社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005－SE－147（1）
2005／3／17

UML記述にUI設計を連携させた開発支援ツールの提案

銅島　康　　上田賀一
茨城大学 工学部 情報工学科

現在のソフトウェア開発では，設計にはUMLを用いるのが一般的である．設計作業にはUML設計とUI設計を含むにもかかわらず，最初からUML設計にUI設計を取り入れた設計作業を支援するようなツールは見当たらない．そこで本研究では，UML設計とUI設計の関連情報にUML設計で用いるメソッドとUI設計で用いるウィジットのリンクを取り上げ，これらの関連情報を入力できるような設計支援ツールを提案する．本支援ツールにより，UML設計とUI設計の関連情報を入力しUML設計の中に残すことができる．その上で設計作業することで，外部仕様と内部仕様のつなぎ目の検討漏れなどを防ぐことができ，設計の精度を高めることができる．

A Support Tool to Make UI Design Cooperate in UML Description

Yasushi Doujima　　Yoshikazu Ueda
Ibaraki University

In current software development, it is general to use UML for the design. Though UML design and UI design are contained in the design work, we have not found the tool which supports the design work that UI design was adopted in the UML design from the beginning. So, we take up the link of a method in the UML design and a widget in the UI design as a related information of the UML design and the UI design, and propose the design support tool which can input this related information in this research. This support tool can keep the input of related information in the UML design. By using this tool, we can prevent the examination omission of the item which becomes the joint of the external specifications and the internal specifications and enhance the precision of the design.

1　はじめに

オブジェクト指向ソフトウェア開発において，分析仕様や設計仕様の記法としてUMLを用いることが一般的となってきている．UMLは，多様な側面からの記述を可能としており，そのため様々な図式が取り入れられている．UMLを用いたソフトウェア設計をここではUML設計と呼ぶことにする．その一方で，ビジュアルプログラミング言語が，ラピッドアプリケーション開発やプロトタイピング手法に広く用いられ，エンドユーザコンピューティングにもその利用が向けられていた．特に，GUIがソフトウェアの当然の機能として取り入れられるようになると，自ずとGUI設計も重要な作業になってきたと言える．ビジュアルプログラミング言語による開発やGUI設計がMVCモデルのViewモデル主導の開発や設計であると捉えると，UML設計はModelモデル主導の設計であると言える．Modelモデル主導であろうが，Viewモデル主導であろうが，最終的にはModel, View, Controllの３つのモデルを完成させなければならない．特定の側面から仕様を固めていく方法，全側面の仕様を同時に少しずつ固めていく方法の両方が，状況に応じて使い分けられるものと考えられる．にも拘らず，前者の支援ツールは多く認められるが，後者の支援ツールはあまり見い出されなかった．

そこで本研究では，GUIに限定せず，UI設計と

いう視点に広げ，UML 設計と UI 設計をもっと連携させて，同時並行的に設計を進められる開発支援ツールを検討する．具体的には本研究では，UML 設計においてアクタとやりとりするメッセージだけでなく，細かい UI 形式まで入力できるようなツールを構築する．また，そこから副次的に，連携を利用して，既存の UML 設計支援ツールではスケルトンプログラムとして生成しない部分を少しでもより多く生成するような開発支援ツールの構築を目指す．

2　UML と UI の連携に関する検討

UML 記述と UI 設計を連携させるにあたり，代表的な UML 設計支援ツールの特徴を調べ，UML 図の各ダイアグラムが UI 情報と関わるか検討する．

2.1　UML 設計支援ツール

現在，UML 設計支援ツールは数多く存在する（表 1 参照）．

これらの UML 設計支援ツールは，UML 図を大別した構造図，振舞い図，実装図の観点では当然，記述可能となっており，スケルトンプログラムを生成できるものが多い．しかし，いずれのツールも UI 設計を支援する機能を持たないものであった．また，UI 設計支援ツールにおいても，GUI を中心としたスケルトンプログラムを生成する機能はあるが，UML 設計を支援することはなく，相互を連携させるものは見当たらなかった．UML 記述と UI 設計は互いに独立したものではなく関連性があることから，その関連情報を入力することで，より価値の高いスケルトンプログラムを生成できると考えられる．

2.2　UML 図と UI 情報の関わり

UML 図それぞれに対して GUI 情報，CUI 情報との連携を考えてみる [14]．

• ユースケース図
1 つ以上のアクタとシステムとの間で交換される一連のメッセージは，UI 情報との間で連携させることができる．

• クラス図
クラス間の関連中には，通常，交換される一連のメッセージを記述しないので，UI 情報との間で連携を行うことはない．

• ステートチャート図
通常，1 つのモデル実体の振舞いを記述し，モデル実体間で交換される一連のメッセージを記述しないので，UI 情報との間で連携を行うことはない．

• アクティビティ図
アクティビティの担当者をレーンによって分けることができる．それによりアクタを明示的に記述できるため，アクタとそれ以外のモデル間の状態遷移と，UI 情報との間で連携を行うことができる．

• シーケンス図
アクタとオブジェクトとの間で交換される一連のメッセージと，UI 情報との間で連携を行うことができる．

• 協調図
アクタとオブジェクトとの間で交換される一連のメッセージと，UI 情報との間で連携を行うことができる．

• コンポーネント図
コンポーネントは，分類子として操作を持ちインタフェースを実現する．このインタフェースと，UI 情報との間で連携を行うことができる．

• 配置図
コンポーネント図と同様に，コンポーネントは，分類子として操作を持ちインタフェースを実現する．このインタフェースと，UI 情報との間で連携を行うことができる．

以上をまとめると表 2 のようになる．

3　本支援ツールの方針

今回構築したツールでは，連携に基づいて詳細なスケルトンプログラムを生成するにあたって，最も有効であると予想されるシーケンス図と GUI 情報との連携について考える．

表 1: UML 設計支援ツールの比較

	入力					出力
	構造図	振舞い図	実装図	GUI 設計	CUI 設計	
Rational Rose	○	○	○	×	×	○
Konesa	○	○	○	×	×	○
Pattern Weaver	○	○	○	×	×	×
WithClass	○	○	○	×	×	○
Describe	○	○	○	×	×	○
Together Control Center	○	○	○	×	×	○
Jude	○	○	○	×	×	○
Enterprise Architect	○	○	○	×	×	○
Poseidon	○	○	○	×	×	○
EclipseUML	○	○	○	×	×	○
野村らのツール [9]	△	△	×	×	×	○

表 2: 連携適用可能 UML 図

UML 図	連携
ユースケース図	○
クラス図	×
ステートチャート図	×
アクティビティ図	○
シーケンス図	○
協調図	○
コンポーネント図	○
配置図	○

表 3: シーケンス図

UML 図要素	保持情報
アクタ	名前
オブジェクト	名前
	タイプ
メッセージ	ラベル
	操作
	作成
	破壊
	リターン矢印
	ステートメント
	条件
	同期
	アトミックデリバリィ
セルフメッセージ	ラベル
	操作
	リターン矢印
	同期
指示	

3.1 UML 図

UML 図の入力可能情報については，EclipseUML とほぼ同じ形式にする [11][12].

シーケンス図についての要素と，その要素が保持する情報を表 3 に示す.

3.2 UI 情報

GUI 情報の入力可能情報については，Visual Editor とほぼ同じ形式にする [13]．GUI 設計要素とそれが保持する情報を表 4 に示す.

入力可能な CUI 情報については，データ入力時は，コマンド引数の有無，出力するプロンプト形式の指定を，データ出力時は，出力形式の指定を入力可能とする.

デフォルトとして，変数名とその変数の保持する値を出力する形式 (変数名:値) を用意する．また，出力形式を他にもいくつか用意しておき，ユーザの出力形式の指定を簡易化する.

3.3 UML 設計と UI 設計の関連情報

UML 設計と UI 設計の関連付けを行う．その関連付けを関連情報として取り扱い，今回の研究では，UML 設計で用いるメソッドと UI 設計で用いるウィジットのリンクのことである.

3.4 UML 設計と UI 設計の関連情報の入力順序

連携するにあたって必要な UML 設計と UI 設計の関連情報をユーザが入力する順序は，シーケンス図と GUI 情報の関連情報の入力順序に限定して

表 4: GUI 情報

GUI 設計要素	保持情報
Selection	
Marquee	
Choose Bean	
Swing Components	JButton
	JCheckBox
	JRadioButton
	JToggleButton
	JLabel
	JTextField
	JTextArea
	JSlider
Swing Containers	JFrame
	JPanel
	JScrollPane
	JSplitPane
	JTabbedPane
	JDesktopPane
	JInternalFrame
	JDialog

考えると，以下に示す 3 つの方法が考えられる．

1. 先にシーケンス図を作成しておき，その後に GUI 情報を取り入れる．
2. 先に GUI 情報を作成しておき，その後にシーケンス図を作成しつつ，その中でアクタとオブジェクト間のメッセージと，先に作成しておいた GUI 情報との関連付けを行う．
3. 先に GUI 情報，シーケンス図を別々に作成しておき，その後にアクタとオブジェクト間のメッセージと，GUI 情報との関連付けを行う．

1，2 は構築するソフトウェアの全体像を把握できている開発者向けの方法である．逆に 3 は，アクタとオブジェクト間のメッセージと，GUI 情報との関連を模索しながらソフトウェア開発を行う開発者向けの方法である．ここでは，今までのツールを利用してきた開発者でも利用し易く，構築するソフトウェアの全体像を把握していなくても利用できると考えられる 3 の順序について考える．

4 本支援ツールの設計

本研究では，連携の結果が分かりやすいと考えられることから，人が介入するようなシステムの設計機能のみを構築した．ここでは，本支援ツールの主要な機能を示し，シーケンス図と GUI 情報を連携させる手順を説明する．

4.1 機能

本研究で構築するツールの主な機能を以下に示す．

- UML 設計機能
  クラス図とシーケンス図の作成機能．
- GUI 設計機能
  GUI 設計を行う機能．
- CUI 設計機能
  CUI 設計を行う機能．人が介入しないようなシステムの UI 設計やネットワーク通信部分の UI 設計も行うことができる．
- UML 設計と UI 設計の関連情報入力機能
  ［UML 設計と GUI 設計の関連］
  例えば UML 設計中のアクタとのメッセージをやりとりする記述部分に，ダイアログボックスやラジオボタンの利用を指定できる．ユーザがボタンなどのウィジットを生成した場合，そのウィジットに対して押下などのアクションを行った時に実行するメソッドを指定できる．
  ［UML 設計と CUI 設計の関連］
  例えばシーケンス図でアクタからオブジェクトへ送信する場合，その送信するメッセージの型を指定でき，送信前のメッセージ入力プロンプトをテキスト形式で入力し，指定することができる．
- スケルトンプログラム生成機能
  対象システムを表す UML 設計と UI 設計の連携から，既存の UML 設計支援ツールがスケルトンプログラムとして生成しない部分の Java スケルトンプログラムを提供する．

4.2 シーケンス図と GUI 情報の連携

シーケンス図でアクタとオブジェクト間のメッセージを指定した後に，そのメッセージと UI 設計で作成したウィジットを関連付けることができる．その時に，メッセージの送信元オブジェクトは送信先オブジェクトのアドレス A を知っている必要が

ある．ただしここで述べているオブジェクトとはアクタ-オブジェクト間のインタフェース，つまり隠されたインタフェース・オブジェクトと，シーケンス図上に記述されているインタフェース・オブジェクトのことである．

送信元オブジェクトが送信先オブジェクトのアドレス A を知らない場合，そのアドレス A を尋ねるオブジェクトをユーザに指定させる．指定先のオブジェクトがアドレス A を知らない場合，そこからさらにアドレス A を尋ねるオブジェクトをユーザに指定させる．このようにアドレス A の流れる経路をユーザに指定させる．

具体的には，

1. 送信元オブジェクト（Class B）が，送信先オブジェクトのアドレス A を得るためにユーザが指定したオブジェクト（Class C）の保持するメソッドを呼び出す．
2. 指定先のオブジェクト（Class C）がアドレス A を知っている場合はそこで現在の処理を終了し，アドレス A を知らない場合はそこからさらにアドレス A を尋ねるオブジェクト（Class D）を指定させる．
3. 2 を再帰的に行う．ただし，ユーザが 1〜3 の処理を拒否した場合は即座に現在の処理を終了し，またユーザが指定したオブジェクトがどのオブジェクトアドレスも知らない場合も終了する．

以上のアルゴリズムを利用しコード生成を行う．

4.3 本支援ツールの UML 設計

本支援ツール自体の UML 設計のクラス図を図 1 に，シーケンス図を図 2 に示す．

4.4 実装環境

開発環境には WindowsXP を用いた．また，より多くの開発者が本支援ツールを利用でき，本支援ツール自体がプラットフォームに縛られないよう開発言語に Java を用いた．

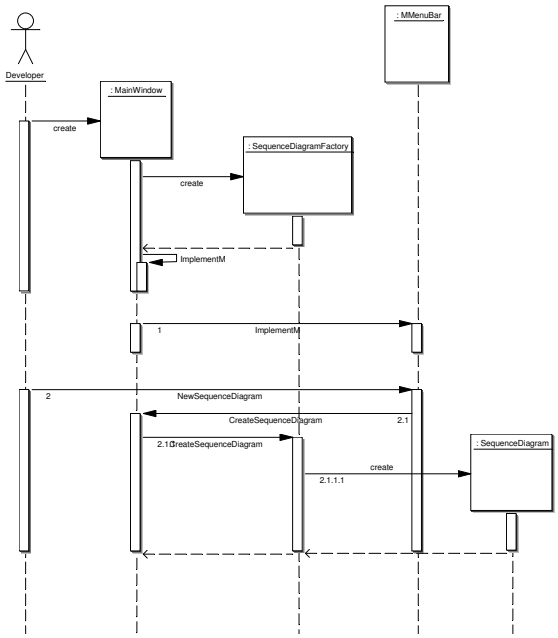

図 2: 本支援ツールのシーケンス図

5 適用例

ここでは，ビデオレンタルシステムを例に取り上げ，本支援ツールを用いたソフトウェア開発の流れに沿いながら実行の様子や結果を示す．

5.1 UML 設計と GUI 設計

ビデオレンタルシステムのユースケース図，クラス図，シーケンス図をそれぞれ，図 3〜5 に示す．

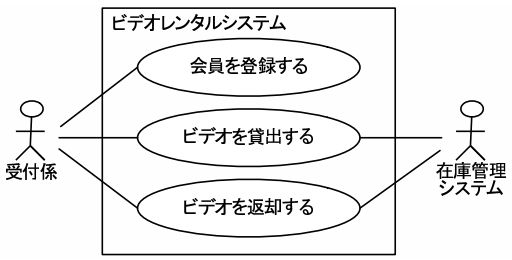

図 3: ビデオレンタルシステム ユースケース図

また，本ツールでシーケンス図と GUI 設計の関連情報を入力している画面を図 6 に示す．これまでの設計支援ツールでは UML 設計と UI 設計が連携していないため，支援ツール上では取り扱われず，開発者の思考の中だけで整理しければならなかった関連情報を扱うことができる．シーケンス図，GUI 設計図に加え，関連情報の表示によって，考えることと書くことの一体化による思考効率の

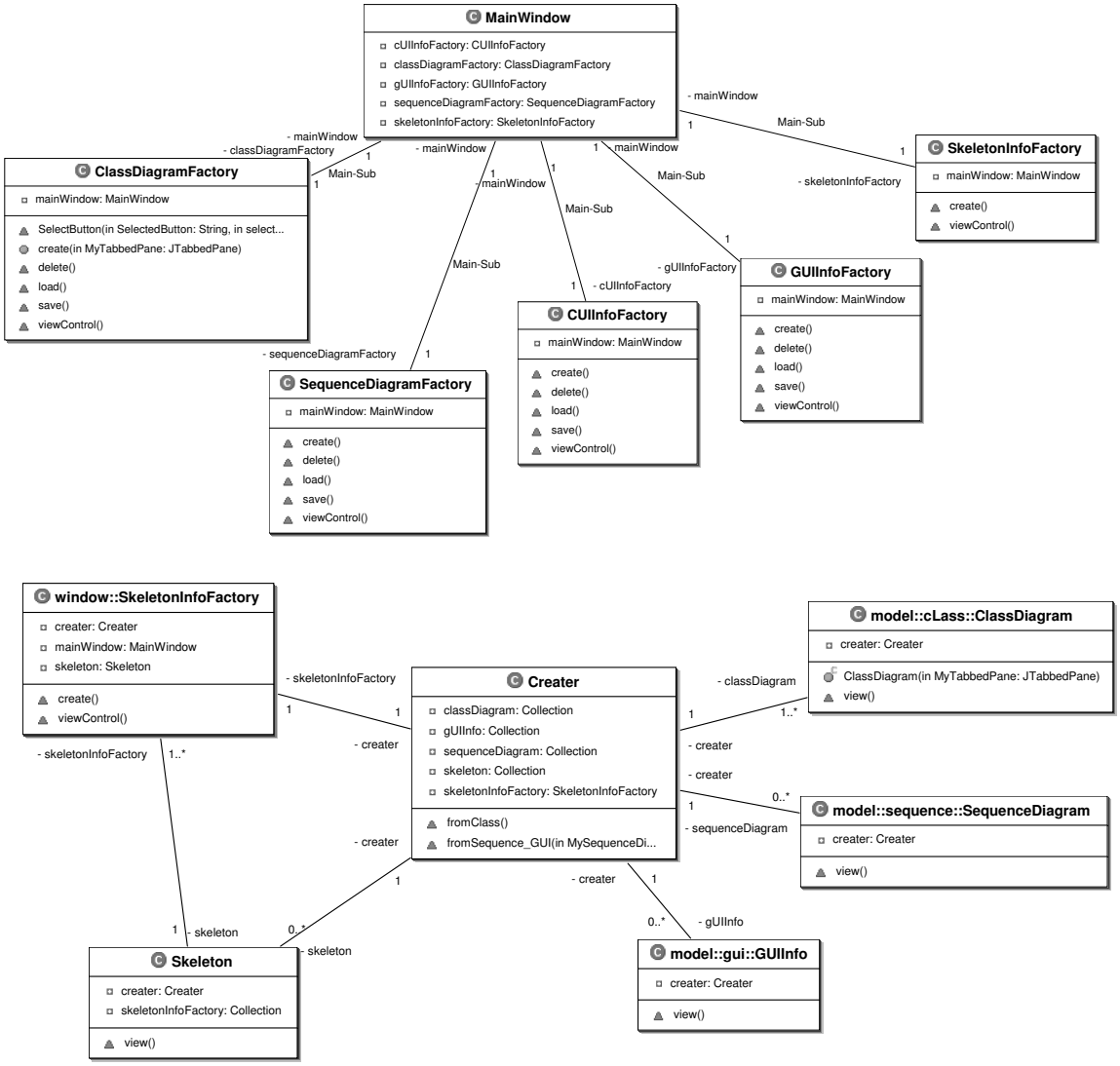

図 1: 本支援ツールのクラス図

向上が図られ，設計を円滑に進めることができると考えられる．

5.2 スケルトンプログラム生成

本支援ツールのスケルトンプログラム生成機能から Java のスケルトンプログラムを生成する．クラス図が保持する情報からはクラスの構造上のコードが生成され，シーケンス図と GUI 情報の関連情報からはシーケンス図で指定したメソッドと GUI 設計で用いるウィジットのリンクコードが生成される．例として,「ビデオレンタルシステム」クラスのスケルトンプログラムを図 7 に，リンクコードが埋め込まれるインタフェースクラス「ダイアログボックス」のプログラムを図 8 に示す．

```
1
2  package VRS;
3
4  public class ビデオレンタルシステム {
5
6  public void 会員を登録する (){
7  }
8
9  public void ビデオを貸出する (){
10 }
11
12 public void ビデオを返却する (){
13 }
14
15 public void 会員番号を発番する (){
16 }
17
18 public void 貸出内容を表示する (){
19 }
20 …
21 }
```

図 7: 「ビデオレンタルシステム」のスケルトンプログラム

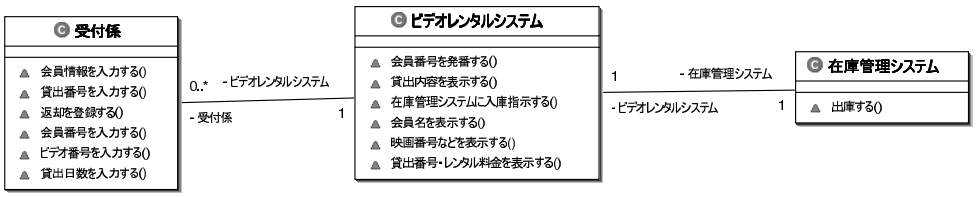

図 4: ビデオレンタルシステム クラス図

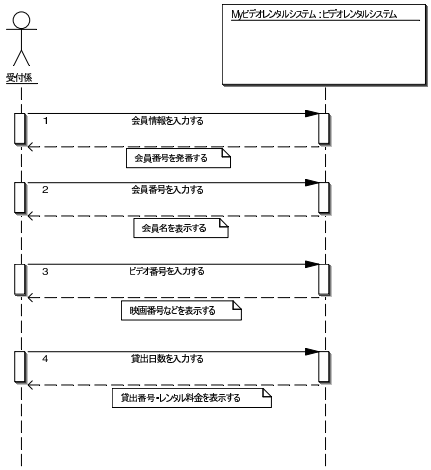

図 5: ビデオレンタルシステム シーケンス図

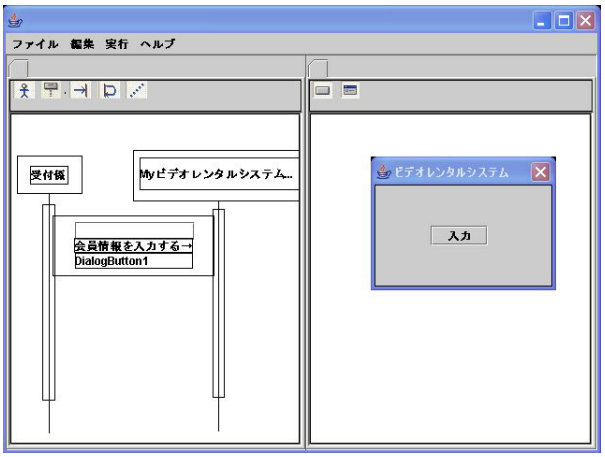

図 6: シーケンス図と GUI 設計の関連情報入力画面

```
1
2  package VideoRentalSystem;
3
4  import java.awt.Dialog;
5
6  public class DialogBox extends Dialog {
7
8  private javax.swing.JButton jButton = null;
9
10 public DialogBox(java.awt.Frame owner) {
11 super(owner);
12 initialize();
13 }
14 ...
15 private javax.swing.JButton getJButton() {
16 if(jButton == null) {
17 jButton = new javax.swing.JButton();
18 jButton.setBounds(73, 101, 149, 27);
19 jButton.setText("入力");
20 }
21 return jButton;
22 }
23
24 public void 会員情報を入力する (){
25
26 MyVideoRentalSystem. 会員情報を入力する ();
27
28 }
29 ...
30 }
```

図 8: 「ダイアログボックス」のプログラム

6 まとめ

本研究では，UML 設計と UI 設計の関連情報を入力することができ，そこから Java のスケルトンプログラムを生成する設計支援ツールを提案した．今回は，関連情報に基づいてスケルトンプログラムを生成するにあたって，最も有効であると予想されるシーケンス図と GUI 情報との連携に着目し，その連携に関する設計機能のみを構築した．

本支援ツールにより，UML 設計と UI 設計の関連情報を入力し UML 設計の中に残すことができる．これは外部仕様と内部仕様のつなぎ目の検討漏れがないか，あるいは MVC モデルの M モデルと V モデルを結び付ける C モデルを記述，確認するといった作業になると考えられる．

このような支援ツール上で設計することで，全体や各部の整合性は確保できているかなど，頭の中だけでは整理しずらい同時並行的に進める幾つかのソフトウェア側面を記述・記録することで整理でき，設計を円滑に進められると考えられる．

また，思考しながらのソフトウェア側面の有機的な結び付きを設計していくスタイルは，アジャ

イル開発に向いた設計支援ツールとしていくことができるのではないかと考えている．

実装に関して，今回はシーケンス図，GUI 情報，シーケンス図と GUI 設計の関連情報を扱うことに留まったが，今後の課題として，人の介入に関わらず，システムのインタフェース部分を作成可能にすること，クラス図，シーケンス図以外の UML 図の入力を可能にすることが上げられる．

参考文献

[1] IBM japan. http://www.ibm.com/jp/.

[2] Ogis-ri object & network group. http://www.ogis-ri.co.jp/otc/index.html.

[3] Technologic arts incorporated, 2001. http://www.tech-arts.co.jp/index.html.

[4] Grapecity.com. http://www.grapecity.com/japan/.

[5] 日揮情報ソフトウェア株式会社. http://www.jsys-products.com/.

[6] Jude uml modeling tool. http://www.esm.jp/jude-web/index.html.

[7] Sparx systems, 2002. http://www.sparxsystems.jp/.

[8] Gentleware. just model, 2000. http://www.gentleware.com/.

[9] 野村 幸司, 小飼 敬, 上田 賀一：アクティビティ図主導のモデリング支援ツールの開発, 情報処理学会, 研究報告 (SE), Vol.2004, No.30, pp.25-32 (2004).

[10] isid. http://www.isid.co.jp/index.html.

[11] eclipse.org, November 2001. http://www.eclipse.org/.

[12] Omondo, 2002. http://www.eclipseuml.com/.

[13] エクリプス. http://eclipsewiki.net/eclipse/.

[14] Object Management Group 著, OMG Japan SIG 翻訳委員会 UML 作業部会訳. UML 仕様書. ASCII, 初版, Nov. 2001.