Vol.2014-SLDM-165 No.6
Vol.2014-EMB-32 No.6
2014/3/15

システム開発のための不確実性フレームワーク

中西 恒夫[1]　馬 立東[1]　久住 憲嗣[1]　福田 晃[1]

概要：システムの開発活動は相互に関連し束縛しあう意思決定の連続である。現実のシステム開発では，情報の不足ゆえに完全な意思決定ができず，不確実性を孕んだままシステムの開発を進めていく必要が生じる。意思決定の結果，システムの要求，特に非機能要求を満足できるかどうかもまた不確実である。本稿では，システム開発における不確実性を定義，分類するとともに，ソフトウェアプロダクトラインのパラダイムを参考にして，不確実性を決定木によりモデリングし，そのシステム開発における影響を評価し，不確実性を解決するためのフレームワークを構築する。

キーワード：不確実性, システム開発, 非機能要件, ソフトウェアプロダクトライン

A Framework to Manage Uncertainty in System Development

Tsuneo Nakanishi[1]　Lidong Ma[1]　Kenji Hisazumi[1]　Akira Fukuda[1]

Abstract: System development is an acitivity of inter-related decision makings. It often happens that sufficient information for decision making is not available in real system development. System developers are often required to continue development with uncertainty. Morever, it is also uncertain whether an instance of decision making satisfies requirements, especially non-functional requirments, or not. This paper gives definition and categorization of uncertainty, and referencing the paradigm of software product line, establish a theoretical framework to model uncertainties by the decision tree, asses their effects, and resolve them.

Keywords: Uncertainty, System Development, Non-Functional Requirements, Software Product Line

1. はじめに

あらゆるシステムの開発活動は相互に関連し束縛しあう意思決定の連続である。過去に類似システムを開発，運用した経験があれば，その経験を活かして，システムの開発から運用に至るライフサイクルの中で生じる種々の意思決定を妥当に行うことができる。システムが開発され運用されるビジネス面ならびに技術面における背景に対する深い理解が，こうした妥当な意思決定に大きく寄与することは論を待たないであろう。

マイクロプロセッサの低廉化や処理能力の大幅な向上により，昨今はあらゆる事業領域においてシステムの機能実現，さらにはそれによる付加価値向上をソフトウェアに大きく依存するようになっている。組込みシステムのように機械的，電気的機構をも含むシステムの場合，制御対象となるそれら機構ができあがるまではソフトウェアの開発を進めていくことが難しい。そのためソフトウェア開発はどうしても遅い出発になりがちである。ソフトウェアの大規模化と複雑化が相当に進んでいることもあって，ソフトウェア開発の遅延がシステム全体の開発の遅延につながるようになってきている。開発の遅延の原因は稚拙なプロジェクト管理，見積もりの甘さ，要件定義の不備など多様に考えられるが，上流工程での不確実性，すなわち手持ちの情報ではシステム開発に係る意思決定を誰も下せない問題を挙げることができる。

不確実性が生じる原因は，単純に不勉強という場合もあるが，開発者の責任にできない原因としては，開発に関わる誰もがシステムの運用されるビジネス面，技術面の環境を理解したり，その将来の変化を予測したりすることが困

[1] 九州大学; Kyushu University, Fukuoka 819–0395, Japan; {tun, malidong, nel, fukuda}@f.ait.kyushu-u.ac.jp

ⓒ 2014 Information Processing Society of Japan

難であることが挙げられる。ビジネス面については環境の理解よりも環境の変化を予測することが難しい。たとえば法令（特に政令）の改正，競合する他社の製品のリリースや特許，通貨レートの変動は，情報収集や動向分析を通してある程度は予測できても，完全かつ正確に予測するのは困難である。技術面における環境の理解が難しいケースは自動車やロボットなど物理環境と密接に相互作用するシステムの開発でよく見られる。システムが運用される物理環境が複雑かつ多様であるためによく理解されておらず，そのモデル化（定式化）がうまくできていなければ，結局，システムを実際に動かしてみるまでわからないことが残りがちである。技術面での環境の変化が予測し難い例としては機械的，電気的機構の設計変更や部品の変更や廃番などが挙げられる。

不確実性は意思決定を行うときのみならず，意思決定を行った結果においても生じる。意思決定の結果，システムの要求，特に非機能要求を満足できるかどうかもまた不確実である。不確実性と意思決定は相互依存の関係にある。

不確実性は後工程での仕様変更，すなわち手戻りのリスクとなるので除去されるべきものである。しかし，システムが運用される環境について，科学的に完全に解明できていないこと，予測できないことがある以上は不確実性を避けることはできない。システム開発の停滞，遅延を避けるためには不確実性と共存しつつ開発を進めていくよりほかはない。また，システムの開発段階で完全に解決できない不確実性については，人手の介在を許す運用時の解決についても検討すべきであろう。そのためには不確実性が認識され，記述され，複雑で大きな不確実性はより単純で小さな不確実性に分割され，システムのライフサイクルを通して開発者の支配下におかねばならない。

本稿では，著者らの不確実性に関する既発表研究 [1] を発展させ，システムの開発，運用における不確実性に関する諸概念の整理とフレームワークを与える。

2. 不確実性の概念

本節では，「不確実性」の概念を定義し，理論的な整理を行う。

2.1 不確実性の定義

システム開発において必要な情報のうち，開発者によって認知され，定義され，命名され，かつ未解決となっているものを**不確実性**と呼ぶ。この定義の意味するところを以下に述べる。

第一に，不確実性は開発者によって認知されなければならない。開発を行ううえで必要な情報であったにもかかわらず後工程において見出されたもの，いわゆるモレヌケを本稿では不確実性とは呼ばない。開発を進めるうえで決めておかなければならないと認知された情報が不確実性である。

第二に，不確実性はつかみどころのない漠然とした情報ではなく，期待されている有り様が明確に記述される，すなわち情報が得られているか否かを明確に判断できるように定義されなければならない。

第三に，不確実性は開発者の間での議論を容易にすべく，情報の内容を抽象化した名前が与えられなければならない。

2.2 不確実性の分類と属性

不確実性はその理由がどこにあるかによって，**外部不確実性**と**内部不確実性**に分類される [1]。前者は理由がシステム開発プロジェクトの責任範囲の外にあるもの，後者は内にあるものである。

システム開発における不確実性は意思決定に必要な情報と意思決定の結果の中に存在する。前者を**一次不確実性**，後者を**二次不確実性**と呼ぶ。一次不確実性はシステム実現の課程における不確実性であり，二次不確実性はシステム実現の結果における不確実性である。システムの非機能要件を満足できるか否かは代表的な二次不確実性である。

不確実性の一般属性として以下が考えられる。

- **名前:** 意思決定または意思決定の結果の内容を適切かつ簡潔に表現する名前。
- **内容:** 意思決定または意思決定の結果の内容を記述する文。疑問文のかたちで記述する。
- **分類:** 外部不確実性か内部不確実性か，一次不確実性か二次不確実性かの区別。

また，一次不確実性を意思決定の期待されている答えの有り様によって，**多肢選択型**，**パラメータ型**，**複合型**とに分類する。

多肢選択型は m（$1 \leq m$）個の**解候補**の中から最小 l，最大 u（$0 \leq l \leq u \leq m$）個を解として選択するような意思決定である。$l = u = 1$ の場合，特に**択一型**と呼ぶ。多肢選択型の不確実性の属性として以下が考えられる。

- **解候補数:** 当該意思決定の選択肢の数。
- **解最小数:** 当該意思決定の解として選択できる選択肢の最小の個数。
- **解最大数:** 当該意思決定の解として選択できる選択肢の最大の個数。

一方，パラメータ型は指定の型（整数，実数，文字列等）の何かしらのパラメータを定めるような意思決定である。パラメータ型の不確実性の属性として以下が考えられる。

- **型:** パラメータの値の型。
- **制約:** パラメータ自体が（他のパラメータとは独立して）満足すべき制約。

複合型は抽象的な不確実性であり，当該意思決定の結果に関係する，ひとつ以上のより具体的な複合型，多肢選択型，パラメータ型の下位の不確実性で構成される。複合型の不確実性はその下位の不確実性がすべて解決されたとき

ⓒ 2014 Information Processing Society of Japan

に解決される。複合型の不確実性に固有の属性は本稿執筆時点では存在しない。

3. 不確実性のモデリング

本節では，不確実性のモデリング，すなわち不確実性の記述と，システム開発の成果物への紐付けについて論じる。

3.1 不確実性の分解

複合型の不確実性はより具体的な不確実性に分解されなければならない。この分解は再帰的に行われ得る。

一定の観点基準に従って分解されるべきである。以下に観点基準の例を挙げる。

一次不確実性と二次不確実性の分離: 意思決定に必要な情報に関する不確実性と意思決定の結果に関する不確実性とを分離して配置する。

プロダクトとプロセスの分離: システムそのものに関する不確実性とシステムの作り方に関する不確実性とを分離して配置する。

フェーズごとの分離: 開発フェーズにおける不確実性と運用フェーズにおける不確実性とを分離して配置する。また開発フェーズにおける不確実性については，要求，設計，実装フェーズにおける不確実性を分離して配置する。（これは構成関係に基づく分解の特殊系である。）

構成関係に基づく分解: モノやコトに関する不確実性の下位に，そのモノやコトの構成要素に関する不確実性を配置する。たとえば，システムの作り方に関する不確実性の下位にそのサブシステムの作り方に関する不確実性を配置したり，あるいは機械的機構，電気的機構（ハードウェア），ソフトウェアに関する不確実性を配置したりする。

汎化関係に基づく分解: モノやコトに関する不確実性の下位に，そのモノやコトを特化したモノやコトに関する不確実性を配置する。

3.2 不確実性の記述

不確実性は決定木として記述できる。図 1 に著者らが開発している農業用自律走行車の操舵機構の開発における不確実性を記述した決定木，すなわち不確実性モデルである。紙面の都合，図には一次不確実性のみ記載しているが，メカ実装容積，回路実装容積，電源容量，マイクロコントローラの使用可能ポート数等が上限に収まっているかといった二次不確実性も存在する。

この自律走行車は緯度，経度の列で与えられた経路を GPS で測位しながら走行する。操舵は左右の無限軌道に動力を伝えるクラッチを電動シリンダおよびコントロールワイヤでつないだり切ったりすることで実現されている。電動シリンダを制御する H 型モータ駆動回路の概略は図 2 に示す通りであるが，この回路の細部の実現には図中の吹出しに示すような不確実性があり，これら不確実性は図 1 の不確実性モデルに整理され記述されている。不確定性には角括弧が添えられその中に不確定性の名前が記されている。また，多肢選択型不確定性の解候補は決定木の葉となっている。

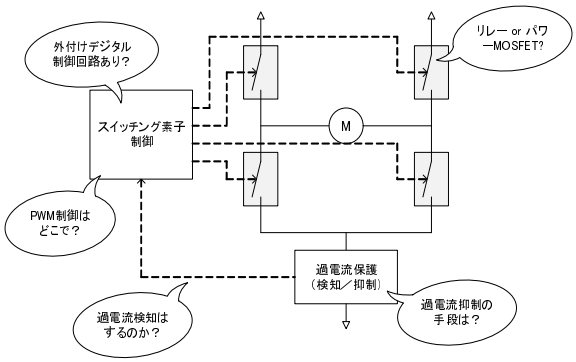

図 2 電動シリンダ駆動回路（H 型モータ駆動回路）

不確実性モデルでは，多肢選択型不確実性の下にその解候補，複合型不確実性の下により具体的な下位の不確実性が配置される。さらに決定木の枝をまたがる形で不確実性と不確実性の間，多肢選択型不確実性の解候補と不確実性，多肢選択型不確実性の解候補と他の多肢選択型不確実性の解候補の間に依存関係を示す枝が張られる。

- ある多肢選択型不確実性 M における解候補 m の選択によって，他の不確実性 C に係る意思決定が必要になる場合は，m から C への依存枝を設ける。
- あるパラメータ型不確実性 P の値 p を引数とする条件 $\mathrm{cond}(p)$ の真偽によって，他の不確実性 C に係る意思決定が必要になる場合は，P から C への依存枝を設け，$\mathrm{cond}(p)$ を併記する。
- ある多肢選択型不確実性 M における解候補 m の選択によって，他の多肢選択型不確実性 N における解候補 n の選択が必要になる場合は，m から n への依存枝を設ける。
- あるパラメータ型不確実性 P の値 p を引数とする条件 $\mathrm{cond}(p)$ の真偽によって，他の多肢選択型不確実性 N における解候補 n の選択が必要になる場合は，P から n への依存枝を設け，$\mathrm{cond}(p)$ を併記する。
- ある多肢選択型不確実性 M における解候補 m の選択と他の多肢選択型不確実性 N における解候補 n の選択とが一蓮托生的／相互排他的である場合は，m と n の間に双方向の枝を設け，*mutually inclusive* ／ *mutually exclusive* と併記する。
- 一次不確実性 C_1 における意思決定の結果が二次不確実性 C_2 の結果に影響する場合，C_1 から C_2 への依存枝を設ける。

4. 不確実性を取り扱うプロセス

本節では，システム開発における不確実性を取り扱うべ

ⓒ 2014 Information Processing Society of Japan

Vol.2014-SLDM-165 No.6
Vol.2014-EMB-32 No.6
2014/3/15

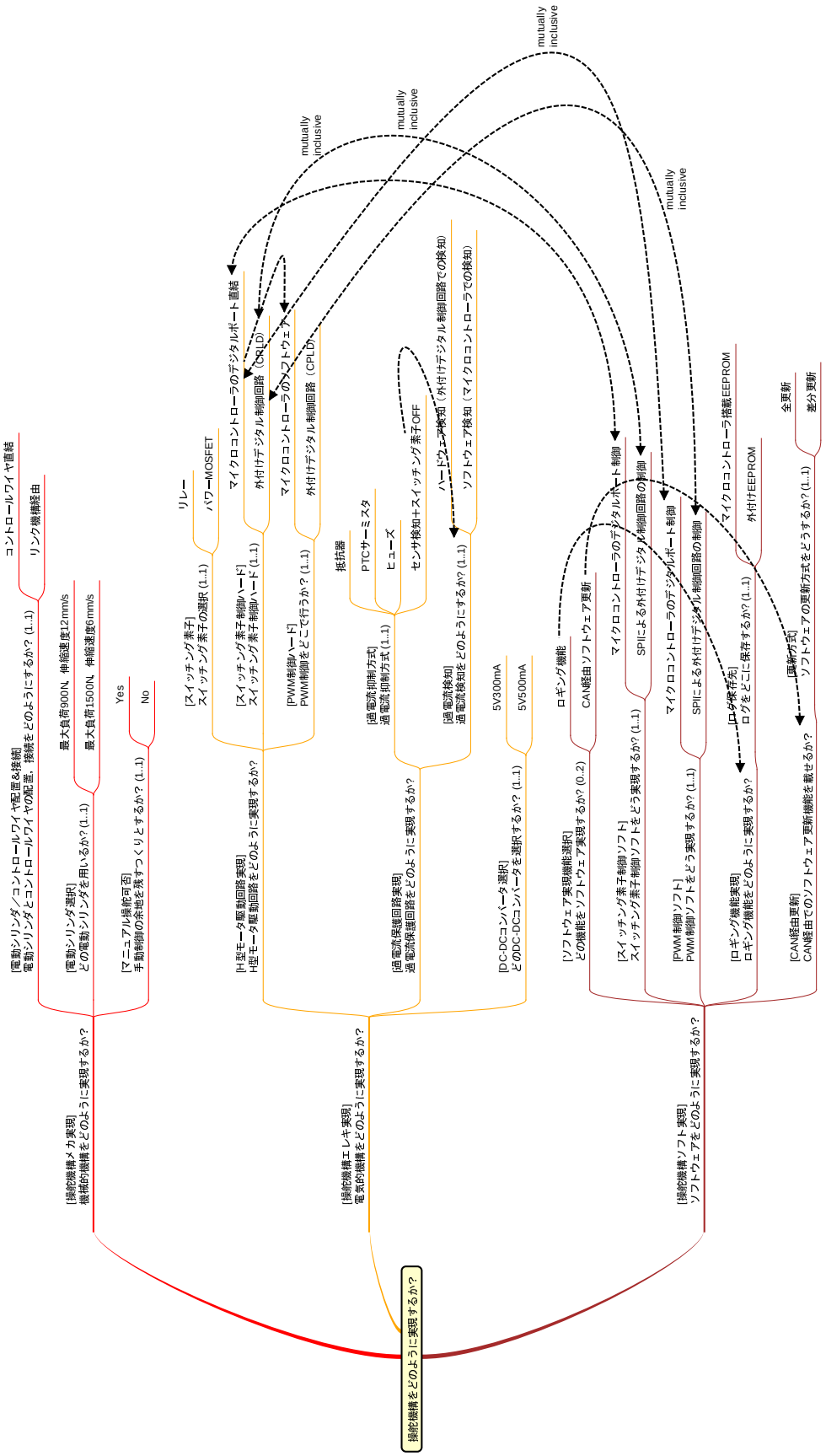

図 1 不確実性モデル

ⓒ 2014 Information Processing Society of Japan

く，要求に始まり，仕様，設計，実装，テストに至るプロダクトに関する開発成果物，ならびに開発から運用に至る諸工程を，不確実性から紐付けるプロセスについて述べる。提案するプロセスの概略は以下の通りである。

(1) 不確実性のモデリング
(2) 不確実性の影響評価
(3) 不確実性の解決策の検討
(4) 不確実性からのトレーサビリティ確保

4.1 不確実性のモデリングと影響評価

第 3 節で述べたとおり不確実性は決定木を用いてモデリングする。

一次不確実性については影響評価を実施し，表 1 に示すような表を作成する。この表は故障モード解析（FMEA: Failure Mode Effect Analysis）**?**で作成される表の書式を参考にして定めたものであり，故障モードのかわりに不確実性を，さらに故障モードの影響のかわりに各不確実性の解決に係る影響をまとめたものである。例では影響の記述を機械的機構，電気的機構，ソフトウェアにわけて記述しているが，分割記述に用いる観点はプロジェクトや対象システムの性格に合わせて定めてよい。

影響評価では一次不確実性における意思決定の結果が二次不確実性への影響も評価する。二次不確実性，多くの場合非機能要求に係る諸特性の値を見積もり，定量的に，それが不可能であれば定性的に記述する。自律走行車の操舵機構の例では，メカ実装容積，回路実装容積，電源容量，マイクロコントローラの使用アナログ／デジタルポート数，CAN や SPI の帯域などが相当する。これらの見積もりは不確実性の解決の基礎資料となる。

4.2 不確実性の解決とトレーサビリティ

一次不確実性に係る意思決定を下し，システム開発における不確実性を排除していく活動を不確実性の**解決**と呼ぶ。不確実性を認知した工程においてその不確実性を解決する**即時解決**と，不確実性を認知した工程より後の工程においてその不確実性を解決する**遅延解決**が考えられる。不確実性の後工程への積み残しは開発の規模と複雑さを膨らますため，即時解決が理想であるが，冒頭に述べたように即時解決できないケースが現実には多く生じる。

不確実性の解決手段を類型化したものを以下に示す。

理論的（演繹的）解決: 何かしらの数学的モデルに基づいて意思決定の結果を見積もり，不確実性の解決を目指す。機能，構造，振舞いを主とするさまざまな観点から開発対象システムの分析を行う分析フェーズの諸工程は，それ自体が要求に潜む不確実性を明らかにし，要求仕様書策定前にそれらの演繹的解決を促すものと言える。

実験的（帰納的）解決: 実験は開発中のシステムやその一部，あるいはそれらを模擬するモノを用いて意思決定の結果を見積もり，不確実性の解決を目指す。プロトタイピングやシミュレーションは，システム開発の振舞いをプロトタイプや計算機上で再現し，不確実性の発見，不確実性の解決に必要な情報の収集，不確実性の実験的解決を促すものと言える。

直感的解決: 理論的ならびに実験的な分析に拠らない，いわゆる勘と経験によって意思決定の結果を見積もり，不確実性の解決を目指す。品質機能展開（QFD: Quality Function Deployment）や階層分析法（AHP: Analytic Hierarchy Process）による意思決定は，意思決定の集約が理論的，実験的分析に拠らないものである以上はこの範疇となる。

包括的解決: 異なる意思決定に対してそれらの結果の共通部の最大化を図り，システムの追加，変更ではなく，コンフィギュレーションによる不確実性の吸収を図る。機械的機構や電気的機構の設計におけるマージンの確保，ソフトウェアにおけるパラメータ化，バッファリング，インターフェース共通化や汎化／特化，中間言語等による抽象化，ソフトウェアプロダクトラインで行われる変化点概念の導入はこの範疇となる。

事後的解決: 不確実性に対して上述の解決策を採ったものの予測を外した場合には事後的な解決，すなわち作り直しを図る。派生開発 [2] は事後的解決の一手段となる。

一次不確実性ならびにその解候補から，一次不確実性を解決した結果生じた成果物，ならびに成果物中の一次不確実性に係る部分へのトレーサビリティを確保する。ここで言う成果物には，要求，仕様，設計，実装，テストに至るシステムの開発プロセスを通して作成されるあらゆる抽象度の文書，ならびに開発や運用のプロセスに関する記述文書が含まれる。

5. 考察

以上，本稿では不確実性概念の定義と理論的整理，不確実性の記述，不確実性の解決手段の体系化について論じてきた。ソフトウェアプロダクトライン [3], **?**に明るい読者はおそらく，これらの内容がフィーチャ概念やフィーチャモデリング [4] と共通点を多くすることを指摘するであろう。実際，著者らは不確実性をフィーチャと擬制することで不確実性の理論的体系化を進めてきた。

しかしながら，著者らは不確実性はフィーチャとは（深い関係はあるものの）異なる概念とする立場を採る。不確実性がシステムの開発における意思決定に名前をつけた概念要素であるのに対し，フィーチャはシステムの機能や特性に名前をつけた概念要素である。フィーチャはシステム開発における意思決定の結果定まっていくものである。さらにシステム開発では，作るモノ（プロダクト）の在り方のみならず，作るスベ（プロセス）に関する意思決定も問われる。これらのことを考えれば，不確実性は明らかに

ⓒ 2014 Information Processing Society of Japan

表 1 不確実性影響評価表の例（一部）

不確実性	解候補	機械的機構への影響	電気的機構への影響	ソフトウェアへの影響
スイッチング素子制御ハード（多肢選択型，1..1）	マイクロコントローラのデジタルポート直結	なし	マイクロコントローラで必要となるデジタルポート数 +（4 * 電動シリンダ数）	デジタルポートを直接制御するコードの記述要
	外付けデジタル制御回路（CPLD）	なし	マイクロコントローラで必要となるデジタルポート数 +4（CE, SCLK, MISO, MOSI）; CPLD の論理回路設計要（SPI インターフェース）; CPLD で必要となる FF 数 +（8 * 電動シリンダ数）	CPLD 中に SPI インターフェースのためのコード要
PWM 制御ハード（多岐選択型，1..1）	マイクロコントローラのソフトウェア	なし	なし	PWM 制御のための周期タスクの実装要
	外付けデジタル制御回路（CPLD）	なし	CPLD の論理回路設計要（SPI インターフェース，タイマカウンタ，制御レジスタ）; PWM タイミング生成用のクロック信号入力要	CPLD 中に SPI インターフェースのためのコード要

フィーチャよりも高い抽象度の概念と言える。

不確実性とフィーチャは異なる概念であり，不確実性の解決の結果フィーチャが定まる，すなわち不確実性が主，フィーチャが従たるモデルとなることが，あえて新たに不確実性モデルを規定する理由である。

6. 関連研究

不確実性は，システム開発に限らない，一般的（あるいは社会的）なリスク解析の文脈で論じられている。Wynne は不確実性を，危害の内容と発生確率の明らかなリスク，発生確率まではわからない狭義の不確実性，危害があるのか否かさえわからない無知，その他非決定性，複雑性，不一致，曖昧性の 7 種類に分類している（訳語は文献 [5] に倣った）[6]。本稿のフレームワークで論じる不確実性は，これらのうちのリスクと狭義の不確実性である。

竹村らはやはり一般的なリスク解析の文脈で，意思決定が行われる環境を確実性下，リスク下，不確実性下（曖昧性下または無知下）に類型化したうえで既存の意思決定手法の説明を試みている [7]。本稿のフレームワークはシステム開発の文脈で構築されたものであり，不確実性の解決策の体系を与えている点で異なる。

一方，著者らの既発表論文 [1] では，システム開発の文脈で，不確実性を孕んだ状況での要求，運用，設計モデリング手法を提案している。同手法では，システムの開発に要する情報のうち不確定となっているものを「不確定要素」と命名，概念化するとともに，その影響を評価し，さらに不確実性を包括するような部分モデルを構築し不確定要素からのトレーサビリティを確保することで，不確実性に因む手戻りの手間を緩和している。この研究では不確実性をフィーチャとしてモデリングしていたが，本稿のフレームワークは不確実性をフィーチャとは異なるより上位の概念として扱うよう拡張されている。

7. まとめ

以上，本稿ではシステムの開発，運用における不確実性に関する諸概念の整理し，不確実性を取り扱うためのフレームワークを定義した。不確実性の定義ならびに分類（外部不確実性，内部不確実性；一次不確実性，二次不確実性；多肢選択型，パラメータ型，複合型）を与えた。また，不確実性のモデリング，影響評価，解決策の検討，不確実性から開発資産へのトレーサビリティ確保を行うプロセスを示した。その中で不確実性を決定木として体系化するモデリングの方法論，影響評価に用いる表の書式，解決策の類型について論じた。

参考文献

[1] 陳 辰ほか,「不確定要素を含む要求・運用・設計モデリング手法」, 組込みシステムシンポジウム 2013 論文集, pp.75–80, 2013 年 10 月.
[2] 清水 吉男,『派生開発を成功させるプロセス改善の技術と極意』, 技術評論社, 2007 年 11 月.
[3] K. Pohl, *et al.*, *Software Product Line Engineering: Foundations, Principles and Techniques*, Springer, 2005.
[4] K. Kang, *et al.*, “Feature-Oriented Domain Analysis (FODA): Feasibility Study,” Technical Report, Software Engineering Institute, Carnegie Mellon University, CMU/SEI-90-TR-222, Nov. 1990.
[5] 平川 秀幸,「リスクの政治学: 遺伝子組換え作物論争のフレーミング分析」(in 小林 傳司（編）,『公共のための科学技術』, pp.109–138）, 玉川大学出版部, 2002 年.
[6] B. Wynne, “Managing and Communicating Scientific Uncertainty in Public Policy,” Background Paper, *Harvard University Conf. on Biotechnology and Global Governance: Crisis and Opportunity*, Apr. 2001.
[7] 竹村 和久ほか,「不確実性の分類とリスク評価: 理論枠組の提案」, 社会技術研究論文集, Vol.2, pp.12–20, 2004 年.

ⓒ 2014 Information Processing Society of Japan