

UX (ユーザエクスペリエンス) 指向開発事例

Case introduction of UX (User Experience) Inclination development

吉原 ヤマト

要約 2008年以降のiPhone, iPadの登場に代表されるように、情報システムは、ユーザに対し豊かな表現方法や利用方法を提供することが必須の要件となってきた。この流れは、業務システムにも影響を与えることは必然であり、これからの業務システムの開発においては、ユーザが真にやりたいことを「楽しく」「面白く」「心地よく」実現させるコンセプトであるUX (ユーザエクスペリエンス) 指向のもとでシステムを構築する必要性が高まる。

S社事例において、UX指向を取り入れた業務システムを構築することが必須要件であり、一般的な開発工程以外に次の2点を実施しシステム開発を成功させている。一つは、曖昧なデザイン要件を具現化するために、通常の開発工程と平行して4段階のUXデザインプロセスを実施したこと、もう一つは、独特なクライアントアプリケーションを実現するために、デザインやインタラクション(動き)とビジネスロジックを完全に分離したフレームワークを作成したことである。

Abstract Since 2008, as represented by iPhone and iPad, the information systems have been required to provide the variety of expressions and usages. This trend has a necessary impact on business system. For the future business system development, we have the increased demands to develop systems under UX (User Experience) design, which allows the users to do what they truly want to do 'pleasurably', 'intriguingly', and 'pleasantly'.

In the case of the system development for Company S, we are required to develop an UX design business system. We executed the following two methods and succeeded the system development. The first method is the implementation of four steps of UX design process to embody ambiguous design requirements under the regular process of development. The second method is the Creation of the framework, which completely separates a design, an interaction, and business logic, to realize a unique client application.

1. はじめに

2008年のiPhone 3G, 2010年のiPadの登場に代表されるように、情報システムとユーザとの関係は進化し続けている。それに呼応して顧客のビジネスも進化し、情報システムは単に早く正確に処理することに加えて、豊かな表現方法や利用方法を提供することが求められてきている。本稿では、日本ユニシスがサービスインテグレーション企業として、顧客のビジネススピードに即応するため、UX (ユーザエクスペリエンス) という最新の開発指向とWPFという最新の技術を利用して、進化したヒューマン・コンピュータインターフェースを実現した事例を紹介する。2章にてUX指向開発について、3章にて事例に取り上げたS社システムについて述べ、4章でUX指向開発の設計プロセスを解説し、5章でアプリケーションフレームワークPWSについて言及する。

2. UX 指向開発とは

UX とは、製品やサービスの使用・消費・所有などを通じて、人間が認知する体験のことであり、製品やサービスを利用する過程を重視し、ユーザが真にやりたいことを「楽しく」「面白く」「心地よく」実現できる点を、機能や結果、あるいは使いやすさとは別の“提供価値”として考えることである。主に一般コンシューマー向けの製品やサービスなどでは常識的に使用されているコンセプトである。

ICT の分野では、コンピュータの使い勝手に関し、不快感を与えたり、面倒な手続きを強いることなく、顧客のニーズを正確に満たすようにデザインすることで、使用すること自体が喜びとなるシステムの構築を目指すことである。よって、UX 指向開発の特徴は、ただ最終的な要求/要件を実行できるものを作るに留まらず、もっと広い視点からアプローチすることで、エンドユーザにとってより目的にかなったものを目指そうという考え方にある。言い換えれば、機能面や合理面のみならず満足感や達成感といった感情面を積極的に考慮することが UX 指向開発の特徴であり、UX 指向開発を目指すものは、エンドユーザが「使える、使いやすい、使いたい」と感じられるアプリケーションの開発である。

3. S 社事例

UX 指向のシステム開発の事例として S 社のシステム構築プロジェクトを紹介する。S 社は主に化粧品を取り扱う大手通信販売会社である。S 社のビジネスは常に進化しており、情報システムは、ビジネスの進化に即応することはもちろん、ビジネスの進化を先導していくことが必要であった。

S 社の基幹システムの全面更改プロジェクトは、2008 年から 2011 年にかけて実施された。その中心的な目的は、オープン化をきっかけとしたデータベースとビジネスロジックの統合化であったが、同時に開発されたコールセンターで注文を受注するオペレータ向けのクライアントアプリケーションは、顧客との接点そのもので、S 社のビジネスに直結する重要な存在であり、ユーザインターフェースについて難易度の高い要件があった。以下にて主要要件の概要と、実現された画面の一部を紹介する。要件 1) の実現例が図 1、要件 3) の実現例が図 2 である。

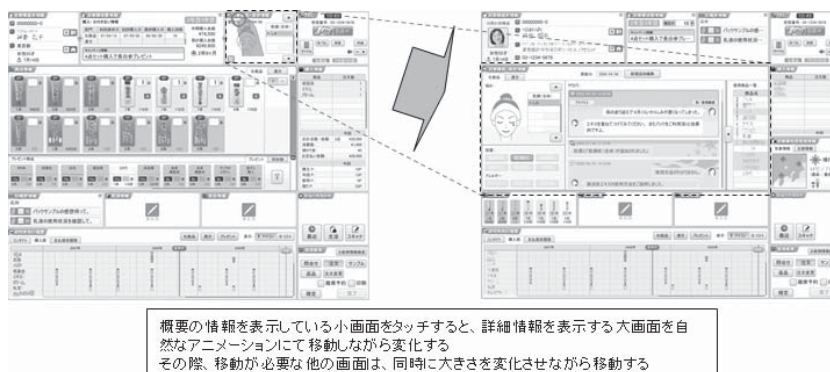


図 1 画面遷移のない画面例



情報を登録する操作は、単にドロップダウンにて選択するのではなく、イメージにドラッグ&ドロップすることにて、より直感的に楽しく操作できる。

図2 「楽しく仕事ができる」画面例

- 1) オペレータが画面に表示される情報にて“お客様を想像できる”ものであること
- 2) オペレータが顧客との会話に集中できるように簡素なタッチパネル操作であること
- 3) オペレータが1日中、気持ち良く、楽しく仕事ができる画面であること
- 4) 常に進化するビジネスに合わせて、容易に画面デザインが変更できること

上記の要件を実現するためには、オープン化開発のプロセスに合わせた通常のクライアントアプリケーション開発プロセスでは実現が難しく、通常のプロセスと平行して、以下の新しいプロセスを実施した。

- ・ユーザインタフェース設計において、通常的设计と平行してUX指向でアプリケーションをデザインする
- ・アプリケーションの独特なインタラクション（動き）を実現し、ビジネスの進化に即応するため、ビジネスロジックと完全に分離させたクライアントアプリケーションのフレームワークを実装する

次章より、UX指向の設計プロセスとフレームワークの実装を紹介する。

4. UX指向の設計プロセス

S社事例では、基本的に日本ユニシスの標準的なシステム構築プロセスを採用している。その中、クライアントアプリケーションの開発、主にUXデザイン部分では、デザイン設計工程が標準的な構築プロセスではそぐわない点もあった。そのため、以下に紹介するUXデザインプロセスを論理設計工程と平行して実施した。

一般的にデザインプロセスの開発手法は持続的、継続的な変更を可能とするアジャイル^{*1}が有効だとされている。もちろん本事例でも、開発後半において持続的、継続的なデザイン変更を行っているが、その変更対象となるベースラインのデザインを初期作成する過程では、図3にて紹介するウォーターフォール型によるデザインプロセスも有効であると考えている。本章の4.1節から4.4節にて、図中のA.クライアントアプリケーション開発手法検討からD.デザイン詳細設計までの各フェーズを説明する。

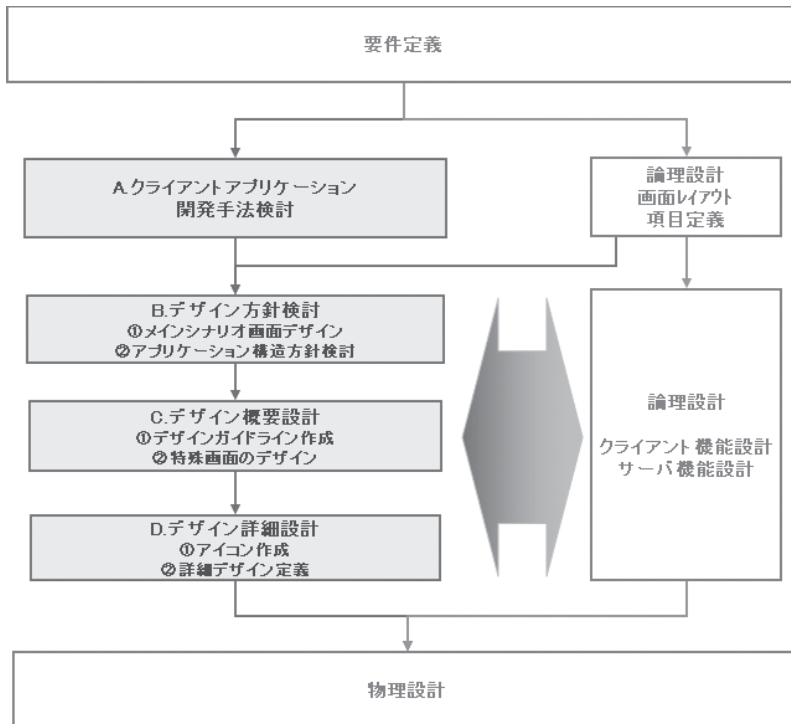


図3 UX デザインプロセス (左) と標準的なシステム構築プロセス (右)

4.1 A. クライアントアプリケーション開発手法検討

要件定義にて明らかにした「お客様を想像できるもの」「気持ち良く楽しく利用できるもの」という高いユーザインタフェースへの要求を実現するため、技術的開発手法を採用した。

当初は Windows フォームを利用したプログラミングを検討していたが、Windows フォームの将来性、グラフィック表現の限界を解決するため、Microsoft の提唱する WPF (Windows Presentation Foundation) を採用した。WPF とは、Microsoft が提供する GUI を開発するためのグラフィックスサブシステムである。3D 表現や動画なども容易に扱え、PC に搭載された GPU を有効利用し、CPU を他の作業に割り当て、最大限に活用する機能を持っており、豊かな GUI 表現と全体的なパフォーマンスの両立が可能である。

4.2 B. デザイン方針検討

論理設計で決定した機能のデザイン要素 (図3の右上側) と WPF を利用することを前提としたデザイン (図3の左上側) を融合して、ビジュアル・デザイン (見た目) とインタラクション・デザイン (動き)、その実現方法を検討した結果を本節にて述べる。

4.2.1 メインシナリオ画面デザイン

100 枚を超えるアプリケーション画面を短時間で効率よく作成するため、まず骨格となるデザインのベースを決定し、それを基に全ての画面をデザインした。骨格となるデザインは、その後のデザイン設計に大きな影響を与える。このため、一番多く利用する業務シーンの画面の動作をシナリオ化し、そこで使用する約 20 画面を重点的に先行してデザインした。前章で紹介

介した図1の画面が代表的な画面であるが、20画面は単にビジュアル・デザインだけではなく、インタラクション・デザインを含めて確認できることが重要であり、実際にタッチパネルで操作できるモックアップ*2を作成し、S社の経営層からシステムの利用者であるオペレータまでにプレゼンテーションして、両デザインについて合意を得た。

4.2.2 アプリケーション構造方針検討

開発するアプリケーションは、極力画面遷移をなくす必要があるため、各画面がサイズと表示位置を変えるという特異なインタラクションを持つ。これを実現するために、画面のサイズと動きを制御するアプリケーションフレームワーク：PWS（パネルウィンドウシステム）を新たに作成した。また、このアプリケーション基盤と各画面の機能プログラムとを分離することで、プログラム製造の生産性/保守性を高めることができる。PWSの詳細は、次章にて述べる。

4.3 C. デザイン概要設計

このフェーズでは、4.2.1項のメインシナリオ画面デザインで決定したことをアプリケーション全体に展開するために、標準化を実施した。

4.3.1 デザインガイドライン作成

メインシナリオ画面デザインで作成したデザイン要素を、アプリケーション全体のデザイン標準とするためガイドラインを作成した。作成したガイドラインの内容は、「機能デザイン基本構成」「レイアウトルール」「カラールール」「基本パーツ一覧」である。このガイドラインに則りアプリケーション全体の画面をデザインすることで、開発メンバー間でより短時間に統一感のあるデザインを実現することを可能とした。

4.3.2 特殊画面のデザイン

標準化されたデザインは開発生産性やデザインの統一性では有効であるが、「楽しく」「面白く」「心地よく」を追求する画面は、ガイドラインにて標準化するべきではないものも発生する。前章で紹介した図2の画面が代表的な例であり、ガイドラインに沿ってデザインするのではなく、ユーザ業務を直感的に、また「楽しく」「面白く」「心地よく」行えるように独特にデザインされたアイコンを用い、タッチパネルのドラッグ操作にてデータ登録を可能とした特殊デザインの画面である。このような画面は数少ないが、これらの特殊デザインはアプリケーション全体の満足度に大きく影響するために、この作業を標準化とは別に定義し、利用者であるオペレータの意見を取り入れ、アプリケーションの満足度を高めるスパイス要素とした。ただし、安易に特殊デザインの画面を増やしていくと、視認性や操作性を落としてしまうことにつながるため、特殊デザインを採用する際の判断は厳密にしていく必要がある。

4.4 D. デザイン詳細設計

このフェーズでは、アプリケーション内の装飾、アイコン作成、RGB値による色設定という詳細なデザイン要素を決定した。この作業は、いかに綿密に実現したとしても、後工程で変更することが想定できたため、ユーザがシステムテスト工程などで実際のデザインを確認する

まで、最小限のデザイン作業体制を維持し続けた。

4.4.1 アイコン作成

限られた表示スペースでオペレータに分かりやすく多くの情報を伝えるために、画面に表示する情報は可能な限りテキスト表記ではなく、アイコン表現とした。アイコンのデザインは正しく情報を伝え、かつ画面全体のデザインの中に調和しつつも瞬時に目視できることを要求される。そのバランスを取ることは、単に美しいデザインのアイコンを作ることより難易度が高い作業である。図4のアイコンはユーザの顧客との接点を時系列に表示する画面にて使用しているものである。視認性を確認するために画面背景上に展開し慎重に検討した。



図4 アイコン例

4.4.2 詳細デザイン定義

主に画面内で使用される色彩を定義した。画面内のテキスト情報等についても、情報の性質、重要度により色に違いをつけ、その色の違いを情報伝達手段として使用している。そのため、アイコンと同じように、画面全体のデザインとの調和、高い視認性、ユーザの長時間使用が可能な色を十分に検討し、最終的には色彩の物理的な定義である RGB 値を決定し、プログラミング工程で問題にならないようにデザインの曖昧さを排除している。

5. アプリケーションフレームワーク PWS (パネルウィンドウシステム)

本アプリケーションの最大の特徴は画面遷移がないことである。画面遷移をなくすためには、依存・非依存の関係にある複数の画面を同時に表示し、ユーザ操作により、各画面の大きさと位置を移動させ、整合性のとれたデータを作成できる必要がある。よって、これらを実現するためのフレームワーク PWS (パネルウィンドウシステム) を作成した。PWS には、本章で述べる三つの機能を実装した。

5.1 レイアウト制御機能

図5で示すように、各画面からのレイアウト変更要求を受け、PWSにてレイアウト変更を実施させている。レイアウト変更はアニメーションにて、各画面の大きさを変更させながら自然な動きで移動させることにより美しいデザインとインタラクションを実現している。また、レイアウト情報は定義ファイルへの設定項目として、レイアウトの変更が生じた場合においても、最低限のシステム改修にて実現できるようにしている。

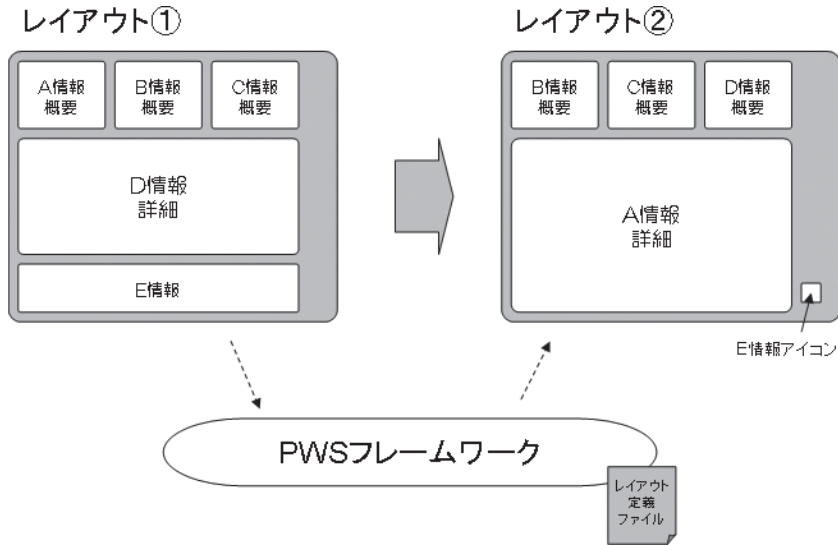


図5 レイアウト制御機能

5.2 プロセス制御機能

本アプリケーションは数多くの画面が同時に動くため、全ての画面がデータを読み込むまで動作をさせないとレスポンスが悪化し、顧客業務上、不都合が発生する。よって、図6に示すように、PWSのプロセス制御機能によって、各画面はマルチスレッドで動かし、他の画面がデータ読み込み中でも動作することを可能にし、多画面にて多量のデータを取り扱っても、レスポンスを保つ構造としている。また、各画面の整合性を保つ必要がある場合は、操作が完結するまでモーダル*3状態を維持させ、整合性が崩れないようにしている。

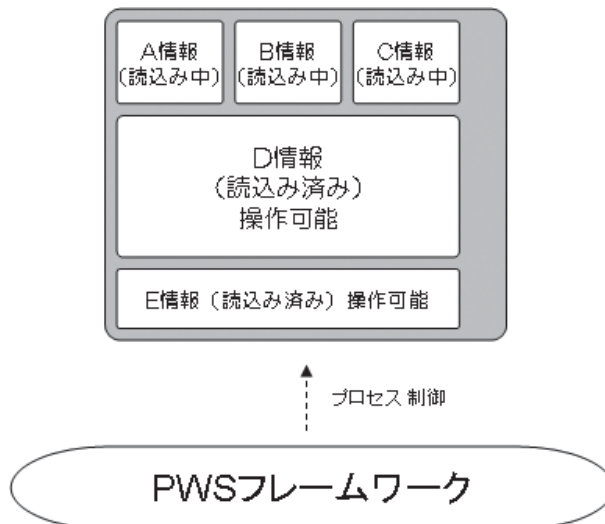


図6 プロセス制御

5.3 データ管理機能

各画面が直接データを取得すると、データ取得が重複して無駄なデータアクセスが発生してしまう。よって、図7で示すように、各画面からのデータ取得要求をPWSにて整理し、必要最低限のデータを取得することを実現している。また、データ更新については、各画面の共有データを保持し、一つの画面より情報更新があった場合は、他の画面に通知し、データの整合性を保たせ、DBへのデータ更新はPWSで一括して行う。

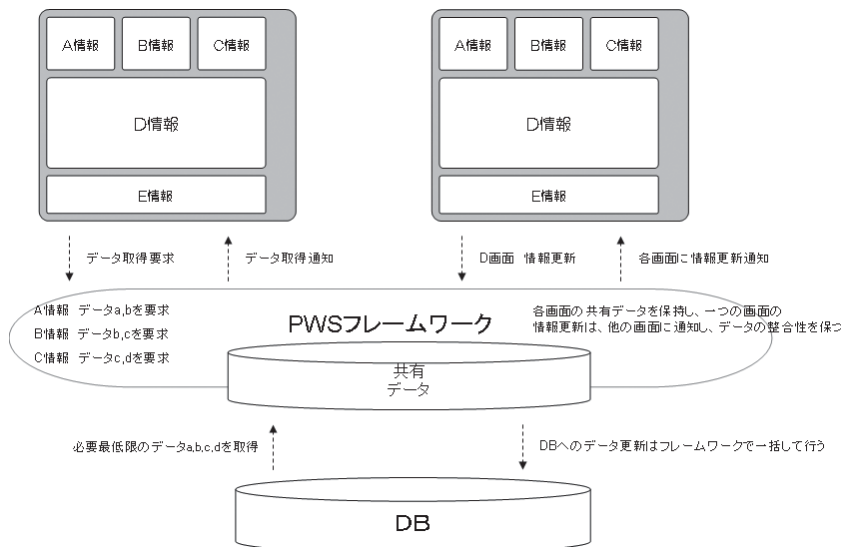


図7 データ管理機能

6. おわりに

事例として紹介したシステムは2011年7月より本番稼働を開始している。WPFを利用したアプリケーションとしては、国内外において最大級のミッションクリティカルなシステムであるという評価を頂いている。これも皆、S社や社内、協力会社を含めたメンバのおかげであり、この場を借りてお礼を申し上げる。

システムの利用形態が変化してきた今日、インターネット、コンシューマー向けのシステムはもちろん、業務システム開発においても、UX指向を利用するような最新の手法、技術を利用すべきシステム要件が多くなることは間違いないであろう。

日本ユニシスのシステム構築も、これまでの機能中心のシステム設計から進化し、顧客ビジネスのあるべき姿を追求し、顧客からの期待を見越してチャレンジングな対応を開始しており、また今後も進化し続けていくと確信している。

- * 1 迅速かつ適応的にソフトウェアを開発する軽量な開発手法群の総称
- * 2 外見と動きを実物そっくりに似せたサンプル画面
- * 3 作り手の意に沿う操作をしない限り、処理を進めないようにウィンドウを制御すること

執筆者紹介 吉原 ヤマト (Yamato Yoshihara)

2001年日本ユニシス(株)入社。主に営業支援システム、コールセンターのシステム構築に携わり、大手自動車メーカー、証券会社、通信教育会社のシステム構築に従事。2008年より製造流通事業部にてダイレクトマーケティング業界向けのシステム開発に従事。PMAJ認定PMスペシャリスト。

