

### I-3 ITを利用した橋梁耐震性能評価システムのオープンシステム化

#### Web-based Seismic Damage State Evaluation System for Bridge Structures

○原田崇司<sup>1</sup>  
Takashi Harada

宮本文穂<sup>2</sup>  
Ayaho Miyamoto

中村秀明<sup>3</sup>  
Hideaki Nakamura

金好昭彦<sup>4</sup>  
Akihiko Kaneyoshi

【抄録】著者らは、従来より、橋梁構造物の耐震性評価を行うための手法として地震動の不確定性を考慮した「損傷状態確率」をその評価指標として提案し、その指標を用いた「橋梁耐震性能評価システム」を構築してきた。このシステムをより良いものへと改良するためには、ユーザからの意見を集約するとともに、実橋梁に関する種々のデータの蓄積が必要である。そこで本研究では、「橋梁の地震時耐震性評価システム」のオープンシステム化を試みた。特に、多くの潜在的ユーザが存在するインターネット上で稼動するようにWWWシステムとして再構築を試みた。本研究では、システム構築の際の留意点を述べるとともに、ユーザ満足度の一指標である応答時間について、既存のハードウェアでの検討を行った。

【Abstract】 The authors have been working for some time on the development of a damage state evaluation system for bridge structures by using the damage state probability as an evaluation index which considers the uncertainty of the earthquake motion. In order to widely apply this evaluation system to existing bridge structures, it is important to operate as an open system on the World Weather Watch (WWW) system. This paper outlines a Web-based seismic damage state evaluation system for bridge structures.

【キーワード】 橋梁全体系, 耐震性能, 損傷状態確率, 評価システム, Web ベースシステム, Java, 応答時間

【Keyword】 Bridge structure, Seismic damage state, Evaluation system, WWW system, Java, Response time

#### 1. はじめに

著者らは従来より、橋梁構造全体系の耐震性能評価を行うための手法として地震動の不確定性を考慮した「損傷状態確率」をその評価指標として提案し、その指標を用いた「橋梁の耐震性能評価システム」を構築してきた<sup>1,2)</sup>。このシステムの特徴は、簡易な計算で、橋梁全体系の地震時における損傷度評価を行うことができることにあり、従来、スタンドアローンで使われていた。このシステムをより良いものへの改良していくためには、できるだけ多くの人に使ってもらい、ユーザからの意見を集約するとともに、実橋梁の

様々な耐震性評価に関するデータを蓄積する必要がある。そこで、潜在的に存在する多数のユーザに使ってもらえるように、本研究では、「橋梁の地震時耐震性評価システム」のオープンシステム化を試みた。インターネット上には、多くの潜在的ユーザが存在しているため、インターネット上で稼動するようにWWWシステムとして再構築を試みた。

本研究では、システム構築における技術的問題の留意点を述べるとともに、ユーザ満足度の一指標である応答時間について、既存のハードウェアでの検討を行った。

<sup>1</sup> 学生員 工学士 山口大学大学院 理工学研究科博士前期課程 (〒755-8611 宇部市常盤台 2-16-1)

<sup>2</sup> 正会員 工博 山口大学教授 工学部知能情報システム工学科 (〒755-8611 宇部市常盤台 2-16-1)

<sup>3</sup> 正会員 博士 (工学) 山口大学助教授 工学部知能情報システム工学科 (〒755-8611 宇部市常盤台 2-16-1)

<sup>4</sup> 正会員 博士 (工学) 株式会社鴻池組 土木本部技術部 (〒541-0057 大阪市中央区北久宝寺町 3-6-1)

## 2. 橋梁耐震性能評価システムの概要

本研究でオープン化の対象としているのは、橋梁構造全体系の耐震性能評価を簡易的に行うことができるシステムである。以下にシステムの概要を述べる。

### 2.1 システムの構成

このシステムでは、橋梁構造全体系をシステム工学的に捉え、地震動が作用することによって進行する各部位部材の損傷を状態遷移確率行列でモデル化している。システムの概略構成を図-1に示す。

- ①まず始めに、橋梁構造全体系を上部工、支承、下部工、基礎などの部位で構成されるシステムとして捉える。
- ②地震力の作用によって各部位部材の損傷が進行していくメカニズムを状態遷移確率行列でモデル化を行い、多段階の損傷状態分類に対する損傷状態確率を算出する。
- ③構成部位部材の損傷度は、損傷状態確率とその損傷状態分類を代表する物理量との期待値と破壊までに許容出来る容量との比として評価している。具体的には、RC製橋脚のように破壊に至るまで非線形挙動する部位部材では、地震力作用により損傷する程度が地震期間中に部位部材に累積される損傷エネルギーで表されるという観点から、損傷状態分類を損傷エネルギーで代表させている。
- ④最終的な出力は、橋梁システム全体の損傷度である

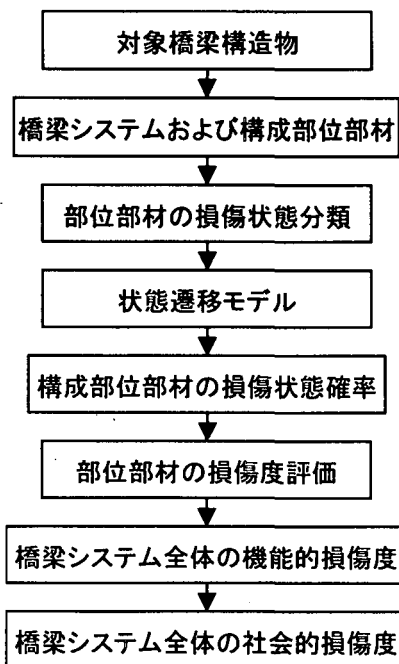


図-1 耐震性評価システムの概略構成

が、このシステムでは、機能的損傷度と社会的損傷度に分けて評価している。

機能的損傷度は、橋梁構造物が有している機能のうち車両通行機能を対象としたもので、次の3つのレベルを想定している。

【損傷状態A】桁の落下、橋脚の崩壊・倒壊、基礎の残留変位などの損傷状況により車両通行が不可能な状態。

【損傷状態B】桁や橋脚で仮受等応急処置を施すことにより制限付きではあるが車両通行が可能な状態。

【損傷状態C】大掛かりな応急処置がなくても車両通行が可能な状態

次に社会的損傷度であるが、社会的損傷度は、地震直後の復旧費用や復旧工期の期待値を橋梁全体系として合計したものをを用いている。

### 2.2 損傷状態確率の算出

このシステムでは、損傷状態確率の算出が必要となる。図-2に損傷状態確率の算出フローを示す。この損傷状態確率は、橋梁データ、地震波形を基に、動的解析を行い、動的解析で得られた応答値を用いて初通過破壊確率を算出し、その値を状態遷移確率行列に代入して求められる。

### 2.3 既存のシステムによる評価結果

2.1節で述べた耐震性評価システムはスタンドアロンでは既に構築されており、橋梁構造物の部位部材の損傷を損傷状態確率で表現する機能、損傷状態確率

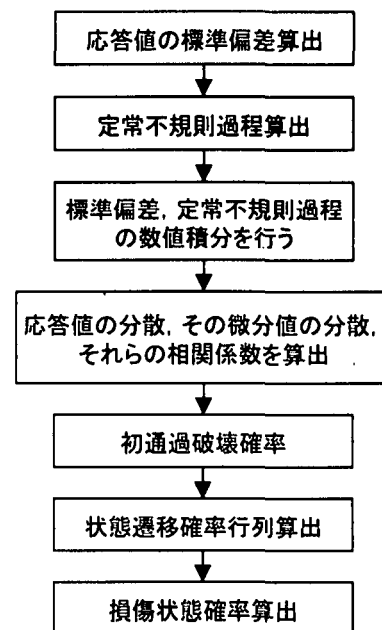


図-2 損傷状態確率算出フロー

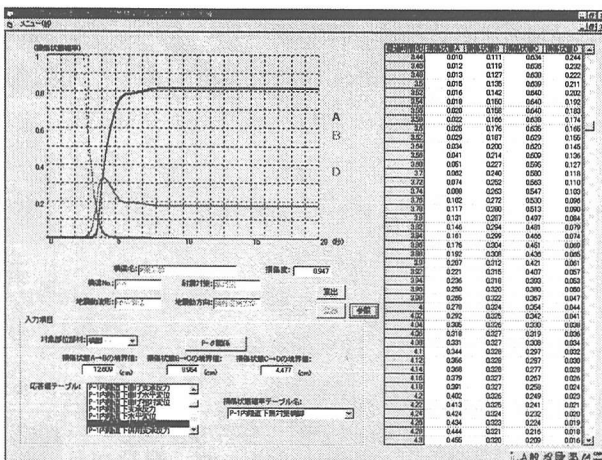


図-3 システム画面の例 (部位部材の損傷度評価結果)

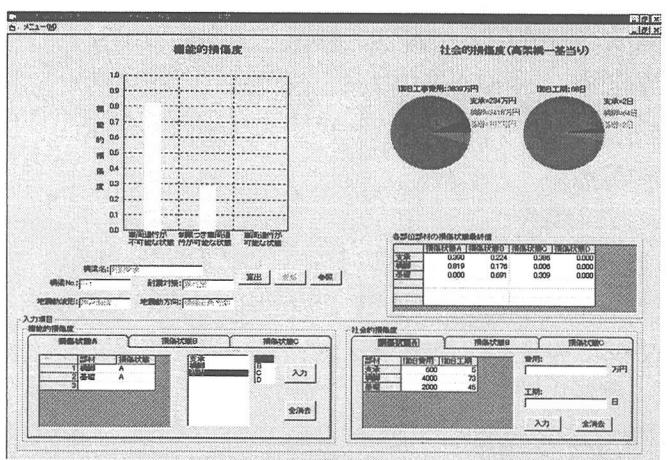


図-4 システム画面の例 (橋梁全体系の損傷度評価結果)

から損傷度や橋梁全体系の機能的損傷度, 社会的損傷度を算出する機能がひとつに統合されている. 図-3, 4にそのシステム画面を示す. ここで, 図-3は, 各部位部材の損傷度の出力画面例を, 図-4は, 橋梁全体系の社会的損傷度の出力画面例をそれぞれ示している.

耐震性評価システムは, 橋梁構造全体系の耐震性能評価を簡易的に行うもので, 橋梁設計の初期段階でまず, このシステムでだいたいの目星を付けておき, 最終段階で詳細な動的解析を行うことにより, 計算コストや設計の手間を省くことが可能となる.

### 3. WWW を利用したオープンシステム化

#### 3.1 WWW 利用の利点

システムをより使いやすく, 実用的なものにするためには, システム開発過程で, 多くのユーザーに試用してもらい, ユーザーの意見をシステムに反映することが必要不可欠である. 本研究では, 従来, スタンドアロンで用いられていたシステムをオープンシステム化することで, ユーザーの意見や要望の集約を行う. さらに, 今後の研究で必要とされる実橋梁の様々な耐震性評価に関するデータを収集する. WWW を利用したオープンシステム化の主な利点を以下にまとめる.

- ①OS や機種に依存しないユーザーの獲得
- ②運用・管理が容易
- ③データ収集・維持管理が容易

まず①の利点であるが, 従来システムは DOS-V 機で Windows 上で開発されていたため, DOS-V 機 Windows でないで使用できなかった. Macintosh や Unix Workstation のユーザーが使用するためには, 新たにシステムを開発する必要があった. オープンシステム化

することにより, WWW を利用できるユーザーであれば, 機種や OS を意識せずに使えるようになり, より幅広いユーザー層の獲得が可能となる.

②の利点であるが, 従来システムはスタンドアロンであり, 使用する全てのユーザーに対して, プログラムファイルやマニュアルを用意する必要があった. しかし WWW システムを構成するプログラムは WWW サーバ上に一元管理されており, WWW を利用できるユーザーであれば誰でも使用することができる. また, プログラムに修正や追加があった場合もプログラムは WWW サーバ上にしか存在しないためにその WWW サーバ上のプログラムを修正・追加するだけで良い. 一方, ③については, 解析に使用する入力データはひとつのデータベースに保存されるために, データの維持管理が非常に容易になる.

以上より, 本研究では耐震性評価システムのオープンシステム化を行う. インターネット上にシステムを構築するための技術は多数存在するが, 本研究では Java 言語の一つであるサーブレット技術を用いてシステムの構築を行う. 今回構築したシステムは, 最終的に「損傷状態確率」を出力する.

#### 3.2 WWW システム構築の要件

システムを構築する際には, そのシステムに対する要件を明確にすることが重要である. それらの要件の内, WWW システムに関しては主に, 機能要件, 性能要件, 信頼性要件, 安全性要件, 運用要件, 拡張性要件, ネットワーク接続要件が挙げられる.

##### ① 機能要件

機能要件とは, 構築するシステムに必要とされる機能を整理することである. 本システムでは「損傷状態

確率算出」が中心的な機能である。また、WWW システムであるために多くのユーザが同一のシステムを使用する。そこでデータをユーザごとに保持する機能が必要である。

## ② 性能要件

WWW システムでは同時にアクセスするユーザ数（同時アクセス数）の増加に伴い、システム応答時間が増大する。よって、システムの性能要件を満たすような仕様を設定するためには同時アクセス数の把握が重要である。しかし、同時アクセス数を把握することは困難であるため、同時アクセス数を若干多めに設定し、ハードウェアの拡張性を確保した上でその同時アクセス数に対するシステム応答速度を満足するようにシステムを構築した。具体的には、本システムのようなマイナーなシステムでは、初期段階では同時アクセス数は若干数であると考え、1~2 程度と想定し、検証において用いる同時アクセス数は最大で5とした。また、システム応答時間は1分未満とした。

## ③ 信頼性要件

信頼性要件とは、WWW システムのサービスを常時提供することを示すものである。したがって、ハードウェアの稼働率が問題となる。サービス停止の要因としてはサーバスペックの浪費によるサービスの停止が挙げられる。本システムではこの観点から信頼性要件を満足させる手法を適用した。

## ④ 安全性要件

安全性要件とは、セキュリティーのことである。本システムは研究に用いるデータを収集することを目的としている。収集したデータが可能な限りインターネット環境下に存在しないようにすることが必要である。

## ⑤ 運用要件

WWW システムの性能はハードウェアのスペックや構成に伴い変化する。よって、ある程度の性能が実現できるようにハードウェアスペックの補強、クラスタリングに伴うサーバ数の増加などが最も簡易な方法である。しかし、これらの性能確保は多大なコストを必要とする。今回は既存のPCサーバ(2台)を用いてWWW システムの稼働環境を構成するものとし、本研究室内で運用を行うものとした。

## ⑥ 拡張性要件

本システムの拡張性はハードウェア、ソフトウェアの両方に必要とされる要件である。ハードウェアに関

しては②の通り、性能要件を満足させるため、柔軟性が必要である。ソフトウェアは今後のシステムの機能拡張などに対し、柔軟性を持つことが必要となる。本システムは現在研究段階である。そこで、この拡張性要件に対してはソフトウェアの拡張性を保持することを目的に、それが満足されるような手法の適用を試みた。

## 3.3 WWW システム化における留意点

3.2節で述べた要件を満足するように、システムの構築（ソリューション）を行う。本節では、システム化における留意点について述べる。

本システムでは、損傷状態確率の算出を行う必要があるが、損傷状態確率の算出には多数の繰り返し演算が必要となる。WWW システムにおいては、従来からアプレットやスクリプト、CGIなどが用いられているが、これらの手法では性能要件で示される応答時間を満足することが困難である。アプレットとスクリプトはクライアントサイドで動作するプログラムであるため、WWW サーバ上に存在するプログラムが HTML ファイルと同時にクライアントに送信される。つまり、計算を行うためには、プログラムの送信が必要であるため、データ量が静的コンテンツに比べ多くなる。データ量が多ければネットワークのトラフィック量が増大するため、データ通信速度が低下し、ブラウザ上にページが表示される時間は増大する。また、アプレットやスクリプトは、OS や WWW ブラウザのバージョンにより挙動が異なるという、依存性が存在する。

これに対し、CGI はサーバサイドプログラムである。サーバサイドプログラムではネットワークを流れるデータはシステムに必要な入力データと、処理により出力されたデータがほとんどであるため、ネットワークのトラフィック量は静的コンテンツとそれほど変わらず、快適な動作を実現できる。また、サーバサイドプログラムではシステムが WWW サーバでのみ動作するため、クライアント側の機種や OS、ブラウザなどに対する依存性がなくなる。さらに、アプレットやスクリプトはクライアント側で動作するため、動作速度は、クライアントのハードのスペックにより決まってしまうが、サーバサイドプログラムでは WWW サーバのスペックに依存するので WWW サーバのスペックを増強することで動作速度を速くすることが可能である。このことは、ハードウェアの拡張性維持の面でも

有効である。しかし、CGIを用いたシステムはその処理速度があまり速くない。CGIでは一つの要求に対して一つのプロセスが作成されるため、多くの要求が集中すると、その数だけプロセスを作成する必要があり、処理速度が低下する傾向がある。

その他の留意点としては、データの漏洩防止手法である。ユーザの個人情報はもちろんのこと、橋梁の諸元データ、応答波形、算出された損傷状態確率などの情報が外部に漏洩することは絶対に避けなければならない。WWWシステムが動作するWWWサーバ等はインターネット環境に公開しなければならず、できるだけセキュリティレベルの高いWWWシステム稼動環境を適用する必要がある。

他の留意点としては、システムの拡張性の問題がある。本システムのように研究を主とするシステムでは、結果の精度向上などを目的に、新機能の付加や入れ替えなど、ソフトウェアの拡張が頻繁に行われる。そこでシステムの拡張が容易に行えることが要求される。

このような留意点に対して効果のある手法の適用を本システムの構築では検討した。次節ではその手法について述べる。

### 3.4 適用手法

3.3節で述べた留意点を考慮し、本システムの構築において用いた手法について以下に述べる。

#### 3.4.1 サーバー・サイド Java<sup>3)</sup>

WWWシステムを構築するために従来から用いられているアプレット、スクリプト、CGIに代わり、本システムではJava言語のサーブレット、JSPを使用する。これらはサーバサイドプログラムであり、CGIのような他のサーバサイドプログラムに比べ有利な点が多数存在する。

Java言語はオブジェクト指向を用いたプログラミング言語であり、共通な機能の一つのオブジェクトとして構築することにより生産性を高めることができる。サーブレットはクライアントからのリクエストに対してレスポンスを返すというインターネットを用いたWWWシステムに必要な共通の機能をもとに作成されたオブジェクトである。サーブレットにはリクエストに対してレスポンスを返すメソッドとして service()メソッドがあり、この service()メソッドをオーバーライドすることでシステムが必要とする機能を構築することができる。しかし、クライアントに返信するデー

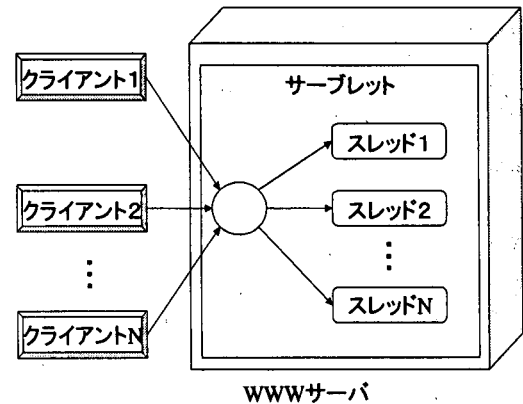


図-5 サーブレットの構成

タはHTML形式で書かれる必要があるために、サーブレットのみでシステムを構築するとロジックとデータが混合し、プログラムの管理が困難になる。この問題に対してはASPのようなHTML内部に簡単なプログラムを記述できる言語を用いることで解決することができる。Java言語を用いてこのような記述を行えるプログラムがJSPである。JSPは一度サーブレットへ自動的に変換され実行される。したがって、プログラム実行時の挙動はサーブレットと同一である。

サーブレットは図-5に示すように、リクエストにより最初の要求が発生した時点でサーバ・プロセスでの一つのスレッド(プログラムの内部実行単位で規模は、プロセスより小さい)として実行される。各スレッドはプロセスの資源を共有するので、生成や切り替えがプロセスよりも軽い。サーブレットプログラムはクライアントとの接続が終わっても、生成されたオブジェクトをWWWサーバ上に残し、データを保持する。つまり、最初の要求以降はそれ以上の速度で処理を行うことができる。したがって、3.2節で述べた複数の要件を満足させることが可能である。

#### 3.4.2 MVCモデル<sup>4)</sup>の適用によるソフトウェアに対する拡張性の取得

MVCモデルとは図-6に示すように、Model(システムの中心的な処理)、View(ユーザインターフェースを構成)、Control(Model・View内のプログラムの制御やユーザから送信されるデータの加工)のようにオブジェクト指向的な考えを基に、各プログラムを分類してシステムを構築する手法(モデリング)である。Java言語を利用した場合、Modelは一般的なJavaクラス(ビーン)であり、ViewはJSP、Controlはサーブレットである。このように領域ごとにプログラムを作成

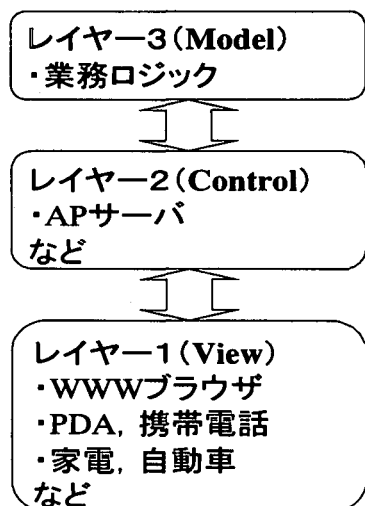


図-6 MVCモデル

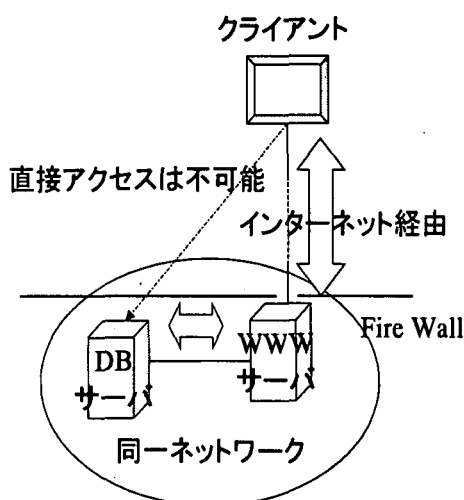


図-7 3階層モデル

することは多くの利点がある。

まず始めの利点は、プログラムの維持管理が容易になることである。Model, View, Controlを一連のプログラムで作成すると、画面描写や出力データの為に作成されるプログラムと制御を行うプログラム、また実際の処理を行うプログラムが混同してしまい、非常に読みづらいプログラムとなってしまう。

2つめの利点は、機能の追加などに伴う人的なコストの低減化があげられる。ロジック部分はサーバ上で動作するために、システムの動作環境には依存しない。したがって、仮にクライアントが使用する端末がパソコンではなく携帯電話となった場合、JSP内部のHTMLタグをCHTMLタグに書き換えるだけでそのシステムは使用することが可能になる。また、同一の入力を用いて異なる処理を行うような機能ならばModel内にその機能を追加することで実現できる。よって、このMVCモデルの適用により、ソフトウェアの拡張性を構築したWWWシステムは取得することができた。

### 3.4.3 WWW・データベースサーバの分割による 負荷分散とセキュリティの向上

本研究でオープンシステム化したWWWシステムは、インターネットに接続されているためデータの漏洩に注意する必要がある、セキュリティを強化する必要がある。

本研究では、WWWシステム構築環境には2台のPCサーバを使用しており、セキュリティ強化のため、3階層モデルを適用する。この3階層モデルとはクライアント(WWWブラウザ)⇔WWWサーバ⇔データ

ベースサーバというようにデータの流れを構成したWWWシステム稼働環境を示す。本システムに適用した3階層モデルの概略を図-7に示す。

クライアントはインターネットを経由してWWWサーバにアクセスする。WWWサーバは処理上必要となるデータをDBサーバから取得する。このとき、クライアントはDBサーバに直接アクセスできないようにすることでインターネットからのクライアント情報の漏洩を防止する。これについては既存のファイアウォールを利用している。また、このことはWWWサーバにDBの維持管理や検索という処理を省くために、負荷を軽減することができ、更にサーバスペックの浪費によるサービスの停止を防止することができる。

セキュリティに関しては、通信中のデータの漏洩防止のため、SSLによる暗号化等も必要であるが、本研究では今のところ実現できていない。

## 3.5 WWWシステムの構築

### 3.5.1 動作の実際

クライアント、サーバ間におけるシステムのフローを図-8に示す。クライアントサイドではHTMLによりユーザインターフェースを提供し、サーバではJSP、サーブレット、ビーンが動作している。以下に動作の概略を説明する。

- 1) システムを利用するにはまずユーザ認証が必要である。ユーザ登録を行っていないユーザは新規登録を行う。
- 2) 新規登録ではユーザのアカウント名、パスワードなどを入力する。そして、これらのデータはWWWサーバに送信される。

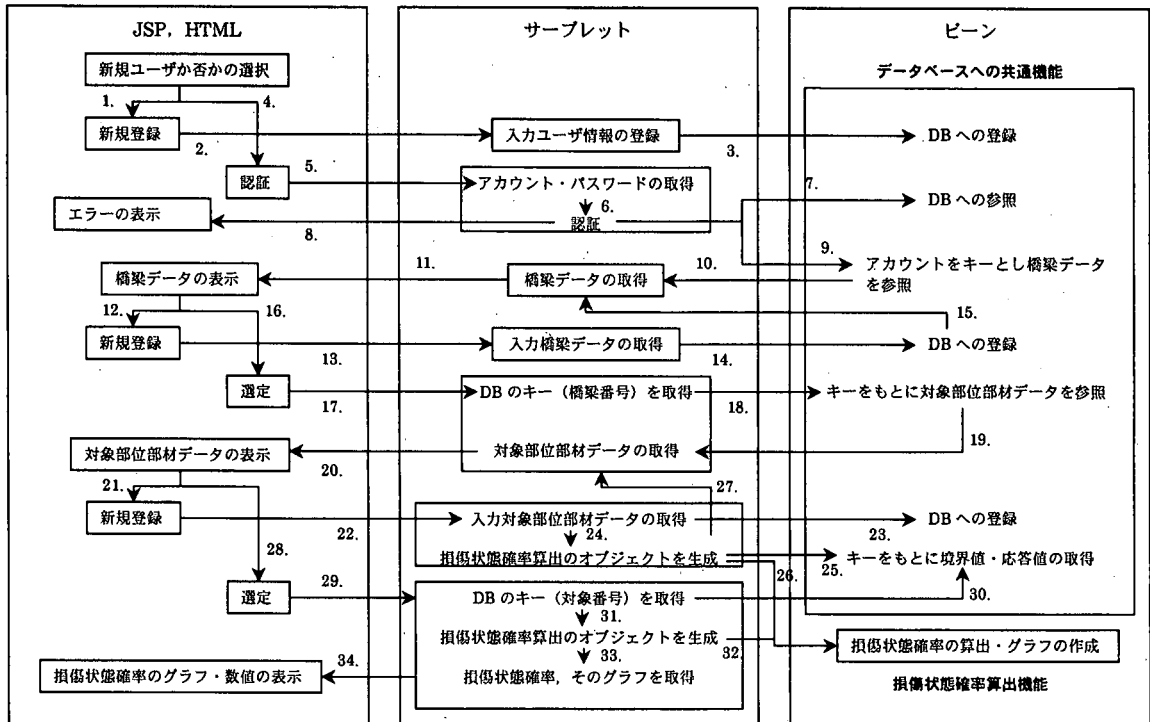


図-8 WWWシステムフロー

- 3) WWWサーバではサーブレットによりこれらのデータが取り出された後、それらのデータを使用してSQL文を作成する。そして、データベースへの操作を行うクラスにそのSQL文を引数として渡すことでデータベースへの登録を行う。
- 4) 登録済みのユーザは認証を行う
- 5) ユーザからアカウント、パスワードがWWWサーバに送信される
- 6) WWWサーバはサーブレットによりアカウント、パスワードを受け取り、認証を行う
- 7) 認証はアカウントをキー項目としてデータベースからパスワードを参照する。
- 8) ユーザが入力したパスワードと参照したパスワードが異なっていればエラーメッセージをクライアントに表示する
- 9) それとは逆に各々のパスワードが一致する場合、アカウントをキー項目として前回までに登録された橋梁データを参照あるいは取得する。
- 10) 橋梁データの取得する。
- 11) それをクライアントに表示する。
- 12) ユーザは表示された複数の橋梁から損傷状態確率を算出する橋梁を選定する。このとき、対象とする橋梁がない場合は新規に作成を行う。
- 13) 新規作成はユーザの新規登録を行う。
- 14) ユーザの新規登録と同様にデータベースへの登録を行う
- 15) 登録されたデータを含めてそのユーザが保持する橋梁データを取得する。
- 16) 橋梁の選定を行う。
- 17) 橋梁の選定を行った場合、WWWサーバには橋梁データのインデックスとなる橋梁番号が送信される。
- 18) 橋梁番号は対象部位部材データに対するキー項目であり、このデータによりその橋梁のうち前回までに算出した部位部材データが取得できる。
- 19) キーをもとに対象部位部材のデータが参照される。
- 20) 対象部位部材データがクライアントに表示される。
- 21) ユーザは表示された対象部位部材データに存在しない部位部材の損傷状態確率を求める場合には新規登録を行う。
- 22) 対象とする部位部材のデータをWWWサーバに送信する。送信するデータには応答値ファイルも含まれる。
- 23) 送信されたデータはサーブレットでプログラム内に読み込まれ、データベースに登録されるがここで、応答値データはファイルとしてWWWサーバ内に保存される
- 24) 登録後に損傷状態確率の算出を開始する。

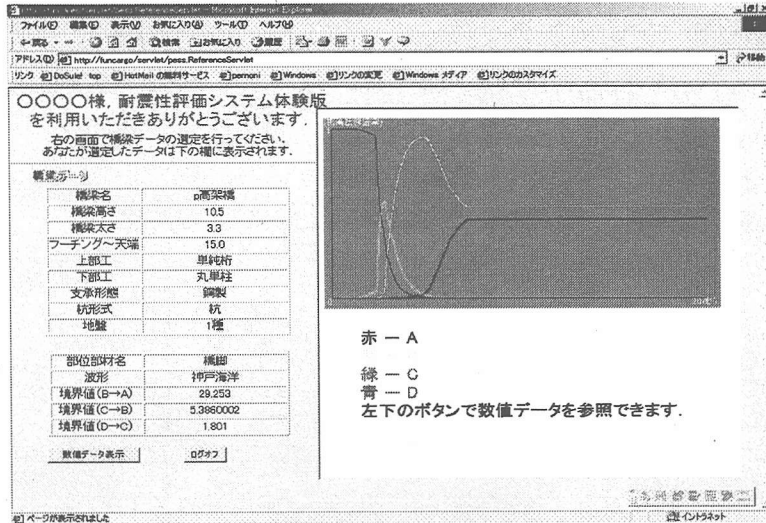


図-9 損傷状態確率の算出例 (グラフ)

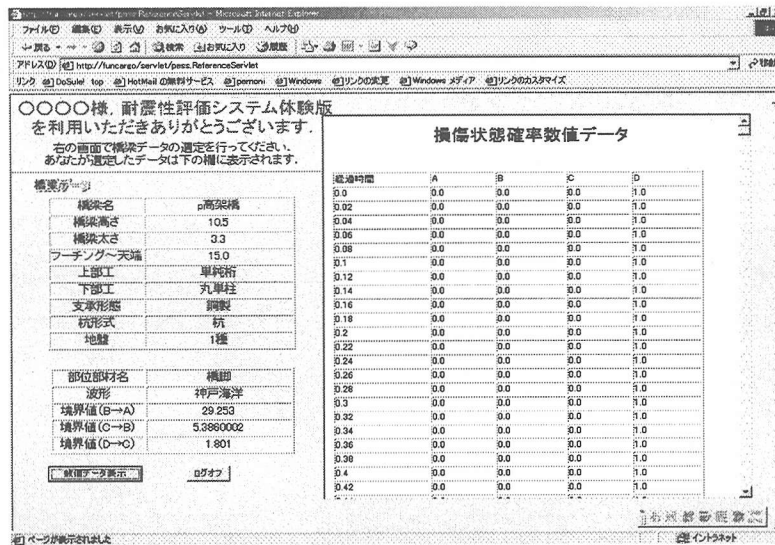


図-10 損傷状態確率の出力例 (数値)

- 25) オブジェクトを生成し、そのオブジェクトに応答値、境界値を設定する。
- 26) メソッドを呼び出すことで損傷状態確率の算出を実行する。
- 27) 損傷状態確率算出後、サブレットは新規に作成された対象部位部材データを含めた結果を取得し、クライアントに表示する。
- 28) 結果の参照は対象部位部材データが表示された状態で参照するデータを選定することで行う。
- 29) 参照データを選定するとクライアントから対象番号が送信される。対象番号とは算出結果にリンクを設定するために設けたインデックスである。
- 30) 対象番号をもとに各損傷状態の境界値、応答値デ

ータを取得する。

- 31) それらのデータは損傷状態確率を算出するために生成されたオブジェクトに格納される。
- 32) メソッドが呼び出され算出が行われる。
- 33) 算出されたデータはクライアントに送信される。
- 34) データは、グラフや数値として表示される。

以上で構築したシステムのユーザインターフェースを図-9~11に示す。WWWシステムではWWWブラウザを使用するのでユーザにとってなじみのあるインターフェースを実装することができる。このことから、本システムの操作性は向上するものと思われる。



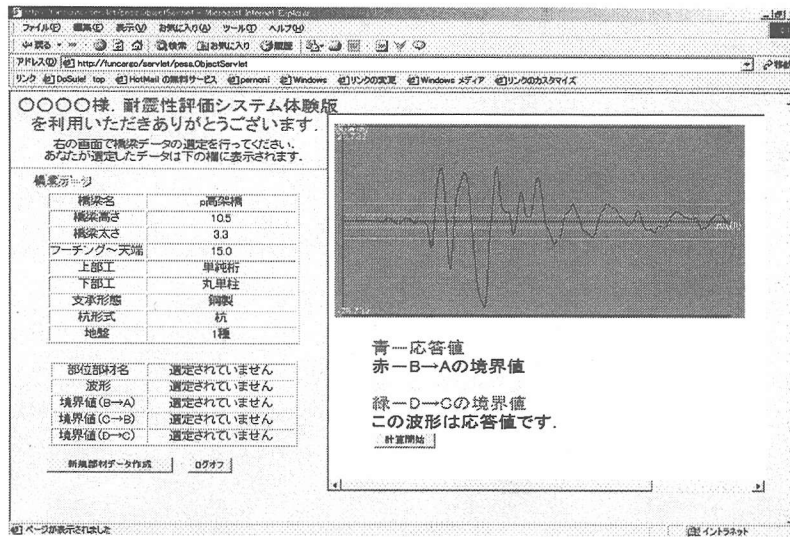


図-11 応答波形データ確認画面

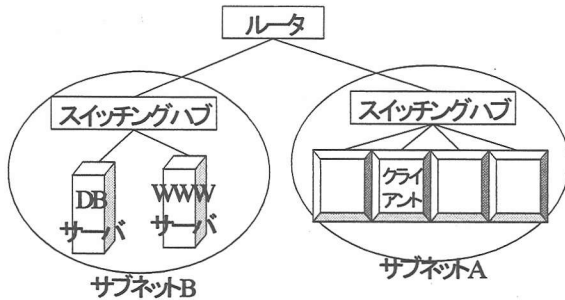


図-12 実験環境

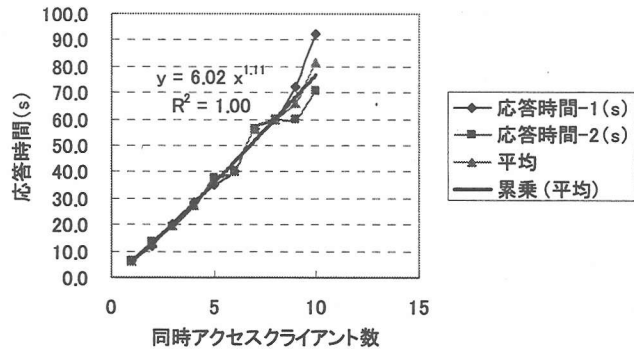


図-13 応答時間の測定結果

### 3.5.2 稼働環境

サーバサイドプログラムによる WWW システムでは中心的な処理はすべて WWW サーバマシンで行う。そこで、WWW サーバ環境に対してボトルネックが発生する可能性がある。ここでは、システムの要求である、インターネット上でも快適なシステム動作を達成するための大きな要因となる WWW サーバの環境および WWW サーバの周囲にあるネットワークインフラについて述べる。WWW サーバは CPU が Pentium III 500MHz、メモリーが 128MB であり、OS は Microsoft 社製の Windows 2000 Server(SP1)を使用した。WWW サーバソフトには同社の IIS (Internet Information Service) 5.0 を使用した。またサーバレットエンジンには Tomcat version 3.0<sup>5)</sup> を使用した。

データベースサーバは CPU が Pentium III 600MHz を最大で二つ使用できるものであり、メモリーは 512MB を搭載する。OS は Microsoft 社製 Windows NT 4.0、また RDBMS (Relational Database Management System) ソ

フトは同社の SQLServer 7.0 を使用した。また、データベースサーバのハードディスクについては RAID コントローラにより RAID0: ストライピングが設定されており、各ハードディスクの容量は 36GB で 10,000 回転する。各サーバは伝送速度が 100Mbps/s であるより対線 (100BASE-TX) を使用したイーサネット (Ethernet) で接続されており、通信プロトコルは TCP-IP 接続である。ハブにはスイッチングハブ (LAN スイッチ) を使用した。また、各サーバは同一サブネット内に存在する。

実験ではその数の 2 倍である 10 までの応答時間を測定した。実験環境を図-12 に示す。ここで、測定する応答時間は、システムが行う処理の中で最も時間を要する「損傷状態確率算出」へと処理を要求し、その結果であるページが完全に表示されるまでの時間とした。また、クライアントはプログラム<sup>6)</sup>によって擬似的に作成されたものであり、その作成されるクライアントの数を同時アクセス数とする。

実験結果を図-13に示す。応答時間-1は同時アクセス数を徐々に増加させ実験を行ったときの応答時間である。そして応答時間-2はその逆で、同時アクセス数を徐々に減少させ実験を行ったときの応答時間である。平均はその二つの値の平均である。また、近似関数は平均の値を基に求めた。R<sup>2</sup>は近似曲線の信頼性を示す決定変数である。まず、応答時間-1と応答時間-2の誤差についてはデータベースサーバの処理時間が大きく関与していると思われる。データベース内のデータ数が多ければデータベースサーバが行う処理時間は長くなる。したがって、応答時間-1の同時クライアント数10の応答時間は応答時間-2のそれよりも高い値を示している。また、同時クライアント数が低い場合はその処理時間による誤差は低いことも分かった。同時アクセス数と応答時間の傾向は次式により表現される。

$$y=6.02x^{1.11} \quad (1)$$

この式から、同時アクセス数が低い場合に1~2程度の微妙なアクセス数の変化があったとしても応答時間は6.0秒程度のずれしか生じないが、アクセス数が高い場合はその微妙な増加により応答時間が大幅に長くなる事が分かる。想定した同時アクセス数の最大は5でありそのときの応答時間は約36.4秒である。ユーザがシステムを使用する場合、許容応答時間がシステムへの要求解決において問題となる。寛容的ではあるが1分未満と想定した場合、本システムの同時アクセス数の限界は8程度と考えられる。よって、想定した同時アクセス数に対しては要求を満たすことができるが、複数のルータを経由してアクセスを行う遠隔地のクライアントに対してはより多くの応答時間を要すると思われる。よって、同時アクセス数の限界は8よりも大幅に少なくなると予想される。これについては、サーバの外部公開後に、より詳細な調査を行うものとする。

#### 4. 結論

本研究は昨年度までに、スタンドアロンで使われていた「橋梁の耐震性能評価システム」をより良いものへと改良していくため、インターネット上への移行を行った。

本システムの稼働環境をインターネット上へ移行さ

せるために、インターネットの基本的な仕組みについて、また、システムを稼働させるために必要な、動的コンテンツの生成方法について詳細な調査を行った。その中で、本研究で実現しようとしている性能に対し、適した方法であるサーブレット、JSPについて、また、本システムに適用した二つの3階層モデルについて本論文で言及した。

オープンシステム化に対する結論は以下のようにまとめられる。

- ① 今回想定した同時アクセスクライアント数に対し、現在のWWW・データベースサーバのスペックは十分であると推測される。
- ② セキュリティに対しては無償のチェックツールにより問題はないことが確認された。
- ③ 今回はシステムの稼働環境をインターネットに移行するための準備開発であり、「スケーラビリティ」、「運用・管理」に関しては運用後に詳細に調査する必要があると思われる。

今回はシステムの稼働環境をインターネットに移行するための準備開発であり、「スケーラビリティ」、「運用・管理」に関しては運用後に詳細に調査する必要があると思われる。

#### 謝辞

本研究の遂行にあたり、プログラム開発など多大な援助を頂いた元山口大学大学院理工学研究科大学院生中山英志氏に感謝の意を表します。

#### 参考文献

- 1) 金好昭彦：橋梁構造システムの耐震性評価に関する研究，山口大学博士論文，2000.3.
- 2) 金好昭彦，宮本文穂，中村秀明，中山英志：橋梁全体系の地震時損傷評価のシステム化，材料，第50巻第1号，pp.32-39, 2001.1
- 3) Marty Hall 著，岩谷 宏 訳：コア・サーブレット&JSP Java サーバ技術による Web 開発，SOFT BANK，2001.1.
- 4) サーバー・サイド Java の実力を探る，日経オープンシステム，日経 BP 社，p106, 2000.9.
- 5) The Jakarta Project: Tomcat, <http://jakarta.apache.org/>
- 6) Microsoft : I can't Stress It Enough - ASP アプリケーションのロードテスト, <http://www.microsoft.com/japan/developer/workshop/server/asp/server092799.asp>, 1994.8.