Software Development Technique for WEB Applications with High Reliability and High Maintainability

Kiyoshi Agusa

Dept. of Information Systems, Graduate School of Information Science,
Nagoya University

agusa@is.nagoya-u.ac.jp

Recently, many web applications have been developed and used in various systems. Web applications consist of server side technologies and client side technologies. Examples of server side technologies are Common Gateway Interface, Server Side Includes, Java Servlet, Java Server Pages, Active Server Pages and so on. Examples of client side technologies are JavaApplet, JavaScript and so on. These technologies interact with each other on web applications. Additionally, web applications include dynamic contents. These characteristics make development difficult and increase the cost of maintenance.

When developing web applications, programmers treat dynamic web content as an extension of static web content. They describe only the relations between each content. However the overall structure and behavior of web applications is not described. Therefore, it is necessary to represent the overall structure and behavior of web applications.

Web applications are commonly built on the J2EE platform based on the MVC (Model View Controller) model. The MVC model separates design concerns to improve the overall software quality. However, it does not explicitly provide the behavioral aspect for the development process of web applications, since the model only defines the static architectural configuration of the components. This results in a broad gap between the description of the web application and its behavioral property.

We propose a method to represent the overall structure and behavior of web applications to improve reliability and maintainability. The structure of a web application is a set of programs and connections between each program through links and actions. The behavior of a web application is a set of sequences of transitions between programs. Web applications consist of programs for pages to display at web browsers and actions on the server side.

The structural model is constructed from the descriptions of programs for web applications. We have already developed a Computer Aided Software Engineering tool platform named Sapid (Sophisticated Application Programming Interface for software Development) for each program. Sapid is based on a fine grained repository. Sapid mainly consists of SDB (Software Database) and AR (Access Routines). SDB is a database that stores structural elements of a source program. The AR is a part of Application Program Interface to accesses SDB. The structural model represents the relations between pages and actions. We use Sapid to obtain these relations.

The behavior of web applications is derived from the structure and specifications of web applications. We treat the behavioral model as an automaton that accepts the sequences of requested URLs to the web application. We propose a testing framework for web applications. We show it provides a testing criteria for web applications when we apply our framework to the Jakarta Struts by using Sapid and the configuration schema of Struts.

We expect that the structural and behavioral models will improve the reliability and maintainability of web applications. The structural model helps programmers develop and maintain web applications. The behavioral model is useful for estimating test coverage. We reveal a method to construct the overall structure from the descriptions of web applications. Additionally, we reveal the detail of the testing technique based on the behavioral model.

Software Development Techniques for WEB Applications with High Reliability and High Maintainability

Background

- Web application
    - is Server-Client system.
    - has Dynamic contents.
    - consists of Many technologies.
- These characteristics
    - make development difficult.
    - increase cost of maintenance.

Web Application

HTML

Servlet   HTML

JSP   Servlet

JSP

Java Script

DB
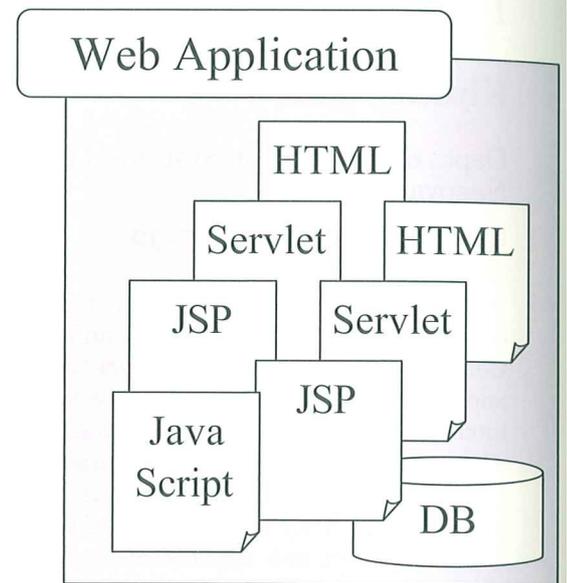
Purpose

We need the whole structure and behavior model of web application for reliability and maintainability.

Structure :
based on Sapid   ⟹   Maintenance

Structure of web application is a set of pages and connections between pages by hyperlink.
We need information about pages and links to construct the whole structure.
We develop models and methods using Sapid to get these information.

Behavior :
based on formal methods   ⟹   Testing framework

Behavior of web application is a set of sequence of transition between pages.
Behavior is derived from structure and specification of web application and represented by web automaton.

Sapid (Sophisticated APIs for CASE tool Development)
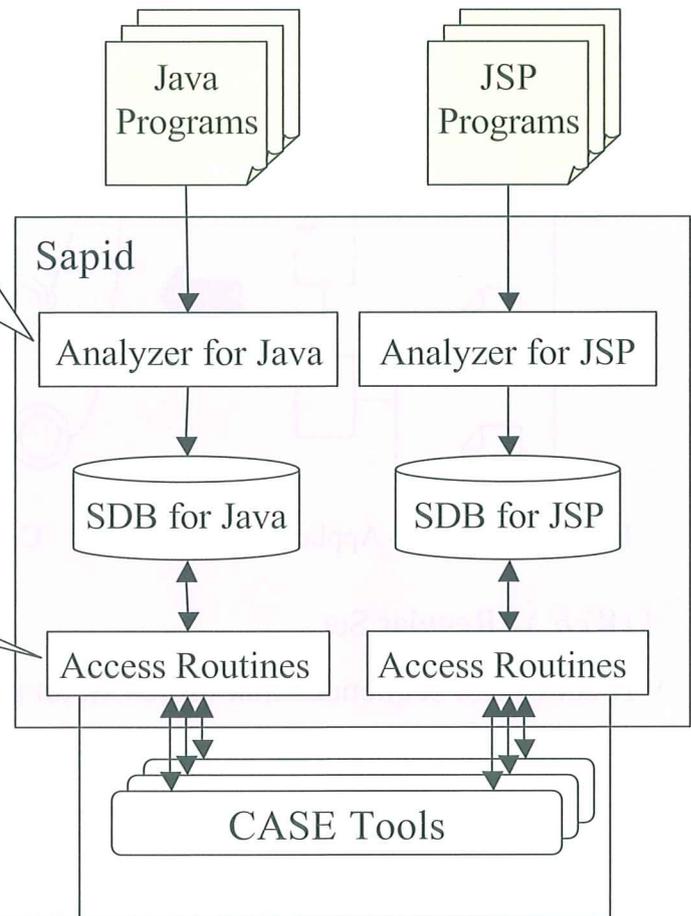
CASE tool platform based on fine grained repository
- Supports for design, implementation, testing, debugging and maintenance.
- Manages data about internal structure and structural elements of modules.

Target languages are C, Java, JSP, JavaScript, HTML and so on.

Analyze source programs based on syntactic elements and put their information into SDB.

Access to SDB and provide stuructural information about target program to CASE tools.

Specification of web applications
Ex.) struts-config.xml
Information about actions and pages

Java Programs

JSP Programs

Sapid

Analyzer for Java

Analyzer for JSP

SDB for Java

SDB for JSP

Access Routines

Access Routines

CASE Tools

Page Transition Graph

A method to represent the whole structure of web application.
Node :
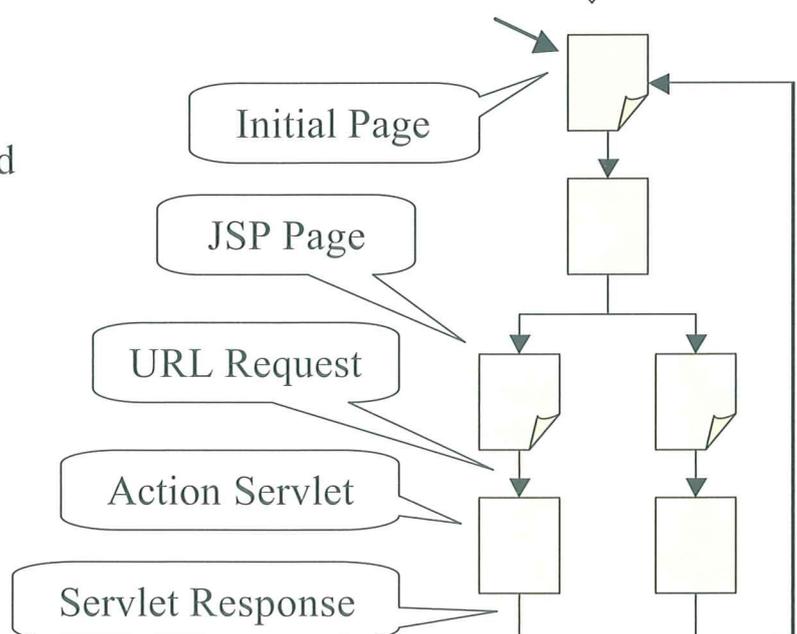JSP page and action servlet.
One of them is an initial page and some are finite pages.
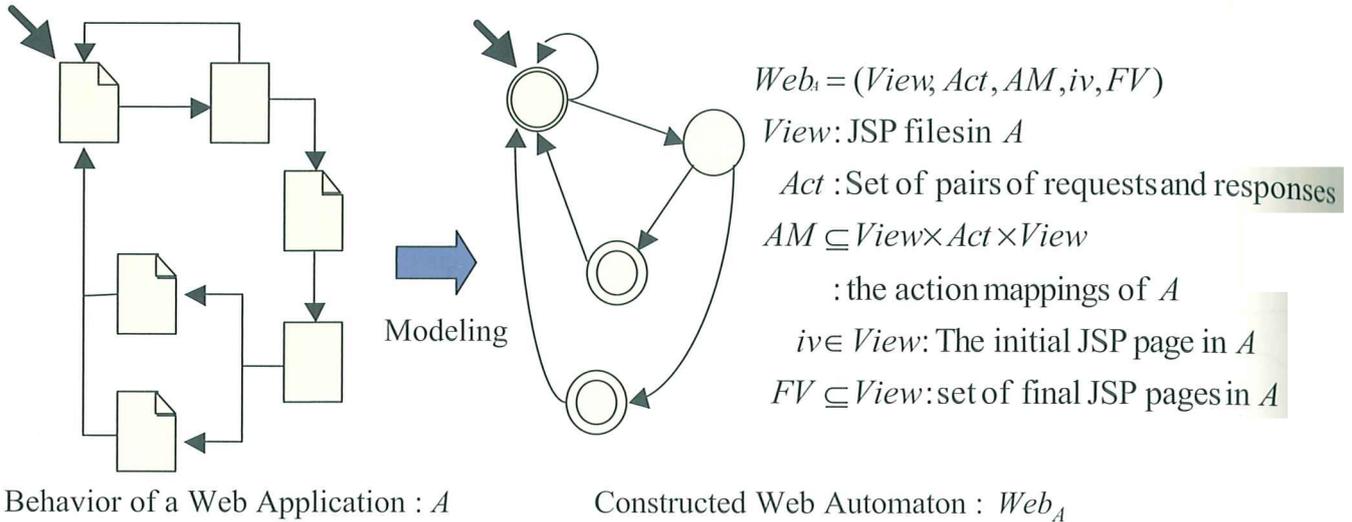
Edge :
URL request or servlet response.

Check followings :
- Reachability
- Deadlink
- Validity of executable path

Initial Page

JSP Page

URL Request

Action Servlet

Servlet Response

A Testing Framework for Web Applications with Behavioral Descriptions

Modeling Web Applications with Web Automaton

$Web_A = (View, Act, AM, iv, FV)$

$View$: JSP files in A

Act : Set of pairs of requests and responses

$AM \subseteq View \times Act \times View$

: the action mappings of A

$iv \in View$: The initial JSP page in A

$FV \subseteq View$: set of final JSP pages in A

Behavior of a Web Application : A Constructed Web Automaton : Web_A

$L(Web_A)$: Regular Set

A set of test sequences that represent an intended transaction or session

Accepted words by Web automaton are Infinite

Testing Criteria :

Categorize test sequences of Web applications

Focus on the loops of the Web automaton

$T_A^n = \{L(e) \mid e \in Exec(Web_A),$

$\forall v \in V(e) : occ(v, e) \leq n+1\}$

$Exec(Web_A)$: the set of all executions of Web_A

$occ(v, e)$: the number of occurences of

v in execution e

$V(e)$: the set of $views$ appering in e

Testing Framework for Web Applications

* Testing Criteria

$RepT_A^1 = \dots (xAct/ok + xAct/ng) \dots$

$RepT_A^2 = \dots RepT_A^1(xAct/ok + \dots)$

....

$RepT_A^i = \dots RepT_A^{i-1} \dots$

....

Behavioral Model   Categorized URL Test Sequences

Extract Test Sequences

Generate Test Data

xxx TestSuite link, form check

request   response

Web server

Consistency Check Between the Architecture and the Implementation

158