p>
<a href="sample_page.html">Home</a>
<br><hr /></div>
```

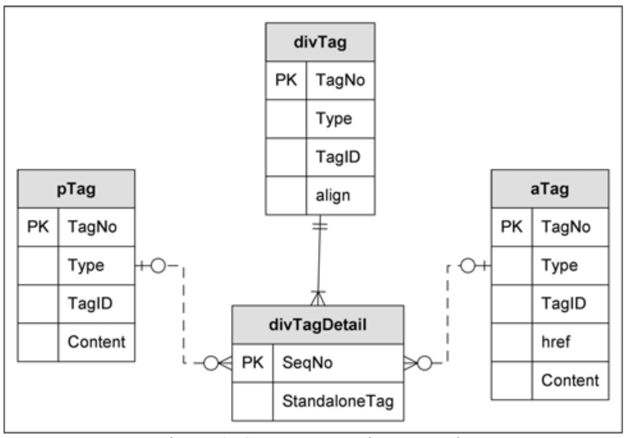

Figure 4. Group Tag Entity Example

In Figure 4, the divTag entity type represents the outer tag <div>, while the divTagDetail entity type represents the collection or group of other tag entity types that are part of the inner tags. The pTag entity type represents the <p> tag in the inner tag group, the aTag entity type represents the <a> tag in the inner tag group, while the <hr /> is the StandaloneTag attribute in the inner tag group. The relationship between the individual tag entity types (pTag and aTag) and divTagDetail entity type is optional-to-optional 1:N, implying that an individual tag (pTag and aTag) instance may be associated with only one instance of divTagDetail instance. The relational model table representation of Figure 4 schema is shown below. In the database tables, primary keys are underlined, and foreign keys are in italics.

TABLE I.
pTAG TABLE

TagNo	TagType	TagID	Content
1	p		Web Design ER

TABLE II.
aTAG TABLE

TagNo	TagType	TagID	Content	href
1	a		Home	sample_page.html

TABLE III.
divTAG TABLE

TagNo	TagType	TagID	Content	Dalign
1	div			center

TABLE IV.
divTAGDETAILS TABLE

divTag	SeqNo	*pTag*	*aTag*	StandaloneTags
1	1	1		
1	2		1	
1	3			br
1	4			hr

C. Site Layout Tag Entity

A Web site is a collection of Web pages. Hence the site layout tag entity represents the entire Web site layout. Also, as any Web page layout is a collection of individual tags and group tags, there will be two entity types to represent the Web site page layout structure. One entity type will store each Web page identification information including the contents of the <head> tag, while the second entity type will contain the references for individual tag entity or group tag entities in the sequence of their occurrence in a page layout. Essentially the site layout tag entity schema (as shown in Figure 5) controls the Web site structure. It will consist of two entity types. The Web page identification information will be represented through the "SiteName" entity type, while the associative (weak) entity type "PageDetails" represents the sequence of individual or group tag entities as they appear in the layout of the Web page, along with the database relationships with such entity types.

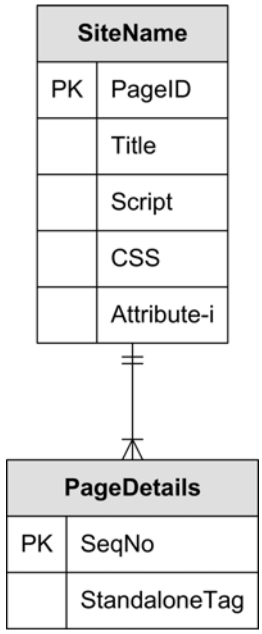

Figure 5. Site Layout Entity Representation

In Figure 5, PageID attibute represents the identifier for each individual Web page. The Title attribute represents the value of the title tag content. The Script attribute represents the value of any script details. The CSS attribute represents the cascading style sheet information. Attribute-i represents other attributes that may be necessary to complete the Web page <head> tag contents. The PageDetails entity type will have SeqNo attribute that sequences the individual or group tag entities as they appear in the layout of the Web page, while the Standalone Tag attribute represent any standalone tag that may also appear in the layout. The PageDetails entity type will have database relationship with other individual or group tag entities. For instance, the Web page schema of Figure 4 can be extended into a complete Web page as shown below. The corresponding entity relationship schema is shown in Figure 6, followed by additional relational database tables. In the database tables, primary keys are underlined, and foreign keys are in italics.

```
<html>
<head><title>Sample Page</title>
</head>
<body>
<div    align="center"><p>Web    Design
ER</p>
<a     href="sample_page.html">Home</a>
<br><hr /></div>
</body>
</html>
```

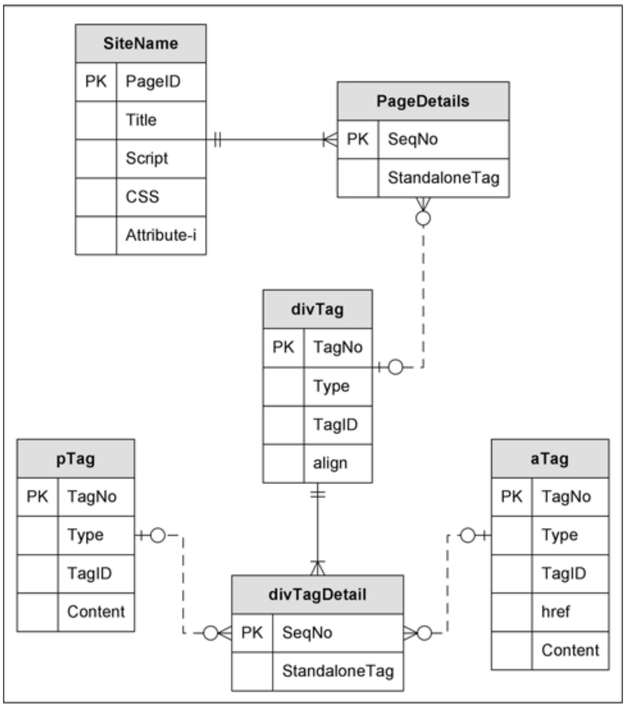

Figure 6. Site Layout Entity Example

TABLE V.
SITENAME TABLE

PageID	Title	CSS	Link	Script	Meta
1	Sample Page				

TABLE VI.
PAGEDETAILS TABLE

PageID	SeqNo	*divTag*
1	1	1

Since the tag details are stored as database tables they can be shared across Web pages. In other words, different PageID values can avail of similar tag row. From a maintenance standpoint, once the tag table contents are updated, the corresponding Web pages are also updated.

III. GENERATE DATABASE CONTENT AND WEB PAGE

The Web page for display in the Web browser can be generated from within the database through a database procedure or some other programming language interface. During generation of Web page, the SiteName and PageDetails entity contents will provide the sequencing of the layout.

As the Web page and Web site structure are represented through database tables, it is possible to embed the contents of the Web pages with database values. Unlike the traditional approach of merging server-side scripting languages with HTML to generate database driven Web pages, the content attributes of the respective HTML tag entities can be directly updated with database values, prior to page layout composition. For instance, as shown in Figure 7, database procedures in Oracle can update the content attributes of the HTML tag entities prior to the structuring of the Web page layout as specified in the PageName and PageDetails entity types.

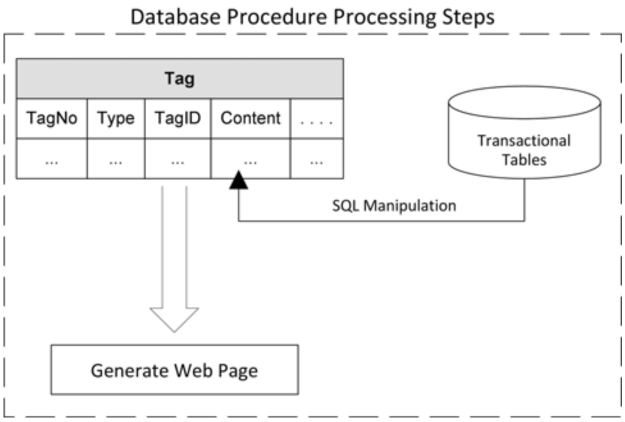

Figure 7. Schematic view of Web Page Generation

IV. WEB SITE MODELING PROTOTYPE

To illustrate the modeling concepts with respect to Web site design, a sample Web site with two Web pages is outlined now followed by their composition through Oracle database procedures. The prototype was installed in Oracle DBMS and implemented through Oracle's PL/SQL language.

Web Page - 1:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01 Transitional//EN">
<html>
<head><title>Sample Page</title>
</head>
<body>
 <div align="center">
 <p>Web Design ER</p>
 <a href="sample_page.html">Home</a>
 <br>
 <hr>
 </div>
 <br>
 <p>This page shows items
available.</p>
 <ul>
 <li>iPhone5</li>
 <li>Lumia928</li>
 <li>Galaxy S4</li>
 </ul>
</body>
</html>
```

Web Page - 2:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
HTML 4.01 Transitional//EN">
<html><head>
<meta http-equiv="content-type"
content="text/html; charset=windows-
```

```
1252"><title>Create Sales
Order</title>
</head>
<body>
<p>Enter Order Details</p>
<br>
<form action="http://www.msu.edu"
method="post" name="form1">
Customer No: <input name="custno_text"
type="text"><br><br>
First Name: <input
name="first_name_text"
type="text"><br><br>
Last Name: <input
name="last_name_text"
type="text"><br><br>
City: <input name="city_text"
value="Springfield" checked="checked"
type="radio">Springfield
<input name="city_text" value="Kansas
City" type="radio">Kansas City
<input name="city_text" value="St.
Louis" type="radio">St. Louis
<br><br>State: <select
name="state_text"><option value="MO"
selected="selected">Missouri</option>
      <option
value="KS">Kansas</option>
      <option
value="IL">Illinois</option>
      </select>
  <br><br><input name="formsbutton1"
value="submit" type="submit">
  <input name="formsbutton2"
value="reset" type="reset">
  </form><br><br>
</body>
</html>
```

The entity-relationship diagram of the Web pages is shown in Figure 8, followed by the database tables representing the ER model. In the Figure, the HomePage entity type represents the Web site. It will have the identification of the two Web pages as reflected in the corresponding tables. The PageDetails entity type will provide the sequence in which the tag entities appear in each of the two Web pages. The PageDetails entity type table consequently contains the foreign key entries of the other related tag entities. In the data model, the FormTag, FormTagDetails, and FormSelectDetails entity types represent nested group tag entities. In the database tables, primary keys are underlined, and foreign keys are in italics.

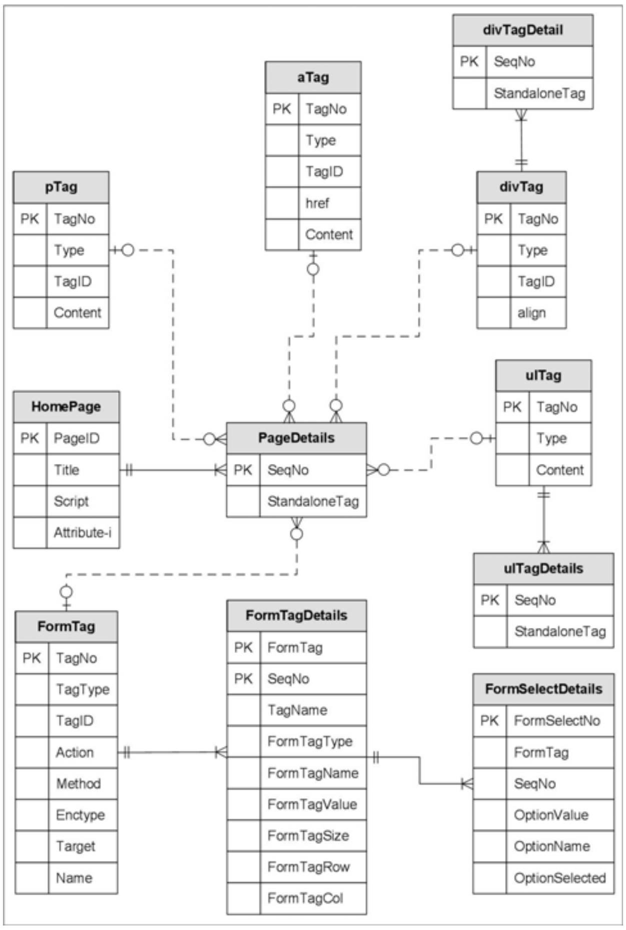

Figure 8. Prototype Web Pages ER Model

TABLE VII.
PTAG TABLE (PART 1)

TagNo	TagType	TagID	CSS
1	p		
2	p		
3	p		

TABLE VII.
PTAG TABLE (PART 2)

TagNo	Content
1	Web Design ER
2	This page shows items available.
3	Enter Order Details

TABLE VIII.
ULTAG TABLE

TagNo	TagType	TagID	Content
1	ul		

TABLE IX.
ULTAGDETAILS TABLE

ulTag	SeqNo	TagType	TagID	Content

TABLE X.
ATAG TABLE

TagNo	TagType	TagID	Content	href
1	a		Home	sample_page.html

TABLE XI.
DIVTAG TABLE

TagNo	TagType	TagID	Content	Dalign
1	div			center

TABLE XII.
DIVTAGDETAILS TABLE

divTag	SeqNo	*pTag*	*aTag*	StandaloneTags
1	1	1		
1	2		1	
1	3			br
1	4			hr

TABLE XIII.
FORMTAG TABLE (PART 1)

Tagno	Tagtype	Tagid	Action
1	form		http://www.msu.edu

TABLE XIII.
FORMTAG TABLE (PART 2)

Tagno	Method	Enctype	Target	Name
1	post			form1

TABLE XIV.
FORMTAGDETAILS TABLE (PART 1)

Formtag	Seqno	Label	Tagname
1	1	Customer No:	input
1	2	First Name:	input
1	3	Last Name:	input
1	4	City:	input
1	5	State:	select
1	6		submit
1	7		reset

TABLE XIV.
FORMTAGDETAILS TABLE (PART 2)

Formtag	Seqno	Formtagtype	Formtagname
1	1	text	custno_text
1	2	text	first_name_text
1	3	text	last_name_text
1	4	radio	city_text
1	5		state_text
1	6		formsbutton1
1	7		formsbutton2

TABLE XIV.
FORMTAGDETAILS TABLE (PART 3)

Formtag	Seqno	Formtagvalue	Formtagsize
1	1		
1	2		
1	3		
1	4		
1	5		
1	6	submit	
1	7	reset	

TABLE XIV.
FORMTAGDETAILS TABLE (PART 4)

Formtag	Seqno	Formtagrow	Formtagcol
1	1		
1	2		
1	3		
1	4		
1	5		
1	6		
1	7		

TABLE XV.
FORMSELECTDETAILS TABLE (PART 1)

Formselectno	*Formtag*	*Seqno*	Optionvalue
1	1	4	Springfield
2	1	4	Kansas City
3	1	4	St. Louis
4	1	5	MO
5	1	5	KS
6	1	5	IL

TABLE XV.
FORMSELECTDETAILS TABLE (PART 2)

Formselectno	*Formtag*	*Seqno*	Optionname	Optionselected
1	1	4	Springfield	Y
2	1	4	Kansas City	N
3	1	4	St. Louis	N
4	1	5	Missouri	Y
5	1	5	Kansas	N
6	1	5	Illinois	N

TABLE XVI.
HOMEPAGE TABLE

PageID	Title	CSS	Link	Script	Meta
1	Items List				
2	Create Sales Order				

TABLE XVII.
PAGEDETAILS TABLE (PART 1)

PageID	SeqNo	*divDetails*	*ulDetails*
1	1	1	
1	2		
1	3		
1	4		1
2	5		
2	6		
2	7		

TABLE XVII.
PAGEDETAILS TABLE (PART 2)

PageID	SeqNo	*pTag*	StandaloneTags	*FormTag*
1	1			
1	2		br	
1	3	2		
1	4			
2	5	3		
2	6		br	
2	7			1

A. Compose Web Page

The Web pages to be displayed in a Web browser are developed through a collection of Oracle database procedures in PL/SQL language through the Oracle's hpt.prn package. Each tag entity structure is composed in its HTML representation through a separate procedure. The logic of Web page composition is shown in Figure 9.

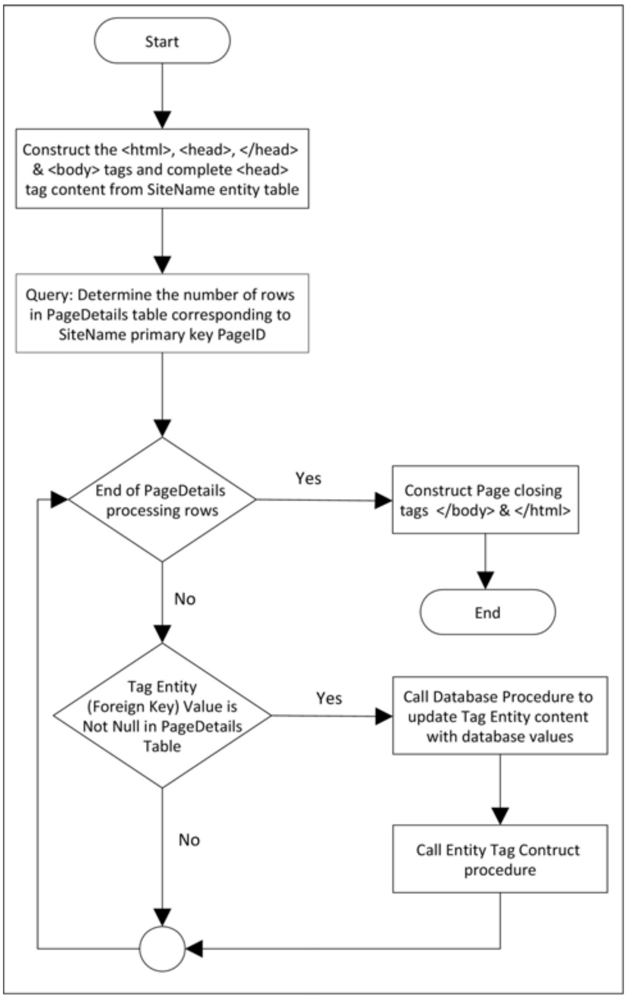

Figure 9. Logic of Web Page Composition

The Web pages utilize a sample transactional entity schema as shown in Figure 10 to illustrate the embedding of database values for tag content. In the prototype, the contents of Web Page -1 within the <ul> tags are derived from the database. Consequently, the ulTagDetails table initially has no data. During the composition of the Web page, the database procedure listed in Appendix-A initially calls another database procedure ul_fill to perform the database SQL operation of inserting values in the ulTagDetails table, and then constructs the <ul> tag for display in Web browser through another procedure ul_tag_expand. Appendix-B shows one of the entity tag construct procedure.

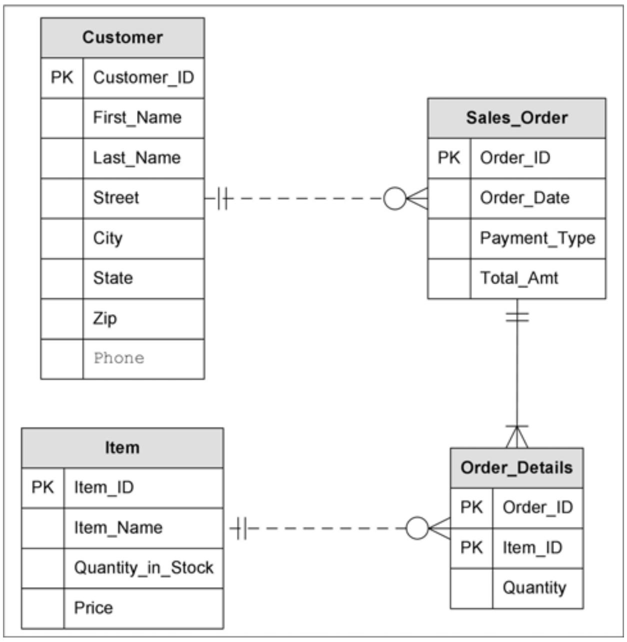

Figure 10. Transactional Database Schema

To generate Web pages, the procedure home_page in Appendix-A is run in Oracle SQL Developer. The procedure will create the two PL/SQL Web procedures pertaining to the two Web pages (Web Page -1 and Web Page - 2), which can then be viewed in a Web browser. A display of the two Web pages in the Web browser is shown in Figure 11 and Figure 12.

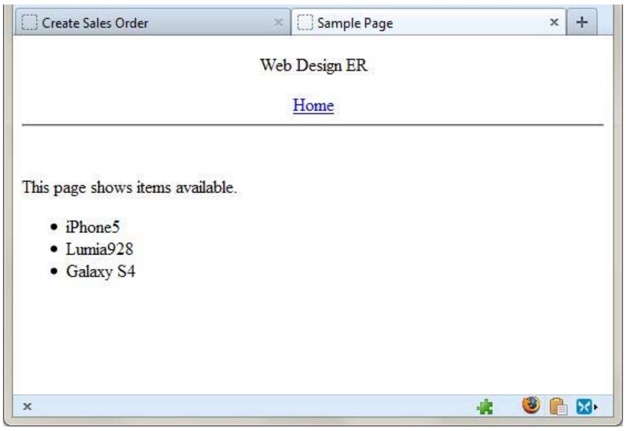

Figure 11. Prototype of Web Page – 1 in Web Browser

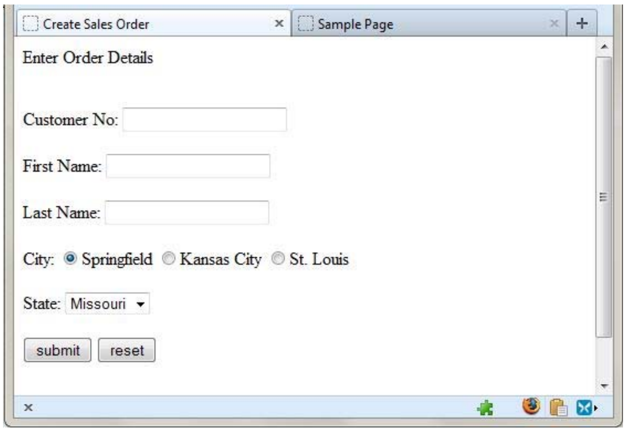

Figure 12. Prototype of Web Page – 2 in Web Browser

V. Conclusion

ER modeling the Web site and Web page design for storage in relational database provides a mechanism to share Web page content or a HTML tag by other Web pages. As the Web site structure is modeled as a relational database, it can be queried and maintained through the industry standard SQL language. The use of relational database features provide for a standardized way to create Web pages, even though the Web pages can also be assembled through any programming or database language.

Further research is ongoing to enhance the data model by including other capabilities. Such capabilities extend the approach with AJAX and CSS features.

Appendix A Web Procedure To Compose Web Page

```
create or replace procedure home_page as
opentag char(1) := '<';
closetag char(1) := '>';
endtag char(1) := '/';
d_title varchar2(512);
d_script varchar2(512);
d_meta varchar2(512);

begin
select title,script,meta
into d_title,d_script,d_meta
from homepage
where pageid=1;
htp.prn('<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01 Transitional//EN">
');
htp.prn('<html>
<head>');
if d_title is not null then
htp.prn('<title>'); htp.prn(d_title);
htp.prn('</title>
');
end if;
if d_script is not null then
htp.prn('<script>'); htp.prn(d_script);
htp.prn('</script>
');
end if;
if d_meta is not null then
```

```
htp.prn('<meta>'); htp.prn(d_meta);
htp.prn('</meta>
');
end if;
htp.prn('</head>
<body>');
 htp.prn('
 ');

for h_row in (select
divTag,ulTag,pTag,StandaloneTags
from pagedetails
where pageid = 1)
loop
if h_row.divTag is not null then
 div_tag_expand(h_row.divTag);
end if;
if h_row.standaloneTags is not null then
 htp.prn(openTag);
htp.prn(h_row.standaloneTags);
htp.prn(closeTag);
 htp.prn('
 ');
end if;
if h_row.ultag is not null then
 ul_fill(h_row.ultag);
 ul_tag_expand(h_row.ultag);
end if;
if h_row.pTag is not null then
 p_tag_expand(h_row.pTag);
end if;
end loop;
htp.prn('
</body>
</html>');
end;
```

APPENDIX B ENTITY TAG CONSTRUCT PROCEDURE

```
create or replace procedure
p_tag_expand(pNo integer) as
opentag char(1) := '<'; closetag char(1) :=
'>'; endtag char(1) := '/';
p_tagtype pTag.tagtype%type;
p_tagID pTag.tagID%type;
p_content pTag.content%type;
begin
select tagType,tagID,content
into p_tagtype,p_tagID,p_content
from pTag
where TagNo = pNo;
htp.prn(openTag); htp.prn(p_TagType);
if p_TagID is not null then
  htp.prn(' ID="'); htp.prn(p_TagID);
htp.prn('"');

end if;
htp.prn(closeTag);
if p_content is not null then
  htp.prn(p_content);
end if;
htp.prn(openTag); htp.prn(endTag);
htp.prn(p_TagType); htp.prn(closeTag);
htp.prn(' ');
end;
```

REFERENCES

[1] B. Biswas,, K. Jain,, V. Mittal, and K.K. Shukla, “Exploiting tree structure of a web page for clustering,” *International Journal of Knowledge and Web Intelligence,* vol. 1, pp. 81-94, 2009.

[2] P. Bohunsky, “Visual structure-based web page clustering and retrieval,” in *Proceedings of the 19th international conference on World wide web*, Raleigh, NC, pp. 1067-1068, 2010.

[3] S. Ceri, P. Fraternali, and A. Bongio, “Web Modeling Language (WebML): a modeling language for designing Web sites,” *Computer Networks*, vol. 33, pp. 137-157, 2000.

[4] J. Conallen, “Modeling Web application architectures with UML,” *Communications of the ACM* vol. 42, pp. 63-70, 1999.

[5] C. Date, *Introduction to Database Systems (8th ed.),* Readings, MA: Addison-Wesley, 2003.

[6] M. Hall, L. Brown, and Y. Chaikin, *Core Servlets and JavaServer Pages*, Upper Saddle River, NJ: Prentice-Hall, 2008.

[7] R. Kaula, *Oracle 11g: Developing AJAX Applications with PL/SQL Server Pages*, New York, NY: McGraw-Hill, 2008.

[8] J. Lin, M.W. Newman, J.I. Hong, and J.A. Landay, “DENIM: finding a tighter fit between tools and practice for Web site design,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, The Hague, Netherlands, pp. 510-517, 2000.

[9] M.V. Mannino, *Database Design, Application Development, and Administration* (3rd ed.), New York, NY: McGraw-Hill, 2006.

[10] M.W. Newman and J.A. Landay, “Sitemaps, storyboards, and specifications: a sketch of Web site design practice,” in *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, Brooklyn, NY, pp. 263 - 274, 2000.

[11] J. Nielsen, *Designing Web Usability: The Practice of Simplicity*, Thousand Oaks, CA: New Riders Publishing, 1999.

[12] A. Sanyal, S. Sanyal, and S. Choudhury, “Physical level implementation of a web data model,” in *Proceedings of the 2011 International Conference on Communication, Computing & Security*, ODISHA, India, pp. 483-488, 2011.

[13] A. Sanyal and S. Choudhury, “An Object Oriented Conceptual Level Design for Web Data Model,” in *Proceedings of the International Conference on Methods and Models in Computer Science*, New Delhi, India, pp. 1-6, 2009.

[14] A. Sanyal, A. Sarkar, and S. Choudhury, “Automating Web Data Model: Conceptual Design to Logical Representation,” in *Proceedings of the 19th International Conference on Software Engineering and Data Engineering*, Bangalore, India, pp. 94-99, 2010.

[15] M.A. Shah and S.M. Deshpande, “Web Page Classification Based on Document Structure without Negative Examples,” *International Journal Of Computer Science And Applications,* vol. 1, pp. 31-35, 2008.

[16] J. Sklar, *Principles of web design*, Boston, MA: Cengage Learning, 2008.

[17] S. Walther, K.S. Hoffman, and N.S. Dudek, *ASP.NET 4 Unleashed*, Upper Saddle River, NJ: Pearson Education, 2011.

[18] Jiashu Xu, “A Dynamic Web Page Modeling Method Based on UML,” in *Proceedings of the International*

Conference on Computing, Measurement, Control and Sensor Network (CMCSN), Taiyuan, China, pp. 324-327, July 2012.

Rajeev Kaula is a professor in computer information systems at Missouri State University at Springfield, MO (USA). Besides teaching database and related topics for the past several years, his areas of research include data modeling, data warehouse, business rules, business intelligence, open information systems, and computer integrated manufacturing. Dr. Kaula has published research papers in many journals and has written books on database design and application, open information systems and Oracle database. Dr. Kaula was also associated with the launch of Oracle Express Edition (XE) DBMS.