

3次元空間を利用したビジュアルな要求仕様の変更管理支援

富 哲男、糸賀 裕弥、大西 淳
立命館大学大学院理工学研究科情報理工学専攻
[tomite|itoga|ohnishi]@selab.is.ritsume.ac.jp

ビジュアルなソフトウェア要求仕様の変更管理支援手法を提案する。ビジュアルな要求仕様の差異を動的に確認するために、ビジュアルな要求定義環境の構成システムの一つであるアニメーションシステムを利用して、ビジュアルな要求の動作の違いを比較できる。また、3次元空間を利用することにより、ビジュアルな要求仕様に表された実体や実体間の関連を容易に把握できる。具体例を用いて手法を紹介する。

Management of Visual Requirements Description with three-dimensional space

Tetsuo TOMI, Hiroya ITOGA, and Atsushi OHNISHI
Department of Computer Science, Ritsumeikan University

This paper describes a management method of visual requirements description. The differences between dynamic behaviors of two visual requirements descriptions can be visualized by using animation system. Moreover, entities and relations among entities in a visual requirements specification can be easily recognized with three-dimensional space. The proposed method will be illustrated with examples.

1. はじめに

システム開発の過程で要求はしばしば変更される。また、過去に開発した要求を用いて新たな開発をする場合もある。現在開発中の要求仕様や過去に開発した要求仕様を管理し、どのように要求仕様に変更されたのかを確認することは要求仕様に誤りが混入することを防ぐのに役立つ。特に、今までに作成された要求仕様がどのように変更されてきたかを確認し、仕様間の差分を簡単に把握できれば、仕様の変更管理が容易となる。

本研究室では、文章だけでなく視覚的にわか

りやすいビジュアルな要求言語[1] (VRDL: Visual Requirements Description Language) と、その処理系から構成されるビジュアルな要求定義環境を開発してきた。フロー要求は文章より図で表したほうが分かりやすく、コンピュータやシステムに関する知識を持たないユーザでも理解しやすいことから、ビジュアルな要求定義環境での記述対象はフロー要求に限定している。ビジュアル化されたフロー要求により、開発者とユーザ間でコミュニケーションが取りやすくなり、効率的にソフトウェアの開発が行える。しかしながら、仕様の変更の履歴や変更箇所を





確認することは難しいという問題があった。本稿ではビジュアルな要求間の動的な差分を容易に確認できる手段を提供することによって、上記の問題点の解決を図る。

2. ビジュアルな要求定義環境

2.1 アイコンと矢印によるシーン

要求記述者にとって自分が思い描いたイメージに合った記号や図を表現でき、要求記述に用いることができれば、視覚的にも分かりやすく、他の人が見ても理解しやすくなる。ビジュアルな要求定義環境では、アイコンの形状を定義できるだけでなく、要求記述に現れる名詞や名詞の型、名称をアイコンの意味として定義できる。アイコンの具体例を表1に示す。

表1 アイコンの具体例

アイコンの意味	アイコンの形状	アイコン役割
お客		データの源泉・吸収
企業		データの源泉・吸収
コンピュータ		やり取りの手段
注文情報		やり取りするデータ

ビジュアルな要求仕様では、表1のようなアイコンを用いて、データの源泉・吸収となるアイコンを配置し、そのアイコン間を矢印で結ぶ。矢印は、データフローがどのように流れるかを表すものであり、始点はデータの源泉、終点は

データの吸収を示す。そして、矢印上に Device 型と Data 型のアイコンを配置する。この作業を繰り返し行うことによって要求を記述していく。記述はひとまとまりの作業を対象とし、それをシーンと呼んでいる。ビジュアルな要求記述例を図1に示す。これは”商品の注文”を対象としている。

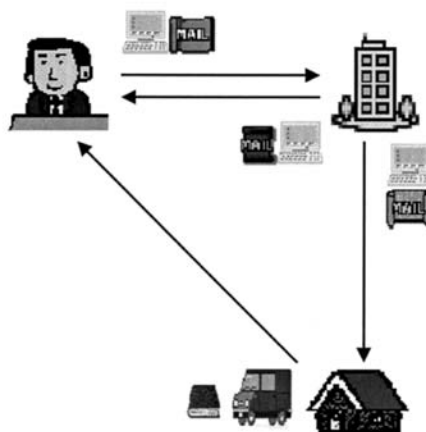


図1 ビジュアルな要求記述(商品の注文)

”商品の注文”のデータフローは、まずお客が企業に何か(例えば書籍)の注文を行う。そして、企業がその注文をもとに出荷指示書を作成し、倉庫へ送り、お客には出荷連絡をする。最後に倉庫からお客へ商品を配送するという流れである。

図1で示したようなビジュアルな要求記述では、記述者のイメージに合ったアイコンを用いて表現でき、視覚的にも分かりやすいので複雑な要求の場合でも、そのシーンをイメージしやすく理解しやすい。実際にフロー要求の中に登場する建物などの写真をアイコンとして利用することにより、記述者はシーンを自分のイメージ通りに記述できると共に、読者はビジュアルな要求を正確に理解できるようになる。

2.2 ビジュアルな要求定義環境

ビジュアルな要求定義環境は、シーンエディタ、シナリオエディタ、シーンフロー図エディタ、アニメーションシステムの4つのシステムからなる。

シーンエディタでは、アイコンと矢印を配置してシーンを記述する。シーンは特定の業務を表し、1つのシーンには複数のシナリオが対応する。ここでのシナリオはシーンに記述されたデータフローを抜き出し時間的な生起順序を明示したものである。シナリオエディタではシーンエディタで記述したシーンをもとにデータフローの生起順序を定義し、シナリオファイルを作成する。シーンフロー図エディタは、システム全体の処理の流れを表現するために、シーン間の生起順序を定義できる。アニメーションシステムは、シーンとシナリオファイルを解釈し、データフローの生起順序に従って、データの流れる様子をアニメーションとして表示する。各システム間の関連図を図2に示す。

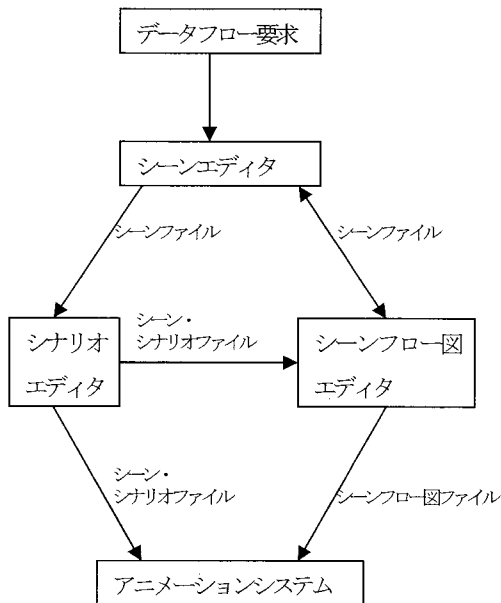


図2 各システム間の関連図

3. ビジュアルな要求のアニメーションシステムによる変更管理支援

3.1 概要

ビジュアルな要求の静的な差分は2つの仕様を対比させて見せれば直感的に把握できる。本稿では特にビジュアルな要求の動的な差分の把握を目標としている。アニメーションシステムを用いて比較したい、それぞれのシーンが持つシナリオを解釈実行することによって、動的にシナリオの差分をアニメーションとしてユーザーに確認させることができれば、視覚的にわかりやすく、直感的に理解できる。これにより、ユーザーは要求定義の差異部分、共通部分を容易に確認できる。

3.2 シーンを比較する手法

アニメーションシステムを用いて動的な振る舞いの差分を比較するためには、それぞれのシーンに対応するシナリオを解析し、ユーザーが差分を確認しやすいようにしなければならない。これを実現するために以下に述べる手法を用いる。比較するシーンが2つの場合と3つ以上の場合で異なる処理をおこなう。また、シナリオに条件分岐が含まれているとシーンを比較することが困難なので、条件分岐は最も選択されやすい条件が選択されると仮定し、条件分岐を含まないシナリオに変換した上で比較する。

シナリオの内容が同じであると判断するには、データの源泉・吸収、そして流れるデータとそのデバイスの定義されている意味がそれぞれ一致していることが条件である。ただし、違う意味を持つアイコンでも、ユーザーはそれらのアイコンを同じ意味として対応づけることにより、一致しているものとみなすことができる。

■ 2つのシーンを比較する場合

2つのシーンに対応するシナリオの違いには

シナリオ 1 (比較もと)

シナリオ 2

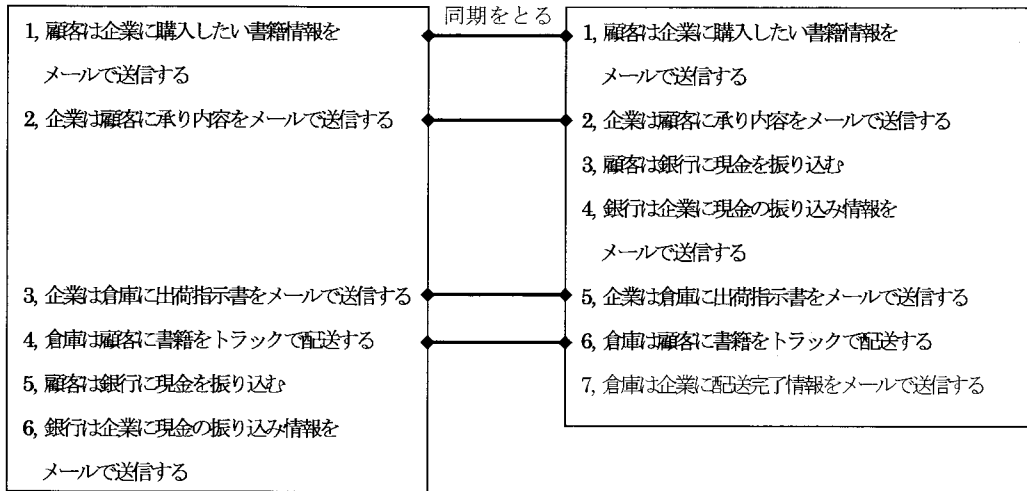


図 4 異なるシナリオ例 (2)

シナリオ 1 (比較もと)

- 1, 顧客は企業に購入したい書籍情報をメールで送信する
- 2, 企業が顧客に承り内容をメールで送信する
- 3, 企業は倉庫に出荷指示書をメールで送信する
- 4, 倉庫が顧客に書籍をトラックで配送する

シナリオ 2

- 1, 顧客は企業に購入したい書籍情報をメールで送信する
- 2, 企業が顧客に承り内容をメールで送信する
- 3, 顧客は銀行に現金を振り込む
- 4, 銀行は企業に現金の振り込み情報をメールで送信する
- 5, 企業は倉庫に出荷指示書をメールで送信する
- 6, 倉庫が顧客に書籍をトラックで配送する

図 3 異なるシナリオ例 (1)

以下のようなものが挙げられる。

・比較もとと同じシナリオ部分の生起順序に変化はなく、新たなデータフローが挿入、又は削除されているだけの場合

このシナリオの例を図 3 に示す。図 3 の下方のシナリオ 2 は比較もとのシナリオ 1 に新たなデータフローが挿入されているだけである。こ

のような場合、同じ内容のデータフロー部分を同時にアニメーションで流し、2 つのシナリオ間での共通部分を確認できるようにする。また、シナリオが異なる部分に達すると、ユーザに“異なるデータフロー部分です”と注意を促し、データフローの変更部分を確認させる。

・比較もとと同じシナリオ部分があってもその生起順序が異なる場合

このシナリオの例を図 4 に示す。このような場合は比較もとのシナリオを基本にして同期をとってアニメーションを流す。比較もとのシナリオを基本に同期をとることにより、比較もとと同じシナリオで、生起順序に変化がない共通部分を確認することができる。しかし、これでは比較もとと同じシナリオ部分があってもその生起順序が変化しているとデータフローの違いとして表れてしまう。この問題を解決するために各シーンのデータフローの生起順序が書かれた図 4 のようなテキストを画面に表示させる。このときに同じシナリオ部分の色を変え、どのように同期をとっているかを表示することにより、シナリオ間の関連を確認する。また、2 つ

のシーン間で共通のデータフロー部分が多数あった場合でも生起順序の異なる部分が多いとアニメーションで同期をとれる部分が少なくなってしまうので、シナリオの差分が分かりづらい。

そこで、2つのシーン間で共通のシナリオ部分を、比較もとのシーンのデータフローの生起順序で再生できるようにする。これにより、シーン間で共通のデータフロー部分を確認できる。

・シナリオの内容が全く異なる場合

この場合は同期をとることができないので、2つのシーンのアニメーションを同時に再生し、データフローの違いを確認する。

■ 3つ以上のシーンを比較する場合

3つ以上のシーンを並べてアニメーションで同時に再生しても、データフローの違いを確認することは困難なので、シーンを1つに統一する。この際、同じ意味を表すアイコンは比較もとのアイコンに統一し、比較もとのシーンと異なる意味をもつアイコンがあった場合は、そのアイコンを半透明にして配置する。そして、それぞれのシーン間で共通のシナリオ部分があれば、比較もとのシーンのデータフローの生起順序で再生する。これにより、シーン間でのデータフローの共通部分を確認することができる。さらに、それぞれのシーンに対応するシナリオが表示されたテキストを並べ、同じシナリオ部分の色を変えて表示することにより、それぞれのシナリオの差異部分や生起順序を確認する。

以上で述べた手法をもとにシーンを比較するシステムを開発する。シーンを比較するシステムでは、画面に多数のシーンを表示する必要があり、3次元空間を用いた。3次元空間では、データフローを様々な視点から眺めることができ、実体間の関連をわかりやすくできる。また、2次元のアイコンが利用できないため、VRMLを用いて作成したものや、立方体にテクスチャ

を貼った3次元アイコンを利用する。

3.3 変更管理支援システムの機能

3.3.1 視点変更機能

3次元空間を利用することにより、様々な視点からデータフロー要求を眺めることができる。視点変更の方法としては、こちらであらかじめ用意した視点をメニューからユーザに選択させるか、キーボード操作によって自由に視点を移動する方法をとる。前者の方法では、メニューを選択することにより、簡単に視点を移動できる。また、メニューに用意されていない視点からデータフローを眺めたいという場合は、キーボードを操作することにより、ユーザは自由に視点を移動できる。メニュー選択による視点変更例を図5に示す。

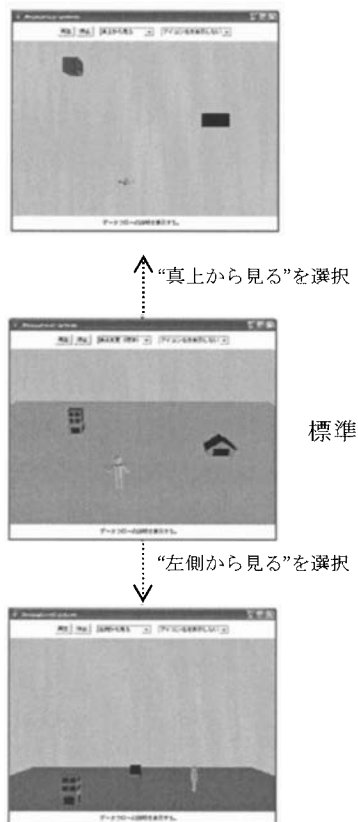


図5 メニュー選択による視点変更

3.3.2 アニメーション再生

3次元空間でのアニメーション再生の動作例として、「書籍の受注」の例を示す。「書籍の受注」のシナリオを図6のように想定する。

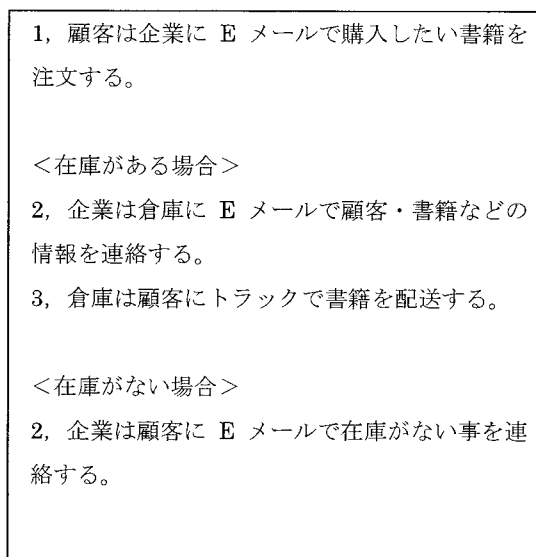


図6 書籍の受注シナリオ

図6のデータフローの場合、源泉格あるいは目標格のアイコンとして顧客と企業と倉庫の3つが必要になる。

図7に動作例を示す。再生ボタンを押すと再生ははじまる。再生中のデータフローの内容は画面下部分のラベルに表示される。条件分岐点に到達すると、ユーザに条件の選択を促し、そこで選択された条件に沿って後のデータフローが再生される。

特定のアイコンに注目する機能も付けたいと考えている。この機能は注目したいアイコンを選択するとシステムの画面の手前にそのアイコンが移動するように視点を移動させる。そして、そのアイコンが源泉格、又は吸収格となっているデータフローだけを表示するというものである。これにより、シーンが複雑になった場合で

も注目したいアイコンが認知するデータフローだけを表示することができる。また、2次元のアイコンと3次元のアイコンを対応付けすることにより、2次元のアニメーションシステムでも再生できるようにする。

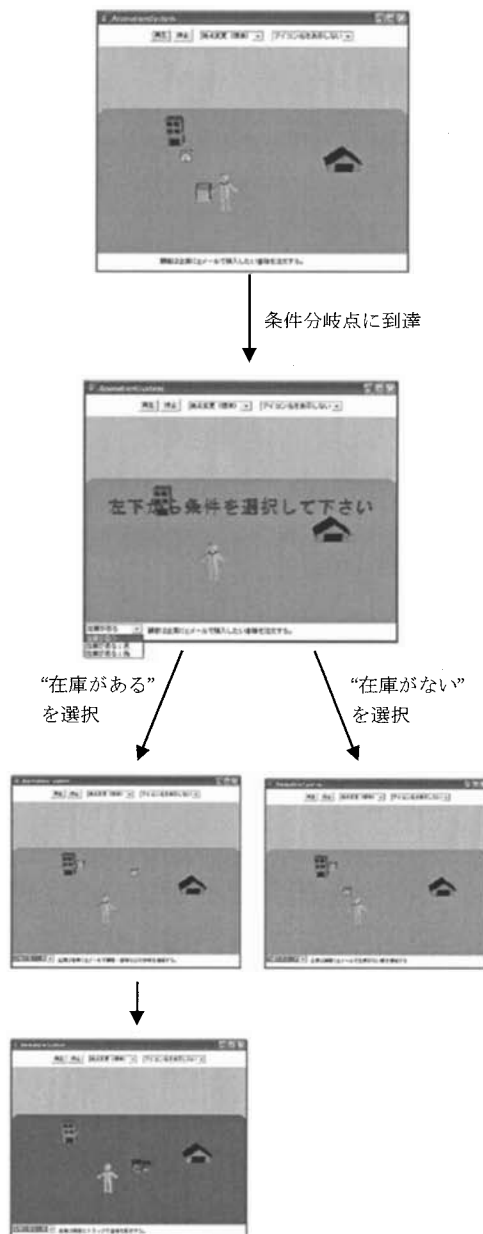


図7 アニメーション再生(書籍の受注)

3.3.3 シーンの比較

3次元空間を利用したアニメーションシステムの操作方法を説明する。まず、シーン・シナリオファイルを選択することにより、シーンを表示する。表示されるシーンはアニメーションが再生される前の状態であり、源泉格、吸収格のアイコンのみが表示されている。ここでは、1つのシーンだけではなく、複数のシーンを選択することができる。シーン・シナリオファイルを複数選択した場合、最初に選択したシーンの上部に土台（地面）を半透明にして順番にシーンを表示させていく。選択したそれぞれのシーンに同じ意味を持つアイコンが配置されている場合は、それらのアイコンの配置位置を最初に読み込んだシーンのアイコンの配置位置に合わせて表示する。アイコンの数が異なる場合もあるので2つのシーン間で位置を合わせるできないアイコンがある場合は、見やすい位置にそのアイコンを配置する。具体例を図8に示す。図8のように、アイコンの配置位置を合わせることで配置されているアイコンの差異が分かりやすくなる。

複数のシーン・シナリオファイルを選択する際に、最新のシーンから順番に過去に作成されたシーンを選択すれば、今まで作成されてきたシーンがどのように変更されてきたのかを時系列順に並べて表示し、確認することができる。シーンを複数選択し、表示した例を図9に示す。

図9では、一番下に表示されているシーンが最新のものであり、上部にいくにつれて順番に過去に作成されたシーンが表示されていると仮定した。これを眺めることにより、配置されている源泉格、吸収格のアイコンの差異を確認することができる。この画面でも3.3.1で述べた視点変更ができるので、ユーザは視点を見やすい位置に自由に移動させ、画面を眺めることがで

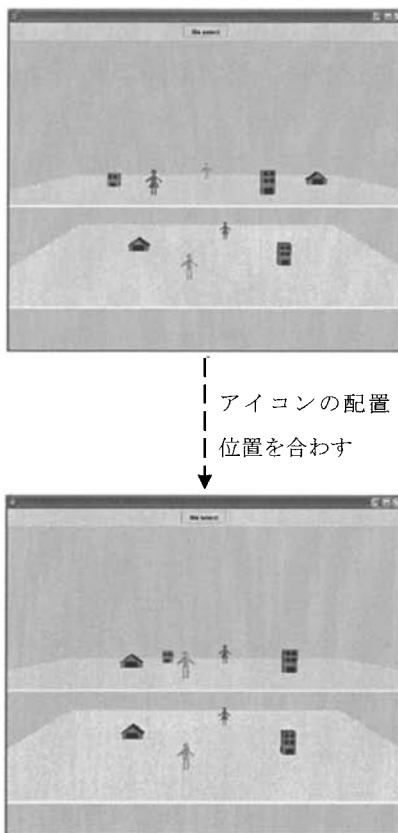


図8 シーン間のアイコンの位置を合わせる例

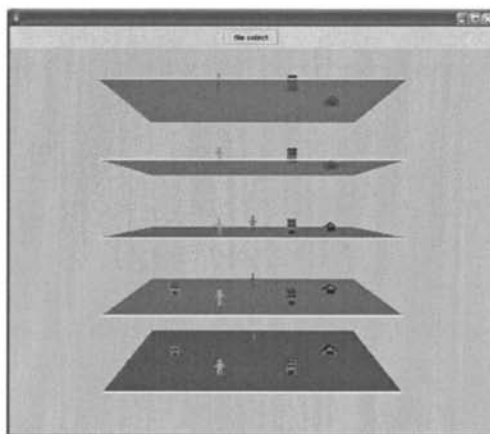


図9 複数のシーンを選択した画面

きる。また、配置されているシーンの土台(地面)にタッチセンサーをつけておき、その土台をク

リックすることにより、画面が 3.3.2 で述べたアニメーション再生のものに切り替わり、選択したシーンのアニメーションを再生できるようにする。この機能により、それぞれのシーンのデータフローの流れを確認することができる。

シーンを比較する際は画面に配置したそれぞれのシーンに対応するシーン・シナリオファイルがメニューとして表示されるプルダウンメニューから比較したいファイルを選択して実行する。ただし、選択できるシーン・シナリオファイルは条件分岐を含まないものに限る。そして、3.2 で述べたシーンを比較する処理をおこない、ユーザにそれぞれのシーンの差分を確認させる。選択されたシーン・シナリオファイルが 2 つの場合は図 8 のようにシーンを上下に並べて表示し、3 つ以上の場合にはそれぞれのシーンが統合された 1 つのシーンを表示して、シーンの比較を実行する。

3.4 考察

現在、既に開発されている要求管理ツールとして、IBM の Rational RequisitePro[2]などが挙げられる。Rational RequisitePro では変化する要求を管理することができ、変更の影響分析機能などがある。しかし、要求の変更を確認する際には、文書で書かれた静的なものを見る。本研究ではフロー要求限定ではあるが、動的に差分をユーザに確認させることにより、容易にフロー要求の差分を理解できるところで優位な点があると考える。

4 おわりに

4.1 まとめ

本研究ではビジュアルな要求定義環境で記述されたフロー要求の動的な差分を容易に確認できるようにするために、アニメーションシステ

ムを利用し、動的にシナリオの差分をユーザが確認できる手法を提案した。また、3 次元空間を用いることにより、様々な視点からデータフローの流れを確認できるようにした。

4.2 今後の課題

今回述べたシステムを実装し、評価することにより、実際にシーンを比較したときに、シナリオの違いが分かりやすく表現されているかということや、3次元空間を利用することにより、実体間の関連が「分かりやすくなったか」、「見やすくなったか」ということを確認する必要がある。

謝辞

本研究及び本論文作成にあたり、多くの助言と励ましをくださった金森氏をはじめとするソフトウェア工学研究室の皆様、またアイコンとしてフリー素材を使用させていただきましたホームページフリー素材倉庫様等にも深く感謝致します。

参考文献

[1]大西 淳:「ビジュアルなソフトウェア要求仕様化技法」、情報処理学会論文誌, 第 36 卷 5 巻, pp.1183-1191, 1995

[2]IBM Rational RequisitePro,
<http://www-06.ibm.com/jp/software/rational/products/req/reqpro/> (2007年8月22日時点)

[3]「Java3D グラフィックス 基礎から立体アニメーションまで」、広内哲夫、セレンディップ、小学館、2004年