SHIBAURA　MOT　DISCUSSION　PAPER

芝浦工業大学大学院工学マネジメント研究科

ディスカッションペーパー

数学的手法による要件定義分析

安岡　孝司

Discussion Paper No. 2013-2

Shibaura Institute of Technology

Graduate School of Engineering Management

芝浦工業大学大学院

工学マネジメント研究科

〒135-8548 東京都江東区豊洲 3-7-5

SHIBAURA MOT DISCUSSION PAPER 2013-2
2013 年 10 月 4 日

数学的手法による要件定義分析

Mathematical Approach for Requirement Definition Analysis

安岡 孝司

Takashi YASUOKA*

Abstract

The aim of this study was to create a tool that allows unimpeded communication between the developers and users of a system and that allows them to share the same recognition of requirements and requirements definitions. For this purpose, we explored a method to elucidate consensus building between users and developers by writing requirements definitions in the users' operating language. At the same time, we specified user requirements with requirements definitions by simply modeling the system concept. Subsequently, we mathematically defined the concepts of a system and of a requirements definition and derived an algorithm that detects the inadequacy of a requirements definition on the basis of a strict theoretical framework.

概要

本稿はユーザとシステム開発者の双方が要求と要件定義を共有認識し、意思疎通をスムーズ化するツールを提供することを目的としている．そのために、要件定義をユーザの業務言語で表記することで、ユーザと開発者との合意形成を明確化する方法を考える。またシステムの概念を単純にモデル化し、ユーザ要求と要件定義を同一視する。次にシステムと要件定義の概念を数学的に定義し、要件定義の不備を検出するアルゴリズムを厳密な論理展開に基づいて導く。

Keywords : requirement definition, system development, project management, mathematical approach

キーワード ： 要件定義、システム開発、プロジェクトマネジメント、数学的手法

* 芝浦工業大学大学院工学マネジメント研究科　　yasuoka@shibaura-it.ac.jp

1　はじめに

システム開発時に発注者とシステム開発者間の意思疎通が不十分なことが原因で、システム開発が失敗する場合があり、意思疎通の正確化のため様々な方法が提唱・研究されている。その方法がいくら優れていても難解だと、システム開発者にとって扱いにくく、発注者も自分の意図が正しく伝わったかをチェックできない。本研究は発注者とシステム開発者の両者が要求と要件定義を共有認識し、意思疎通をスムーズ化するツールを開発することを目的としている．そのツールでは特殊な学問や専門用語を使わず、発注者の業務言語で表記できるものにすれば、双方の合意形成を可視化できる。ツールに搭載するプログラムで、要件定義の論理的な不備をチェックすれば、両者のコミュニケーションの場で要件定義をチェックできる。図１はそのコンセプトを表したものである。

このアプローチの第一歩として、本稿で扱うシステムは大規模な人数の会員を管理するもので、個々の会員の属性によってオペレーションをひとつ定める静的なものである。そしてシステムと要件定義の概念を数学的に定義し、厳密に理論体系を構築する。このフレームワークの下で要件定義が無謬であるための数学的条件を導き、無謬性の判定法を実行可能な形で表す。しかしこのアルゴリズムには無駄が多いので、さらに計算を簡略化する方法を示す。

これらの結果は直感的に既知のことかもしれないが、本稿では数学的に厳密な議論ができるフレームワークを構築しており、より複雑なものに発展させられるものである。論文中で扱う数学は大学１,２年次程度の集合論のみであり、専門的な知識は不要である。

2　システムと要件定義

コンピュータシステムには様々なものがあるが, 本稿では大量の個人データに対して各個人の属性に依存するオペレーションを対応させる静的なシステムを扱う．このようなシステムを数学上の概念として定義することから議論を始める．

X をデータの集合とし, $x \in X$ をその元とする．O をオペレーション全体の集合とし、その元を o で表す。オペレーションという概念がどこまで含むかを厳密に定義することは難しい。ここでは通常のシステム開発時の要求のようなものを意味していることにしたい。(x, o) は x にオペレーション o が対応すること、つまり、o は x に何らかの行為を行ったり、行わせたりすることを意味する。(x, o) の全体を (X, O) で表す。(X, O) には一つの元 x に複数のオペレーションが対応する場合があったり、オペレーションが紐付けられていない元 x の存在もあり得る。

例 2.1　あるクラブの会員管理システムの場合, $x \in X$ は個々の会員データを表し, X はその全体集合である．o_1 は会員にDMを送る．o_2 はDMを送らないというオペレーションとする．そして $O = \{o_1, o_2\}$ とおく．(x_1, o_1) は x_1 にDMを送ることを表し、(x_2, o_2) は x_2 にDMを送らないことを表す。対応がこの２件のみであれば

$$(X, O) = \{(x_1, o_1), (x_2, o_2)\}$$

である。

会員管理システムで, 時刻 t における会員データの集合を D_t で表すと、入退会によって D_t は時刻とともに変動する. システム開発時 $t=0$ にわかっているデータ D_0 でシステムが要求どおりに稼働しても, その後会員が増加したとき, D_t に対して要求通りに稼働するとは限らない. したがって, 想定しうる最大規模会員の集合 X に対してシステムが発注者の要求どおりに機能する必要がある. たとえば, 国内店舗で使えるポイントカードの会員システムを扱うとき, X は国内在住者全体を想定することになろう.

定義 2.1 (X, O) が以下の i)〜iv) をすべてみたすとき、(X, O) は”半完全システム”であるという. また i)〜v) をすべてみたすとき、”完全システム”であるという.

i) $i \neq j$ のとき $o_i \neq o_j$

ii) すべての $o \in O$ は X を変えない

iii) すべての $o \in O$ は O を変えない

iv) オペレーション O はある時刻に同時に X に作用する.

v) すべての $x \in X$ に対してあるオペレーション o_i が一意的に定まる.

たとえば預金管理システムの場合, $x \in X$ は各会員の預金残高を表し, O は預金者の現金を引き出すオペレーションとする. このとき引き出しによって残高 x が変わる. 条件 ii) はこのようなオペレーションを除外している.

$x_1, x_2 \in X$ の各元にそれぞれ o_1, o_2 のオペレーションが作用するとしよう. o_1 は x_1 にピザを届ける, o_2 は x_2 にそのピザを途中で奪わせる, としよう. 結果的に x_1 はピザが届かないので o_2 は o_1 を変えたことになる. 条件 iii) はこのようなオペレーションを除外している.

会員数や属性は時刻とともに変化するが, 上の条件は固定された状態ですべての x になんらかのオペレーションを一意的に行うことを意味する. 一般のシステムでは時間経過やオペレーションによって X の内容や O が変化するが, 各時刻ごとに更新後の X と O を固定して考えれば定義 2.1 の条件でも実際のシステムの多くに適用できるであろう. 本稿では (X, O) は常に半システムの条件を満たすものとして議論を進める。(X, O) が完全システムのとき (x, o) から写像 $f(x) = o$ を定義でき、f は X から O への全域写像である。文脈によっては f を完全システムと呼ぶ場合もあるが、混乱は生じないであろう。

以下では $C = \{C_i\}_i$ は X の部分集合の族とする. F を C から O への写像とし, (F, C) を”要件定義”という. 本稿の目的は要件定義 (F, C) がどのような条件を満たせは完全システム f を構築できるかを示すことである.

定義 2.2 要件定義 (F, C) が次の条件を満たすとき (F, C) は X 上で”無謬”という.

i) $\cup_i C_i = X$

ii) $C_i \cap C_j \neq \phi$ なら, $F(C_i) = F(C_j)$ である.

命題 2.1　X, O, F, C が与えられているとする．(F, C) が X 上で無謬のとき，X から O への全域写像 f が一意的に存在する．

証明　定義 2.2.i) より，任意の $x \in X$ についてある C_i が存在し，$x \in C_i$ である．f を $f(x) = F(C_i)$ と定義する．x を含む C の元が 2 個存在するときはそれらを C_i, C_j とすると，定義 2.2.ii) より $F(C_i) = F(C_j)$ である．これによって $f(x)$ が一意的に定まるので，f は X から O への写像である．$f(x)$ が X からの全域写像であることは条件 i) からわかる．証明終.

この命題により，要件定義 (F, C) が無謬か否かをチェックすれば，完全システム f を構築できる．

例 2.2　X は日本在住の成人全体の集合，$D(\subset X)$ はあるクラブの会員とする．発注者の要求を"男性会員には誕生日にネクタイを贈り，女性会員には誕生日に花を贈る"としよう．$C = \{C_1, C_2\}, O = \{o_1, o_2\}$ とし，$C_1 =$ "男性会員", $C_2 =$ "女性会員" とおくと、$C_1 \cup C_2 = X$ を満たす．次に $o_1 =$ "誕生日にネクタイを贈る", $o_2 =$ "誕生日に花を贈る" として，$F(C_1) = o_1, F(C_2) = o_2$ と定めると，定義 2.2.ii) から (F, C) は X 上で無謬である．この要件定義では会員の増減によらず，すべての会員にオペレーションを一意的に定義できることもわかる．

実務で要件定義をチェックするときも，c や o などの記号を使う必要はなく，例 2.2 で扱ったような文字列をそのまま使えばよい．たとえば $F(C_1) = o_1$ は

"男性会員" $\to$ "誕生日にネクタイを贈る"

と表現すればよい．このようにすると発注者の要求が業務上の言葉（文字）で表現されるためわかりやすく，発注者とシステム開発者の間で認識を共有できる．

次に集合 X とその部分集合族 $C = \{C_1, C_2, \cdots\}$ が存在し，F は C から O への写像とする．また $\tilde{X}, \tilde{C} = \{\tilde{C}_1, \tilde{C}_2, \cdots\}, \tilde{F}$ も同様とする．2 組の要件定義 (F, C) と $(\tilde{F}, \tilde{C})$ は同じオペレーション O を値域に持つ．

定義 2.3　次の 3 条件をすべて満たすとき，(F, C) と $(\tilde{F}, \tilde{C})$ は"O-同型"という．

i) C から $\tilde{C}$ への全単射が存在する．簡単のため任意の $i = 1, 2, \cdots$ に対して C_i は $\tilde{C}_i$ に対応するものと仮定する．

ii) $C_i \cap C_j = \phi$ が成立するのは，$\tilde{C}_i \cap \tilde{C}_j = \phi$ が成立するときのみである．

iii) すべての $i = 1, 2, \cdots$ に対して $F(C_i) = \tilde{F}(\tilde{C}_i)$ が成立する．

明らかに O-同型は同値関係である．次の命題は，同値な要件定義が 2 種類存在するとき，データ構造の簡単な方で要件定義が無謬か否かを判定すればよいことを意味する．

命題 2.2　$\cup_i C_i = X$ と $\cup_i \tilde{C}_i = \tilde{X}$ であり，(F, C) と $(\tilde{F}, \tilde{C})$ は O-同型と仮定する．このとき，(F, C) が X 上で無謬なら $(\tilde{F}, \tilde{C})$ は $\tilde{X}$ 上で無謬であり，この逆も成立する．

証明　仮定より定義 2.2.i) は満たされているので, 定義 2.2.ii) を示せばよい. $\tilde{C}_i \cap \tilde{C}_j \neq \phi$ とする. このとき, 定義 2.3.ii) から $C_i \cap C_j \neq \phi$ である. (F, C) が X 上で無謬なら, $F(C_i) = F(C_j)$ である. 定義 2.3.iii) から $\tilde{F}(\tilde{C}_i) = \tilde{F}(\tilde{C}_j)$ で, $(\tilde{F}, \tilde{C})$ は $\tilde{X}$ 上で無謬である. 逆も同様である.

定義 2.3 において C と $\tilde{C}$ の各元の添え字は自然数タイプのものに限っているわけではなく、なんらかの集合として定義されるものであればよい。その場合にも命題 2.2 が成立することは明らかである。
$\{S_1, \cdots, S_m\}$ は $i \neq j$ のとき $S_i \cap S_j \neq \phi$ を満たす集合族とし, $X = \prod_{i=1}^{m} S_i$ とする. つまり $x \in X$ は $x = (x_1, \cdots, x_m)$ で表され, $x_i \in S_i, i = 1, \cdots, m$ である. 経験的に, 会員管理システムのデータはこのような直積型の構造である（例??.1 を参照）. この節では, 直積型の構造をもつ X 上の要件定義をチェックする方法を構築する.

定義??.1　$C_i = \{C_{i1}, \cdots, C_{i\kappa_i}\}$ を S_i の部分集合族とする. ここで $\kappa_i \geq 2$ とする. 次の 3 条件をすべて満たすとき, C_i は S_i の"直和分割"という.

i) $\cup_{j=1,\cdots,\kappa_i} C_{ij} = S_i$

ii) すべての $j = 1, \cdots, \kappa_i$ について, $C_{ij} \neq \phi$, ここで κ_i は S_i の分割数.

iii) $l \neq k$ のとき $C_{il} \cap C_{ik} = \phi$

本稿では S_i を"属性", C_{ij} を S_i の"セグメント"と呼ぶ. 以降では C_i はいつも S_i の直和分割になっているとする. また以下の議論で S_i 全体を扱うことがあるので, $C_{i0} = S_i$ とする. C_{i0} は C_i には含まれないことを注意しておく. 次に λ を次式で表される添え字の組とする.

$$\lambda = (\lambda_1, \cdots, \lambda_m), \quad 0 \leq \lambda_i \leq \kappa_i$$

λ に対応する X の部分集合 E_λ を次式で定義する.

$$E_\lambda = \prod_{i=1}^{m} C_{i\lambda_i} = \{(x_1, \cdots, x_m) \in X;\ x_i \in C_{i\lambda_i}\} \tag{2.1}$$

λ の集合を Λ で表し, E_λ の全体からなる部分集合族を E で表す. つまり $E = \{E_\lambda\}_{\lambda \in \Lambda}$ である. E_λ を"要件セグメント"と呼ぶことにする。個々の λ は発注者のさまざまな要求に対応し, Λ は要件セグメントの全体に対応している.

例??.1　S_1 は会員番号, S_2 は名前, S_3 を住所の属性をもつ集合とする. 個々のデータは

$$x = (\text{会員番号}, \text{名前}, \text{住所})$$

の形をしている. とくに C_{31} は首都圏, C_{32} は首都圏外というセグメントを考えれば,

$$C_3 = \{C_{31}, C_{32}\} = \{\ \text{首都圏}, \text{首都圏外}\ \}$$

は S_i の直和分割である. $\lambda = (0, 0, 1)$ のとき, 会員番号, 名前の属性は無視され,

$$E_{(0,0,1)} = \{\ \text{住所が首都圏}\ \}$$

という部分集合を意味する．

次に E から O への全域写像 F による要件定義 (F,E) を考える．このとき定義 2.2 から次の命題を得る．証明は省略する．

命題 2.3 次の *2* 条件を満たすとき，(F,E) は X 上で無謬である．

i) $\cup_{\lambda\in\Lambda}E_\lambda = X$

ii) $\lambda,\gamma\in\Lambda$ に対して $E_\lambda\cap E_\gamma\neq\phi$ なら,$F(E_\lambda)=F(E_\gamma)$ である．

実務で (F,E) の無謬性を検証できるかという観点で考えると，命題 2.3 は実現可能なアルゴリズムを導かない．その理由は以下のとおりである。X を日本在住者とし,時刻 t での会員を $D_t\subset X$ とする．$t=0$ に (F,E) が D_0 上で無謬だとしても，その後の時刻 t に会員増が生じた時，(F,E) が D_t 上での無謬性を $t=0$ の時点では判定できない．また X は非会員を含むが非会員のデータは入手できないため，条件 ii) をチェックできない．

次に実現可能な判定法を構築しよう．Γ は $\gamma_i, i=1,\cdots,m$ が正整数をとるすべての添え字の組み合わせの集合とする．これは次で表される．

$$\Gamma=\{\gamma=(\gamma_1,\cdots,\gamma_m);\quad 1\leq\gamma_i\leq\kappa_i\}$$

G_γ を次で定め，$G=\{G_\gamma\}_{\gamma\in\Gamma}$ とする．

$$G_\gamma=\prod_{i=1}^{m}C_{i\gamma_i}=\{(x_1,\cdots,x_m)\in X; x_i\in C_{i\gamma_i}\} \tag{2.2}$$

このとき，$\cup_{\gamma\in\Gamma}G_\gamma = X$ であり，$\gamma\neq\gamma'$ に対して $G_\gamma\cap G_{\gamma'}=\phi$ であることから，G は X の直和分割である．またこれ以上細かい要件セグメントは存在しないので，G を C から生成される X の"最細分割"と呼ぶ．明らかに，C から生成される X の最細分割は一意的である．$m=2$ なら G は表などで可視化されるが，実務のデータ構造において m は数十以上の規模を想定すべきであり、とても可視化できない．そのため要件定義分析は極端に困難になる．

次の定理は要件定義の実現可能なチェック法を導くものである．その理由は D や X を扱わずに最細分割を使っていることによる．

定理 2.4 (F,E) が X 上で無謬であるための必要十分条件は次の 2 条件が成立することである．

i) 任意の $\gamma\in\Gamma$ に対して，ある $\lambda\in\Lambda$ が存在し，$G_\gamma\subset E_\lambda$ である．

ii) 任意の $\gamma\in\Gamma$ に対して $G_\gamma\subset E_\lambda\cap E_\xi$ を満たす $\lambda,\xi\in\Lambda$ が存在するとき，$F(E_\lambda)=F(E_\xi)$ である．

証明 i) と ii) が必要条件であることは明らかなので，十分条件であることを示せばよい．G は X の直和分割なので，任意の $x\in X$ に対してある γ が存在し $x\in G_\gamma$ である．条件 i) より，ある $\lambda\in\Lambda$ が存在し，$x\in G_\gamma\subset E_\lambda$ である．したがって $\cup_{\lambda\in\Lambda}E_\lambda = X$ がわかる．

$x \in E_\lambda \cap E_\xi$ が存在するならある γ が存在し, $x \in G_\gamma$ である．G は最細分割なので G_γ は E_λ と E_ξ 双方の部分集合である．つまり, $G_\gamma \subset E_\lambda \cap E_\xi$．条件 ii) より $F(E_\lambda) = F(E_\xi)$ である．命題 2.3 より (F, E) は X 上で無謬である．証明終.

定理 2.4 によって要件定義 (F, E) の無謬性は次の手順によってチェックできる．（　）内は不備が検出されたときの対応法を示す.

要件定義チェックの手順

(1) $\gamma \in \Gamma$ を固定し, $G_\gamma \subset E_\lambda$ となる $\lambda \in \Lambda$ が存在するかを判定する.

(2) 上の λ が存在すれば条件 i) をみたす．もし λ が存在しなければ, (F, E) は無謬ではない．（最細セグメント G_γ に関連する属性セグメントとオペレーションを追加する）

(3) $\gamma \in \Gamma$ に対して $G_\gamma \subset E_\lambda \cap E_\xi$ を満たす $\lambda, \xi \in \Lambda$ が存在するかを判定する

(4) (3) の λ, ξ が存在する場合は $F(E_\lambda) = F(E_\xi)$ であるかをチェックする.

(5) $F(E_\lambda) \neq F(E_\xi)$ のときは, E_λ, E_ξ の要件セグメントに対するオペレーションが不整合である．（該当する要件セグメントとオペレーションを見直す）

(6) すべての $\gamma \in \Gamma$ に対して (1) から (5) を繰り返す.

このアルゴリズムのうち (1) と (3) のプロセスは次の命題によって実行可能である．証明は省略する.

補題 2.5 任意の $\gamma \in \Gamma, \lambda \in \Lambda$ に対して $G_\gamma \subset E_\lambda$ である必要十分条件は, すべての $i = 1, \cdots, m$ で $\gamma_i = \lambda_i$ か $\lambda_i = 0$ が成立することである.

実務で扱う会員データの属性は非常に多く, オペレーションの内容によって要件定義に使用する属性は異なっていることが一般的である．例 2.2 でも, 会員に誕生日プレゼントを贈るというオペレーションにおいて性別の属性しか使っていない．また会員にDMを送るかどうかというオペレーションを定義するときは別の属性で判断するであろう．すべての属性データを使って定理 2.4 を適用すると、明らかに計算の無駄が多いので効率化したい.

直積型のデータ X, S, C, E, Λ が与えられ, (F, O) は E から O への要件定義とする．X の最細分割 G, Γ は S, C から自動的に生成される．固定された自然数 $l < m$ があり, Λ の元はすべて $(\lambda_1, \cdots, \lambda_l, 0, \cdots, 0)$ の形をしているものとする．$\tilde{\Lambda}, \tilde{\lambda}, \tilde{E}, \tilde{E}_{\tilde{\lambda}}$ を以下で定義する.

$$\begin{aligned}
\tilde{\Lambda} &= \{\tilde{\lambda} = (\lambda_1, \cdots, \lambda_l); (\lambda_1, \cdots, \lambda_l, 0, \cdots, 0) \in \Lambda\} \\
\tilde{E}_{\tilde{\lambda}} &= \prod_{i=1}^{l} C_{i\lambda_i} \\
\tilde{E} &= \{\tilde{E}_{\tilde{\lambda}}\}_{\tilde{\lambda} \in \tilde{\Lambda}}
\end{aligned}$$

このとき, $\tilde{F}(\tilde{E}_{\tilde{\lambda}}) = F(E_\lambda)$ と定義すると, $(\tilde{F}, \tilde{E})$ は O を値域とする要件定義である．そしてこの属性セグメント $\tilde{E}$ は, 属性 $S_j, j > l$ を参照しない構造である.

命題 2.6 $(\tilde{F}, \tilde{E})$ と (F, E) は O-同型である.

証明　任意の $\tilde{\lambda} \in \tilde{\Lambda}$ から $\lambda \in \Lambda$ を $\lambda = (\lambda_1, \cdots, \lambda_l, 0, \cdots 0)$ で対応づける. この関係は $\tilde{\Lambda}$ と Λ の間の全単射なので, $E_\lambda \in E$ に $\tilde{E}_{\tilde{\lambda}}$ を対応させれば E から $\tilde{E}$ への全単射が自然に導かれる. したがって定義 2.3 の i) と iii) が成立する.

次にある $\tilde{\lambda}, \tilde{\gamma} \in \tilde{\Lambda}$ が存在して $\tilde{E}_{\tilde{\lambda}} \cap \tilde{E}_{\tilde{\gamma}} \neq \phi$ であるとしよう. このとき $\tilde{x} = (x_1, \cdots, x_l)$ が存在し, 各 $i, 1 \leq i \leq l$ について $x_i \in C_{i\lambda_i} \cap C_{i\gamma_i}$ である. 次に $x = (x_1, \cdots, x_l, x_{l+1}, \cdots, x_m)$ とする. ここで $x_j, j > l$ については $x_j \in S_j$ であれば何でもよい. すると $x \in E_\lambda \cap E_\gamma$ なので $E_\lambda \cap E_\gamma \neq \phi$ である. 逆も同様に示すことができる. 証明終.

集合 $\tilde{X}$ を $\tilde{X} = \prod_{i=1}^{l} S_i$ で定めると, 次の定理を得る. この結果は (F, E_λ) の無謬性をチェックするには $(\tilde{F}, \tilde{E})$ をチェックすればよいことを意味する. つまりオペレーションの決定に必要な属性のみを抽出して属性セグメントを構成すればよい.

定理 2.7 $(\tilde{F}, \tilde{E})$ が $\tilde{X}$ 上で無謬なら (F, E) は X 上で無謬である.

証明　仮定より

$$\tilde{X} = \cup_{\tilde{\lambda} \in \tilde{\Lambda}} \tilde{E}_{\tilde{\lambda}} \tag{2.3}$$

が成立している. 任意の $x = (x_1, \cdots, x_m) \in X$ に対して $\tilde{x} = (x_1, \cdots, x_l)$ を考えると, (2.3) より $\tilde{\lambda} \in \tilde{\Lambda}$ が存在し, $\tilde{x} \in \tilde{E}_{\tilde{\lambda}}$ である. $\lambda = (\lambda_1, \cdots, \lambda_l, 0, \cdots 0)$ とおくと, $x \in E_\lambda$ であり, $X = \cup_{\lambda \in \Lambda} E_\lambda$ がわかる. 命題 2.2 と命題 2.6 から (F, E) は X 上で無謬である. 証明終.

例??.1　S_1 は住所, S_2 は性別, S_3 は名前という属性の集合とする. 個々のデータは

$$x = (\text{住所}, \text{性別}, \text{名前})$$

の形をしている. 全会員の誕生日に何かひとつプレゼントを送るという発注者要求を考える. 発注者要求は"首都圏の人にネクタイを贈る", "女性には花を贈る"としよう. この場合名前は使われていないので, $\tilde{X} = \{(\text{住所}, \text{性別})\}$ の上で要件定義の無謬性をチェックすればよいことを定理 2.7 は示している.

定理 2.7 に従ってこの要件定義を展開すると図 2 のようになる. この要件定義では, 首都圏外の男性にはプレゼントが贈られないので定義 2.2.i) を満たさない. また首都圏の女性にネクタイと花を贈ることになり定義 2.2.ii) を満たさない.

このような簡単な例では要件定義が図 2 のように可視化されるのでわかりやすいが, 定理 2.4 と定理 2.7 は大規模データであっても、この図の考え方で要件定義の無謬性をチェックできることを示している.

3　終わりに

定理 2.4 はエクセル VBA などでも実現できる。各属性、セグメント、オペレーションを文字入力できるようにすれば、要件定義に関する専門知識がなくても発注者要求をツール上に記述していくだけで、要件定義の無謬性をチェックできる。図 2 はそのように試作したエクセルシートの入出力例である。

実務ではひとつの元に複数の独立なオペレーションを対応させる場合が一般的だが、その場合は目的別に (F, E) の無謬性を判定すればよい。

本稿では同時刻に行うオペレーションを一意的に定めるシステムを対象としたが、複数のオペレーションの中から優先度の高いオペレーションを対応させるシステムに発展させることもできるであろう。例えば買い物で、ある要件セグメントは 1 割引きのサービスを受け、別の要件セグメントは 2 割引きのサービスを受けられるとする。そして双方のセグメントに含まれる会員は 2 割引きのサービスを受けられるような場合である。

他に、会員のクレジットスコアを決めるような、加法的なオペレーションにも発展させることも考えられる。

謝辞

本研究のモチベーションとなる要件定義の課題は平田貞代准教授（芝浦工大 MOT) からご教示いただいた.

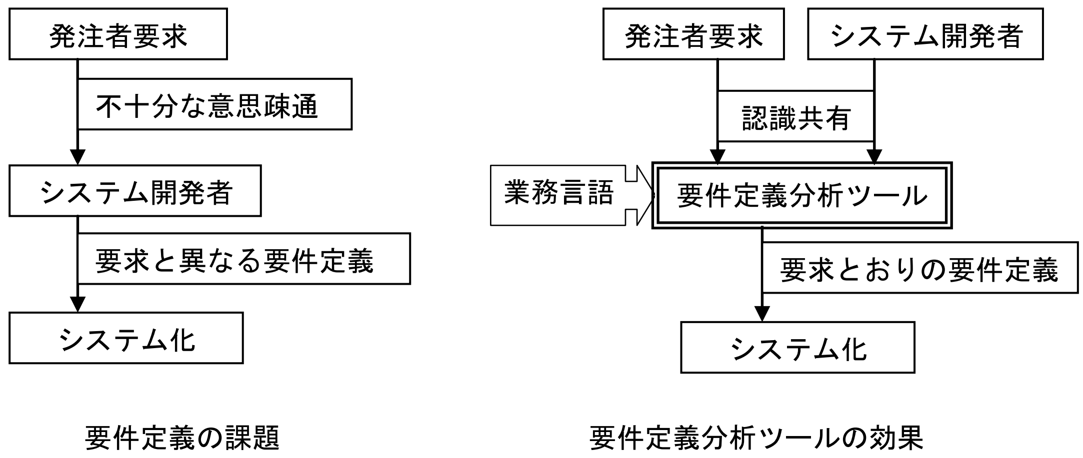


図１　　要件定義の課題と本研究の考え方

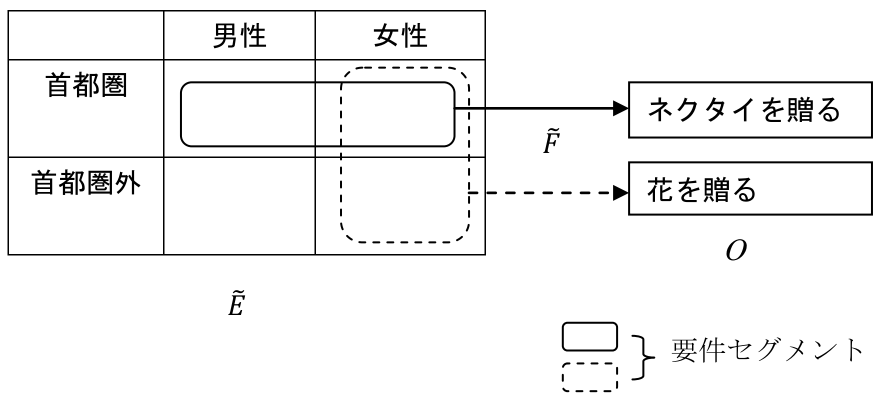
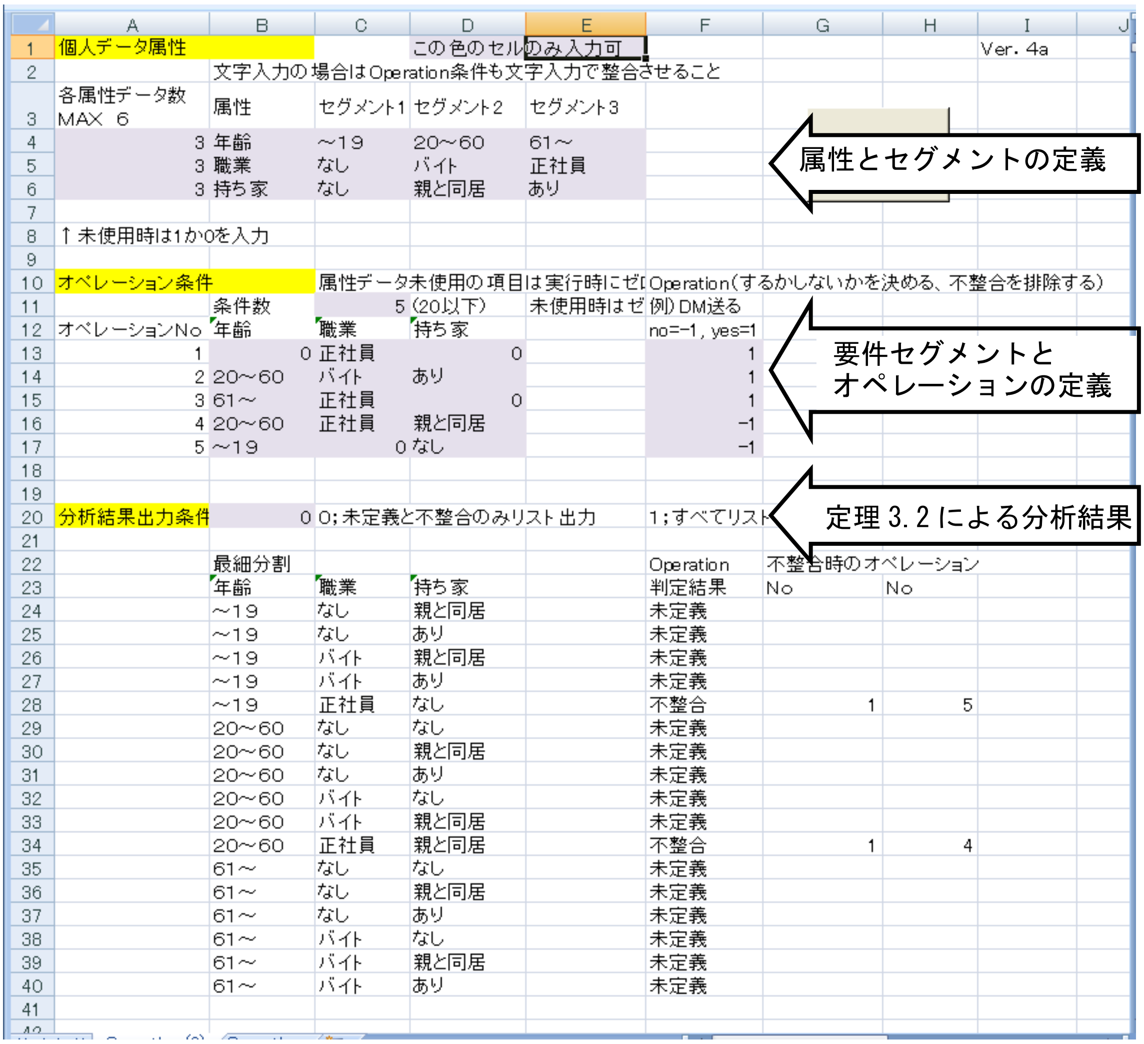


図２　要件定義の例
表の各枠は最細セグメント



	A	B	C	D	E	F	G	H	I	J
1	個人データ属性			この色のセルのみ入力可					Ver. 4a	
2		文字入力の場合はOperation条件も文字入力で整合させること								
3	各属性データ数 MAX 6	属性	セグメント1	セグメント2	セグメント3					
4	3	年齢	～19	20～60	61～					
5	3	職業	なし	バイト	正社員					
6	3	持ち家	なし	親と同居	あり					
7										
8	↑未使用時は1か0を入力									
9										
10	オペレーション条件		属性データ未使用の項目は実行時にゼ			Operation(するかしないかを決める、不整合を排除する)				
11		条件数	5	(20以下)	未使用時はゼ	例) DM送る				
12	オペレーションNo	年齢	職業	持ち家		no=-1, yes=1				
13	1	0	正社員	0		1				
14	2	20～60	バイト	あり		1				
15	3	61～	正社員	0		1				
16	4	20～60	正社員	親と同居		-1				
17	5	～19	0	なし		-1				
18										
19										
20	分析結果出力条件	0	0;未定義と不整合のみリスト出力			1;すべてリスト				
21										
22		最細分割				Operation	不整合時のオペレーション			
23		年齢	職業	持ち家		判定結果	No	No		
24		～19	なし	親と同居		未定義				
25		～19	なし	あり		未定義				
26		～19	バイト	親と同居		未定義				
27		～19	バイト	あり		未定義				
28		～19	正社員	なし		不整合	1	5		
29		20～60	なし	なし		未定義				
30		20～60	なし	親と同居		未定義				
31		20～60	なし	あり		未定義				
32		20～60	バイト	なし		未定義				
33		20～60	バイト	親と同居		未定義				
34		20～60	正社員	親と同居		不整合	1	4		
35		61～	なし	なし		未定義				
36		61～	なし	親と同居		未定義				
37		61～	なし	あり		未定義				
38		61～	バイト	なし		未定義				
39		61～	バイト	親と同居		未定義				
40		61～	バイト	あり		未定義				
41										

図３　要件定義分析ツール　Exel VBA での実装例