



UI (ユーザインタフェース) 設計とシステム設計を結ぶ モデリング技術の可能性について

尾形 慎哉*¹ 鱗原 晴彦*¹ 青島 寛太*¹

Possibility of modeling technology which connects user interface design and system design.

OGATA Shin-ya*¹, UROKOHARA Haruhiko*¹ and AOSHIMA Kanta*¹

Abstract - In software development, "Makeup Time" which returns from a detailed design to a basic design may occur. Developers take in modeling technology, such as UML, and are furthering development efficiently. Since UML inquires focusing on a function, it can advance a system design efficiently. The defect of UI design is one of the causes of "Makeup Time". Then, an U'eyes flow is proposed as the technique of UI design.

Keywords - User Interface, System Development, Modeling, Usability, U'eyes flow

1. はじめに

ソフトウェア開発において、“いかに手戻りをなくすか”ということは、プロジェクトを成功に導くための重要なリスク管理のひとつである。手戻りとは「もう済んだ」と思っていることをまたやらなければならないことで、全開発費の30～50%を消費すると言われている^[1]。また、プロジェクトが進んでから分かった欠陥を修正する費用は、早期に修正する費用よりもはるかに多額となる^[2]。

そのため、想定外の大幅な手戻りを未然に防ぐには、開発プロセスのより早い段階で致命的な“欠陥”、つまり“ユーザのシステムに対する機能要求（システムの振る舞い）と機能外要求^[3]（使いやすさ、信頼性、性能、サポートのしやすさ）への期待に対する不備”を起こさないことがポイントとなる。

現在のソフトウェア開発では、UML (Unified Modeling Language) などのモデリング技術を活かして、効率的に開発を進める環境が整備されつつある。しかし、これらのモデリング技術は機能要求を対象とした検討・設計を行うことに重きが置かれ、製品品質の要素として「使いやすさ」の重要性が著しく高まっているにも関わらず、ユーザインタフェース (UI) 設計についての語彙が全く欠落している。また、そもそもUIの標準化はソフトウェア開発手法としてもプロセスとしてもまだ確立されたものがない^[4]。

これまで弊社では、UI設計を行うベースとして操作フロー図^[5-7]を活用してきた。操作フロー図とは、ユー

ザがシステムを操作する過程を時間軸で管理、情報整理を行うモデリング技術のことで、使いやすいUIを構築するための基本的な考え方を整理するためには非常に有効なツールであったと実感している。

本論では、ソフトウェア開発において操作フロー図の考え方を踏襲し、機能（システム設計に関わる検討）と使いやすさ（UI設計に関わる検討）を結ぶモデリング技術として進化させたU'eyesフローの可能性について考察する。

2. ソフトウェア開発の現状

2.1 開発プロセスモデル

現在までに確立している開発プロセスモデルとしては、ウォーターフォールモデル、スパイラルモデル、反復型プロセスなどがある。顧客のニーズに効率よく対応するため、開発プロセスモデルも進化しているのだが、現在でも伝統的なウォーターフォールモデルを採用しているところは多い（図1）。

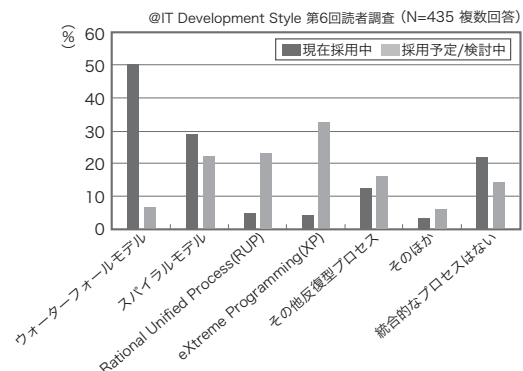


図1 開発プロセスの採用状況

Fig.1 Adoption situation of a development process.

*1: 株式会社ユー・アイズ・ノーバス

*1: U'eyes novas Inc.

2.2 開発手法

開発プロセスの中で採用している各手法の利用状況は図2の通りである。UMLをはじめとしたモデリング技術を利用して、ソフトウェア開発を行っている。

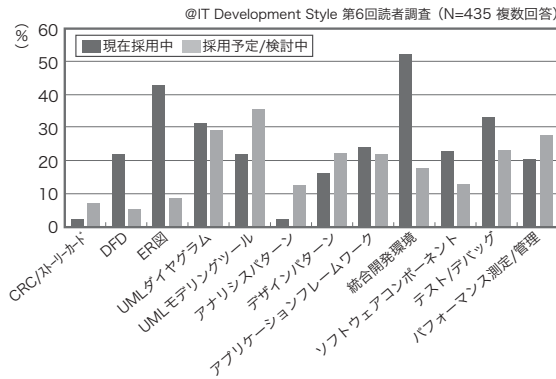


図2 ソフトウェア開発手法/ツールの利用状況
Fig.2 Software development technique/
The use situation of a tool.

3. 手戻りの要因

3.1 UI設計の不備

モデリング技術を利用して開発を進めることで、開発チーム内のコミュニケーションロスを低減することができる。また、UMLのように標準化された表記法であれば、どのような開発でも内容を把握することが容易になる。

表1はUMLを利用した開発プロセス⁸⁾である。UMLでは、STEP2でユースケース図などを用いて、ユーザの視点で要求について明確化しようとしている。STEP3では、ユーザの要求をシステム設計に落とし込む準備として、クラス図などで視覚化しようとしている。しかし、この時点でユーザの要求は機能としては反映することはできるものの、情報の主従関係や頻度・緊急度といったUI設計要件（使いやすさに対する要求）の多くが欠落してしまう。

表1 開発フェイズと利用されるUMLダイアグラム
Table 1 Development phase and
the use situation of a UML diagram.

開発フェイズ	代表的なUMLダイアグラム
STEP1 業務分析	アクティビティ図、など
STEP2 要求分析	ユースケース図
STEP3 システム分析	クラス図、オブジェクト図
STEP4 システム設計	ステートチャート図、シーケンス図
STEP5 実装・テスト	なし

現状では基本的にUI設計は、機能外要求として別途検討することになっているが、時間とコストに追われた実際の開発現場では、UI設計の検討がおろそかになってしまうことが多い。弊社のこれまでの経験では、UI設計とシステム設計のプロセスは、次のように行われている。

パターンA：「UI設計」から「システム設計」へ

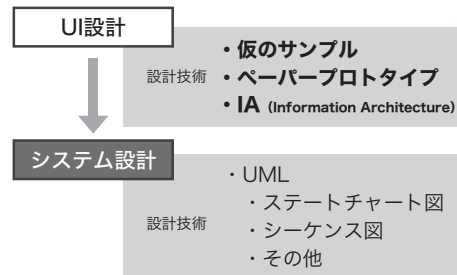


図3 UIとシステムの設計手法（パターンA）
Fig.3 The design technique of UI and a system (A).

・仮のサンプル

システム設計の前に、アウトプットイメージとして仮に作成したUIサンプル。使いやすさを十分考慮したとしても、システムの操作に関する全体像が見えなく、場当たり的なサンプルとなることが多い。時間やコストの制約などから、結局システム設計後にできる画面は、仮のサンプルに近いものになってしまう。このタイプは、「作っては直し」の連続で手戻りが非常に大きくなる可能性が高い。

・ペーパープロトタイプ

低コストで何度もユーザからのフィードバックをUIに反映することができる。部分的には非常に有効な手法ではある。さらに、事前にシステムの全体像を把握できればより効率が上がる。

・IA (Information Architecture)

コンテンツの意味やユーザのニーズにより情報をカテゴライズし、構造を設計することが可能となり有効な手法である。しかし、時間軸を含めてのインタラクションの管理をすることは難しい。

パターンB：「UI設計」から「システム設計」へ

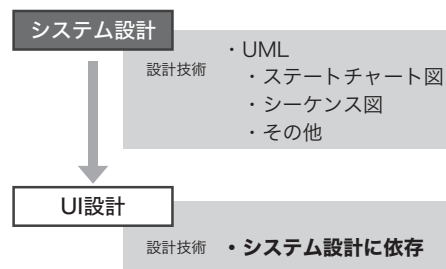


図4 UIとシステムの設計手法（パターンB）
Fig.4 The design technique of UI and a system (B).

・システム設計に依存

システム設計の結果からUIが必然的に決められる。その結果、ユーザの使いやすさに対する要求が反映されにくいいため、UIに不備が出る可能性が高い。

4. U'eyesフローの役割

4.1 U'eyesフローの目的

前述のように、操作性に対するユーザの視点（User's eyes）を盛り込んだUI設計が十分されないまま、システム設計に移行してしまう。結果的に、UI設計やシステム設計の想定外の大幅な手戻りが発生してしまう。この問題を解決するモデリング技術が、U'eyesフローの目的である。

4.2 U'eyesフローの特徴

図5のようにUI設計やシステム設計を行う前に、基本操作設計を行うことが、ユーザの要求をUI設計やシステム設計へ反映することにつながる。

基本操作設計（U'eyesフロー）

- ・システムに関わる操作の全体像を明らかにする

ユーザがシステムを操作する時のタスクの開始から終了までを記述することで、操作の全体像を把握することができる。そうすると、操作全体を通して整合性や一貫性を管理できるので、基本的な操作性を確保することができる。

整理された情報に従って、画面構成を検討することで、場当たりの画面サンプルを作成することは少なくなる。また、ペーパープロトタイプを作成するにしても、基本的な考え方は整理されているので、修正コストが通常よりもさらに少なく、一定の品質を保ったUIが設計できるものと思われる。

- ・時間軸を設けた情報整理をする

タスクの開始から終了までを時間軸に沿って記述することで、静的なインタラクションの管理を行うことができる。こうすることで、画面遷移に必要な要素の抽出をできる限り盛り込むことが可能なため、動的なプロトタイプを作成する前に、インタラクションの検討ができる。

- ・情報の主従関係を明確にする

ユーザに対する情報の重要度を整理することで、煩雑な情報提示を防ぐことができる。メインタスクとサブタスクなどを整理することが、基本的な操作性を確保する第一歩で、U'eyesフローでは、それを検討することができる。

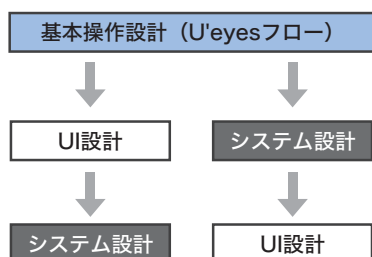


図5 U'eyesフローの位置づけ
Fig.5 Positioning of a U'eyes flow.

5. U'eyesフローを導入した開発事例

5.1 開発対象システム

弊社が携わったシステム開発支援業務において、U'eyesフローを用いたUI設計、システム設計のプロセスについての事例を報告する。

開発するシステムは、タッチパネル式の業務用POSで、異なる2つのシステムを統合した新しい概念のものである。

5.2 開発プロセス

開発プロセスは、次のようになっている（図6）。

STEP1 フィールド調査

はじめに、業務を取り巻く環境の実態を把握するためのフィールド調査を行った。

STEP2 要件整理

次に、ユーザの使いやすさに対する要求やシステムの要求・制約などを要件として文書でまとめた。

STEP3 概念的なU'eyesフロー

文書化されている要件を、ユーザの視点でダイアグラムを作成。操作性に関わる情報を可視化した。

STEP4 基本画面構成

質の同じ情報ごとにグルーピングしたものを、画面の基本的な画面構成に落とし込んだ。

STEP5 詳細なU'eyesフロー

操作の流れに合わせて必要な機能を組み込んで整理したダイアグラムを作成。インタラクション検討のベースとしても活用した。

STEP6 UI設計（画面仕様）、システム設計

詳細なU'eyesフローをベースに、各画面のUI設計（ペーパープロトタイプでの検討）とシステム設計（スタートチャート図やシーケンス図などでの設計）を並行して作業を進めた。

5.3 開発工数

従来であれば、STEP3～5までの検討がなく、UI設計・システム設計を行うことになる。しかし、今回はU'eyesフローを用いたことで、UI設計・システム設計の前にユーザの視点を組み込んだ基本操作の検討を行うことができた。また、詳細なU'eyesフローができた時点で、画面仕様が固まっていない段階でもシステム設計を進行することもできるメリットが確認できた。

システム設計担当者の話によると、基本設計の段階で、U'eyesフローによる検討を行っただけ工数はかかったが、実装するための設計に関して感覚的には従来の1/2～1/3の時間でできた、というコメントが得られた。

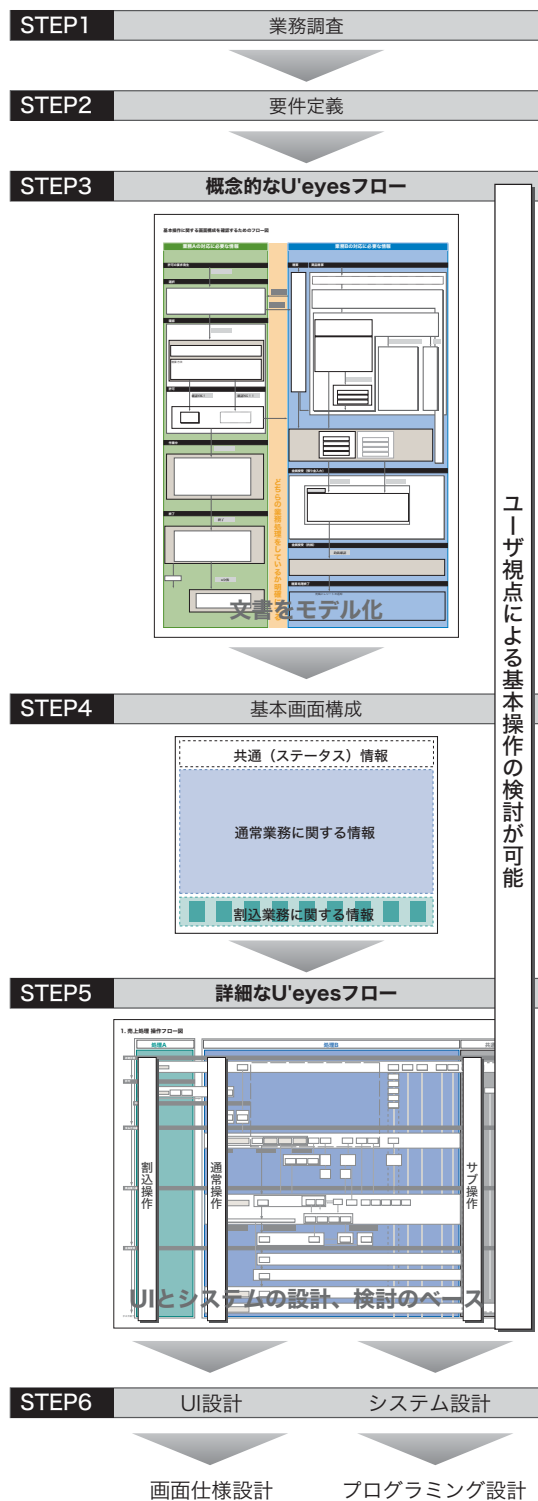


図6 U'eyesフローを用いた開発プロセス
Fig.6 The development process using the U'eyes flow.

6. まとめ

6.1 U'eyesフローの可能性

これまで、UI設計とシステム設計をつなぐモデリング技術としてのU'eyesフローの可能性について考察してきた。その結果、以下のような役割を担う可能性がある。

- ・要件定義の内容をユーザの視点を欠落させずに、UI設計やシステム設計につなげることができる。
- ・UI設計の基本となる考え方を整理することができる。
- ・UI設計とシステム設計が同時に進めることができ、時間を有効に使うことができる。

6.2 今後の課題

今のところU'eyesフローの作成は、基本的な考え方があるものの、作成者のスキル（ユーザビリティ概念と開発経験）に依存している。今後、標準化などについても考えて行く必要がある。

また、U'eyesフローのみではなく、IA（Information Architecture）など情報デザインの考え方なども含めて活用することで、よりユーザ視点での設計が可能になるものと思われる。

謝辞

本論の作成にあたり、開発の現状について貴重なご意見を頂戴いたしました、コマタ株式会社様へ深く感謝いたします。

参考文献

- [1] Boehm, Barry W., and Philip N. Papaccio.: Understanding and Controlling Software Costs; IEEE Transactions on Software Engineering, 14(10), pp.1462-1476 (1988).
- [2] Grady, Robert B.: An Economic Release Decision Model: Insights into Software Project Management; In Proceedings of the Applications of Software Measurement Conference, pp.227-239, (1999).
- [3] Grady, Robert B.: Practical Software Metrics for Project Management and Process Improvement; Englewood Cliffs, NJ; Prentice-Hall, (1992).
- [4] ラリー・コンスタンチン：ソフトウェア開発のカオス, 構造計画研究所, p.228-229, (2003).
- [5] 鱗原：操作フロー図によるタスク整理と仕様整理; Human Interface News and Report 計測自動制御学会ヒューマンインタフェース部会, Vol.11, pp.75-78, (1996).
- [6] 吉橋, 鱗原: インタフェース・デザインにおける道具とプロセスの研究 「操作フロー図」のはたらきと特性について; 日本デザイン学会デザイン学研究, pp.308-309, (1998).
- [7] Urokohara,H.: Method to build good usability Task analysis and user interface design using operation flowcharts; Proceedings of HCI International '99, Vol.1, pp.928-932, (1999).
- [8] 萩本: 初歩のUMLモデリング; 技術論評社, p.140-141, (2004).