

## オーディオ・ビジュアルな要求仕様化技法

大西 淳

立命館大学 理工学部 情報学科

525-77 滋賀県草津市野路東 1-1-1

e-mail: ohnishi@selab.cs.ritsumeai.ac.jp

ビジュアルなソフトウェア要求言語によるフロー要求の仕様化において、音情報を併せて仕様化する。フロー要求の実行順序を指定したものをシナリオとしてまとめ、シナリオを解釈実行することによって、音情報を付加したアニメーションを提示でき、利用者にとっては、より分かりやすい仕様化ができる。ソフトウェア要求言語 VRDL と仕様化技法を例を用いて説明する。

## Audio-Visual Software Requirements Definition

Atsushi OHNISHI

Department of Computer Science, Ritsumeikan University

Kusatsu, Shiga 525-77, Japan

The author proposes an audio-visual software requirements definition method including audio-visually describing data flow requirements, describing icons' movements as a scenario, executing the requirements as an animation by interpreting the scenario. This animation provides users with the behavior of a target system and enables users to confirm the correctness of the requirements more easily. A visual requirements language named VRDL and tools are illustrated with examples.

## 1 はじめに

要求記述としてはテキストだけでなく、構造化分析技法 [3] に基づいた DFD(Data Flow 図) や業務フロー図といった図式の形態のものも読みやすさや書きやすさの点からよく用いられている。特に DFD の処理系は上流 CASE ツールとして各社から製品が出されている。

DFD はデータの流れを名前付きの矢印で、機能を円で、ファイルを直線で、データの源泉と吸収を四角形であらわし [3]、記号の種類が少ないので覚えやすい。

しかしながら、能大式の業務フロー図のように 30 以上の多種の記号を使う図に慣れた人にとっては DFD は単純化しすぎて使いにくく、記号の種類が少ないので名前や説明を詳細に記述しなければならない。また場合によっては利用者は機能を円でなく四角で、またファイルを直線でなく JIS の情報処理用流れ図記号の直接アクセス記号を使いたいかも知れない。

我々は要求定義者の頭に描いたイメージをできるだけ忠実に仕様化することを目標にしてビジュアルな要求言語 VRDL を開発してきた。VRDL については文献 [7] に述べているが、その後も記述能力の拡大や使い勝手の向上を目指して改善を進めている。今回は文献 [7] 以降の改善点を中心に紹介する。

## 2 要求言語:VRDL

VRDL(Visual Requirements Description Language) は、(1) データや制御の流れに関するソフトウェア要求を要求定義者の頭に描いたイメージにできるだけ忠実に表せるようにする、(2) ビジュアルな要求を記述した本人だけでなく他の人間にもわかるようにする、(3) 要求フレームによる誤り検出を可能とする [6]、(4) ビジュアルな要求仕様におけるデータの流れをアニメーション表示することによって誤りを検出することを目標に我々の研究室で開発してきたビジュアルな要求言語である。VRDL の特長を以下に示す。

1. アイコンの形状と意味を定義できる
2. アイコンと矢印をエディタ上で配置していくことによって要求を記述する
3. VRDL による記述をあらかじめ用意した標準的なアイコンを用いた記述に変換できる

4. VRDL 記述は要求フレームに基づく言語に共通の内部表現 CRD に変換される
5. VRDL 記述に用いられたアイコンの動作を与えることによって、記述を実行できる
6. より複雑な仕様のために VRDL 記述のアイコンを階層化して仕様化できる。
7. VRDL 記述の流れを表す要求に音情報を付加でき、その場合実行時には動きと音をアニメーションとして表現する
8. サーバー/クライアント方式でネットワークを介して VRDL 記述を共有でき、任意の利用者が変更権を主張した上で VRDL 記述を変更できる。またアイコンの形状と意味をクライアントごとに独自に定義できる。

上の特長のうち 6, 7, 8 が文献 [7] 以降に付加されたものであり、1~5 についても改善されている。

### 2.1 アイコンの定義











記述者にとっては、自分のイメージにあった記号をそのまま要求記述に用いることができるならば、要求を記述しやすいし、また理解しやすい。このためには自分でアイコンの形状を定義して要求記述に用いることができるようにすればよい。一方、要求記述は記述者以外にも設計者など開発に携わる人によって参照される。他人の描いた図を理解するには、そこで使用されたアイコンの意味を的確に把握する必要がある。記述者以外の人にとってアイコンが別の意味にとられると正しく要求を理解できない。

このため VRDL ではアイコンの形状を pixmap ファイルや GIF フォーマットのイメージとして定義するだけでなく、要求フレームモデル [6] に対応して、要求記述に現れる名詞や名詞の型をアイコンの意味としてアイコン辞書に定義するようになっている。表 1 にアイコン定義の具体例を示す。

### 2.2 アイコンと矢印の配置による仕様

例えば表 1 のようにアイコンが定義されているとき、図 1 の上側のフローは「酒販売店から倉庫へ注文書をファックスで流す」という要求を表す。このように定義

表 1: アイコンの定義例

アイコンの形状	アイコンの意味(名称、型)	
	顧客	human
	注文 DB	file
	秘書	human
	受付	human
	倉庫	function
	酒販売店	function
	注文書	data
	酒	data
	ファックス	device
	トラック	device

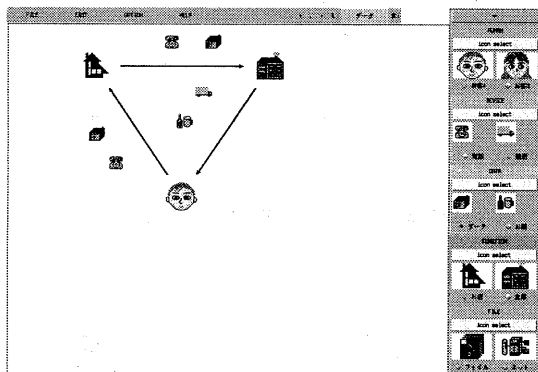


図 1: VRDL による記述例

表 2: 要求文の関係データモデル表現

関係：DFLOW

動作主格	源泉格	目標格	道具格
注文書	酒販売店	倉庫	ファックス

したアイコンを矢印で結びながら、フロー要求を簡単に表現できる。

### 2.3 標準的なアイコンへの変換

VRDL では記述者がアイコンを定義できるが、他人が定義したアイコンを正確に理解するのは難しい場合がある。例えば①というアイコンを記述者は「ファックス」のつもりで用いたのに読者は「電話」と誤って解釈するかもしれない。

このような場合の一つの解決法として、あらかじめ標準アイコンを用意しておき、記述者が独自に定義したアイコンを全て標準アイコンに変換すれば、標準アイコンの意味を理解している利用者には正しく解釈できる。

### 2.4 共通の内部表現への変換

要求フレームに基づいた要求文の動作概念を関係、必須格を属性とみなすことによって格フレームを関係スキーマと見ることができる。例えばデータの流れを表す概念 DFLOW の格フレームは DFLOW を関係、動作主格・源泉格・目標格・道具格といった必須格を属性とする関係スキーマと見ることができる。これにより「酒販売店から倉庫へ注文書をファックスで流す」という要求文は表 2 のように関係表のテーブルと見なされる [9]。

このように VRDL で表した要求文は関係データモデルに容易に変換される。要求フレームモデルに基づいたもう一つの要求言語である日本語要求言語 X-JRDL による要求文も同様にして関係データモデルに変換できる。換言すれば、日本語で書きたい部分とビジュアルに書きたい部分を切り分けて、より書きやすい言語を用いて要求を記述でき、これらの言語による要求文は容易に関係データベースに格納できる [9]。

### 2.5 VRDL 記述の実行

VRDL によるフロー要求に順序付けをしたものをシナリオとし、そのシナリオを指定した順序に従うとともに、流れるデータの様子をアニメーションとして表示することができる。フロー要求に対する順序付けに関して、当初はフローをイベントとしたイベント駆動の状態遷移に対して順序付けをすることによってシナリオを作成していたが、現在では VRDL によるフロー要求記述

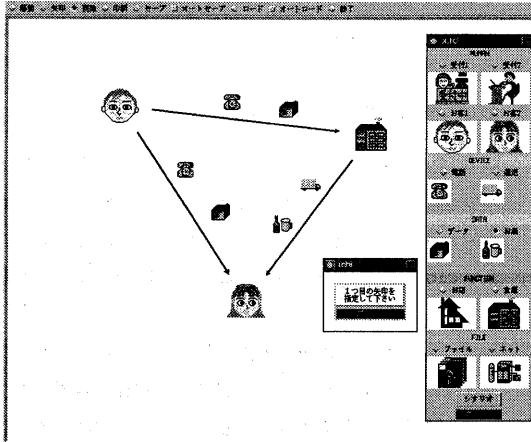


図 2: データフローの順序付けによるシナリオ作成

に直接順序を指定できるようにしている。図 2 に作成過程を示す。またアニメーションのスナップショットを図 3 に示す。アニメーションによって利用者はデータの流れる方向と順序の確認ができる。

### 3 VRDL 記述の階層化

従来、VRDL では 1 つのウィンドウでビジュアルな要求を仕様化していたが、大規模な要求になると 1 つのウィンドウだけで仕様化するのは難しかった。そこで VRDL の処理系の機能を拡張し、DFD と同様に階層による仕様化を可能とした。

図 4 はビジュアルな要求記述のエディタ VRDA による階層化された VRDL 記述の例である。倉庫アイコン (■) を展開した様子が左下のウィンドウ内に示されている。このようにビジュアルな要求仕様中のアイコンを展開して階層的に記述できるようになっており、大規模・複雑な要求の仕様化にも対処できる。

### 4 音情報の付加

ビジュアルな要求仕様のデータフローに音情報を併せて定義し、仕様実行時に該当するデータフローの音情報を再生することによってより効果的・実際のなプロトタイプイングが可能となる。現在は、音情報を格納したオーディオファイル名をデータフロー記述時に指定して、ア

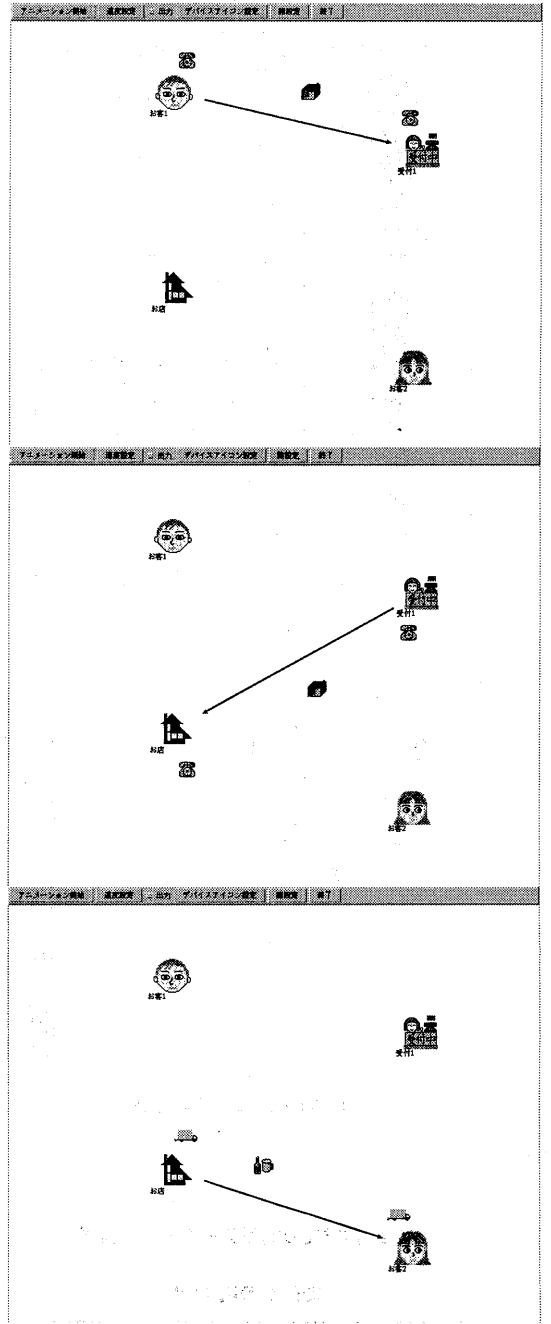


図 3: アニメーションのスナップショット

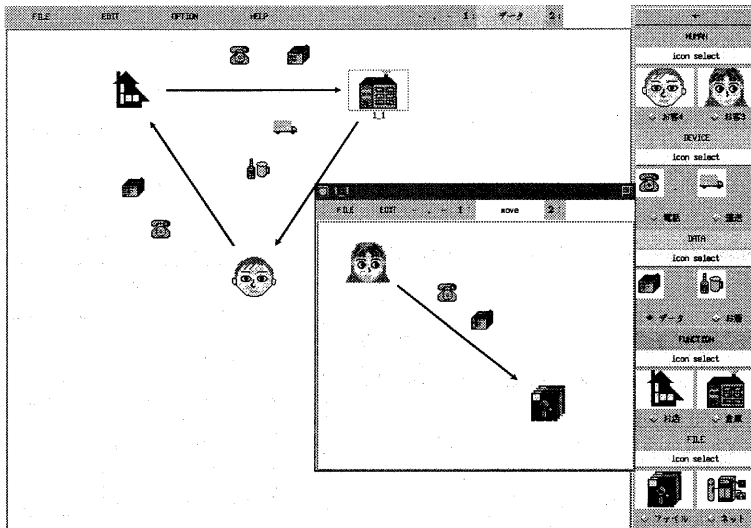


図 4: VRDL による階層化された記述

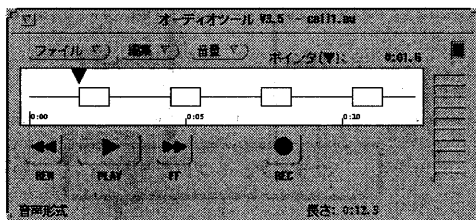


図 5: オーディオツールによる音情報の編集

アニメーション時にそのファイルを `audioplay` コマンドによって実行することによって音情報を再生している。トラックの走行音と電話のベル音のファイルの 2 種類を用意しているが、試作システムの試用からアニメーション時には音情報が再生されると、より印象を深められることが分かった。

音情報の作成にはオーディオツールを用いて、ワークステーションのマイクを通して録音し、それを編集し再生している。図 5 に再生編集過程を示す。アイコンと同様に標準的な各種の音源を用意して仕様に付加できるようにするといった使い勝手の改善を進めている。

## 5 ネットワークを介した VRDL 記述の共有



VRDL の処理系をネットワーク対応化して構築した。ネットワーク機能の特長は

1. アイコンの形状と意味をサーバー/クライアントで独自に定義できる。
2. ビジュアルな要求記述をサーバーから複数のクライアントに対して提供するが、任意の利用者が変更権を主張した上で VRDL 記述を変更できる。
3. 複数の人間が協同して要求を記述する場合、他人の定義したアイコンは自分の定義したアイコンもしくはシステムであらかじめ用意した標準的なアイコンに変換して表示できる。他人の定義したアイコンをそのまま表示させることもできる。

図 6 では記述者によって、表 3 のように同じ「受付」というオブジェクトに対して異なるアイコンを定義している場合のビジュアルな仕様がそれぞれの固有なアイコンを使って表示されている様子を示している。このようにアイコンでの `synonym` (異なる形状で同じ意味) や `homonym` (同じ形状で異なる意味) を処理できる。

サーバーとクライアント間のデータの流れを図 7 に示す。また、サーバーとクライアント間での処理の流れを

表 3: ローカルなアイコンの異なる定義

記述者	アイコン	意味
記述者 A		受付・人間型
記述者 B		受付・人間型

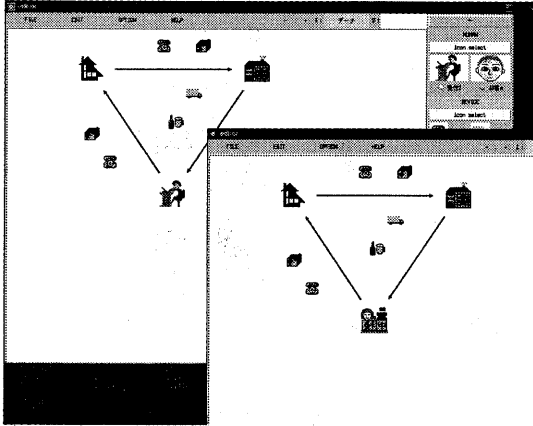


図 6: 固有アイコンによる VRDL 記述

図 8 に示す。

## 6 要求定義環境: CARD

ソフトウェア要求定義のための環境として CARD (Computer Aided Requirements Definition) を開発している [8]。CARD は要求定義のためのモデル「要求フレーム」と要求言語に基づいた要求定義支援技法を実現した環境である。要求分析、要求記述、記述の検証、記述の検索、記述の実行(アニメーション表示)、設計支援をサポートしており、より高品質な要求仕様を得ることを目標としている。CARD の構成を図 9 に示す。

CARD は

- 要求記述解析系: READ
  - 日本語要求言語解析系: X-JRDA
  - ビジュアルな要求言語解析系: VRDA
- 要求記述精製系
  - 要求記述検証系: RDV

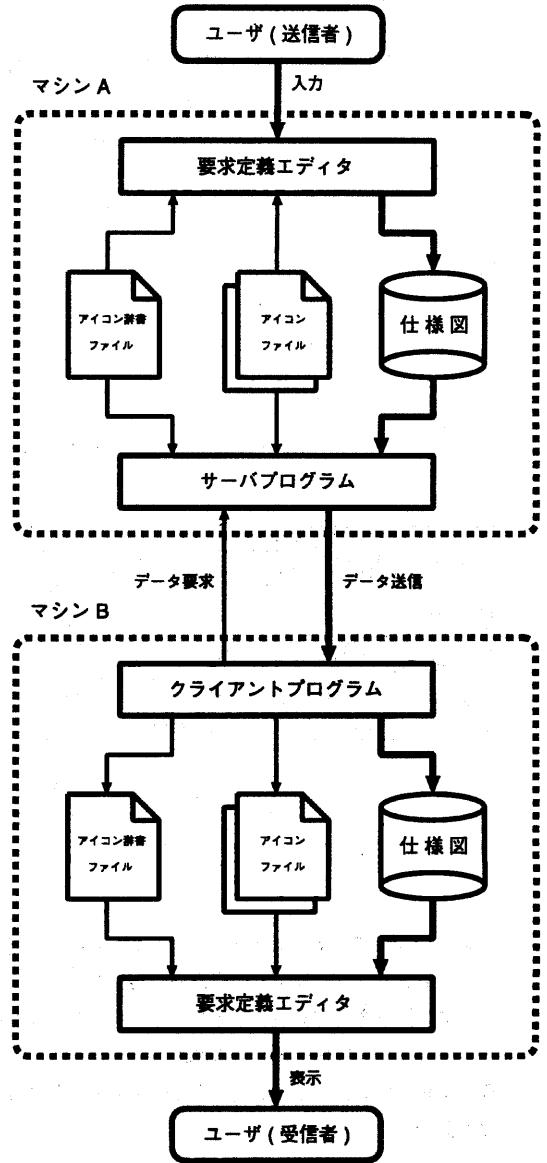


図 7: サーバとクライアント間のデータの流れ

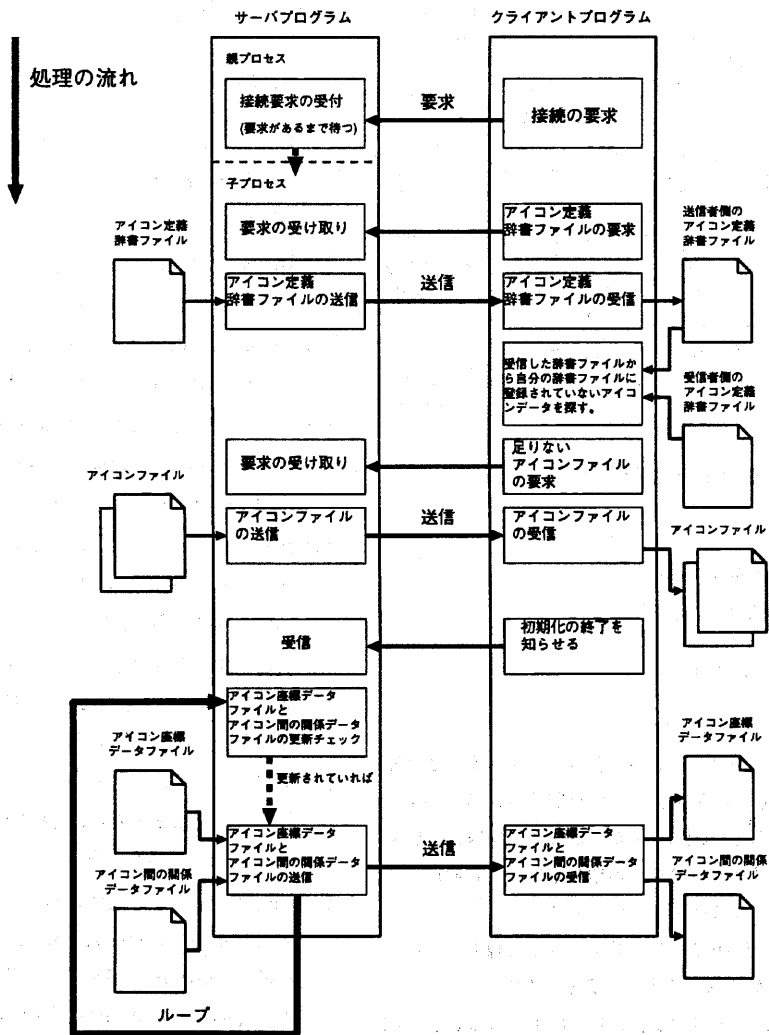


図 8: サーバとクライアント間での処理の流れ

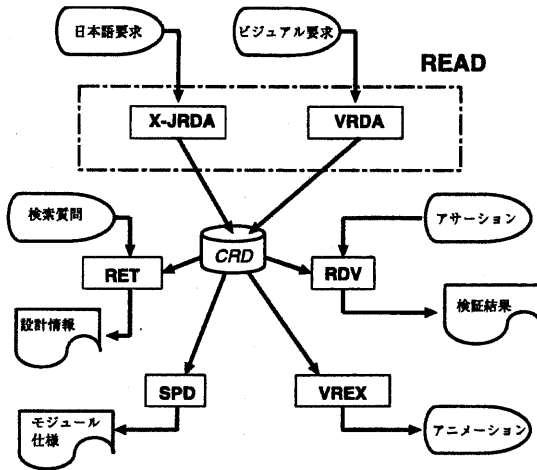


図 9: 要求定義環境: CARD

- 要求記述検索系: RET
- 要求記述実行系: VREX

● 設計支援系: SPD

から構成されている。

要求は日本語かビジュアルな言語のいずれかにより記述される。どちらの言語による要求記述も、解析の結果として CRD 表現に変換されて、要求記述精製系や設計支援系で利用できる。要求記述精製系は、主に記述の正当性 (correctness) の向上を目的としている。

CARD の中で VRDA がビジュアルな仕様のエディタを中心とした解析系であり、VREX がアニメーションを用いた実行系に相当する。これらは主に Tcl/Tk、一部 C 言語によって実現されている。

## 7 おわりに

ビジュアルな要求言語について強化された機能を含めその概略を紹介した。試作したツールの使用経験から

- シナリオでのフローの順序指定を逐次だけでなく並列、同期のサポート
- アイコン辞書作成支援
- 音情報のパレットを用いた指定

● 日本語要求からビジュアルな要求への変換

などを今後の課題として進めている。

## 参考文献

- [1] Chang, S.K.: "A Visual Language Compiler," *IEEE Trans. Softw. Engrn.* Vol.15, No.5, pp.506-525 (1989).
- [2] Davis, A.M.: *Software Requirements, Objects, Functions, & States*, Prentice-Hall, Inc., NJ (1993).
- [3] DeMarco, T.: *Structured Analysis and System Specification*, p.352, Prentice-Hall (1978).
- [4] Hsia, P., Yaung, A.T.: "Screen-Based Scenario Generator: a tool for scenario-based prototyping," *Proc. IEEE HICSS*, Vol.2, pp.455-461 (1998).
- [5] Marca, D.A., McGowan C.L.: *SADT: Structured Analysis and Design Technique*, McGraw-Hill Book Co. NY (1988).
- [6] 大西 淳、阿草清滋、大野 豊: 「要求フレームに基づいた要求仕様化技法」、*情報処理学会論文誌* 31 巻 2 号, pp.175-181 (1990).
- [7] 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* 36 巻 5 号 (仕様記述特集), pp.1183-1191 (1995).
- [8] Ohnishi, A.: "CARD: an Environment for Software Requirements Definition," *Proc. of the second Asia-Pacific Software Engineering Conference (APSEC'95)*, pp.420-429 (1995).
- [9] Ohnishi, A.: "Software Requirements Specification Database based on Requirements Frame Model," *Proc. IEEE 2nd International Conference on Requirements Engineering (ICRE'96)*, Colorado Springs, Colorado, pp.221-228 (1996).
- [10] Shu, N.C.: *Visual Programming*, Van Nostrand Reinhold Co., NY (1988).
- [11] St-Denis, R.: "Specification by example using graphical animation and a production system," *Proc. IEEE HICSS*, Vol.2, pp.237-246 (1990).