

ITSS 診断システムの開発

前田陽平† 櫻井一欽† 串松慎太郎† 安藤晴仁‡ 古石雄輝‡ 大谷真†

湘南工科大学 情報工学科† 国際テクノロジーセンター株式会社‡

1. はじめに

ITSS とは 2002 年に経済産業省が策定した IT 関連サービス分野における職種と必要とされる技能体系で、スキルの集合と習熟度に応じて 1~7 までのレベルが定義されている。ITSS 診断システムとは ITSS のレベル判定ツールのことを指す。

本研究では OSS ベースで ITSS 診断システムを Web アプリケーションとして新規開発した。開発にあたっては中心的なフレームワークに Struts を採用し、最新の ITSS に準拠する事、実用上十分な性能であることを方針に開発した。

2. ITSS と ITSS 診断システム

(1) ITSS コンテンツ

スキルの集まりと、レベル判定基準を定めている。スキルの集まりはスキル項目が約 7000 個、質問項目は各スキル項目に対しておおよそ 5 個定義されている。コンテンツが格納されているテーブルは 8 階層の木構造になっている。本研究では 5337 項目を使用した。

(2) ITSS 診断システム

診断方法は図 1 に示すように、利用者はブラウザ画面を通して ITSS コンテンツにしたがって表示されるスキルに対し習得状況を回答する。すべての回答が完了したら、回答結果から 1~7 までのレベルを算出され、キャリアフレームワーク上に表示される。

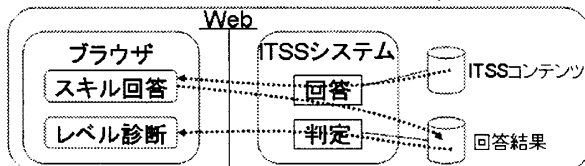


図1. ITSS診断システムの概要

3. 開発方針

1) 使用言語とフレームワーク

ITSS システムを構成する言語には java を採用し、Web アプリケーションのフレームワーク

Development of ITSS Judgment System

Yohei Maeda†, Kazuyoshi Sakurai†, Shintaroh Kushimatsu†,

Haruhito Ando‡, Yuki Fuluishi‡, Makoto Oya†

Shonan Institute of Technology†

Kokusai Technology Center Co., Ltd. ‡

には商用アプリケーションとして最も普及している Struts を採用した。

2) D/B との連携の方法

開発は MVC モデルにより行ったが、画面と D/B との接続方式をテーブル毎にモデルを作成するのではなく、ページ毎にモデルを作成した(図 2)。ページに関係しているテーブルの量が多い為、テーブル毎にクラスをインスタンス化した場合メモリの無駄になり、クラスの依存関係も複雑になるからである。

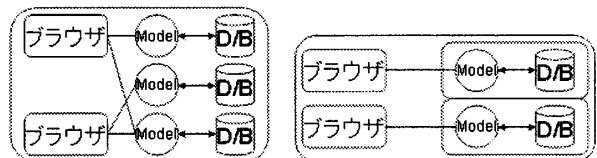


図2. D/Bとモデルの連携方式

3) 画面動的構成

スキルを選択する画面は JSP による動的生成方式を採用した。これはスキルコンテンツが変更されても画面のプログラムを変更する必要がないからである。

4) その他

Struts が提供する機能のエラーチェックのバリデーション機能や、xml での記述によるフォームの自動生成機能。JSP 内で使用する Struts 固有のタグライブラリの有効活用。D/B との接続情報を xml に記述し、再コンパイルの必要がないコネクタとする、そしてパフォーマンスの重視を方針として設計・開発した。また、各クラスの依存性を極力排除したクラス構造をとることも重要な方針とした。

4. 実装

(1) プログラム構成

同システムの判定ロジックは、Web アプリケーションとは別に単一のパッケージとして作成した。Web 上で同時に、あるいは判定中に別ユーザーが判定データを出した場合、5337 の回答結果をユーザー毎に参照し、計算結果を保存するのは D/B に多大な負荷を与え、判定とは無関係なページを参照中のユーザーにも影響を与える。したがって一定時間置きにバッチ処理をした方が効率的に資源を活用でき

るからである。

プログラムは全部で 42 クラスで構成されており、機能別にパッケージとして分割した。主なパッケージは図 3 の 5 種類である。また、図 3 の分類を用いて画面毎のクラス関係を表 1 に示す。



図3. 機能別パッケージ構造

表 1. ページ毎のクラス関係

機能	クラス				
	Action	From	Model	DAO	Logic
Login	○	○	○	○	○
Select	○	○	○	○	○
Question	○	○	○	○	○
Answer	○	○	○	○	○
Career	○	○	○	○	○
Decide	○		○	○	○
Logout	○				
共通	utility, msg				

さらにパッケージ間の関連性は図 4 の通りである。ユーザーからのリクエストや、その時に送られてくるパラメーター等を処理するクラスを Action 関係のパッケージにまとめた。その際、ユーザーに処理した結果のやりとりをするのを Form 関係にまとめ、Action 関係と Form 関係の媒介として Model 関係を作成した。また、D/B と直接やりとりするクラスを DAO 関係とし、Logic 関係では DAO 内で発生した例外処理や、D/B から受け取った値の処理をするクラス群とした。

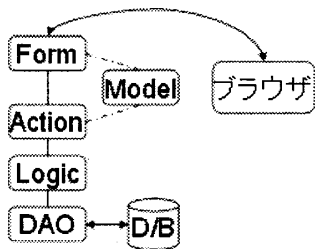


図4. パッケージ間の関連

(2) 主要な工夫点

a) シングルトンパターンの採用

D/B にアクセスするクラスは全てシングルトンパターンを採用した。初回のアクセス時にクラスをインスタンス化する為、それ以降クラスをインスタンス化する必要がないので高速化が期待できるからである。

b) 例外のキャッチ

Java のプログラミングの D/B とのやりとりについて、各変数ないで発生した例外を必ず検出し、対処できる構造にした。図 5 にその例を示す。

```
try{
    con = ds.getConnection();
    try{
        sql = con.prepareStatement("SELECT skill
        sql.setInt(1,empid);
        try{
            for(rs=sql.executeQuery();rs.next()
            }finally{
                rs.close();
            }
        }finally{
            sql.close();
        }
    }finally{
        con.close();
    }
}
```

図5. 例外のキャッチの例

5. 開発結果

今回 ITSS 診断システムを開発することができ、スキル診断機能（判定機能、質問・回答機能）、キャリアフレームとランク表示機能を動作させることに成功した。ITSS 準拠については最新の ITSS バージョン 2 2006 に準拠させたが、ITSS ユーザー協会で定義されている厳密な準拠性[1]に対しては 14 個の内、完全に準拠できたものは 3 個にとどまった。コメントを除くソースコードの総ライン数 3106 行、総クラス数 42 個となった。最も時間のかかる処理である判定処理の測定結果を図 6 に示す。登録ユーザー数 100 ユーザーで、一人にあたり 1.4 秒で処理することができた。この他に 1000 ユーザーでも 13.8 秒の速度となった。これにより小～大企業で ITSS 診断をする場合、実用上十分な性能であるといえる。

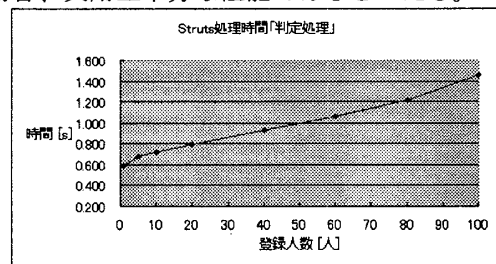


図6. 判定処理の時間

6. まとめ

今回開発した ITSS 診断システムは ITSS 診断システムとしては OSS ベースで開発された初めてのシステムである。今後は ITSS ユーザー協会が定義している ITSS 準拠に全て対応する為に機能を追加していく。特に今回対象としなかった約 2000 個のスキル項目に対するコンテンツの追加は必須である。また、ユーザー管理機能の追加、性能の向上も重要な課題である。

謝辞

本システムを構築するにあたり、ITSS ユーザー協会 殿には、ITSS スキルコンテンツ V2_2006 をご提供していただきました。

参考文献

1. 診断ツールガイドライン

http://www.itssug.org/docs/guideline/guideline_res.pdf