

システム構築の上流工程における要求定義の効率化について

石野 正彦[†] 上田 翔太[†]

About promotion of efficiency of the requirements definition in the upper reaches of the systems construction

Masahiko Ishino and Shota Ueda

Abstract

In the present system development, the requirements definition is very important. Many causes are it in a cause for requirements definition by the project break of the system development. In addition, improvement of precision and the speed of the requirements definition is promotion of efficiency of the overall work and reduction of the cost. This report examined improvement of the promotion of efficiency about the requirements definition.

Keyword: Information system construction, Requirements definition, Unified Modeling Language

1. はじめに

システム構築において、仕様の要求定義は非常に重要である。構築上でのプロジェクト崩れの多くが要求定義フェーズに原因がある場合が多い。また、要求定義での仕様の精度向上は、全体工程の効率化やコストの削減に繋がる。本研究は要求工学の見地から効率化の改善を対象とした。

2. 要求工学の領域

要求工学は、システム構築におけるユーザの要求を、科学的に定義するための方法や考え方であり、“ユーザは何が目的か”、“どんな機能が必要か”などの要求をまとめ、システムに正しく反映させる手法全般を対象にする。主にシステム構築の最上流の要求定義の領域を対象とし、要求の変更を管理する手法や、要求が適切に反映されているか検証する手法もある。要求工学は、米国で1970年代後半から研究が進められてきた。日本では2004年に本格的な取り組みが始まり、情報処理推進機構（IPA）ソフトウェア・エンジニアリング・センター（SEC）が研究部会を立ち上げ、日本で初めて要求工学国際会議が開催された。要求工学では一般的に、要求に関する工学的な手法を①要求の獲得、②定義、③検証、④管理の4領域に分類する。①は主にインタビューなどでユーザから目的や必要な機能を引き出すことであり、「ソフト・システムズ方法論（SSM）」や「ゴール指向分析」などの手法がある。②はUML（統一モデリング言語）などを使って要求を可視化し、要求仕様書を作成する手法である。③はレビューなどでソフトが要求を正しく反映しているかどうかを確認する手法がある。④は要求変更時に上流からの一貫性を保ち、要求定義内容を再利用する手法である[1]。

[†] 経営情報学科

3. システム構築上の課題

システム構築では、様々なユーザ要求仕様が重なり、度重なる修正が発生する。その原因として、上流工程の要求定義フェーズでの仕様の確定が不十分であることが多い。この要求定義フェーズでの仕様の欠落は、ユーザの要求を正しく定義できていない。それはユーザの言い忘れや、意識していなかった部分で漏れの要素[2]が発生するからである。また、定性的で相対的な情報やイメージのままの表現が多く、あいまいさが残る要素がある。この二つの要素が、必要以上の仕様の修正やロスコストを生み出す。漏れやあいまいさが、図1のような弊害を及ぼす。

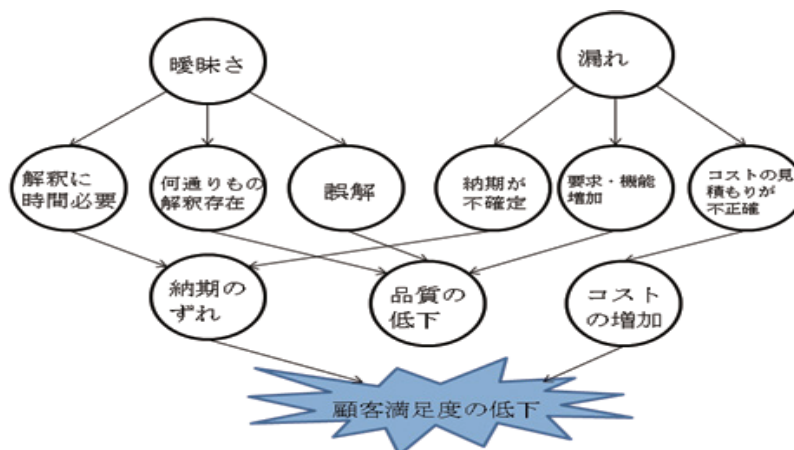


図1 仕様の漏れとあいまいさによる弊害

4. システム構築におけるプロジェクト崩れ

図2に示されるように、プロジェクトの遅れの主な原因として下記が挙げられる[3]。

- 要求定義不十分[36.7%]
- テストが不十分 [41.7%]
- システムの開発の質が悪い[31.7%]
- システムの設計が不正確[31.7%]

国内のシステム構築プロジェクトの成功率は 26.7%である。すなわち、約7割以上のプロジェクトが失敗している。その原因として、約4割が上流工程で要求定義の不十分が原因である。

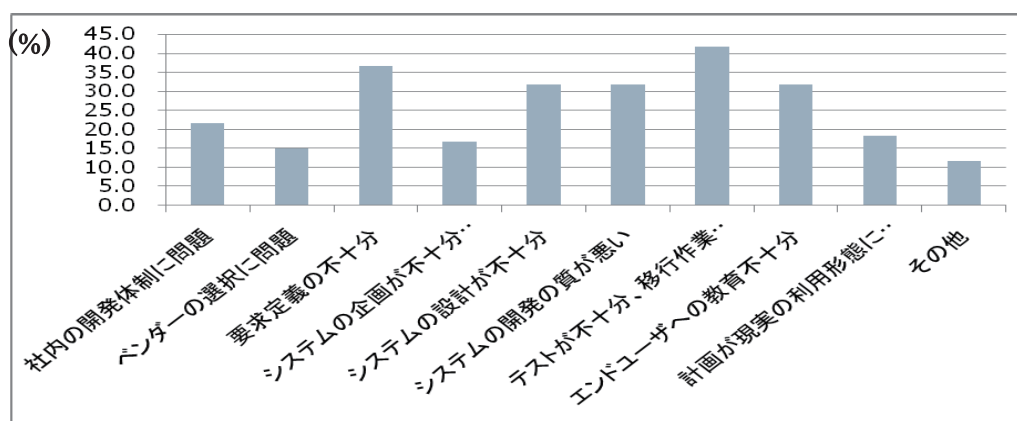


図2 プロジェクト崩れの課題項目

5. 要求分析

システム要求分析は、抽出されたか、要求の正しさ、望まれているかという成分で構成される。システム要求分析によって抽出される要求成分を表1と図3に示す。Xの部分の望ましい要求が正しく抽出されていることが理想であるが、実際には、抽出された要求の中には、Wの部分の望ましい要求が抽出されているが正しくないの修正対象、Yの部分の抽出された正しい望ましくない要求、Zの部分の誤った不要な要求で修正の対象が含まれる。

表1 システム要求分析によって抽出される要求

| 記号 | 望まし しさ | 抽出 | 正しさ | 説明 |
|----|-----------|----|-----|------------------------------------|
| U | x | x | — | 望ましくない要求が抽出できていないので、正しいのか確認できない |
| V | 0 | x | — | 要求の漏れである。抽出できていないので、正しいかどうか確認できない |
| W | 0 | 0 | x | 望ましい要求が抽出されているが正しくないの、修正対象である。 |
| X | 0 | 0 | 0 | 望ましい要求が正しく抽出されている |
| Y | x | 0 | 0 | 正しいが不要な要求である。要求漏れではないが、要求変更の対象となる。 |
| Z | x | 0 | x | 誤った不要な要求である。要求漏れではないが、修正対象である。 |

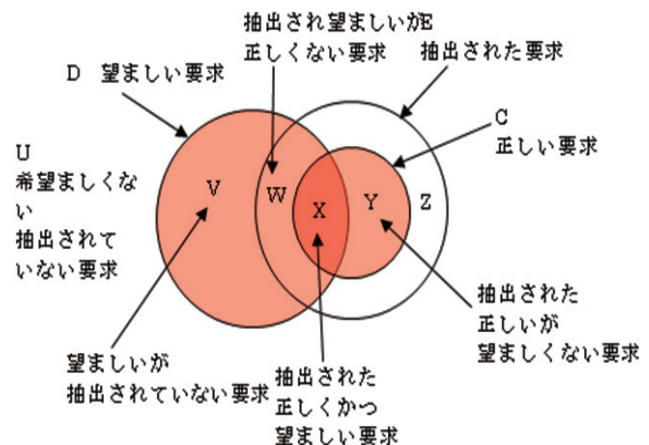


図3 システム要求分析によって抽出される要求

6. 現状の課題提起

① 開発工程と上流工程

上流工程で図4の要件と計画部分での要求定義の改善を図った[4],[5]。

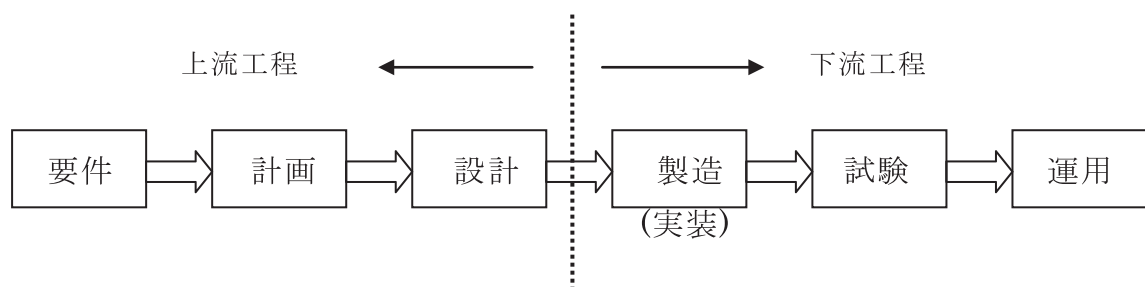


図4 情報システム構築の上流工程

② 要求定義の前提条件

要求定義では、図5のように情報システムの「あるべき姿」に近づける努力をする。さらに、図6の示す、現状の業務の内容(As Isモデル)の改善すべき点(To Beモデル)へテキスト化する。また、改善すべき場所、改善された場所をピックアップして具現化する[6],[7],[8]。

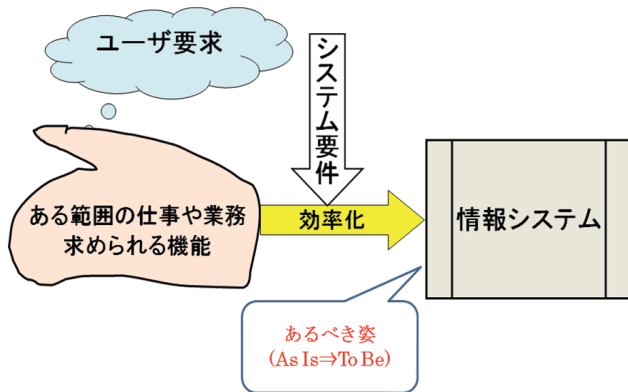


図5 現状業務からあるべき姿への到達

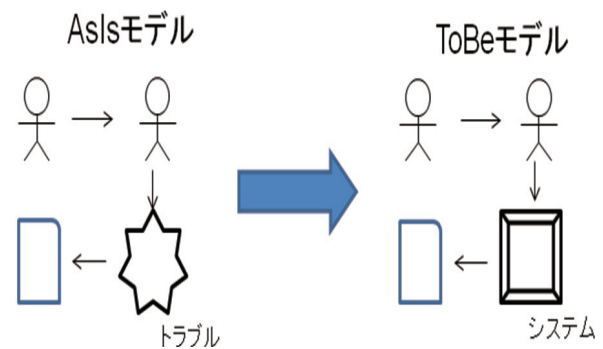


図6 As Is モデルから To Be モデルへ

③ユーザ側の問題

プロジェクト崩れが起こる理由として、以下の項目が挙げられる。

- ・業務上何を欲しているかわからない
- ・ソフトウェアの知識がないため、アルゴリズムに近づこうとしない。

④システム構築担当者の問題

- ・ユーザとの意思の疎通がうまくできていない
- ・技術者が要求分析を行うための業務知識に欠け、正確な要求分析ができていない。

7. 要求定義の効率化策

要求定義の効率化策として①定義の省力化と②視覚的にわかりやすいツールや手法が挙げられる。

① 要求定義の省力化

要求定義の省力化を行う方法[5]として、オブジェクトやノードなどの一定のデータを入力し、関係図を作成する。そうすることによってユーザ側にも視覚的に分かりやすく、仕様のあいまいさや齟齬の削減に繋がる。ユーザと開発側の齟齬が減少できれば、要求分析の精度が向上する。また、統一様式や方法[9],[10]によって、分析するので工数のロスコストは少なくなる。一般的に MIND-SA などの上流 CASE ツールがある。

② 視覚的にわかりやすいツールや手法

視覚的に分かりやすくすることにより、IT 技術者以外の人やユーザに分かりやすくできる。比較的簡単に確認できるようになれば、漏れやあいまいさによる齟齬を削減できる。しかし、簡略的にすることにより、分かりやすさの向上、工数のロスコストを削減はできるが、詳細な仕様についても簡略化しなければならない。一方で、詳細な仕様書の作成は工数コストの増加や漏れ等に気付かなくなる恐れがあるというトレードオフの関係にあるため、簡易的な確認のための仕様書と適確な仕様書の双方の機能を備えることが必要となる。現状の方法として統一モデリング言語 UML(Unified Modeling Language)がある。多種のツールによって、システムの機能を視覚的に分かり易くできる[11],[12]。

③ UML ツールの活用

現状では、表 2 のような UML のツールを現場で活用することが進められている。(図 7,8)

表 2 UML ツールの種類と特徴

| 分 析 | ツール | 特 徴 |
|---------|--|--------------------|
| 機能モデル | ユースケース図,DFD(データフローダイヤグラム) | 機能把握 |
| 動的論理モデル | アクティビティ図,ステートマシン図,シーケンス図,コミュニケーション図,タイミング図 | オブジェクト指向 |
| 静的論理モデル | クラス図,オブジェクト図,パッケージ図,コンポジットストラクチャ図,コンポーネント図 | データ構造 コンポーネント構造 |
| 実装モデル | システム実装構成図 | モジュール構成 |

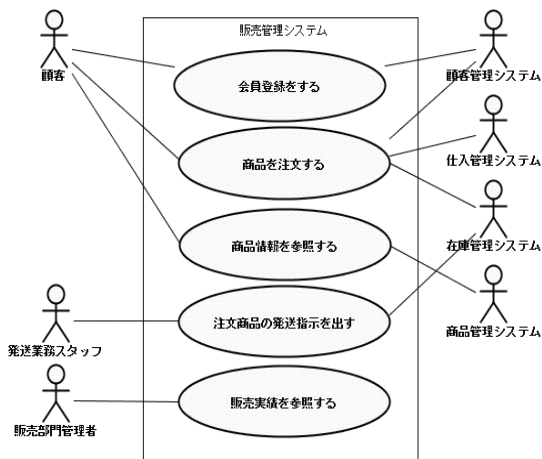


図 7 ユースケース図(例)

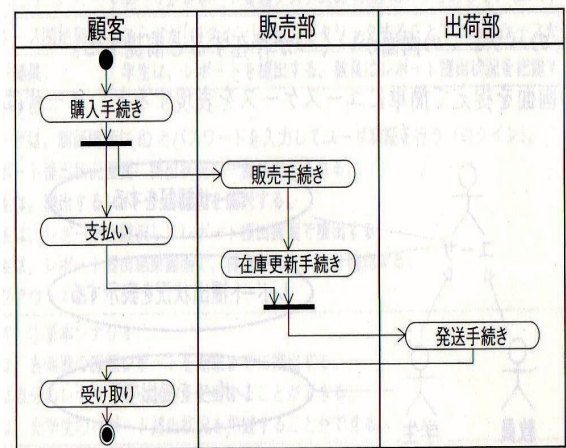


図 8 アクティビティ図(例)

8. 効率化の改善策

効率化方法として、ユーザ側の視点に立った手法が好ましいと考える。重視する点として、①1日のデータの流れ図、②月次のデータの流れ図、③ユーザの入力画面の3次元(3D)表記である。しかし、記述が多い仕様書は理解が難しく、細心の注意が必要となる。①と②で年間のシステムの処理機能を詳細に把握できるようにする。①と③でユーザ側に立った説明ができ、保守的な役割も持てる。②と③で季節変動するようなシステムの説明ができる。また、仕様の変更前、変更後の仕様書履歴を残すことで、仕様変更工数をビジュアル化できる[5]。同時に表記することで変更前と変更後の仕様を色でクローズアップして表記し、より分かり易くする。また、ユースケース図のような簡易的で大筋の流れを表現し、その部分をクリックすることにより詳細なデータや時系列などへ展開するといったツールを提供すれば、ユーザ側、開発側の仕様の齟齬を軽減することができる。要求分析としては完全性が保持できない場合には、確認用ツールとして使用できる。図9の三次元(3D)表記は3つの要素から確認できる。①関連性の可視化、②目標への到達度、③業務改善を図10のようなグラフのイメージで表せる。

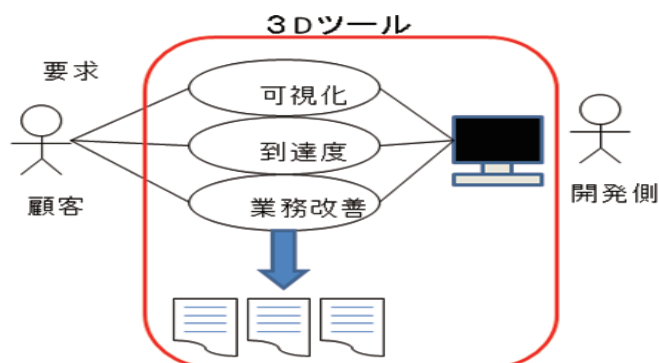


図9 三次元(3D)ツール

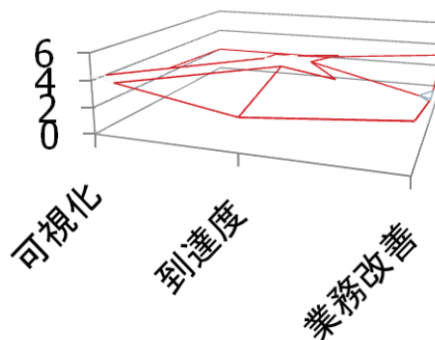


図10 評価グラフィメージ

要求定義ツールによって、工数コストの削減を狙い、処理機能や、論理的な矛盾などをチェックすることを実現する。ユーザ要求情報を入力することにより、関係図の作成やシステム提案をする。開発側とユーザ側の意思の疎通が改善される。視覚的に打ち合わせし易くなれば、ミスがあったとしても初期工程で認識し易い。システム使用の詳細ではなく、ステークホルダの関係をビジュアル的に表現する。レビューの流れ(図11)とステークホルダの関係(図12)も含まれる。

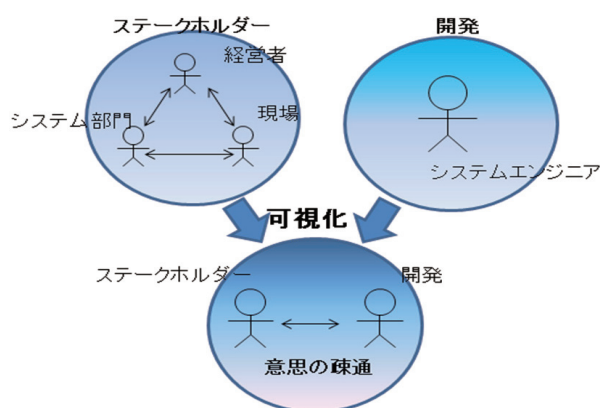


図11 ステークホルダの関係

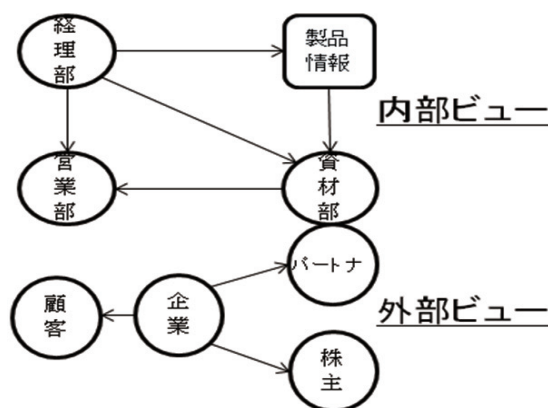


図12 レビューの流れ

図13の検索ツールの条件入力にシステムの概要、各関係図、可視化、ステークホルダ、要求定義対象の理解度、既存のシステムの有無、業務の实地調査の許可などがある。条件を入力し、手法を検索する。そして、テンプレートを表示し、手法の説明や記述方法などを結果出力する。

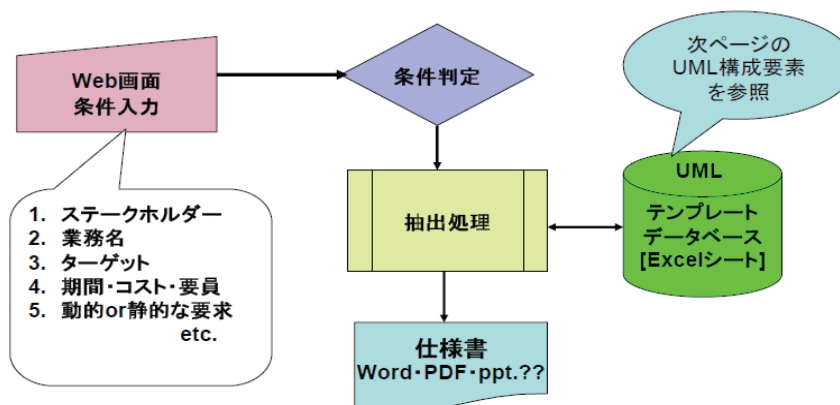


図13 要求定義手法検索ツールのしくみ

表 3 要求定義手法検索ツール(試作中)

| No. | UML | 概要 | 要求 | 分析 | 設計 | URL |
|-----|-----------|---------------------|----|----|----|---------------------------|
| 0 | ユースケース図 | システムの範囲と機能などの定義 | 3 | 0 | 3 | ユースケース図 |
| 1 | アクティビティ図 | 処理の流れを記述 | 2 | 0 | 1 | アクティビティ図 |
| 2 | クラス図 | クラスをその関係を表現 | 0 | 3 | 3 | クラス図 |
| 3 | シーケンス図 | オブジェクトの連携を時間軸に沿って表現 | 0 | 3 | 2 | シーケンス図 |
| 4 | コラボレーション図 | オブジェクトの相互作用を表現 | 0 | 3 | 2 | コラボレーション図 |
| 5 | ステートチャート図 | オブジェクトの状態遷移を表現 | 2 | 0 | 1 | ステートチャート図 |
| 6 | 配置図 | ハードウェアなどの構成を表現 | 0 | 0 | 3 | 配置図 |
| 7 | コンポーネント図 | プログラムのモジュールの構成を表現 | 0 | 0 | 3 | コンポーネント図 |
| 8 | オブジェクト図 | オブジェクトの構造を表現 | 0 | 0 | 1 | オブジェクト図 |
| 9 | パッケージ図 | システムの構造を表現 | 0 | 2 | 2 | パッケージ図 |

● 到達度 (バロメータ表記)

三次元(3D)ツールでは、要求定義の到達度を提供する。評価項目を用意してユーザがチェックし、集計処理をして到達度を推定する。ユーザと開発者が合意した評価基準を設定し、簡単なプロトタイプを作成し、要求定義での仕様の問題点や矛盾する部分の抽出と到達度を表記する。

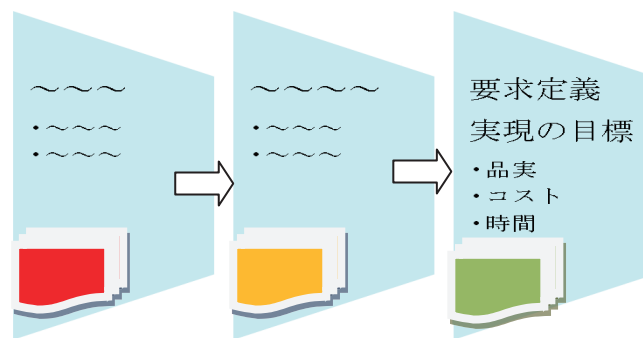


図 14 到達度(バロメータ)表記

● 実現機能 (試作計画)

① 3D ツール

- ・ 関連性の可視化
- ・ 目標への到達度
- ・ 業務改善

② 検索ツール

- ・ プロトタイプの手法選択ツールの作成
- ・ テンプレート作成

③ 重視する点

- ・ 1日のデータの流れ図→ルーチンワークを正確に記述する。
- ・ 月次のデータの流れ図→月次の決算用に締日処理をする。
- ・ ユーザの入力画面→エンドユーザ用に操作性を考える。

10. 今後の研究計画

本研究では、システム構築において、要求定義の重要性を提起し、仕様の精度向上を目指している。後工程への影響度が高く、要求定義の効率化はシステム構築の全体工程の品質向上や構築コストの削減に繋がるものと確信する。また、大学院生の研究テーマとして、実際にどのような手法で解決していくかなどの具体的な手法やツールを準備している。3D ツールと仕様の可視化を進めていく中で、システム構築の現場の実務関係者や学生向けにアンケート調査を実施することを計画している。今後、システム構築における要求定義の効率化について、更なる実用化研究を進めていきたいと考える。

参考文献

- [1] 日経 BP 社：日経コンピュータ,情報システムハンドブック,日経 BP 社,2010.
- [2] 佐川博樹：要求定義の基本と仕組み,秀和システム,2010.
- [3] (社)情報サービス産業協会：情報サービス産業白書,2010.
- [4] 高橋真吾,衣川功一,野中誠：情報システム開発入門, 共立出版,2008.
- [5] 松永俊雄,中村太一,亀田弘之：コンピュータシステム開発入門,オーム社,2008.
- [6] システム開発ジャーナル編集部：システム開発上流工程,毎日コミュニケーションズ,2010.
- [7] 山岸耕二, 安井昌男, 他：要求開発価値～ある要求を導き出すプロセスとモデリング, 日経 BP 社, 2006.
- [8] 神崎善司：要件定義マニュアル,秀和システム,2011.
- [9] 山本修一郎：ゴール思考によるシステム要求管理技法,ソフト・リサーチ・センター,2007.
- [10] 河村美嗣：UML を入力とするソースコード自動生成ツールの開発, 情報処理学会第 72 回全国大会, 2010.
- [11] 清水吉男：要求を仕様化する技術・表現する技術,2010.
- [12] 情報処理推進機構 ソフトウェアエンジニアリングセンター：IT プロジェクトの「見える化」上流工程編, 日経 BP 社,2007.

(平成 23 年 3 月 31 日受理)