B-018

VDMJ と Apache Axis2 を用いた上流工程における モデル実行環境の構築

Model Executing Environment in Upstream Process using VDMJ and Apache Axis2

村林 慧 †　多田 圭佑 †　和崎 克己 ††
Kei Murabayashi　Keisuke Tada　Katsumi Wasaki

1　はじめに

システム設計において，仕様の定義は最も重要な工程である．現在主流である自然言語による仕様記述には仕様自体の曖昧さ，欠陥発見の遅れによるコストの増大，など多数の問題点がある．このような問題を取り除く有用な方法として形式手法が存在する．形式手法の一つとして，仕様を形式的に記述し，仕様自体の曖昧さ，欠陥を排除する仕様記述言語 VDM++ がある [1][2][3].

VDM++ で記述した仕様を検証する際に，オープンソースの VDM 向け援用ツールである VDMJ が用いられている．VDMJ は構文解析器，型チェッカー，インタプリタ，デバッガ，および証明課題生成機能を持つ．しかし VDMJ はコマンドライン・インタフェースのみの提供となっており，利用者は，妥当性確認の際にモデルの詳細について把握している必要がある，多人数での妥当性検証に不向きである，妥当性確認が直感的ではない等の問題点が存在している．

上記の問題点に対し，本研究ではオープンソースの VDM 向け援用ツールである VDMJ と，SOAP 実装である Apache Axis2 を用いてモデル実行環境 [4] を構築した．通信プロトコルに SOAP を用いているため，分散環境下でモデル実行を行うことが出来る．ユーザは Apache Axis2 により Web サービスとして展開されている VDMJ に接続し，サーバーに VDM++ ファイルをアップロードすることで，モデル実行を行う．

また Java 向けに提供されている Web API を用いることで，GUI で構築された外部アプリケーションとの通信を行い，VDM++ の知識を必要とすることなく利用者が仕様の妥当性をテスト工程時において容易に確認することが出来る．モデル実行環境の評価を行うため，VDM++ で記述された仕様と，最初に仕様に対応する Java Servlet で記述された外部アプリケーションを手書きで試作し，その後，著者らが開発しているアクティビティ図からの Java Servlet コード自動生成器 [5] によって生成された Java Servlet を利用しモデル実行環境の評価を行った．具体的な例として，ある予約システムの仕様を対象とした妥当性確認を行った．

2　仕様記述言語 VDM++

2.1　形式手法と仕様記述言語 VDM++

形式手法とは，信頼性の高いシステムを作るために用いる仕様記述・設計開発・検証の技術である．形式手法の一種として形式仕様記述が存在する．形式仕様記述は理論に基づいた形で仕様を厳密，正確に書く．基本設計の段階でモデル実行を行い誤りを排除することで，手戻りによるコスト増大を防ぐことが出来る (図 1)．形式仕様記述言語 VDM++ は，Vienna Development Method（ウィーン開発手法）の基盤言語 VDM-SL を，オブジェクト指向に基づいたモデル化を扱えるように文法を拡張したものである [1]．VDM++ には，不変条件，事前条件，事後条件を制約条件として記述可能な支援ツールが用意されており，これによって仕様を検査，テストすることが出来る．VDM++ には開発援用ツールが存在する．厳密さでは数学的証明を用いた定理証明系に劣るものの，専門的知識がなくとも仕様実行を用いて妥当性を確認出来る．

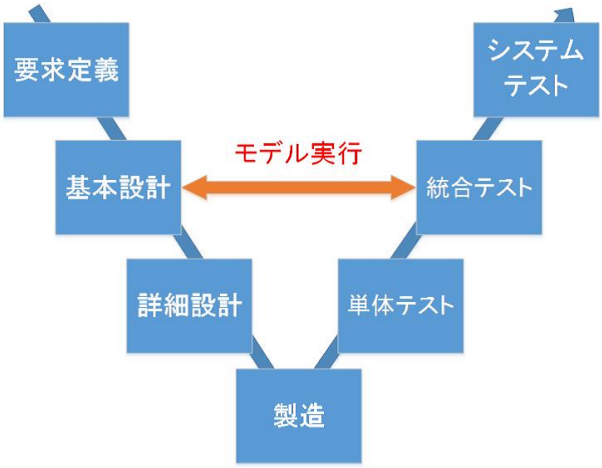

図 1　V 字モデルにおけるモデル実行によるテスト工程

2.2　VDMJ

VDMJ とは Java で記述された仕様記述言語 VDM-SL，VDM++，VDM-RT（VDM++ の非同期リアルタイム向けシミュレーションモデル用拡張）のための VDM 開発援用ツールである [2]．構文解析器，型チェッカー，インタプリタ，デバッガ，および証明課題生成機能を持つ．Overture という VDM 向け Eclipse プラグインに用いられている．オープンソースであり，自由な改変が可能である．

3　上流工程におけるモデル実行

3.1　GUI を用いたモデル実行

仕様記述言語 VDM++ を用いたモデル実行の際，GUI を構築し，その GUI から VDM インタプリタを呼

† 信州大学大学院理工学系研究科，Graduate School of Science and Technology，Shinshu University.
†† 信州大学工学部，Faculty of Engineering，Shinshu University.

び出すことにより「要求に合致したシステムが作られているか」という妥当性をソフトウェア開発や VDM++ の知識を持たない実際のユーザによって確認することが出来る [3]．本研究では GUI は Java Servlet で構築されることを想定している．外部アプリケーションを Java Servlet で記述することは，Web ブラウザや Tomcat 等の既存のツールを用いることが出来るため，妥当性確認の際にかかるコストの削減が期待できる．更にネットワーク間で妥当性確認を行えるため，チェックを行うテスターに外部アプリケーションを配布する必要が無く，テスターは Web ブラウザを用意すればよい．内部的な通信に注目することの無いテスターからは，Java Servlet が動作しているかのように見え，Web ブラウザを用いて，より簡単に妥当性確認を行える [4]．また，妥当性確認を行う仕様として Web システムを想定している．

3.2 上流工程における妥当性確認の全体フロー

本研究で対象とする，上流工程における妥当性確認の全体フローを図 2 に示す．モデル実行の際，GUI を構築することは妥当性確認を容易にするが，妥当性確認のためのコード記述コストが増大するという欠点が存在する．そこで仕様記述の際に記述する，クラス図やアクティビティ図から，著者らが開発している Java Servlet スケルトンコード生成器で Java Servlet や HTML，JSP を半自動生成し記述コストを削減する．半自動生成されたコードは Web API を用いて VDMJWeb サービスと通信を行う．

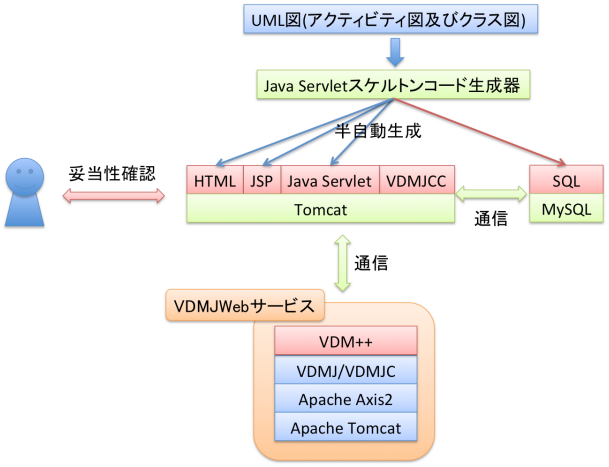

図 2 上流工程における妥当性確認の全体フロー

3.3 Java Servlet スケルトンコード生成器

Java Servlet スケルトンコード生成器とは著者らが開発している，UML アクティビティ図を用いて対象システムの画面遷移条件およびロジックフローを記述し，その成果物を用いて VDM++ で記述された仕様と一対一に対応した Java Servlet スケルトンコードを自動生成するツールである [5]．VDMJWeb サービスと通信を行う Java Servlet スケルトンコードを自動生成することで，プロトタイピングにおける記述コストを削減し，上位設計モデルの再利用性向上に寄与する．

4 Apache Axis2 を用いたモデル実行環境

4.1 SOAP

SOAP は XML ベースの RPC プロトコルである．拡張可能で分散的なフレームワークであり，データ構造のみが規定されているため，HTTP や SMTP など様々な通信プロトコルで利用することが出来る．メッセージの表現に XML を使用し，言語やプラットフォームに依存しない．また XML と HTTP を用いることによりファイアウォールを超えてオブジェクト間で通信することが出来る．主要な実装として Apache Axis2[6] が存在する．Apache Axis2 は Java と XML 技術に基づいた Web サービス/SOAP/WSDL エンジンである．

4.2 動作の仕組み

通信の流れや各自の役割分担を，図 3 に示す．VDM++ で記述された仕様は VDMJWeb サービス上で動作し，SOAP を用いて，Java Servlet と通信をする．Java Servlet は HTTP を用いて，Web ブラウザと通信をする．テスターはネットワーク越しに Web ブラウザを用い，妥当性を確認することが出来る．外部アプリケーションは，VDMJWeb サービスと通信するために提供されている Web API を利用する．外部アプリケーションは Web API で提供されているメソッドを介して VDMJWeb サービスに命令を送る．VDMJWeb サービスは命令の処理結果を外部アプリケーションへと送信する．通信は SOAP 及び HTTP を利用する．メッセージの表現に XML を使用しているため，オブジェクトの受け渡しが可能であり，命令の処理結果は VDMResult 型というオブジェクトで取得することが可能である．

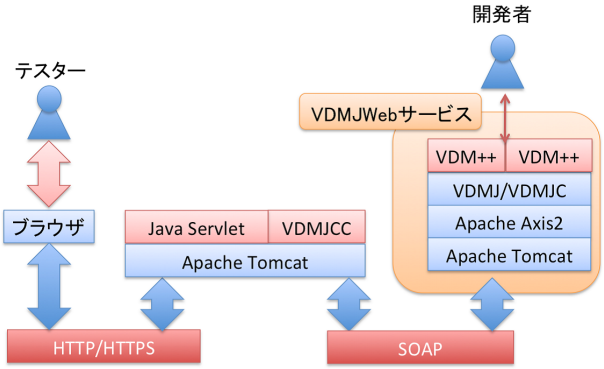

図 3 VDMJWeb サービスの動作の流れ

4.3 VDMJWeb サービスの構成

VDMJWeb サービスは，実際のインタプリタ機能を有する VDMJ，VDMJ のクライアントとして外部と通信を行うための機能が備わっている VDMJC から構成されている．VDMJ のクライアントである VDMJC を Apache Axis2 を用いて Web サービスとして公開する．Web コンテナとして Apache Tomcat を用いている．Web サービスの機能を利用するため提供されている Web API を用い，外部アプリケーションは VDMJWeb サービスと SOAP 通信を行うことが出来る．Web ブラウザから送られてくるリクエストは Tomcat 上で Java Servlet で受け，Web API を介して VDMJ サービスへ送られ処理される．VDMJ インタープリタによる処理

結果は，VDMJC から Tomcat 上に実装されたクライアント API(VDMJCC) が受信し，Java Servlet まで送られ Web ブラウザに表示される．その結果を見てユーザは妥当性確認を行う．ユーザが行った妥当性確認の結果を踏まえ，VDM++ を記述する開発者は再度 VDM++ を記述する．

4.4 コマンド一覧

使用可能なコマンドの一覧を表 1 に示す．

表 1 コマンド一覧

load	VDM++ ファイルをロードする
reload	ロード済みの VDM++ ファイルを再ロードする
unload	ロード済みの VDM++ ファイルを破棄する
create	クラスのインスタンスを生成する
print	VDM++ の式を実行，またはオブジェクトの関数，操作呼び出しを行う
classes	ロード済みのクラス一覧を取得する
init	インタプリタを初期化する
coverage	カバレッジを取得する
quit	インタプリタを終了する

現在 VDMJWeb サービスで対応しているコマンドは load, print などが存在する．妥当性確認を行うテスターは複数のコマンドを用いて VDMJWeb サービスと通信し，モデル実行をする．次節で具体例を交えたコマンド及び VDMJWeb サービスの利用方法を解説する．

4.5 Web API を用いた VDMJWeb サービスとの通信

VDMJWeb サービスと接続するためには，Java のクラス VDMJCC のコンストラクタを呼び出し，VDMJCC オブジェクトを用いなければならない．引数として，VDMJWeb サービスの stub，そしてスレッド内でユーザの判別用に用いるユニークな文字列を必要とする．このクラスはメソッド makeNewVDMJCThread を提供する．このメソッドを用いることにより VDMJWeb サービスにて，専用のスレッドが立てられ，処理を行うための準備が整う．サービス内でスレッドが立った後は，サービスの機能にアクセスすることが可能である．

List 1 コネクション開始時におけるインスタンス生成

```
Vdmjcc vdmjcc = new Vdmjcc(stub,name);
boolean bool = obj.makeNewThread();
```

List 2 VDMJC への処理指示コマンド

```
try{
  String{} cmd = {``load a.vdmpp'',``create a :=
       new a()''};
  obj.init(cmd);
  EvalMessage msg = obj.EvalCmd(``print a.op(1)'')
  System.out.println(``Result :'' + msg.getResult()
       .getResultInt())
}catch(VDMException e){
  System.err.println(e.getErrorLine());
  System.err.println(e.getErrorMsg());
  System.err.println(e.getErrorNumber());
}
```

サーバの機能にアクセスする場合通常，テスターは VDM++ ファイルを読み込むことから開始する．これはサーバー側に既にアップロード済みの VDM++ ファイルを load コマンドを用いることで出来る．その後実行のためのオブジェクトを生成するため create コマンドを用いる．init メソッドでは String の列を引数として受け取り，String で定義された命令式を順次実行する．同じように命令式を実行する方法として EvalCmd メソッドを使用する方法がある．EvalCmd メソッドは，VDMJWeb サービスへの命令を引数としてとる．返り値として結果やメッセージ等が格納されている EvalMsg オブジェクトを返す．EvalCmd は，例外として VDMException を投げることが出来る．またエラーオブジェクトを参照しエラーメッセージ，エラー番号，エラー発生箇所を取得することが出来る．

5 VDM モデル実行と妥当性確認例

5.1 ホテル予約システム

VDMJ サービスの評価を行うため，ホテル予約システム (図 4) を作成した．一般的なホテル予約システムは，宿泊予約，宿泊予約変更，宿泊予約取り消し等様々な機能があるがここでは「宿泊予約」のみに注目してプロトタイピングを行う．主な要求として以下のものが挙げられる．

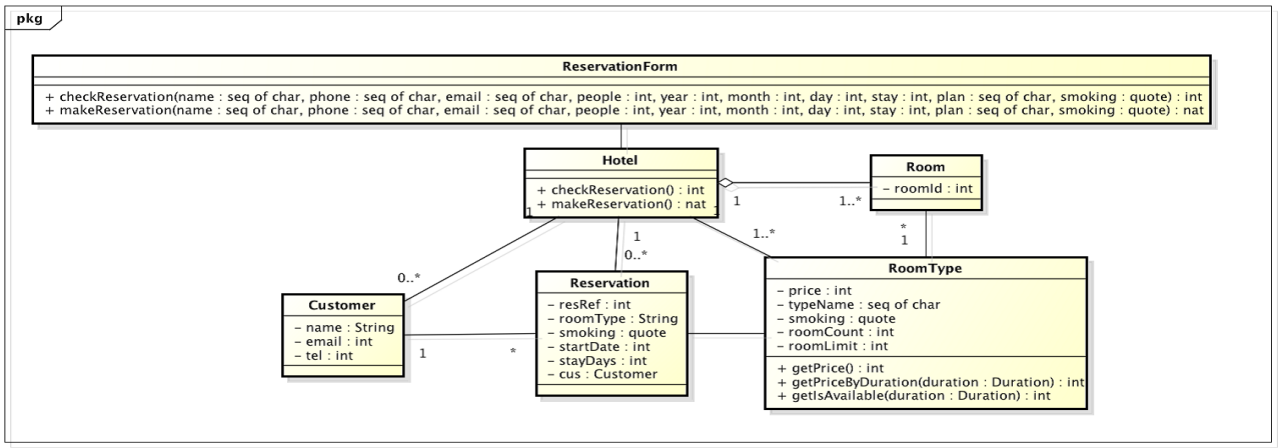

図 4 ホテル予約システムのクラス図

- ユーザは，氏名，メールアドレス，電話番号，宿泊日数，チェックイン予定日を入力し，宿泊プランと禁煙部屋か否かを選択する．
- ユーザは予約内容を確認後，予約を確定する．
- 宿泊プランは 5 つある．シングル，ツイン，ダブル，デラックスツイン，そしてスイートである．
- ホテルの予約は予約日から 3 日前とする．
- 日帰り (0 泊) はできない．
- 禁煙ルームと喫煙可能ルームが選択できる
- チェックイン時に顧客を判別できるようシステムは予約番号を発行する

アクティビティ図から生成したスケルトンコードに追記を行い，モデル実行を行った．図 5 にツールを直接用いた妥当性確認，図 6 に外部アプリケーションを利用した妥当性確認の例を示す．外部アプリケーションを用いることにより，妥当性確認が容易かつ直観的に行えることが確認できた．

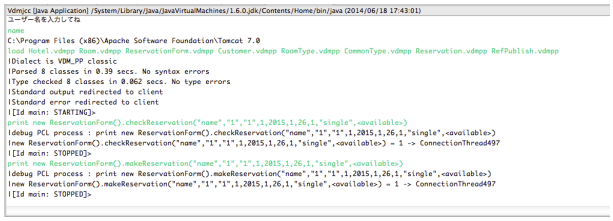

図 5　CUI クライアントを用いた妥当性確認

図 6　GUI(ブラウザ) を用いた妥当性確認

5.2　実行性能評価実験

VDMJWeb サービスの処理性能を確認するため，単純な処理を大量に行うものと，複雑な処理を行った．実行計算機の性能は OS : Mac OSX 10.9，CPU : Intel Core i5-3210M 2.5GHz，メモリ : 8.0GB である．

List 3　引数に 1 を足して返す

```
public op : int ==> int
op(dx) == return dx+1;
```

リスト 3 に示す，ごく単純な処理を百回試行した結果，処理終了までの時間は 4,026ms であった．つまり一回当たりの処理時間として，おおよそ 40ms かかることがわかった．

List 4　冪集合の取得

```
public getAvailableSeats : EventID * nat1 ==> set
    of Seat
getAvailableSeats(eventID, num) ==
  let s in set power getEventSeats(eventID) \
      getReservedSeats(eventID) be st card s = num
      in return s
```

リスト 4 の，座席集合から 2 席の冪集合を取得する処理を試行した結果を図 7 に示す．総座席数の増大によって，実行時間が指数的に増えていく．VDMJWeb サービスにおいて，冪集合の取得は不向きであることがわかった．

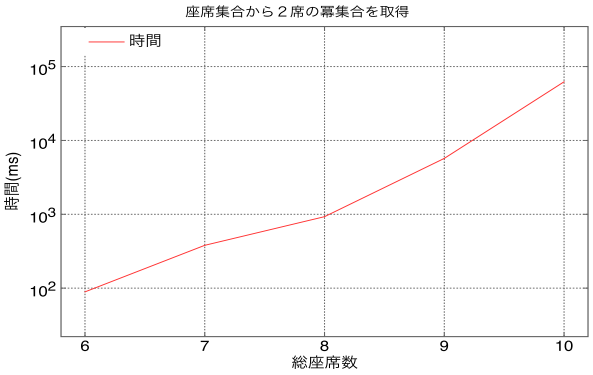

図 7　冪集合取得を VDMJWeb サービスで実行した場合の例

6　まとめと今後の課題

VDMJ を Web サービスとして公開し，Apache Axis2 を用いた SOAP 通信を外部ツールと行うことが出来た．またホテル予約システムの仕様と Java Servlet スケルトンコード生成器によって生成された外部アプリケーションを用いて，VDMJWeb サービスの評価を行った．また複数の処理を行い性能を確認した．

Web サービスとして公開するにあたり，複数のユーザからの処理に対応するため，サービスを利用するユーザの使用メモリや連続使用時間等を管理する機能を導入する必要がある．また現時点では処理を中断するたびに，最初から処理をやり直さなければならないため，複雑な処理や，下準備が必要な処理には不向きである．そこで処理途中で中断した場合，どの VDM ファイルがロードされているか，どのようなインスタンスが生成されているかなどをデータベースに保存し，再読み込み出来る機能の実装を考えている．

参考文献

[1] 石川冬樹 : VDM++ による形式仕様記述，近代科学社，2011.
[2] VDMJ — Free Development software downloads at SourceForge.net, http://sourceforge.net/projects/vdmj/
[3] ジョン・フィッツジェラルド，他 : VDM++ によるオブジェクト指向システムの高品質設計と検証，翔泳社，2010.
[4] 村林慧，和崎克己 : 仕様記述言語 VDM++ と Java Servlet を用いた Web アプリケーションのプロトタイピング ; 平成 25 年度電気関係学会東海支部連合大会講演論文集，(O3-1)，1page(CD-ROM)，2013.
[5] 多田圭佑，和崎克己 : VDM++ を用いたラピッドプロトタイピング向けの Java スケルトンコード生成器 ; 平成 26 年度電気関係学会東海支部連合大会講演予定.
[6] Apache Axis2 - Apache Axis2/Java - Next Generation Web Services, http://http://axis.apache.org/axis2/java/core/