

あいまいさを扱う論理に基づく要求仕様の高品質化

杉本 英昭, 大西 淳

立命館大学大学院 理工学研究科 総合理工学専攻

525-8577 滋賀県草津市野路東 1-1-1

e-mail: {sugi,ohnishi}@selab.cs.ritsumeai.ac.jp

要求仕様中で矛盾する複数の文を検出し、もっともらしさを Dempster & Shafer の理論に基づいて定量化して解釈する手法について述べる。本稿で述べる手法はあいまいさを扱う理論として Dempster & Shafer の理論を適用しているが、その他のあいまいさを扱う理論を要求仕様の矛盾の解消手法に適用した時の得失を比較し、我々が提案する手法の有用性を述べる。

Making a Software Requirements Specification of Good Quality based on a Logic for Ambiguous Informations

Hideaki SUGIMOTO, Atsushi OHNISHI

Department of Computer Science, Ritsumeikan University

1-1-1 Noji-Higashi, Kusatsu, Shiga 525-8577, Japan

We propose a supporting method of detecting and interpreting the inconsistency of a software requirements specification (SRS) based on the Dempster & Shafer's theory to our method and illustrate our method with an example. We also describe the reason why we adopt the Dempster & Shafer's theory by evaluating several logics handling ambiguous informations.

1 はじめに

我々はソフトウェア要求仕様書に含まれる誤りを分類、定義し、それらの解釈、修正支援手法を提案してきた [13, 14, 16, 15]。また、我々はあいまいさを扱う理論である Dempster & Shafer の理論を、ソフトウェア要求仕様の矛盾の解釈、解消支援手法に適用した [15]。本稿では、あいまいさを扱う理論を要求仕様の矛盾を解消する手法に適用した場合の得失について述べ、我々の提案した手法の有用性を示す。

2 矛盾の分類と検出手法

本研究では要求フレームモデルと名付けた要求記述のためのモデルとそれに基づいて開発した日本語要求言語 X-JRDL を要求仕様化言語として採用している [12]。その理由は、要求フレームモデルを用いることによって、矛盾の検出が簡単に行なえるからである。日本語要求言語 X-JRDL はファイル処理システムを記述対象としており、そのための動詞や名詞が使えるようになってきている。しかしながら X-JRDL は動作概念とそれに対応した動詞を拡張可能であり [13]、その他の分野の記述にも対応できる。X-JRDL で書かれた要求仕様に現れる矛盾を以下の 2 つの場合に分類する。

1 要求文での矛盾 1 つの文単体で矛盾した要求となっている

2 つ以上の要求文間の矛盾 個々の要求文だけを取り出すと、矛盾しないが他の要求文とつき合わせると矛盾した要求となっている

それぞれの場合に対して、どのような矛盾となるかを定義し、さらにそれらの検出手法について述べる。

2.1 1 要求文での矛盾とその検出

要求フレームモデル [12] では、各動作概念に対して格構造が定義されており、該当する格と、格に該当する名詞の型が定められている。この定義に反した格や名詞の型が記述されているとき、すなわち、動作概念の格構造に合わない名詞が記述されているとき、この要求文は単独で矛盾していると見なす。次の要求文をもとに説明する。

在庫管理係は在庫マスタファイルから出庫先を受けとる。

X-JRDL で提供している「受けとる」という動詞に対応する DFLOW という動作概念は表 1 に示すような格構造をもっている。源泉格には、人間型か機能型の名詞があてはまるが、上の例では、「在庫マスタファイル」というファイル型の名詞が記述されている。このように、動詞の格構造に合わない名詞が記述されている時、要求文は矛盾していると見なす。X-JRDL で書かれた仕様を、その解析系によって処理する段階で、格構造に合わない矛盾は自動的に検出される。上の例では矛盾の他にも道具格に相当する名詞が抜けているが、「抜け」という誤りは矛盾と区別しており、本稿では扱わないが、「抜け」を解析系によって解釈する技法は既に確立済みである [13]。

表 1: DFLOW 概念の格構造

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

2.2 2 つ以上の要求文での矛盾とその検出

個々の要求文を取り出すと矛盾しないが、要求文を 2 つ以上組み合わせて解釈すると矛盾が生じる場合がある。X-JRDL による要求仕様においてこのような矛盾が生じる可能性がある、

- 2 つ以上の同一オブジェクト定義での矛盾
- システムの階層構造の階層間での矛盾
- ファイル構造定義とファイル利用での矛盾

について説明する。

2.2.1 同一オブジェクトの定義に関する矛盾

同一の機能の構造やファイルやデータの構造を定義する要求文が複数存在する場合、全く同一の要求であれば「冗長」となるが、同一のオブジェクトの

定義にも関わらず、内容が異なっているならば、異なる定義が2種類以上存在することになり、矛盾と見なす。

以下の2つの要求文について述べる。

文A: 出庫依頼票は 出庫品名と 出庫数と 出庫先から 構成される。

文B: 出庫依頼票は 出庫品名と 出庫数と 依頼者名から 構成される。

これらの2文は「出庫依頼票」という同一のデータを定義しているが、その3番目の構成要素がA、Bそれぞれ「出庫先」、「依頼者名」と異なっている。従って、これらの文は矛盾していると見なす。X-JRDLにおいて「構成される」という動詞は、機能、データやファイルを定義するANDSUBという動作概念に対応し、ANDSUBは図1のような格構造を持っている。この矛盾を検出するには動作主格に同一の名詞が該当するANDSUB文で目的格の個数や内容が異なっているものを検出すればよい。

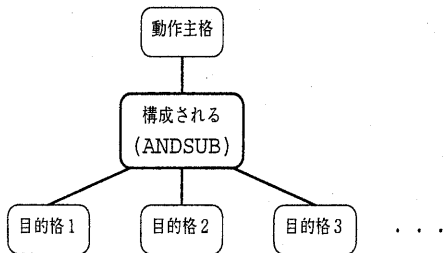


図1: ANDSUBの構造

2.2.2 機能の階層構造に関する矛盾

機能が階層構造をとる時、親となるレベルと子となるレベルの間で、入出力と機能の対応関係がうまく取れないような記述となる場合に、この記述は矛盾している。例えば、図2に示すように、システムAが記述されており、そのサブシステムとしてA1、A2が記述されているとする。システムA,A1の入出力関係が、図2のように記述されている時、すなわち、それぞれの入力が入力I1,I2と記述されて、しかもI1とI2の関係が記述されていない場合、本来は同じ入力にならなければならないところが、違

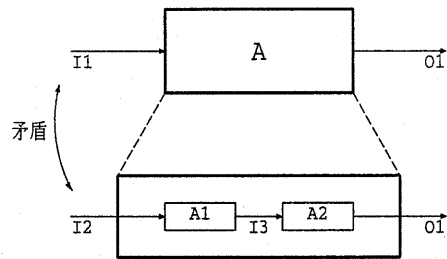


図2: システムの階層構造

う入力記述されている。このような記述がされた場合、これらの要求文は矛盾している。

X-JRDLでは、機能階層の定義、機能階層間のデータ構造の定義、データと機能との関係を表す要求文をつき合わせることによって、これらの間の矛盾を検出している。

2.2.3 ファイル定義とファイル利用に関する矛盾

ファイルの構造を定義する要求文と、そのファイルを利用する要求文が存在し、しかもファイル利用で表れる属性とファイル構造で表れる属性が異なる場合に矛盾と見なす。例えば、

文A: 在庫マスタファイルは品名と数量と 出庫先からなる。

文B: 在庫管理係は在庫マスタファイルから 出庫先を依頼者名で検索する。

で、文Aの構造定義に表れない属性「依頼者名」が文Bでは用いられているため、矛盾となる。X-JRDLでは「検索する」という動詞はRETという動作概念に対応し、これは表2に示す格構造を持っている。従って、RET文に表れる源泉格と同じ源泉格を持つANDSUB文をつき合わせて、RET文の目標格、キー格に該当する名詞が、ANDSUB文の目的格に表れるかどうかを調べれば良い。このように、今まで紹介した要求仕様に表れる矛盾は要求フレームモデルを用いることによって、容易に検出できる。

表 2: RET 文の格構造

動詞	概念	格	名詞の型
検索する 検索される	RET (データの検索)	動作主格	function
		源泉格	file
		目標格	data
		キー格	data

3 矛盾の解釈支援手法

3.1 1 要求文での矛盾の解消

1 要求文での矛盾は要求フレームモデルを用いて解析することによって、格構造に矛盾した要求文が検出されるので、格とそれに該当する名詞の型の矛盾をエラーとして記述者に提示し、修正してもらう。

3.2 2つ以上の要求文間の矛盾の解釈支援

個々の要求文は正しいが、2つ以上の要求文をつき合わせて初めて生じる矛盾を以下のように解釈する。

1. まず2つ以上の要求文間の矛盾を検出する
2. 矛盾した要求文の確からしさを調べる
3. 確からしさを定量化し、その結果に基づいて解釈する
4. 記述者は解釈結果を吟味して、矛盾の解消の参考とする

このうち2番目の「確からしさを調べる」ために、矛盾する要求文の差分を取る。差分で表れた語を含むような要求文は、その差分を含んだ矛盾する要求文を支持すると考えて、支持する要求文がどの位出現するかを調べる。矛盾する要求文が要求仕様の異なる節に現れる時は、現れた節ごとに矛盾する要求文のそれぞれの確からしさを調べ、最終的に確からしさを統合するが、その統合には Dempster & Shafer の確率理論 [3, 8] を適用する。以下では、本研究で提案する解釈支援手法を具体例を用いて説明する。

3.2.1 Dempster & Shafer の確率理論

ある事実が確かである確率と確かであるかどうかを決めかねる確率を計算する理論として、Dempster & Shafer の確率理論がある。この理論は、ある事実を真であると確信する度合い、偽であると確信する度合い、真であるか、偽であるか不明な度合いの3種の度合いで確からしさを表現する理論であり、どちらともいえない、あいまいさを含んだ情報を扱うのに適している。

A.P.Dempster は Bayes 確率に適さない主観にかかわる不確実性を扱うために、下界および上界確率と名付けた新しい考え方を提示した。G.Shafer は Dempster 理論を洗練させ、主観的な意味を付与するために、元来の下界、上界確率をそれぞれ belief function, plausibility と言い換え、それらを基本確率を通じて定義した。Shafer によれば、基本確率 $m(A_i)$ は部分集合 A_i に閉じ込められているが、 A_i 内の各点に自由に動ける半可動確率質量としてイメージを描くことができる。

A_0 を有限な全集合、 $A_i (i = 1, 2, \dots)$ をその部分集合とすると、基本確率 $m(A_i) (i = 0, 1, \dots)$ は $[0, 1]$ の値をとり、次の条件を満たす。

$$\begin{cases} m(\phi) = 0 & (\phi: \text{空集合}) \\ \sum_{A_i \subseteq A_0} m(A_i) = 1 \end{cases}$$

$m(A_i) > 0$ のとき A_i は焦点要素とよばれる。

さらに、独立な証拠から推論された基本確率を統合する方法として、 m_1, m_2 を独立な証拠に基づいて得られた基本確率、 $A_{1i}, A_{2j} (i, j = 0, 1, 2, \dots)$ をそれぞれの焦点要素として、次の式、

$$m(A_k) = \frac{\sum_{A_{1i} \cap A_{2j} = A_k} m_1(A_{1i}) m_2(A_{2j})}{1 - \sum_{A_{1i} \cap A_{2j} = \phi} m_1(A_{1i}) m_2(A_{2j})}$$

があるが、本稿では矛盾する部分の確率を正規化しない(つまり上の式で分母を取った)

$$m(A_k) = \sum_{A_{1i} \cap A_{2j} = A_k} m_1(A_{1i}) m_2(A_{2j})$$

とした式を採用する。この理由は、統合された基本確率の値が正規化したことで小さい値から大きくなることもあり、最初から大きい値の場合と区別して解釈するためである。

$\{\bar{A}, A\}$ 0.3	0.18	0.12
$\{A\}$ 0.7	0.42	0.28
	0.6	0.4
	$\{\bar{A}\}$	$\{A, \bar{A}\}$

図 3: 有罪、無罪の合成尤度を求める例

3.2.2 DS 理論の適用例

DS 理論の適用例として、心証を得るプロセスを例にとる [8]。

ある人 X は有罪である (A とする) か、無罪である (\bar{A} とする) かのいずれかであるとす。また、A を支持する側を検事側、 \bar{A} を支持する側を弁護士側とする。

検事側から出されてきた証拠 1 は 0.7 だけ有罪であることを示していて、(基本確率は $m_1(A) = 0.7$, $m_1(A, \bar{A}) = 0.3$) 弁護士側から出された別の証拠 2 が 0.6 だけ無罪であることを示している (基本確率は $m_2(\bar{A}) = 0.6$, $m_2(A, \bar{A}) = 0.4$) とき、図 3 を参考に合成尤度を求める。ここで、合成尤度はある候補を含むすべての集合の基本確率の総和を表す。また、図 3 の網掛けされた部分は矛盾を、斜線の部分は A または \bar{A} であるので有罪か無罪か決めかねることを表す。

証拠 1、2 を統合して合成尤度を求めると、

- ・有罪である合成尤度は、 $0.28 + 0.12 = 0.40$
- ・無罪である合成尤度は、 $0.18 + 0.12 = 0.30$ になる。

3.2.3 要求文の合成尤度を求める手順

要求文の合成尤度を求める手順を以下に示す。ここで述べる手順は 2 つの要求文の組合せにおいて矛盾が検出された場合の手順であり、3 つ以上の要求文の組合せにおいて矛盾が検出された場合も同様に

して合成尤度を求めることができる。

STEP0: 要求仕様中で 1 つの要求文単独で矛盾している文を 2.1 の手法によって検出し、記述者に修正させる。この時点で単独で矛盾する要求文は全て修正されていると仮定する。

STEP1: 与えられた要求仕様中で単独では矛盾していないが、2 つ以上の組合せで矛盾する要求文の組合せを 1 組検出し、要求文同士を付き合わせて差分をとる。例えば、2 つの矛盾する要求文 A と B で α と β がそれぞれの差分項目とする。

STEP2: 要求仕様の中で矛盾していない要求文の基本確率を全て 1.0 とする。

STEP3: 差分項目を含んだ要求文で矛盾しない要求文を調べる。具体的には、差分項目を含む矛盾しない要求文の出現頻度を各節ごとに求める。差分項目 α を含む他の要求文は A を支持し、差分項目 β を含む他の要求文は B を支持すると考える。差分項目 α を含む文がある節 X で 1 文あって、差分項目 β を含む文が同じ節 X で 5 文あったとする。差分項目 α を含む文が別の節 Y で 10 文あって、差分項目 β を含む文が節 Y で 5 文あったとする。

STEP4: 各節における差分項目の出現頻度の合計を求め、合計の最大数に対してその比が 0.5 より小さい合計が得られた場合は確からしさを表す指標が同程度とは言えないので、小さい節での出現頻度を整数倍してその和が 0.5 以上となるように正規化する。差分項目 α を含む文の出現頻度と β を含む文の出現頻度の和の比は節 X と Y とで、 $6/15 = 0.4$ と 0.5 より小さいので、節 X での出現頻度を 2 倍して、 α を含む文が 2 文・ β を含む文が 10 文あったとしてやる。出現頻度を整数倍するということは、要求文が整数倍した回数だけ要求記述に出現する場合と等価であるが、この様な冗長性は要求記述の意味を変更するものではない。

STEP5: $1 - \prod_i (1 - 0.3 \times p_i)$ (但し、 p_i は差分項目を含んだ要求文 i の基本確率) を計算し、矛盾する要求文の基本尤度とする。上の式で 0.3

は仮に定めた数値である。ある要求仕様において $p_i = 1.0$ であるとしたとき、この式で $i = 3$ の時、式の値は 0.657、 $i = 10$ の時、式の値は 0.972 となる。 $i = 0 \sim 20$ の数値をこの式に代入し、その結果が著者の直観と合うことを確認しているが、さらに利用者の主観に合うよう 0.3 という数値を調整できる。

STEP6: STEP5 で得られた基本尤度から、基本尤度はその候補を含む集合すべての基本確率の和になるように基本確率を割り当てる。

STEP7: 別の節で異なる基本確率が得られた場合は、DS 理論に基づいて矛盾する要求文の合成尤度を計算する。

STEP8: 得られた合成尤度を解釈して、その結果を記述者に提示することにより要求仕様の解釈と修正を支援する。この解釈は絶対的なものではなく、あくまで記述者の参考にしてもらうためのものである。

【得られた合成尤度を解釈する方法】 得られた合成尤度によって次のように場合分けをしてその結果を解釈する。いま、文 A と文 B が矛盾しているとする。上記の手順で得られた合成尤度を、L(ow), M(edium), H(igh) の 3 段階に以下の式に基づいて分ける。

$$0 \leq L < 0.3 \leq M < 0.5 \leq H \leq 1$$

と表し、以下の表 3 の場合分けに従い解釈を行う。ただし、表 3 は (文 A の合成尤度) \geq (文 B の合成尤度) の場合を示しており、(文 A の合成尤度) $<$ (文 B の合成尤度) の場合は、表 3 の A、B を入れ替えることにより解釈する。なお、上記の合成尤度の場合分け、及び表 3 の解釈結果については矛盾項目の出現回数をランダムに 20 例ほど取り上げて実際に手法に適用し、その結果が著者の直観と合うことを確認している。

4 あいまいさを扱う論理

あいまいさを扱う論理を要求仕様の矛盾を解釈する手法に適用する場合を想定して、その得失を比較・検討する [10]。

表 3: 得られた合成尤度を解釈する方法

文 A	文 B	解釈結果
H	H	A か B のどちらかは正しく、どちらかの主格の書き間違いの可能性がある。
H	M	A のほうが正しそうだが、B も支持されているので、間違いとは言いきれない。
H	L	A が正しく、B が間違っている可能性が高い。
M	M	どちらが正しいともいいかねる。片方が正しく片方が間違っているかあるいは両方とも間違っているか。
M	L	B は間違っている可能性が高い。A は決めかねる。
L	L	どちらも間違っている可能性が高い。

4.1 非単調論理

古典論理では公理の増加に従い定理も単調増加するのに対し、非単調論理では公理の増加により定理が減少することがある。この論理が非単調と呼ばれる理由は、公理 (知っている事実) の増加が定理 (それらから証明できる事実) の増加を意味しないからである。すなわち、これまで正しいと考えられていたことが新しい事実の発見により覆ることがある。

要求仕様においてこれを適用してみると、新しい公理が増加しても、定理の増加は意味しない。すなわち、新しい要求文が追加されても、その要求文が常に正しいとは限らない。したがって、この非単調論理は要求仕様においても適用できそうである。

4.2 デフォルト推論

デフォルト推論は非単調論理のうちの一推論方式である。

デフォルト論理では、「逆の情報が成り立たないとき、常識的な結論を導け」という推論規則 (デフォルトルール) を付加することにより、一階論理を拡張する。

たとえば、デフォルトルールで「鳥は通常飛ぶ」という知識を表すと、

$$\frac{Bird(x) : Fly(x)}{Fly(x)}$$

となる。このルールを用いて、結論を導く対象となるインスタンス *Inst* について、

「 $Bird(Inst)$ が成り立ち、 $Fly(Inst)$ と仮定してもそれを否定する情報が導かれないうとき、 $Fly(Inst)$ を結論しよう」

ということを実行するのである。大半の鳥に関しては、「飛ぶ」という常識的な結論を導け、一方、ペンギン、ダチョウ、ひな鳥、羽の傷ついた鳥など、例外的な鳥や例外的な状況という不完全な知識に対しても上のルールで対応することができる。

この推論方式は、常識的な結論を導くことが出来る一方、例外的な、不完全な知識にも対応できるということで、要求仕様に適用することを考えると、新しい要求文が記述された時は常識的な結論を導き、例外的な要求文が記述されたときにも対応できるので一見、矛盾の解釈、解消に適用できそうである。

しかし、要求仕様においては、「常識的な結論」、「例外的不完全な知識」の判断、定義が困難であるという問題がある。また、デフォルト推論は一階述語計算のもとでは、計算不能であるという問題点がある。

4.3 TMS

非単調論理では、新しい知識が追加されるとそれまで成立すると考えられていた命題が成立しなくなる。そのために、新しい知識が追加されるたびに信念の状態が矛盾のないように知識を維持しておかなければならない。

信念の状態を維持するための1つの方法として「真理性維持システム」(TMS: Truth Maintenance System)がある。非単調な推論機構を持つ知識ベースシステムにおいては、

1. 非単調論理で得た結論が新しい知識の追加で成立しなくなる
2. 後に、その知識と矛盾するような新しい知識が得られる

という場合が考えられる。そのために、知識ベースの論理的整合性を回復する必要がある。知識ベースの論理的整合性を回復するためには、不都合となった知識を取り除き、それを導き出す根拠となった知識及びそれから導き出された別の知識を訂正する必要がある。

TMSにおいては、新しい知識(新しい要求文)が必ず正しいものとして知識ベースの論理的整合性を回復するようになっている。しかし、要求仕様の記述においては、新しい要求文(新しい知識)が必ずしも正しいとは限らないので矛盾の解消には適さない。

4.4 サーカムスクリプション

サーカムスクリプションは不完全な知識をもとに結論に飛びつく方法を定式化したものである。当然、数学的な意味での演繹ではなく、サーカムスクリプションの結論は、単なる仮説に過ぎないし、時には矛盾を導入することもある。

この論理の結論は、「単なる仮説に過ぎないし、時には矛盾を導入することもある」ということで、推論方式には適しない。

4.5 ファジィ推論

ファジィ推論は、知識ベースとして保存されている推論規則(これを単に知識と呼ぶ)と与えられた知識から新しい結論を導き出すものである。通常の推論と異なるのは、命題中の変数がすべてファジィ変数、すなわち、あいまいな言語情報で構成されている点である。

要求仕様の「矛盾」を解釈、解消する手法に適用することを考えると、このファジィ推論は適当ではない。「あいまいな言語情報」とは、「もっと多くの」などの表現のことであり、本研究で扱う要求仕様の「矛盾」とは異なるものである。

4.6 Dempster & Shafer の確率理論

Dempster & Shafer の確率理論はある事実が確かである確率と確かであるかどうかを決めかねる確率を計算する理論であり、ある事実を真であると確信する度合い、偽であると確信する度合い、真であるか、偽であるか不明な度合いの3種の度合いで確か

らしさを表現する点に特徴がある。この理論は、どちらともいえない、あいまいさを含んだ情報を扱うのに適しており、本研究で提案してきた手法でもこの理論を適用して、要求仕様中の矛盾する複数の要求文のもっともらしさを定量化して解釈している。

矛盾する要求文の基本尤度を計算する式において利用者の主観に合うように定数を調整する必要があるが、矛盾する要求文のもっともらしさを定量的に表現できる点で他の理論より優れている。

5 おわりに

あいまいさを扱う理論を、要求仕様中で矛盾する複数の文を解釈する手法に適用する場合の得失を論じ、我々が提案した手法の有用性を示した。

今後は手法を現実的な要求仕様に適用して、手法の有用性を確認し、さらに洗練させていく予定である。

参考文献

- [1] Anthony C. W. Finkelstein, et al, "Inconsistency Handling in Multiperspective Specifications" IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL.20, NO.8, AUGUST 1994, pp.569-578
- [2] A. Finkelstein, D. Till, G. Spanoudakis, "Managing Inconsistencies" Joint Proc. SIGSOFT'96 Workshops, 1996, pp.172-174.
- [3] 浅見 直紀: 「エキスパート・システムのあいまい処理にデンプスタ・シェーファ理論を利用する」, NIKKEI ELECTRONICS, no.424, pp.104-106(1987).
- [4] Bashar Nuseibeh, et al, "A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification" IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 20, NO. 10, OCTOBER 1994, pp.760-773
- [5] Easterbrook,S., Nuseibeh,B., "Using ViewPoints for inconsistency management" Software Engineering Journal, Vol.11, 1996, pp.31-43
- [6] Easterbrook,S., "Learning from inconsistency" IEEE Computer Society Press, Proceedings of 8th International Workshop on Software Specification and Design, 1996, pp.136-140
- [7] Nuseibeh,B., "To be and not to be : on managing inconsistency in software development" Proceedings of 8th International Workshop on Software Specification and Design, 1996, pp.164-169
- [8] 石塚 満: 「Dempster & Shafer の確率理論」, 電子通信学会誌, Vol.66, No.9, pp.900-903(1983).
- [9] John Grundy, et al, "Inconsistency Management for Multiple-View Software Development Environments", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL.24, NO.11, NOVEMBER 1998, pp.960-981
- [10] 中島秀之: 「知識表現と Prolog/KR」, 産業図書, 昭和60年
- [11] 大西 淳、阿草清滋、大野 豊: 「要求定義のための要求フレーム」, 情報処理学会論文誌、Vol.28, No.4, pp.367-376(1987).
- [12] 大西 淳、阿草清滋、大野 豊: 「要求フレームに基づいたソフトウェア要求仕様化技法」, 情報処理学会論文誌、Vol.31, No.2, pp.175-181 (1990).
- [13] 大西 淳: 「ソフトウェア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、Vol.33, No.8, pp.1064-1071(1992).
- [14] 大西 淳、杉本 英昭: 「誤りを含んだ要求仕様の検出と解消」, ソフトウェア工学の基礎 IV, 近代科学社, 1997年, pp.83-86
- [15] 大西 淳、杉本 英昭: 「矛盾する要求の解釈支援手法」, 情報処理学会 ソフトウェア工学研究会, 123-9, 1999年, pp.59-66
- [16] 杉本 英昭、大西 淳: 「矛盾を含んだ要求仕様の解釈手法」, ソフトウェア工学の基礎 V, 近代科学社, 1998年11月, pp.106-115