IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4110

Applying JAVA-Triggers for X-Link Management in the Industrial Framework

Abraham Alvarez and Youssef Amghar
Laboratoire d'Ingénierie des Systèmes d'Information, INSA de Lyon, France, 7 avenue Jean Capelle, F-69621, Villeurbanne
Tel: +33 4 72 43 85 88, Fax: +33 4 72 43 87 13, aalvarez@lisi.insa-lyon.fr, amghar@if.insa-lyon.fr

ABSTRACT

In the industrial context, the life cycle of a document plays a central role describing the "steps out" of a product. structured or semi-structured, these documents are related by different kinds of links. Users realize some manipulations like: creation, edition, suppression and querying under a multi-user environment, risking possible destruction or the alteration of the document's integrity . A classical impact is the infamous "Error 404 file not found". However, the user needs a notification Alert Mechanism to prevent and warrant the coherence of manipulations over all the life cycle process of a product. The main objective of this paper is to provide a generic relationship validation mechanism to remedy this shortcoming. The paper focuses in referential link integrity aspects. We believe that by the combination of some standard features of XML specifically XLL specification as a support for integrity management and JAVA-Triggers approach as an alert method.

INTRODUCTION

In this study, the coherence integrity problem means that a link depends on the "state" of the referenced document. When documents are edited or revised, locations can probably be altered , with the potential risk of destroying the record of a linked text. The main purpose of this paper is to provide a generic relationship validation mechanism to keep the links references in a "coherent state" over all process involved. Therefore, some standard features of XML and highlighted in the XLL specifications (x-link & x-pointer) presented by XML-Linking working group, will be considered in this work as a support for integrity management and JAVA-Triggers approach as, an alert method.

Following this introduction, the related work section specially analyzes the different approaches for managing link integrity, in addition, presents briefly a survey for the similar application domains as: hypermedia, hypertext systems and digital libraries referring to link integrity problem. The motivation and background of this study are explained in section 3. Section 4 illustrates and examines with an example the validation process. Section 5 concerns the algorithm and the application presentation. Finally, a conclusion and future perspectives are discussed.

RELATED WORK

In the last few years, a lot of time has been dedicated to the problem of maintaining links. Different approaches for managing link integrity are categorized and exposed in this section. Two main solution categories are discussed concerning the referential integrity problem: the preventive and corrective solutions. In the recent surveys Ashman [Ashman 19999] and [Asham 2000] present some preventive solutions: Forbidding changes, this strategy, the modification or deletion of documents is not authorized. Another preventive solution is proposed by Davis [Davis 1999] introducing a concept called "publishing" solution, which consists in preventing the content reference problem prohibiting users from updating documents once they have pointers into them.

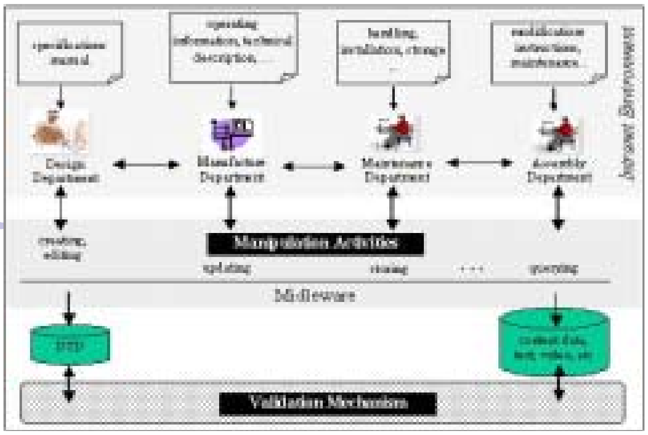
In the context of link attributes [Oinas-Kukkonen 1998] proposes to integrate link attributes like semantic link types and link keywords. Link attributes properties associated with links provide knowledge about interrelationships between pieces of information. This preserves the context of information units and increases the local coherence of an information collection link; attributes also provide the user a way to know or preview the target before activating the link, [Thüring and al. 1995].

Finally, DeRose presents the most up-to-date overview of the Xlink, Xpointer, and Xpath recommendations, explaining important features of hypertext systems not seen previously, e.g. one-to-many links, out-of-line or externalized links, and calculations of links positions [DeRose 2001]. A literature survey related to links history is presented by Lewis and al, in [Lewis 1999] the authors make a distinction between navigation and retrieval information handling. Nowadays, in the area of hypermedia research systems, we have found powerful systems MicroCosm [Hall 1993], Hyperware [Maurer 1996], linking Systems Dexter [Halasz 1990], [DeBra 1999], Intermedia [Haan and al. 1992]. However, still constrained by the limitations imposed by HTML. The OpenDoc architecture[1] [Killpack 1994] is an answer to satisfy the interoperability problems in hypermedia system.

MOTIVATION

In the industrial context, the on-line technical documentation is a support describing and defining department activities overall manufacturing process, executed by several departments such as designing, manufacturing, maintaining and assembling areas. Considering this context and using Intranet technology to deliver, share and connect documents to other departments, any change in the on-line documentation can impact over other department activities, notably in the whole cycle of a product's development. Under the previous context, the main motivation of this study is to provide a generic relationship mechanism to maintain the integrity of links. The technical documentation projects in organization require coherent manipulations on the overall process involved. The Figure 1 depicts this scenario.

Figure 1: Major phases of product manufacturing



This paper appears in *Issues and Trends of Information Technology Management in Contemporary Organizations,* the proceedings of the Information Resources Management Association International Conference. Copyright © 2002, Idea Group Inc.

Technical Documentation

Definition: technical documentation is the creation, control, delivery, and maintenance of distributed information across the extended enterprise and a network that includes sources and users. The technical documentation is not a recent concept in the Intranet environment [Albin 1996]. The on-line technical documentation serves as an interface between the user and the product.

World Wide Web, Intranet, hypertext links and hypermedia systems, are key concepts in the way we read, write and access information and have created a radical way of reading and writing office documents.

Using Intranet technology, document workflow relates to the product's life cycle. The set of relevant information for a particular task or a set of related tasks during the life cycle of the product. e.g. manual specifications for a design department, modification instructions for maintenance department and so on.

- Users who create, use and maintain the information set are working under Intranet environment. User manipulations are concurrent.
- All documents are interrelated by links over all process
- Changes can impact notably in other department activities.

Background of Active Database and Triggers Approach

This section shows the basic knowledge of Active Databases and Triggers approach. An important functionality supported by many of the most recent relational and object-relational database systems is represented by triggers, that enhance the database with reactive capabilities and which can be used to support integrity constraints i.e. to maintain the integrity of the data, and maintaining long-running transactions. Event-Condition-Action (ECA) formalism, where the Event specifies when a rule should be triggered, the Condition is a query that is evaluated when the Event occurs, and the Action is executed when the Event occurs and the condition is satisfied.

The notion of trigger appeared in the seventies, and has been generalized with the notion of active rule that is based on the Event-Condition-Action (ECA) formalism. *The semantic of an ECA rule is noted as follows: when an event E is produced, if the condition C is satisfied, then the action A is executed. Actions are initiated by the DBMS when appropriate events occur, independently of external requests.* These rules allow database designers to specify the active behavior of a database application that provides the enforcement of database integrity. In the literature, several approaches were proposed to integrate active concepts into databases.

Several commercial database management systems (DBMS) include event/trigger mechanism that has been proposed by [Kotz 1988], such as the Postgres rule system [Stonebraker90], Starburst's production and alert rules [Lohman 1991], Ariel's production rule system [Hanson 1989], the ECA model of HiPAC [Dayal 1988], and the event-action EA model of Ode [Gehani 1992]. Few researches have addressed the conceptual specification of behavioral aspects of applications independently from any active DBMS. To our knowledge, only IDEA [Ceri 1993] and SORAC [Peckham 1995] projects have treated the design of active database. However IDEA methodology is strongly linked to the Chimera DBMS. The SORAC model permits the schema designer to specify enforcement rules that maintain constraints on object and relationships to facilitate the task of the designer.

VALIDATION PROCESS OF XML DOCUMENTS

As a part of the preventive solution category, this study is notably related in the works of [Thüring and al. 1995], [Davis 1999], [Oinas-Kukkonen 1998]. We believe that these preventive techniques are an answer to link integrity problem but not a full solution. However, these works focus only on representation aspects. Our contribution proposes both coherence representation point of view and algorithmic points of view. This work allows the development of a new fundamental to enhance midleware of document.

XML Specification

XLL allows XML tags to refer to other XML or HTML documents, as well as specific elements or data within those documents. In addition, X-link specifies a set of attributes in X-link namespace, one can add element tag references to other documents, while x-pointer specifies a syntax to select specific elements within a document.

The mechanism to maintain the link integrity is described in this section. After the hypertext document creation, the next important steps are: validating the document by a classical parser for xml-documents, making a recognition of links and finally, the referential links validation.

Step 1 - Parser validation. This step consists in verifying the syntax and structure of documents before applying the process of validation (step 3), we are using an evaluation copy of XML Spy 3.5 developed by IBM.[2] Two kinds of parsers are available for the validation of documents: a "non-validating processor" that makes sure that xml documents are "well-formed-syntactically" correct. The second, is a "validating processor" that checks for much more, particularly the document structure against a set of declarations to be sure that the structure contains all the parts required by these declarations [Laurent99]. Unfortunately, both parsers are far less performing when it comes to checking the relationship coherence of structured documents. The importance of this step is to reduce potential syntax errors and check the document structure, for a more powerful analysis to the next step.

Step 2 – Links recognition. In order to keep the reliability recognition, let us consider only "well-formed-syntactically" documents (step 1). The next step is to detect the tags between the documents, this means, to identify the type of link. The simple and extended links are the classical links embeded in most of the structured documents. The simple link is used to "surf" or to navigate between the documents (transversal notion), this link is unidirectional. Extended link is multi-directional, allowing a multi-directional navigation and reference. The other types of links are not considered for the moment.

Step 3 – Referential links validation. The existence of a link is asserted by a linking element. Linking elements are reliably recognized by software applications. When the user inputs the URL or file location, the application searches within documents all occurrences of type: "A HREF=", "xmlns:xlink=", "<link target=", "<IMG SRC=", "<ID Ref=", etc.

We begin the validation link process "if the target link exists": validating the resources (start-point) to targets (end-point) for each type of link, i.e. the links relationships in the same document, in other documents. This step uses Java application displaying the valid and wrong links. i.e. Find out, if the links are valid or no longer valid.

To have a general panorama of x-link properties, a summary is considered. In this way, XLL language has two types of links: internal and external links, internal links are "on-line" links (within the document), an advantage lies in the use of X-pointer, this means that we can point any portion of the document and identify the end-points of the link, that work on the tree nodes, (child, descendant, antecessor,etc) e.g. HREF=http://lisi.insa-lyon.fr/~aalvarez/ #id(publications).child (3,item). External links are "out-of-line" (stored in a database).

An Example

From now on, we will focus on describing the example. A small technical manual was chosen to illustrate the repercussion of any change during the life cycle of a product. The main purpose of producing the on-line technical documentation is to provide an easy way to find and up-to-date the information that they needed.

In this example, three types of links considered in this paper: composition, sharing and referential links respectively.

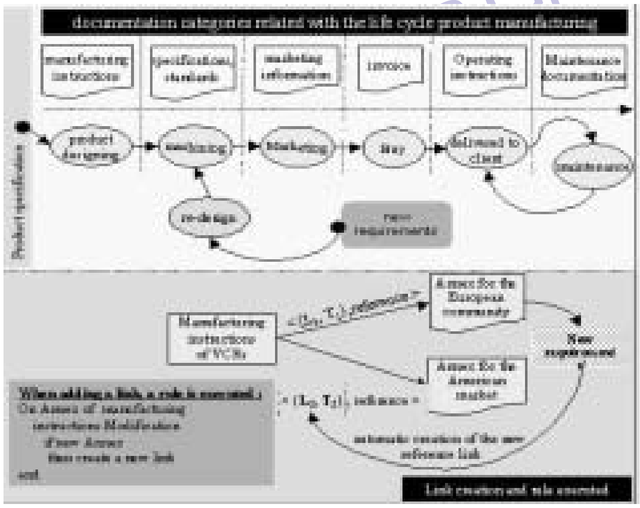
Composition links: they represent the composition of a document structure by elements called nodes. The source corresponds to root node, the target being a component of the source. Each anchor of link is both source and target. This link is bi-directional. Moreover, they also show their hierarchical organization in order to retrieve the document structure. In the figure below, Manual represents the document type and the root element of the document: an article is composed by child elements called nodes, in this case, a manual is composed by a {ManualTitle} containing "Operating Information Manual", a {Body element} and afterwards a {Reference element}. {Section I},{Section II} are children nodes of {Body element} and so on. Each anchor of the link is both source and target. This link is bi-directional and implicit. Moreover, shows the hierarchical organization generally in order to retrieve the document structure.

Reference links: they allow to establish a reference between documents and documents fragments. These kind of links are unidirectional. They are often represented by known expressions such as "see section 1", "for more information", etc. *Information sharing link*: allows to point the same information between 2 or more fragments. This link is unidirectional and explicit.

Considering an European manufacturing enterprise of (VCRs) Video Cassette Recorders, its products are manufactured with European electric characteristics i.e. 220volts & 60Hz frequency. All the VCRs operating manuals have a similar structure: a manual title, a body and its sections. Section I contains the Introduction and General Information, Section II Basic Operations, Section III corresponds to Technical Specifications, and so on.

A new project of expansion focuses on the North American area. Considering this expectation and the difference of American electric characteristics, i.e. 110volts & 50Hz. The company needs to think about the repercussions implied by this change in the related technical documentation across the product life cycle. Figure 2 describes the processes of a product life cycle and the creation of a new link , when a new requirement is claimed. If we consider that case, i.e. the quality service of documentation requires the addition of a new annex for previous manufacturing instructions manual.

Figure 2: Product life cycle related with document generation


documentation categories related with the life cycle product manufacturing
manufacturing instructions
specifications, standards
marketing information
invoice
Operating instructions
Maintenance documentation
Product specification
product designing
machining
Marketing
Buy
delivered to client
maintenance
re-design
new requirements
Manufacturing instructions of VCRs
<(L1, T1)> reference =
Annex for the European community
Annex for the American market
New requirement
<(L2, T2)> reference =
When adding a link, a rule is executed :
On Annex of manufacturing instructions Modification
if new Annex
then create a new link
end
automatic creation of the new reference link
Link creation and rule executed

This annex shall contain the new product specifications concerning the electric transformation to involve the North American sector. This new change on the technical annex has to be generated by the creation of a new "reference link" i.e. The original annex makes reference to a new technical annex. When the user modifies the original annex, the validation mechanism has to be executed by the system automatically creating the new reference link. Regarding the figure below, a rule has been executed to create the link. Now, the challenge is to keep the link references in a "coherent state" before and after manipulations; in addition, when documents are edited or revised, locations change, with the risk of potentially destroying a record of a linked text. In this level, our mechanism verifies if all links are valid or no longer valid.

Algorithm

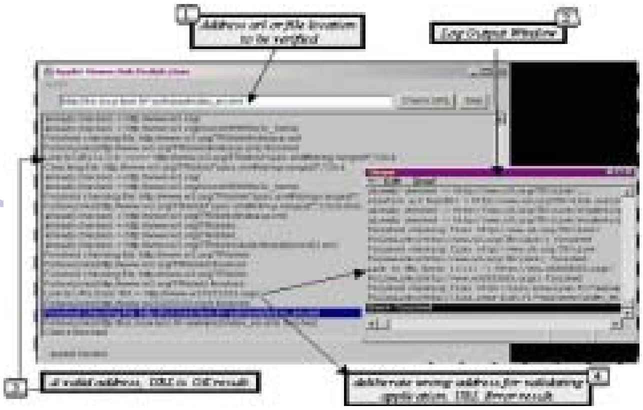
The algorithm starts from an initial pass through an input url-location or a file location (establish a connection to the specified URL, i.e. var stringURL) in order to search any text chains containing the referential tags as "<A HREF=","XLINK:HREF"," <LINK TARGET=","<IMG SRC=","<ID REF=" into document in a recursive way. The verified links are stored in a table (Hastable class provided by Java.util;).

```
   LinkValidation
   Begin
    /* Variables definition */
    String stringURL;
    int i=1, line=1, lineNo=1;
    String[]linkStrings={"<A
    HREF=","XLINK:HREF","<LINK TARGET= ","
<IMG SRC=","<ID REF=");
   /* Getting url-location */
    void clicked() {
    if ( stringURL == null ){
   showStatus("URL ENTRY IS EMPTY !");
      return;}
    . . .
    startIndex=ret+1;
   lineNo++;}
   void CompareTo
   for (int int i = 0; i < hash.size(); i++)
{URL storedU = (URL)en.nextElement();
   if (u.sameFile(storedU) == true )
   {s = new String ("already checked=>"+u );
    return}}
   end
```

Application

The next figure depicts the parts of running application. The input field takes the url or file location to be verified i.e. URL: http://lisi.insa-lyon.fr/~aalvarez/document_a. xml. (1); The Output window shows the status of each link and the links contained in the same document (2), when the link address is valid, an Ok message is displayed.

Figure 3: Running application


1
Address url or file location to be verified
2
Log Output Window
3
A valid address, URL is OK result
4
deliberate wrong address for validating application, URL Error result

CONCLUSION AND FURTHER WORK

We have locked at the cost-benefits using Intranet Technology to distribute electronics documents during life cycle product process provided by hypertext systems. The links role is multiple (structural, navigational, spatial, etc). A reliable documentation is a significant factor in a product's success, since users are often learning how to use and to maintain the product, its manufacturer and its supplier. We believe that companies need to document manufacturing process, the notification systems need to keep production employees up-to-date on procedural changes and approval workflows necessary for ensuring a successful technical documentation. This success depends on being able to maintain referential integrity, and hence link integrity when the documents to which links refer are altered in some way.

Link integrity and Trigger concepts are the issue considered in this paper. We have shown the different approaches to avoid the link reference problem. Two categories of solutions (the preventive and corrective) were discussed. The actual solutions are concentrated in representation aspects and are not full solutions. We have proposed a preventive mechanism to maintain the link reference integrity supported by a strong algorithm, moreover considering some XLL specifications from Xml-Linking workgroup. Further research work should contain at least two future directions: management of document and fragments versions. Another direcendnotestion is the DTD translation to schema XML.

ENDNOTES

1 OpenDoc is a registered trademark of Apple Computer, Inc

2 XML Spy 3.5 is an Integrated Development Environment (IDE) for the eXtensible Markup Language (XML). http://www.xmlspy.com

REFERENCES

Albing Bill. 1996. "Process Constraints in the Management of Technical Documentation", SIGDOC'96.

Ashman Helen and Rosemary Michelle S. 1999. "Computing Survey's Electronic Symposium on Hypertext and Hypermedia: Editorial." *ACM Computing Surveys,* Vol. 31.

Ashman Helen. 2000. "Electronic Document Addressing: Dealing with Change", ACM *Computing Surveys,* Vol. 32.

Ceri, S. & Manthey, R. 1993. "First Specification of Chimera", IDEA DD.2P.004.01.

Dayal, U. et al. 1988. "The HiPAC Project : Combining Active Databases and Timing Constraints", *SIGMOD Record*, pp. 51-70,Vol. 17, N° 1.

DeBra, Paul, Geert-Jan Houben and Hongjing Wu, "AHAM: A Dexter-based Reference Model for Adaptative Hypermedia". 1999. *In Proceedings of the 10th ACM Conference on Hypertext and Hypermedia*: Returning to Our Diverse Roots, Darmstdt, Germany. Feb: pp.147-156.

DeRose J Steve, Eve Maler, David Orchard and Jamcracker. 2001. « XML Linking Language (XLink) » Version 1.0, W3C Recommendation 27 June 2001. [Online:http://www.w3.org/TR/2001/REC-xlink-20010627].

Gehani, N.H. et al. 1992. "Composite Event Specification in Active Databases : Model & Implementation", *Proceedings of the 18th VLDV Conference*, Vancouver, Canada.

H Paul. Lewis, Wendy Hall, Leslie A. Carr, and David de Roure. 1999. "The significance of linking", *ACM Computing Surveys*, Vol. 31, Number 4es, December.

H. Maurer. Hyperware, 1996. *The Nexy Generation Web Solution*, Addison- Wesley, Reading, Massachusetts.

Halasz Frank G. and Mayer D. Schwartz. 1990."The Dexter Hypertext Reference Model" in Proceedings of the Hypertext Standarization Workshop by the National Institute of Science and Technology, *Communications of the ACM*, Vol. 37, N° 2, Feb: pp. 30-39.

Hall Wendy, Gary Hill and Hugh Davis, 1993. "The Microcosm Link Service", Technical Briefing. *In Proceedings of the ACM Hypertext '93*, November:pp. 256– 259, Seattle, Washington, USA.

Hanson, E.N. "An Initial Report on the Design of Ariel : A DBMS with and Integrated Production Rule System", 1989. *SIGMOD Record*, pp. 12-19, Vol, 18, N° 3.

http://www.w3.org/XML/

Hugh C. Davis. 1999. "Hypertext Link Integrity", *ACM Computing Surveys*, Vol. 31, Number 4es.

Kotz, A.M., & Dittrich, K.R., & Mulle J.A. 1988. "Supporting Semantic Rules by a Generalized Event/Trigger Mechanism", *Advances in Databases Technology, EDBT'88*, pp. 76 –91, Vol. 303, Venice, Italy.

Laurent Simon St and Biggar R., 1999.*Inside XML DTDs*, McGraw-Hill Editor.

Lohman, G.M. et al. 1991. "Extensions of Starburst: Objects ,Types, Functions and Rules". *Communications of the ACM*, pp. 94 – 109, Vol. 34, N° 10.

Oinas-Kukkonen Harri, 1998. "What is Inside a Link?", *Communications of the ACM*. July: pp. 57-66, Vol. 41.

Peckham, J. & MacKeller, B., & Doherty, M. 1995. "Data Model for Extensible Support of Explicit Relationship in Design Databases", *VLDB Journal*, pp. 157 191, Vol. 4.

Robbi Bean Killpack and Jaelynn Williams, 1994. "Interoperability: Rethinking The Documentation Paradigm", *ACM, SIGDOC94-10/94,* Alberta, Canada.

Stonebracker, M. et al. 1990. "On Rules, Procedures, Caching and Views in Database Systems", *Proceedings of the ACM SIGMOD,* pp. 281–290, Atlantic City, New Yersey.

Thüring, M., Hannemann, J., and Haake, J.M. 1995. "Designing for comprehension: a cognitive approach to hypermedia development", *Communications of the ACM*. August :pp. 57-66, Vol. 38, No.8.