

組込みシステム向け HMI ツールのモデルベース開発について

On Model-based Development of a HMI Tool for Embedded Systems

新 吉高[♥] 鯨井 俊宏[♦]
 土井 敬司[♦] 深谷 直彦[♦]
 原 隆浩[♦] 西尾 章治郎[♦]

Yoshitaka ATARASHI Toshihiro KUJIRAI
 Keiji DOI Naohiko FUKAYA
 Takahiro HARA Shojiro NISHIO

組込みシステムは製品の多機能化により、HMI 開発工数が増大している。市販ツールによる設計は個々の製品の特性に合わせ、使い方に制約を持たせる必要があったため、製品系列や世代を超えた HMI 設計情報の共有が難しかった。そこで、本稿では組込みシステム向け HMI ツールのモデルベース開発について議論する。開発したツールでは、市販ツールからの変換規則を定め、データベースでモデルを統合管理することで、市販ツールの使い方に依存しない設計情報の管理を可能とした。

Man-month of HMI development for embedded systems is increasing rapidly because a wide variety of functions are required. While it is easy to design a HMI specification using standard office applications, it becomes difficult to share HMI specifications among the product models and generations. In this paper, we propose a model-based development method of a HMI tool for embedded systems. The proposed tool can transform HMI specifications designed by standard office applications into XML-based models which do not depend on how to use the applications.

1. はじめに

カーナビ、携帯電話に代表される情報系の組込みシステムは、本来の機能だけでなく、地上波デジタル放送受信機能や音楽プレイヤー機能などを取り込み、多機能化が進んでいる。このため、組込みシステムにおける HMI (Human Machine Interface) 開発は、PCと比べて画面が狭いという制約を考慮しつつ、多様な情報提供、組込みシステム特有の特殊なハードウェアデバイスとの接続に対応する必要がある。

例えば、一般的なウィンドウシステムでは、それぞれのアプリケーションに独立したウィンドウが与えられているため、アプリケーションの画面もそれぞれ独立している。一方

カーナビを例に取ると、地図表示・誘導といった基本的な機能以外に、音楽再生や交通情報など様々なアプリケーションが実行される。ユーザにとって見やすい情報提供を行うためにはレイアウトに工夫が必要で、各アプリケーションが独立して画面を設計できないという課題がある。携帯電話も近年の画面解像度の向上により、同時に複数アプリケーションの情報を提供するようになり、同様の問題が発生しつつある。

この結果、組込みシステムのハードウェア制約、多機能化に起因する要求仕様の調整が複雑化し、製品系列や世代を超えた HMI 設計情報の流用が難しくなったため、製品開発期間にも影響を与えるようになった。

これを解決するためには、仕様調整の結果を共有し、仕様変更の影響範囲を追従可能とする必要がある。HMI 設計情報を、製品系列や世代に非依存なメタモデルを用いてモデル化し、モデルを統合的に管理するデータベースを用意することで、上流工程から下流工程に速やかに情報が流れるようになり、開発期間短縮が期待できる。

そこで本稿では、組込みシステムに適した HMI 開発のために、様々なツールの設計情報を柔軟に扱える HMI ツールのモデルベース開発について議論する。開発したツールでは、市販ツールからの変換規則を定め、データベースでモデルを統合管理することで、市販ツールの使い方に依存しない設計情報の管理を可能とする。

モデルベース開発の有効性と課題を明らかにするため、カーナビの HMI 設計情報を用いたプロトタイプによる評価を実施した。

2. 関連研究

組込みシステム向け HMI 開発に関し、XML によるモデル化のアプローチを取る研究がいくつか行われている。

文献[1]はモデルベース HMI 開発の事例である。HMI 開発における自動車メーカーとサプライヤの関係を示し、従来紙ベースで仕様を取り交わしていた HMI 設計から、モデルベース開発により HMI シミュレーションモデルを互いにブラッシュアップすることで開発プロセスが改善すると主張している。一方、本研究では、要求仕様が必ずしもモデル化されている必要はなく、様々な要求に対応できるようにモデルを自由にカスタマイズすることで、多様な要求仕様のモデル化を実現するアプローチを取る。

文献[2]は車載機器における、XML による HMI 設計情報の標準化に関して報告している。これは、車載マルチメディアインタフェースの標準化を目標とする AMI-C (Automotive Multimedia Interface Collaboration) の成果である。Java VM と OSGi (Open Services Gateway Initiative) を前提ソフトウェアとして要求するため、HMI 開発効率の向上には寄与しても、組込みシステムとしてのコスト削減要求には応えられない。一方、本研究では、このような組込みシステム上での前提ソフトウェアは要求せず、モデルから必要な情報を抽出し、組込みシステムの制約に合わせた変換ツールを個別開発するアプローチを取る。

モデルベース HMI 開発を支援する製品として、独自のデザインツールと CASE ツールを提供し、設計情報からシミュレーション実行、さらには実行形式への変換を行えるものが既に市販されている。ツールがサポートするモデルの記述能力がそのまま設計能力となるが、複雑な組込みシステムの仕様に対応可能なツールは今のところ見当たらない。

EMF[3] (Eclipse[4] モデリングフレームワーク) はツール

[♥] 学生会員 大阪大学大学院情報科学研究科マルチメディア工学専攻 博士後期課程, (株)エイチ・シー・エックス yoshitaka.atarashi.uw@hitachi.com

[♦](株)日立製作所 ftoshihiro.kujirai.ek.keiji.doi.zm.naohiko.fukaya.ha@hitachi.com

[▲] 正会員 大阪大学大学院情報科学研究科マルチメディア工学専攻 hara.nishio@ist.osaka-u.ac.jp

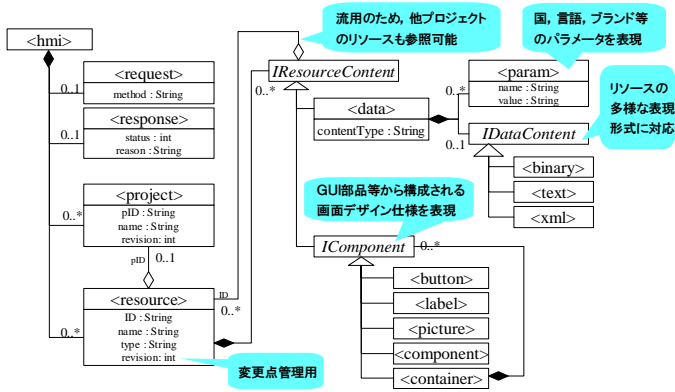


図 1 メタモデルのクラス図表現
Fig.1 Meta-model represented by class diagram

のモデルベース開発を支援するフレームワークである。メタモデルECoreを定義し、これを中心としてJava, XML, UMLを統合して扱うことが可能となっている。EMFは開発途上であるものの非常に強力であり、将来はモデルベース開発を行う上で必須になると期待される。しかし、Eclipse以外のツールとの連携を実現するには敷居が高いのが難点である。

3. 提案ツール

組込みシステム向け HMI ツールが対象とする HMI 設計情報は、紙ベースや市販ツールによる仕様書からの変換、協調開発のための設計情報の一元管理、OS 非依存、プログラム言語非依存、ハードウェア非依存であることが求められる。

XML は木構造で表現可能なモデルやドキュメントを扱うシステムとの親和性が高いことから、メタモデルを XML スキーマで定義する XML 中心の設計手法を採用する。

この手法を用いた場合、モデルの操作を XML バインディング、すなわち XML スキーマから自動生成されるプログラム言語で行うことで、ツールの開発工数を削減可能である。

HMI 設計情報のメタモデルは XML スキーマの一種である RelaxNG[5]を用いて設計する。RelaxNG はシンプルな仕様ながら表現能力が高く、強力な Java バインディングツールが存在することが特徴である。

RelaxNG とクラス図の対応を以下に示す。

- <element> ⇒ クラス
- <attribute> ⇒ 属性
- <optional> ⇒ 0..1
- <zeroOrMore> ⇒ 0..*
- <choice> ⇒ インタフェース
- <ref> ⇒ 集約
- 子エレメント ⇒ コンポジション

3.1 メタモデルの設計

図 1は、メタモデルのクラス図表現の主要な部分を抜粋したものである。hmiタグはルートタグである。HMI 設計情報の一部または全部を表現することができる。

requestタグは、後述のリソース管理DBに対する要求を表現するためのもので、最大1つ存在できる。method属性は操作の種別である。リソース管理系の操作としては、getList (リソース一覧の取得)、get (リソースの取得)、insert (リソースの新規登録)、update (リソースの更新)、delete (リソ

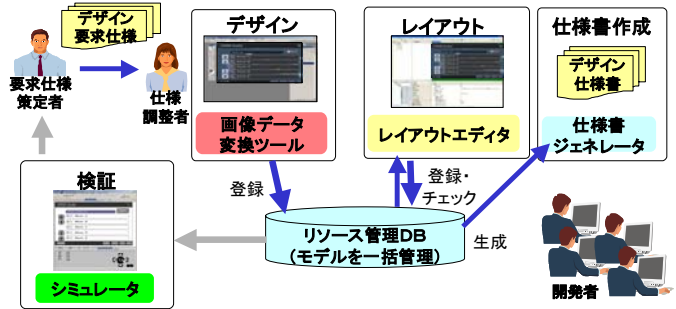


図 2 提案ツールの構成
Fig.2 Structure of the proposed tool

スの削除)を用意している。他にも、リビジョン管理、検索に関するメソッドがある。responseタグは要求に対する返信を表現するためのもので、最大1つ存在できる。status属性は要求の成否を判断するためのもので、HTTP [6] のstatusと同等のコードを採用した。projectタグはHMI 設計情報を纏める単位であるプロジェクトを表現する。pID属性を主キーとしてリソース管理DB で管理される。name属性はプロジェクト名、revision属性はプロジェクトの変更に応じて増加するリビジョン情報を表す。

resourceタグはHMI 設計情報の管理単位であるリソースを表現する。ID属性を主キーとしてリソース管理DBで管理される。name属性はリソース名、type属性はリソースの種類、revision属性はリビジョン情報を表す。また、リソースを管理するプロジェクトのpIDを参照として持つ。

IResourceContentはリソースの中身を抽象化したインタフェースを表し、データとコンポーネントから構成される。resourceタグへの0個以上の参照は、リソースから他のリソースへの参照が含まれる可能性があることを示す。dataタグはデータを表し、データの中身そのものと、データの性質を表す情報から構成される。

IDataContentインタフェースで表されるデータの中身は、文字列、フォント、色、画像、音声等、単体で意味を持つ情報が対象である。テキスト形式、バイナリ形式(XML内部ではBASE64形式で表現)、XML形式のいずれかで保持する。paramタグはデータの性質を表すための情報で、データ1つに対し複数持つことができる。例えば、スキン、言語、昼または夜、製品の納入先などの違いで画像だけが入れ替わる、というような場合に用いる。

IComponentインタフェースは、button タグ、label タグ、picture タグで表現される GUI (Graphical User Interface) 部品、component タグで表現される対象ハードウェアのCPU, OS, プログラム言語等に依存する実装部品、container タグで表現される部品の集合を表すコンポーネントの抽象化表現である。画面レイアウトは container によるツリー構造で表現する。

3.2 提案ツールの構成

図2に、提案ツールの構成を示す。HMI設計情報を統合管理するリソース管理DB は、Servletをインタフェースとし、HTTP/POSTでXMLを送受信する通信プロトコルにより、モデルの統合管理を可能とする。XML管理部は、リレーショナルデータベースMySQLによる実装とバージョン管理ツールSubversionによる実装を行った。

画像データ変換ツールは画像データからモデルへの変換

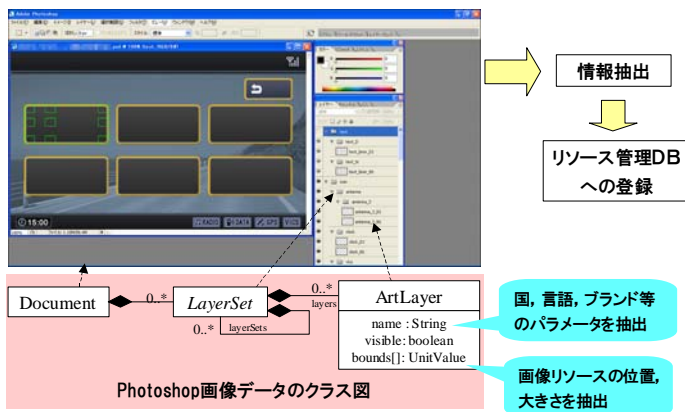


図 3 画像データのモデル
Fig.3 Model of image data

機能を持つ。スクリプト言語[8]によりレイヤー単位で画像を抽出し、リソース管理DBに登録する。

レイアウトエディタはリソース管理DBと連携し、レイアウト編集、コンポーネント定義によるモデル操作が可能なEclipseプラグインである。

仕様書ジェネレータはリソース管理DBからモデルを抽出し、PDF形式またはHTML形式の仕様書を生成可能なアプリケーションである。どのツールで作られた設計情報かという区別なしに仕様書を自由に生成できるため、提案ツールの構成に対して新たなツールを追加しても、モデルの設計が変わらない限りは本ツールを変更する必要はない。

シミュレータはリソース管理DBからモデルを抽出し、製品での見栄え、振る舞いをシミュレート可能である。振る舞いについては、図2のツールに加えて、画面遷移設計ツール、コンポーネント設計ツール等が必要となる。

3.3 画像データのモデル化

図 3は、Photoshop[7]の付属マニュアルに記載されているJavaScript Scripting Reference[8]を参考に、Photoshop画像データのモデルの一部をクラス図表現したものである。

Document はドキュメント全体、LayerSet はレイヤーの集合、ArtLayer は画像を管理する単体のレイヤーを現す。ArtLayerのname属性はレイヤー名、visibleはレイヤーの表示状態、boundsは画像が定義されている範囲を表している。

画像データ変換の方法は、レイヤー名の命名規則を定め、メタモデルに変換することである。ここでは、レイヤー名として、「コンポーネント名_ボタン状態_昼夜状態_スキン名」のように_ (半角のアンダーバー) をデリミタとする命名規則を定義する。

まず、モデルのコンポーネントへの変換を行う。レイヤー名でボタン状態が省略された場合はpictureに変換する。ボタン状態が定義された場合はbuttonに変換する。コンポーネント名はresourceタグのname属性にマッピングする。boundsはPhotoShopの管理する単位(UnitValue)として原点からの相対座標と幅、高さにマッピングする。

次に、モデルのデータへの変換を行う。boundの範囲内の画像をレイヤーからPNG形式で切り出し、dataタグの下のbinaryタグにマッピングする。レイヤー名ボタン状態、昼夜状態、スキン名はそれぞれ、dataタグの下のparamタグにマッピングする。コンポーネントに対してはこのdataタグへの参照を持つよう設定する。

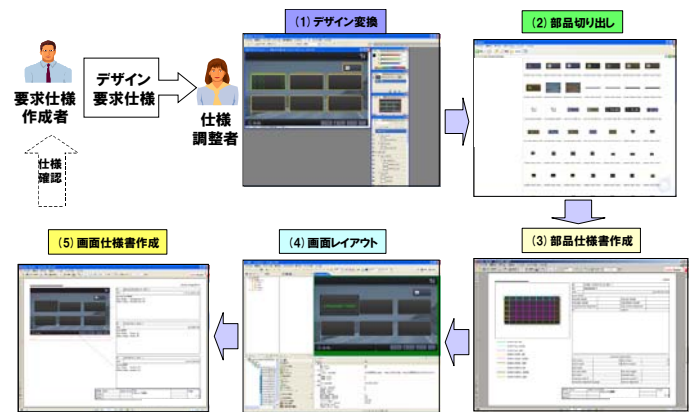


図 4 画面デザイン仕様策定の流れ
Fig.4 Flow of creating screen design specifications

以上の変換を行う際に、画像データ変換ツール内でレイヤー名命名規則とモデルへの変換表を定義し、それに従って変換することで、Photoshopの仕様に依存しないモデル化が実現できる。一般的には、定義すべき部品情報は製品種類や世代、派生によって異なることが多いため、この変換表そのものをプロジェクト単位で管理する方法が有効である。

4. 有効性評価

画面デザイン仕様策定における、提案ツールの工数削減効果を示す。測定は2名で行い、1画面当たりの平均時間を計った。以下、図 4に示す画面デザイン仕様策定の流れに従って説明する。

(1) デザイン変換

要求仕様策定者が作成したデザイン要求仕様を、仕様調整者が受け取る。組込みシステムの場合色数に制限あることが多いため、デザイン要求仕様からの減色、色補正をここで行う。提案ツールではこの作業工数は削減できない。測定の結果22分であった。

(2) 部品切り出し

画像データ変換ツールによって画像部品への変換を行い、リソース管理DBに登録することで、作業が自動化される分工数が削減する。測定の結果8分であった。

(3) 部品仕様書作成

仕様書ジェネレータにより、部品仕様書を自動生成することで、提案ツールによって作業が自動化される。測定の結果1分であった。

(4) 画面レイアウト

登録された画像部品を元に、レイアウトエディタを用いて必要な画面数分のレイアウトを行い、リソース管理DBに登録する。提案ツールにおいても、この作業工数は削減できない。測定の結果49分であった。

(5) 画面仕様書作成

仕様書ジェネレータにより、画面仕様書を自動生成することで、提案ツールによって作業が自動化される。測定の結果1分であった。

比較対象として2005年の開発事例を調査した。290枚の画面デザイン仕様書作成について、部品仕様書と画面仕様書を市販のドローツールで作成しており、1人1枚当たりの平均時間は、(1)(2)(3)を合わせて66分、(4)(5)を合わせて199分

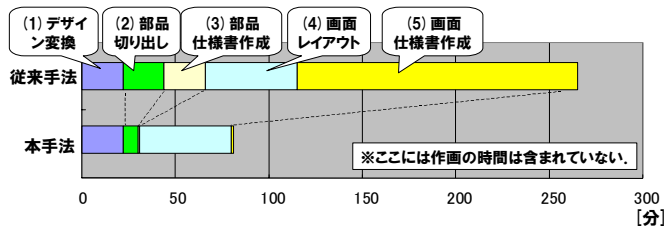


図 5 画面デザイン仕様策定の工数削減効果見積もり
Fig.5 Estimation of efficiency improvement of creating screen design specifications

であった。これらを纏めると図 5 のようになる。要求仕様の画面仕様書化にかかる工数が7割近く削減できることから、仕様調整のTAT(Turn Around Time)の短縮の見通しが得られた。

5. まとめと今後の課題

本稿では、組込みシステムの特徴を踏まえ、モデルベースによる HMI ツールの開発手法を提案した。実際の組込みシステムの HMI 設計情報をプロトタイプ適用し、モデル化を通じて開発工数削減の見積もりと、モデル化工数そのものの負荷に関する見通しが得られた。また、多様な要求仕様に対応するために、市販ツールの活用だけではなく、モデルベース設計による内作ツール開発の有効性が見えてきた。

今後は、製品開発全体を見据えた工数削減を目指し、モデルの進化によるツールへの影響の解析、開発フェーズにおける設計情報の検索性能を考慮したリソース管理 DB の評価、HMI 設計情報と実装ソースコードの間のギャップの解析を行う予定である。

【謝辞】

これまでモデルベース HMI ツールの研究・開発に従事してくださった皆様に深謝する。

【文献】

- [1] Carsten Bock: Model-Driven HMI Development: Can Meta-CASE Tools do the Job?, Proc. of HICSS (2007).
- [2] Laci Jallis, Frank Szczublewski: AMI-C Content-Based Human Machine Interface (HMI), Proc. of SAE (2004).
- [3] Frank Budinsky et al. : Eclipse Modeling Framework: A Developer's Guide, Prentice Hall Ptr (2003).
- [4] Eclipse, <http://www.eclipse.org/>.
- [5] J Clarck, M Murata: RELAX NG Specification, <http://www.relaxng.org/spec-20011203.html> (2001).
- [6] RFC2068 Hypertext Transfer Protocol - HTTP/1.1, <ftp://ftp.isi.edu/in-notes/rfc2068.txt>.
- [7] Photoshop, <http://www.adobe.com/jp/products/photoshop/>.
- [8] JavaScript Scripting Reference, Photoshop CS2/スクリプティングガイド/JavaScript Reference Guide.pdf.

新 吉高 Yoshitaka ATARASHI

(株) エイチ・シー・エックス主任技師。大阪大学大学院情報科学研究科博士後期課程在学中。1997 大阪大学大学院工学研究科博士前期課程修了。車載情報システムの研究・開発に従事。情報処理学会正会員。日本データベース学会学生会員。

鯨井 俊宏 Toshihiro KUJIRAI

(株) 日立製作所中央研究所研究員。1997 東京大学大学院工学系研究科修士課程修了。音声認識及び車載情報システムの研究・開発に従事。日本音響学会正会員。

土井 敬司 Keiji DOI

(株) 日立製作所日立研究所研究員。2002 東京大学大学院工学系研究科修士課程修了。車載情報システムの研究・開発に従事。

深谷 直彦 Naohiko FUKAYA

(株) 日立製作所中央研究所企画員。2004 名古屋大学大学院工学研究科博士前期課程修了。車載情報システムの研究・開発に従事。情報処理学会正会員。

原 隆浩 Takahiro HARA

1997 年大阪大学大学院工学研究科博士前期課程修了。同年、同大学院工学研究科博士後期課程中退後、同大学院工学研究科助手。同大学院情報科学研究科助手を経て、2004 年より同大学院情報科学研究科准教授となり、現在に至る。工学博士。データベースシステム、分散処理に興味をもつ。IEEE, ACM, 情報処理学会、電子情報通信学会、日本データベース学会の各会員。

西尾 章治郎 Shojiro NISHIO

1980 年京都大学大学院工学研究科博士後期課程修了。工学博士。京都大学工学部助手、大阪大学基礎工学部および情報処理教育センター助教授、大阪大学大学院工学研究科教授を経て、2002 年より大阪大学大学院情報科学研究科教授、2007 年より大阪大学理事・副学長となり、現在に至る。2000 年より大阪大学サイバーメディアセンター長、2003 年より2007 年まで大阪大学大学院情報科学研究科長を併任。データベース、マルチメディアシステムの研究に従事。現在、Data & Knowledge Engineering 等の論文誌編集委員。本会理事を歴任。電子情報通信学会、情報処理学会フェローを含め、ACM, IEEE など 9 学会の会員。