

要求仕様データベースシステムの開発

大西 淳

立命館大学 理工学部 情報学科
525-77 草津市野路町 1916

膨大な要求仕様から参照すべき箇所を的確に引き出すとともに分かりやすく提示することを目標としたソフトウェア要求仕様のデータベース化に関する研究を紹介する。

要求フレームと名付けた要求を表現するためのモデルとそれに基づいた要求言語を開発しているが、要求フレームモデルは容易に関係データモデルに変換可能であり、要求文は動詞を関係とする関係表のタプルに変換できる。

要求仕様のためのデータベースに対する質問言語ならびに検索システムを試作した。質問に答える際に、要求文を具体例もしくは反例として提示するとともに自然な日本語で表現する特長があり、利用者は適切な質問を用意することによって要求仕様の正しさを確認できる。

Development of Software Requirements Specification Database System

Atsushi OHNISHI

Department of Computer Science, Ritsumeikan University
1916 Noji, Kusatsu, Shiga 525-77, JAPAN
e-mail: ohnishi@cs.ritsumei.ac.jp

The author proposes a method to build a relational database of software requirements specifications (SRSs) from textual SRSs automatically. The author has been developing both a requirements model and requirements languages based on the model. Since the model can be transformed into relational data model, each of requirement sentences can be regarded as a tuple of a relational table.

The author has been developing a relational DBMS for an SRS DB. One of the features of the SRS DB system is to show an answer with a requirement sentence as an example/counter-example. This feature contributes to verify the SRS from developer's own viewpoints.

1 はじめに

ソフトウェアの開発・保守において要求仕様は様々な場面で参照される。本稿では膨大な要求仕様から参照すべき箇所を的確に引き出すとともに分かりやすい表現で提示することを目標としたソフトウェア要求仕様のデータベース化に関する研究を紹介する。

最初に要求フレームモデルと名付けた要求を表現するためのモデルとそれに基づいた要求言語について説明する。次に要求フレームモデルから関係データモデルへの変換と要求仕様のデータベース化について述べる。さらに試作した要求仕様データベースシステムとその動作について例を用いて説明する。

2 要求フレームモデルと要求言語

2.1 要求フレームモデル

要求フレームモデル [6] は要求仕様の枠組を与えるものであり、

- ① 名詞と名詞の型
- ② 動詞・形容詞と概念
- ③ 文と格フレーム
- ④ 文章と機能フレーム

のそれぞれについての対応関係を定める。

名詞は人間 (human) 型、機能 (function) 型、データ (data) 型、ファイル (file) 型、制御 (control) 型、装置 (device) 型の 6 種の型のいずれかを持つ。

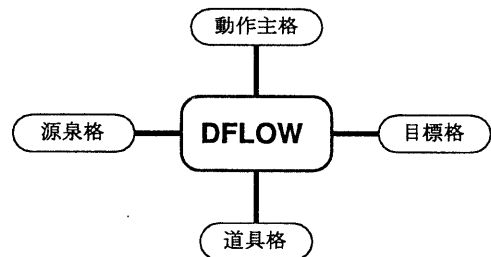
動詞はデータの流れ、データや機能の構造、ファイルの操作など 10 種の動作に関する概念のいずれかに分類される。表 1 に概念と対応する動詞を示す。「入力する」、「(データを) 渡す」、「出力される」といった動詞はすべてデータの流れに相当する DFLOW という概念に分類される。

形容詞は「大きい」、「小さい」、「等しい」といった 6 種の比較演算に関する概念のいずれかに分類される。名詞の型と動作概念については要求定義の対象をファイル処理分野に限定して用意している。なお、動詞については後述するように、記述者が新たに概念を定義することにより、それに対応した新規の動詞も利用できるようにしている。

文は一つの動詞とその動詞に対応する動作概念の必須格に当てはまる名詞を中心として構成される。文には比較条件を表す形容詞を含んだ節を加えてもよい。

表 1: 概念と動詞

概念	対応する動詞
データの流れ	入れる、受け取る、等
制御の流れ	渡す、受け取る、移す、等
機能、データ、ファイルの and 木構造	構成する、成る、含む、分割される、含まれる、等
機能の or 木構造 かデータ種別	構成する、成る、含む、分割される、含まれる、等
ファイルの検索	検索する、検索される
ファイルへの挿入	挿入する、加える、等
ファイルの更新	更新する、更新される
ファイルのデータ削除	削除する、除く、等
ファイル処理	ソートする、アペンドする、マージする、複写する、等
データ生成	作成する、変換する、等



概念	格	名詞の型
DFLOW	動作主格	data
	源泉格	functionかhuman
	目標格	functionかhuman
	道具格	device

図 1: 概念 DFLOW(データの流れ) の格フレーム

格フレームは動作概念と必須格に対する枠組である。各概念はそれぞれ異なった格構造を持っている。

例えばDFLOWの場合は図1のように動作主、源泉、目標、道具という格を持っている。それぞれの格に当てはまる名詞の型も限定される。DFLOWの場合は動作主格は流れるデータが相当し、当てはまる名詞もデータ型となる。源泉・目標格はそれぞれ流れるデータを送り出す・受けとる主体が相当するため、人間型または機能型の名詞が当てはまる。道具格は装置型の名詞となる。10種の動作概念に該当しないような動作を記述したい場合は、記述者が新たにその概念と対応する格構造を定義することによって利用できる。

この格フレームによって、格の抜けや不正な型の名詞の使用を検出できる。さらに、代名詞が使われたり、格が省略された場合に、文脈から用いられるべき名詞を推定できる [7]。

例えば、「利用者は検索コマンドを入力する」という要求文は「入力する」の対象である「検索コマンド」がデータ型の名詞であることから、動作概念がDFLOWであるとみなされ、助詞をもとにDFLOWの格フレームに基づいて判断した結果、表2のように解析される。

表 2: 要求文の解析結果

要求文「利用者は検索コマンドを入力する」

動作概念	DFLOW
動作主格	検索コマンド
源泉格	利用者
目標格	** 不明 **
道具格	** 不明 **

この例文では、目標格(どこに対して)と道具格(何を以て)が抜けていることが判明する。DFLOWの格フレームから目標格は機能か人間型の名詞、道具格は装置型の名詞が埋まるべきであり、この文の以前に表れた要求文や要求文の置かれた段落の見出し語などから該当する型の名詞を埋まるべき名詞の候補として記述者に問い合わせる。候補が見つからない場合は抜けと判断する [6, 7]。

このように要求文は動作概念とその格フレームに基づいた内部表現CRD (Conceptual Requirements Description) に変換される。ここで“conceptual”と呼んでいる理由は、元の要求文が受動態や能動態であろうと語句

の順番が異なっていようと同一要求を表す場合は同一の内部表現に変換されるからである。

機能フレームはシステムが備えるべき一般的な性質を規定するものであり、「外部からの入力と外部への出力は、それぞれ少なくとも一つ存在する」、「作成されたり検索されて得られたデータは一度は参照されなければならない」といった10個の性質について、要求記述が満たしているかどうかをチェックするために用いられる [6]。機能フレームによって機能単位の抜けや矛盾を検出できる。

2.2 要求言語

要求フレームモデルに基づいた要求言語として

- 日本語要求言語 X-JRDL
- ビジュアルな要求言語 VRDL

を開発している。

X-JRDL (eXtended Japanese Requirements Description Language) は以下の特徴を持つ。

1. 単文だけでなく重文や複文(主語節、対立節、連体修飾節)が書ける。
2. 文脈から前方照応可能な名詞を省略したり、代名詞が利用できる。
3. 表1に示した概念とそれに対応する動詞を利用できる。さらに新規の動詞も格構造を定義することによって利用できるようになる。

重文や複文のように1つの文の中に複数の動詞がある場合は、一つの動詞しか含まない単文に分割し、次に代名詞や省略された名詞を格フレームに基づいて補ってからCRD表現に変換する [7]。

VRDL (Visual Requirements Description Language) は以下の特長を持つ [9]。

1. アイコンの形状と意味を定義できる。
2. 複合的なアイコンを容易に作成できる。
3. 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フロー要求を記述する。
4. VRDL記述に用いられたアイコンの動作を与えることができ、その動作記述を解釈実行することによって、記述の中のアイコンの動作をアニメーションとして表示できる。

- VRDL による記述を標準的なアイコンを用いた記述へ変換できる。




VRDL では、名詞に対するアイコンの形状をビットマップデータとして定義しその意味を要求フレームに基づいて、

- ① 名詞の型、あるいは
- ② 名詞の名前と型

のいずれかでもって定義する。

表 3 にアイコン定義例を示す。最初のアイコンは人間型の名詞の総称を意味する。このような総称的なアイコンには記述に表すたびに、異なる名前を名称として与えることができる。2 番目のアイコンは装置型の「ファックス」を意味する。3 番目のアイコンはデータ型の「メッセージ」を意味する。

表 3: アイコンの定義例

アイコンの形状	アイコンの意味
	人間型
	「ファックス」、装置型
	「メッセージ」、データ型

データや制御の流れに関する要求を、定義したアイコンを矢印で結んでいくことによって表す。図 2 に表 3 で定義したアイコンを用いた簡単な要求文を示す。この要求文は「富市さんから Bill さんへメッセージをファックスで送る」ことを表している。

このように動作概念の必須格にアイコンを、概念に矢印を対応させて、要求文をビジュアルに表せる。データの流れを表す矢印には始点、終点、ラベルとしてデータと装置に関する 4 つのアイコンが用いられるが、これらは図 1 に示した DFLOW の格フレームを忠実に反映している。従ってビジュアルな要求記述中の矢印単位で CRD 表現に容易に変換できる。

2.3 関係データモデルへの変換

要求フレームにおいて動作概念を関係、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 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「富市さんがファックスを用い

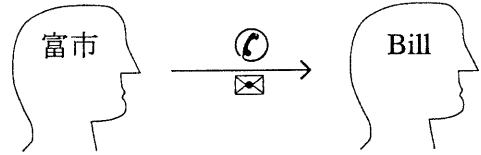


図 2: ビジュアルな要求文

て Bill さんへメッセージを送る」という要求文は表 4 のように関係表のタプルと見なされる。

表 4: 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
メッセージ	富市さん	Bill さん	ファックス

日本語要求文の解析結果として示した表 2 の枠組みを表 4 のように変換するのは容易であり、日本語要求言語やビジュアルな要求言語で記述された要求はそれぞれの概念に対応した関係スキーマのタプルの集まりとみなすことができる。換言すれば、日本語要求言語やビジュアルな要求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。

また要求記述の内部表現では、表層の格ではなく深層の格をもとにしている。例えば「富市さんが Bill さんへメッセージをファックスで送る」という要求文も「富市さんから Bill さんへファックスによってメッセージが送られる」という要求文も表 4 の同じ内部表現に変換される。このため、元の文では目的語となっていた名詞を主語にして表現したり、また逆に主語となっていた名詞を目的語にして表現することは容易にできる。従って、機能を中心に記述していた要求ではある特定のデータは様々な箇所にも目的語として表れるかも知れないが、該当する文を集めてきて、そのデータを主語とする文章を再構成できる。

3 要求仕様データベースシステム

CRD 表現の要求仕様をデータベースとみなして検索するための言語と処理系 RET を試作した。

3.1 質問言語

質問言語では集合の定義と質問が記述できる。集合の定義では動作概念の集合、名詞の集合、名詞の型の集合を要素の性質の記述、もしくは要素の列挙によって定義する。さらに定義済みの集合に対して演算子『かつ』(\cap)、『または』(\cup)、『～の否定』(\neg)を適用できる。

また効率良く定義するために、動作概念を表す動詞以外に『到達可能』、『祖先』、『子孫』という予約語が用意されている。機能 A から機能 B へデータフローが存在するとき A から B へ『到達可能』とする。また『祖先』と『子孫』はデータ構造と機能構造に対して、親より上すべてを『祖先』として、また子供より下すべてを『子孫』として表すために用いられる。

以下に定義例を示す。

- ・入力コマンドと解析データを受動オブジェクトと呼ぶ。
- ・受動オブジェクトを受け取る機能を入力プロセスとする。
- ・ユーザとインタフェース部を外部オブジェクトと呼ぶ。
- ・外部オブジェクトの否定を内部オブジェクトと呼ぶ。
- ・計算部から到達可能な機能をシステム到達機能と呼ぶ。

定義文に表れる名詞や動詞は要求記述に用いられたものと、定義文によって定義済みのものと『～とする』、『～と呼ぶ』に限られる。最初の定義文では「入力コマンド」と「解析データ」という2つの要素からなる集合「受動オブジェクト」が定義される。

2番目の定義文では1番目で定義した「受動オブジェクト」を入力とする機能型の名詞を要素とする集合「入力プロセス」が定義される。3番目でも2つの要素からなる集合が定義される。4番目の定義文では3番目で定義した集合の否定が定義される。否定の場合は元の集合の要素の名詞の型を持つ名詞の集合との差がとられる。この場合では要求記述中の「ユーザ」以外の人間型の名詞と「インタフェース部」以外の機能型の名詞を要素とする集合を「内部オブジェクト」として定義したことになる。

5番目の定義文では「計算部」からの出力が直接もしくは一つ以上の機能を介して流れ込む機能の集合が「システム到達機能」として定義される。このように要

素を列挙するか、集合の性質を示すことによって集合を定義する。

次に定義した集合に対する性質を質問する。集合に含まれる要素にどういったものがあるかを質問したり、集合の要素すべてが特定の性質を満たすかどうか、集合の要素の少なくとも一つが特定の性質を満たすかどうかといった質問ができる。

以下に具体的な質問文例を示す。

- ・解析データが入力される機能はなにか？
- ・少なくとも一つのシステム到達機能の要素を目標格とするDFLOW文があるか？
- ・入力系かつシステム到達機能は存在するか？
- ・すべての入力プロセスはファイルを検索するか？

最初の質問は要求記述中で「解析データ」を入力とする機能型の名詞を問うものである。このような質問に対しては該当する名詞を列挙する。

2番目は先に定義した集合「システム到達機能」の要素の中でDFLOW文の目標格になっているものが少なくとも一つ存在するかどうかを問うものである。このように Yes/No 型の答を要求する質問の場合は該当する要求文、もしくは反例となる要求文を提示する。

3番目は集合「入力系」と「システム到達機能」の共通する要素を問うものである。この場合も該当する名詞を列挙する。

4番目は集合「入力プロセス」の要素すべてはファイルを検索する動作主格に当てはまっているかどうかを問うものである。この場合も該当する要求文、もしくは反例となる要求文を提示する。

3.2 要求仕様 DB 検索系: RET

要求仕様の精製とソフトウェア設計時や保守時の要求仕様の検索と修正を支援するために、CRD 内部表現に対する検索系 RET を WS 上で KCL (Kyoto Common Lisp) を用いて開発した。RET の特長を以下に示す。

- ・質問に対して要求文を具体的な例証もしくは反例として示す。
- ・要求フレームで定められた枠組みを意味制約としている。

最初の特長は要求仕様の正しさの確認に有効である。利用者は要求仕様中の名詞や動詞の集合に対して、そ

れが満たすべき・満たすべきでない性質を質問できる。もし要求仕様が誤っているならば、具体的な例証もしくは反例が得られるので誤っている箇所が判明できる。

2番目の特長は効率的な検索に有効である。格フレームはある動作概念の持つ特定の格にはどのような型を持った名詞が当てはまるかを規定するものであるが、質問文で誤った型の名詞が表れたならば、検索することなしに答を返すようにしている。

4 検索例

要求仕様の解析と質問に対する検索例を示す。

4.1 要求仕様の解析

X-JRDLによって書かれた在庫管理システム要求の一部を示す。

在庫管理システムは 入庫管理部、出庫管理部、在庫管理部から構成される。

1. 入庫管理部

在庫管理係から制御を受け取り、彼からキーボードを通して品名と数量からなる入庫票を受け取る。そして、それを用いて在庫マスタファイルを更新し、在庫管理係に制御を返す。

2. 出庫管理部

検索部と 出庫部と 発注部からなる。

最初に要求仕様の解析系は辞書に未登録の単語の意味を問い合わせる。具体的には名詞はその型を、動詞・形容詞はその概念(と格構造)を問い合わせる。次に複文・重文を概念が一つしか含まない単文に分割する。その過程を以下に示す。なお明朝体は入力された要求記述を単文に分割したものを、ゴシック体は解析系からのメッセージを、イタリック体は利用者の応答を示す。

これから単文に分解します。

在庫管理システムは 入庫管理部と 出庫管理部と 在庫管理部から構成される。

見出し語：入庫管理部

在庫管理係から 制御を受け取る。

“入庫管理部”が“受け取る”のですね。

入庫票は 品名と 数量からなる。

彼から 入庫票を キーボードで受け取る。

“彼”の指す名詞は“在庫管理係”のですね。

”入庫管理部”が“受け取る”のですね。

在庫マスタファイルを それで更新する。

“入庫管理部”が“更新する”のですね。

”それ”の指す名詞は何ですか。

RET: “入庫票” 1: “数量” 2: “品名”

==> 2

在庫管理係へ 制御を 返す。

“入庫管理部”が“返す”のですね。

見出し語：出庫管理部:

検索部と 出庫部と 発注部からなる。

“出庫管理部”が“なる”のですね。

この過程で省略された格に相当する名詞や代名詞は具体的な名詞に置き変わる。もし置き換える名詞の候補が複数存在する場合は記述者に問い合わせる。最後に必須格がすべて揃った単文を提示する。各単文は格構造をそのまま反映しており内部表現に変換される。内部表現は2章に示したように関係データモデルに変換できる。

表 5: 要求仕様の関係データモデル表現

関係：ANDSUB

動作主格	目的格 1	目的格 2	目的格 3
在庫管理システム	入庫管理部	出庫管理部	在庫管理部
入庫票	品名	数量	—
出庫管理部	検索部	出庫部	発注部

関係：CFLOW

動作主格	源泉格	目標格
制御	在庫管理係	入庫管理部
制御	入庫管理部	在庫管理係

関係：DFLOW

動作主格	源泉格	目標格	道具格
入庫票	在庫管理係	入庫管理部	キーボード

関係：UPDATE

動作主格	源泉格	目標格
入庫管理部	品名	在庫マスタファイル

結果を表 5 に示す。表で ANDSUB は機能やデータの and 木構造を、CFLOW は制御の流れを、UPDATE はファイルの更新を、それぞれ表す動作概念である。ANDSUB は 2 つ以上の目的格を子として持てば良く、「在庫管理システム」と「出庫管理部」は 3 つのサブシステムを持つが、「入庫票」は 2 つのデータより構成されるため、目的格 3 は空欄となっている。

このように単一の動詞しかもたない単文に分割された各要求文は、その動詞の動作概念を関係とする関係表のタプルとみなされる。

4.2 要求仕様の検索

ここでは質問と検索結果を示す。なお質問を明朝体で検索系からの出力をゴシック体で示す。

在庫管理システムの子孫を子システムと呼ぶ。

在庫管理係から到達可能な機能はなにか？
入庫管理部 です。

すべての子システムの要素は受け取るか？

いいえ。出庫管理部、在庫管理部、検索部、出庫部、発注部は条件を満たしません。

少なくとも 1 つの子システムの要素は受け取るか？

はい。子システムの要素のうち、入庫管理部は以下のように条件を満たします。
「入庫管理部は在庫管理係から入庫票をキーボードで受け取る。」

子システムの要素を動作主格とする ANDSUB 文はあるか？

はい。子システムの要素のうち、出庫管理部は以下のように条件を満たします。
「出庫管理部は 検索部と 出庫部と 発注部からなる。」

最初の質問文は集合の定義であり、「在庫管理システム」のサブシステムである「入庫管理部」、「出庫管理部」、「在庫管理部」、「検索部」、「出庫部」、「発注部」を要素とする集合を「子システム」という名前で定義している。

2 番目の質問では「在庫管理係」からデータフローによって到達する機能について該当する「入庫管理部」が答えられている。

3 番目では「子システム」の要素がデータもしくは制御を受け取るかという質問に対して、受け取らない要素（つまり DFLOW, CFLOW の目標格に該当しない要素）が示されている。4 番目の質問では「子システム」の要素でデータもしくは制御を受け取るものを一つ例示している。5 番目でも質問を満たす要求文が一つ具体的に示される。

このように利用者は必要とする要求文を検索できるだけでなく、要求記述に誤りがある場合でも、適切な質問を用意し、その結果例証や反例となって提示される要求文を吟味することによって要求記述の正当性を検証できる。

5 要求定義環境: CARD

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 の構成を図 3 に示す。

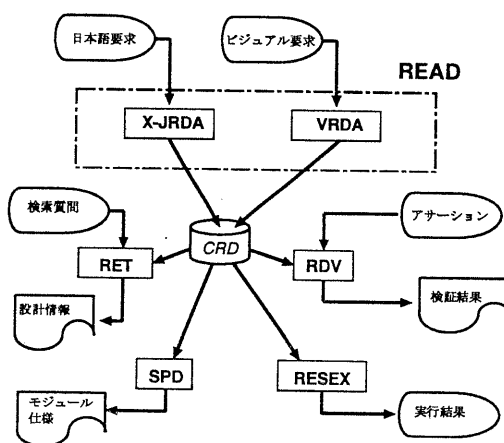


図 3: 要求定義環境: CARD

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
 - 要求記述検索系: RET

● 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。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記述できる。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。今回紹介した検索系のほかにアサーション記述を用いた検証系とデータのフローを利用者からの指示に基づいて対話的に提示していく実行系がある。

設計支援系は、複合/構造化設計技法 [4] の STS 分割に基づき、要求仕様からのモジュール設計を支援する。

CARD 環境は WS 上で KCL と C と X Window System を用いて開発されている。

6 おわりに

要求フレームモデルに基づく要求言語で記述された要求仕様から要求仕様データベースを構築し、またその検索システムを試作した。検索の結果として具体的な要求文を例証あるいは反証として提示するので、要求仕様の正しさの検証にも有効であることを示した。また表層の表現ではなく、深層の表現としてデータベースに格納されるため例えば機能中心に書かれた要求から、特定のデータに関する要求を、そのデータを主語とした要求文に変換して提示することができる。

今後の課題であるが、要求仕様データベースには単文単位で要求が格納されており、上に示した特定のデータを主語とした要求文を羅列しただけではわかりにくい場合もあるので、複数の単文を組み合わせ、重文や複文を再構成し、文脈から推定可能な名詞は省略することによって自然な文章として表現する研究を進めていく予定である。

謝辞

討論に参加してくれる本学理工学部情報工学科 4 回生の谷口由美君と長谷川誠君をはじめとする研究室の学生諸君に感謝する。

本研究は一部文部省科学研究費補助金一般研究 (C)06680318 による。

参考文献

- [1] *Guide to Software Requirements Specifications*, ANSI/IEEE Std. 830-1984, 1984.
- [2] 中所武司:「エンドユーザコンピューティング」、情報処理学会誌 32 巻 8 号, 1991, pp.950-960.
- [3] Davis, A.M.: *Software Requirements, Objects, Functions, & States*, Prentice-Hall, Inc., NJ, 1993.
- [4] Myers, G. J.: *Composite/Structured Design* (国友、伊藤訳ソフトウェアの複合/構造化設計)、近代科学社, 1979.
- [5] 永田 勇: 国文法の基礎、洛陽社, 1970.
- [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, 1990 (pp.175-181).
- [7] 大西 淳:「要求定義のためのコミュニケーションモデル」、情報処理学会論文誌 33 巻 8 号, 1992 (pp.1064-1071).
- [8] Ohnishi, A., Agusa, K.: "CARD: A Software Requirements Definition Environment," Proc. of IEEE Int'l Symp. Requirements Engineering, San Diego, CA, U.S.A., Jan. 1993, pp.90-93.
- [9] Ohnishi, A.: "A Visual Software Requirements Definition Method," Proc. of the IEEE first International Conference on Requirements Engineering (ICRE'94), Colorado Springs, Colorado, U.S.A., April 1994, pp.194-201.
- [10] Thayer, R., Dorfman, M.: "System and Software Requirements Engineering," IEEE Computer Society Press Tutorial, Los Alamitos, CA, 1990.
- [11] Ullman, J.D.: *Principles of Database Systems* 2nd ed., Computer Science Press, Inc. Maryland, 1982.