

フレキシブルな UI 開発方法論とその支援環境の構成

徳田佳一[†] 李 殷碩^{††} 白鳥 則郎[†]

[†] 東北大学電気通信研究所 / 情報科学研究科

^{††} 韓国成均館大学工科大学情報工学科

近年の多様化するユーザインタフェース (UI) への要求に対処するために、UI 開発の知識や経験があまりない一般ユーザでも、容易に UI の設計、修正、拡張ができる UI 開発支援環境への要求が高まっている。本研究では、エンドユーザによる UI 開発を支援するために、i) ユーザの所有知識あるいはメンタルモデルに近い形でのユーザフレンドリな要求具体化、ii) 設計事例と部品事例による UI 設計知識の積極的な再利用、iii) ユーザによる要求確認や新たな要求の具体化を促進する視覚的なラピッドプロトタイプを提供を特徴に持つ、事例を用いた UI 開発環境、CUIDE を提案する。また、CUIDE のプロトタイプを試作し、その有効性を評価する。

A Methodology of Flexible UI Development and Construction of Its Support Environment

Yoshiichi Tokuda[†], Eun-Seok Lee^{††} and Norio Shiratori[†]

[†] Research Institute of Electrical Communication
/ Graduate School of Information Sciences,
Tohoku University, Sendai, Japan

^{††} Department of Information Engineering,
Faculty of Engineering,
Sung-Kyun-Kwan University, Seoul, Korea

Recently, the necessity of UI development support environments, which enable end users to design, modify and expand UI easily, has heightened, to cope with various requirements of UI. In this paper, in order to develop UI by end users, who have little knowledge and experience of UI development, UI development support environment needs to have especially the following features; i) user friendly embodiment of users' requirements in a form close to their own knowledge or mental model, ii) active reuse of knowledge for UI design, and iii) visualized rapid prototyping. We propose Case-based UI Development Environment, CUIDE, and evaluate its effects with a prototype system of CUIDE.

1 はじめに

コンピュータを利用するユーザ層の拡大に伴い、コンピュータシステムにおけるユーザインタフェース (UI) の果たす役割がますます重要になってきている。その一方で、UI 開発は高度な技術と経験を要するので、優れた UI はこれらを有するエキスパートによってのみ開発可能であり、従来の UI 開発支援もエキスパートユーザの支援にその重点をおいている [2]-[4]。しかし従来のようにエキスパートに依存した UI 開発では、幅広いユーザ層からの多様化する UI に対する要求への対処に限界がある。

本研究ではこのような問題点に対する一つの解決策として、UI 設計に関する専門的知識をあまり持っていないエンドユーザでも容易に UI を設計、保守かつ拡張することを可能にすることを目指す。そのために非エキスパート向けのフレキシブルな UI 開発支援法と支援環境 CUIDE (Case-based UI Development Environment) を提案する。CUIDE では特に、ユーザの所有知識に近い形式で要求を具体化し、設計事例及び部品事例を効果的に活用し、要求を即座にかつ視覚的にユーザに確認させるなどの支援を行なう。これにより、UI 開発経験のない非エキスパートユーザが、新たに UI 設計のための言語やツール等の学習に時間をさくことなく、効率的に望みの UI を開発することが可能になるなど、非エキスパートユーザによる UI 開発効率の飛躍的向上が期待できる。

2 非エキスパート向け UI 開発支援法

エンドユーザによる UI 開発プロセスを、i) ユーザの UI イメージからユーザ要求を抽出し、UI 記述言語などを用いてそれを記述する、要求具体化・設計プロセスと、ii) 設計結果に基づき UI を実装する実装プロセスに大別する。この中で、ii) での支援は、従来のエキスパート向けの UI 開発支援法において自動化など様々な支援法が検討されている [4],[5]。しかしながら、i) での支援については、その作業の難しさから検討例が少ないのが現状である。

本研究では、ユーザの UI の設計知識や経験の

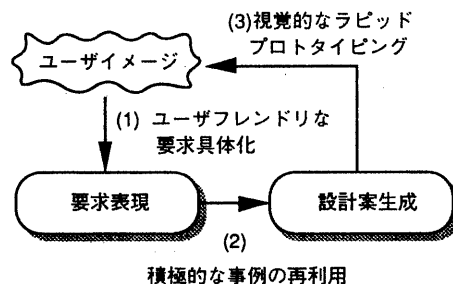


図 1: 要求具体化・設計プロセスのモデル

不足のため難があると言われている要求具体化・設計プロセスに特に重点をおいて支援することを考えている。非エキスパートによる要求具体化・設計プロセスを効果的に支援するために、図1のように、このプロセスを(1) ユーザフレンドリな要求具体化、(2) 積極的な事例の再利用、および(3) 視覚的なラピッドプロトタイピングの3つのステップの反復的なプロセスとしてモデル化する。(1)では、ユーザの持つ UI に対するイメージを、ユーザに分かり易い形で具体化し、(2)では、(1)で獲得したユーザ要求(初期要求)をもとに、事例として表現かつ蓄積された既存の UI 設計例を活用して設計案を生成する。(3)では、その設計案をもとに視覚的かつシミュレーション可能なプロトタイプをユーザに提供する。また、(1)では、このプロトタイプに対する拡張、修正などの詳細化要求の具体化も行なわれる。

2.1 ユーザフレンドリな要求具体化

非エキスパートユーザによる要求具体化タスクを容易にするために、個々のユーザがもつ UI に対するイメージに近い形での要求の具体化を支援する。このような要求具体化支援法を提案するにあたり、UI 開発の非エキスパートを対象に以下のようなユーザ要求獲得実験を行なった。まず、被験者が開発したい UI のイメージを自由な発話により具体化させ、その記録を行なう(初期要求の獲得)。次に、実験者がこれらの初期要求に基づき UI の設計案を図を用いて被験者に

提示する。さらにその設計案に対する修正、拡張などの詳細化要求を同様の方法で獲得し、被験者が満足する設計案が得られるまで、この詳細化要求の具体化と設計案の提示を繰り返す。ユーザ要求獲得実験後、記録したデータをもとにプロトコル分析を行なった。その結果、ユーザの初期要求は、

- UI オブジェクトの機能的な特徴
- オブジェクトを構成する構成要素や構成要素間の関係
- ユーザ操作に応じた UI オブジェクトの遷移関係

などに関する断片的な要求が混在していることが確認された。また、初期要求に応じて具体的な UI レイアウトをユーザに提供すると、非エキスパートユーザでも比較的容易に、設計案と自分のイメージとの相違点を発見し、それを詳細化要求として具体化できることも確認できた。

本研究では、この分析結果に基づき、ユーザがイメージする UI に対する要求を、次のような UI 要求モデルで表現する。UI 要求モデルは、(1)UI オブジェクトの機能的、表現的特徴、(2)UI オブジェクトの構成構造、(3)UI オブジェクト間のインタラクションからなる。(1)は個々の UI オブジェクトに関する機能的あるいは表現的な特徴についての要求である。(2)は UI オブジェクトの組織的構造、あるいは視覚的な関係についての要求である。(3)はオブジェクトと環境とのインタラクションに関する要求で、ユーザの操作に対する応答の形式表現される。また、(1)～(3)のユーザ要求の表現を次のように定式化する。

- UI Requirement Model
= {Feature, Organization, Interaction}
- Feature = {function | representation}*
function ∈ Function,
representation ∈ Representation
- Organization = {Components, Rel}

Components = {objects}*
Rel = {comp-i, rel, comp-j}*
objects ∈ UObjects,
comp-i, comp-j ∈ Components,
rel ∈ Relation

- Interaction = {operation, obj1, reaction, obj2}*
operation, reaction ∈ Function,
obj1, obj2 ∈ UObjects

以上のようなモデルに従い、非エキスパートユーザでも容易に要求を具体化できるように、上記の Function, Representation, UObjects, Relation について、予め知識体系を整備し、UI 設計知識構造として表現する。

2.2 事例の再利用

前述の分析結果が示すように、UI 設計の知識や経験を十分持たない非エキスパートユーザでも、利用経験などをもとに開発目標の UI に対するイメージを持っている。しかし、実際に UI を設計する際には（たとえ、UI 部品が提供されていたとしても）、次のような設計知識、

- i) 機能知識
- ii) レイアウト知識
- iii) 属性知識

が要求される。i) はユーザの機能的な要求を満たす UI 部品を決定するための知識である。ii) は選択された UI オブジェクトを組織化し、かつレイアウトするための知識である。iii) は各々の UI 部品の属性を詳細に設計するための知識である。このようにユーザの所有知識と UI 設計に必要な知識との間の大きなギャップを埋めるために、単に UI 部品をユーザに提供するだけでは、非エキスパートが提供された UI 部品を用いてイメージどおりの UI を設計、開発することは困難である。

本研究では、UI 設計のエキスパートが所有すると思われる上記のような UI 設計知識を設計事例と部品事例の 2 種類の事例に分類し、ユーザ

要求に応じてこれらを積極的に活用する。設計事例は UI 部品を用いた設計手法に関する知識で、部品事例は UI を構成している構成要素としての UI 部品に関する知識である。これらの事例は、ユーザにより具体化された要求と、それに依じて行われた UI の設計に関する情報、あるいは部品の属性情報で表現される。エンドユーザは、具体化されたユーザ要求をもとに、既存の類似したユーザ要求に応じて設計された設計例や部品の詳細化の例を再利用して、UI の設計を行なうことができる。

本研究で、UI 設計知識を表現するために事例を用いた理由は次のとおりである。一般に、エンドユーザは UI 開発の開始時に、UI に対する完全なイメージを持っているとは限らず、断片的でかつ不明瞭なイメージしか持っていないことの方が多い。このような要求に対し、設計知識をルールとして表現した手法では、要求の不完全さに対する対処の柔軟性が乏しい。また、十分な設計を行なうためには大量のユーザ要求の獲得やルールの獲得を行う必要があり、プロトタイピングの効率やユーザの要求具体化の容易さの点においても問題が多いと考えたからである。一方、本研究での事例の活用は、UI 開発の効率化のみならず、UI 開発の望ましい方向を示唆するためにも有効であると考えられ、経験的な UI 設計知識として、ユーザ要求に類似した既存の優れた設計例を積極的に再利用する手段を提供する。

2.3 視覚的なラピッドプロトタイピング

設計事例と部品事例の活用により、ユーザの UI 設計知識や経験の不足が補われるが、これだけでは設計結果がユーザの設計目標とする UI の条件を満たしているかどうかを、非エキスパートユーザが確認することは困難である。

本研究では、個々の UI 部品についてだけでなく、事例を用いて設計された設計案をもとに、獲得されたユーザ要求に対する UI イメージを即座に生成、表示することによりユーザに視覚的に設計結果を確認させる。また、ユーザに提供する UI イメージは、見た目だけでなく、UI オ

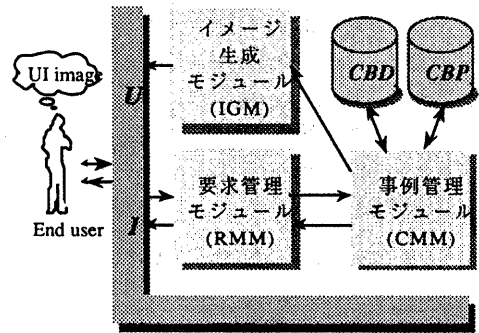


図 2: CUIDE のシステム構成

ブジェクトとのインタラクションなども確認できるように、ユーザが自由に操作可能なシミュレーション機能を有している。このように、UI 開発プロセスの初期段階でユーザ要求の確認のための手段を提供することで、UI 開発の容易化および効率化を図る。

3 事例を用いた UI 開発支援環境 CUIDE

本研究では、エンドユーザによる UI 開発を効果的に支援するために、上述の非エキスパート向け UI 設計方法論にもとづき、事例を用いた UI 開発支援環境 CUIDE を提案する。CUIDE は、(1) 要求管理モジュール (RMM)、(2) 事例管理モジュール (CMM)、(3) イメージ生成モジュール (IGM)、(4) 設計事例ベース (CBD) および (5) 部品事例ベース (CBP) から構成される (図 2)。

3.1 機能モジュール

(1) 要求管理モジュール (RMM)

RMM は、i) 要求具体化支援サブモジュール (REM)、ii) 要求分析サブモジュール (RAM)、iii) 設計管理サブモジュール (DMM) からなる。REM は、エンドユーザでも容易に要求を具体化できるように、機能要求や修正要求など様々なユーザ要求を、その要求に応じて、メニューによるキーワードの選択、図的表現形式、UI オブジェクトに対するダイレクトマニピュレーションなどの方法で獲得する。RAM は、REM でユーザ

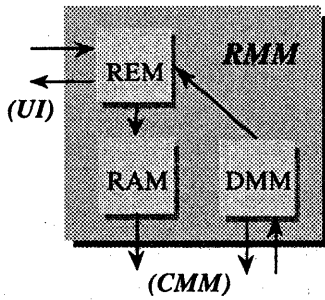


図 3: 要求管理モジュール

から獲得された要求を分析し、事例検索情報あるいは事例修正情報の形式に組織化する。これらの分析・組織化の効率化のために、本研究では UI 設計知識と UI 部品を予め UI 設計知識構造として組織化し、これを用いてユーザ要求の分析・組織化を行なう。なお本 UI 設計知識構造は、ユーザ要求の獲得機構を設計する際の論理的ベースを提供する。DMM は、設計中の UI に関連する事例の管理を行う。

(2) 事例管理モジュール (CMM)

CMM は、i) 事例検索サブモジュール (CRM) と ii) 事例適用サブモジュール (CAM) からなる。CRM は、獲得されたユーザ要求に類似の事例を、事例検索情報に基づき CBD/CBP から検索する。また、CAM で修正された事例の再保存も行う。CAM は、事例修正情報に応じて、検索された事例の修正や拡張を行い、ユーザ要求に適用する UI 設計案を生成する。

(3) イメージ生成モジュール (IGM)

IGM は、i) UI 生成サブモジュール (UGM) と ii) イメージ表示サブモジュール (IPM) からなる。UGM は、個々の UI オブジェクトの生成をはじめ、CMM で設計された UI 設計案に基づき UI 部品を組み合わせより複雑な UI オブジェクトを自動生成する。IPM は、UGM で生成された UI オブジェクトを、視覚的かつ自由にシミュレート可能な UI イメージを生成し、ユーザに提供する。

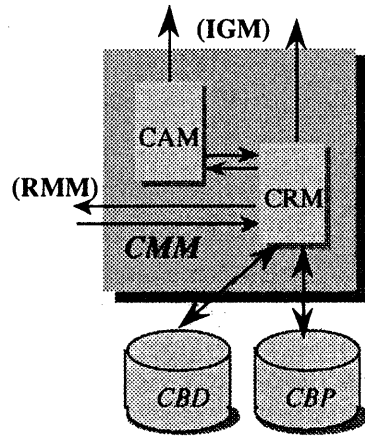


図 4: 事例管理モジュール

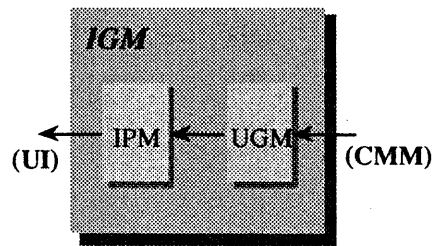


図 5: イメージ生成モジュール

3.2 事例ベース

(1) 設計事例ベース (CBD)

CBD は、ユーザの初期要求に応じて UI の構成部品を選択かつ構造化したり、詳細化要求に応じて設計を修正、拡張したりするための経験的な UI 設計知識を蓄積する。設計事例は、UI 設計者により設計された優れた設計例や個々のユーザにより生成された好みの設計例からなるレイアウト事例と、設計例に対する詳細化要求に基づいて施された修正の際のノウハウ的な知識である修正事例から構成される。

(2) 部品事例ベース (CBP)

CBP は、ユーザの機能要求や詳細化要求に基づいて個々の UI 部品を選択したり、UI 部品の詳細な属性を修正、拡張したりするための知識

を蓄積する。部品事例は、UI 部品の属性に関する一般的な知識からなるイメージ部品事例と、特定のユーザ要求に応じてカスタマイズされた UI 部品例からなる機能部品事例から構成される。

4 実装と評価

我々は 2 章で提案した非エキスパート向け UI 開発支援方法論の有効性を評価するために、基本的な UI 部品として Motif ウィジェットセット、UI の設計記述言語として Motif の UIL を用いて、C 言語で CUIDE のプロトタイプシステムを試作した。また CUIDE のモジュールのうち CRM と 2 つの事例ベース、CBD と CDP は、本研究室で開発され、事例ベースの構築、事例の蓄積、検索を容易に行うことができる事例検索支援環境、CSSE を用いて構築、運用されている。

現在このシステムを用いて、非エキスパートユーザを対象に次のような手順で、CUIDE による UI 開発支援の有効性の評価を進めている。

(1) 代表的な UI 構築ツールとの比較

Motif[6]、UIL などを比較対象とし、1) 設計の際に必要とされる知識群や設計/構築プロセスの比較、及び 2) サンプル UI の開発における設計時間、修正時間などの定量的評価を行なう。

(2) 従来の部品を用いた開発環境との比較

IntelligentPad[5]、UIM などを比較対象とし、部品の選択や修正方法などの特徴的な要素からの推定比較による定性的評価を行なう。

本稿では、UI の利用経験はあるが設計経験はないエンドユーザを支援対象として評価を行ない、その中間結果として (1)-1) について報告する。このようなユーザは Motif や UIL などのツールによる支援では、UI 設計知識や提供された部品に関する詳細な知識が必要とされるため、UI 構築が思い通りに進まないのが観測された。一方 CUIDE を用いた開発では、UI の利用知識を活用することで、望みの UI に対する要求を段階的に具体化し、設計を洗練していくことが可能であることが確認できた。

5 まとめ

本研究では、非エキスパートユーザによる UI 開発を支援するために、ユーザフレンドリなユーザ要求具体化、積極的な UI 設計知識の再利用、視覚化されたラビッドプロトタイピングの特徴を持つ UI 開発支援方法論を提案し、その支援環境 CUIDE を試作した。また、CUIDE を用いた本方法論の評価の中間結果を報告した。

今後の課題としては、UI 設計の非エキスパートのエンドユーザを対象に評価実験を行ない、定性的かつ定量的に評価を行なう必要がある。特に、ユーザフレンドリな要求具体化支援法及びそれに関する CUIDE の支援機構については重点的に評価を行ない、UI 設計知識構造などの再検討を行なう予定である。

参考文献

- [1] Y. Tokuda, E. S. Lee and N. Shiratori, "User Interface Development Environment for End Users : CUIDE", In *Proceedings of HCI International '95(to appear)*, 1995
- [2] 木下哲男, 菅原研次, 白鳥則郎, "コンピュータコミュニケーションシステム設計のための知識型要求定義方式", 電子情報通信学会論文誌 A, Vol.J76-A, No.3, pp.528-539, 1993
- [3] 神場知成, 橋本治, "マルチビューモデルに基づくユーザインタフェース設計ツール U-face", 情報処理学会論文誌, Vol.34, No.1, pp.167-176, 1993
- [4] P. Sukaviriya, J. D. Foley and T. Griffith, "A Second Generation User Interface Design Environment : The Model and The Runtime Architecture", In *Proceedings of the ACM INTERCHI'93*, pp.375-382, 1993
- [5] 長崎祥, 田中譲, "シンセティック・メディアシステム : IntelligentPad", コンピュータソフトウェア, No.11, Vol.1, pp.36-48, 1994
- [6] Open Software Foundation, *OSF/Motif スタイルガイドリリース 1.2*, 株式会社トッパン, 1993