特集論文

組み込み機器向けUI設計ツール“Edamame”

中川隆志*
岡本啓嗣**
小中裕喜***

User Interface Development Tool for Embedded Systems "Edamame"

Takashi Nakagawa, Hirotsugu Okamoto, Hiroki Konaka

要　旨

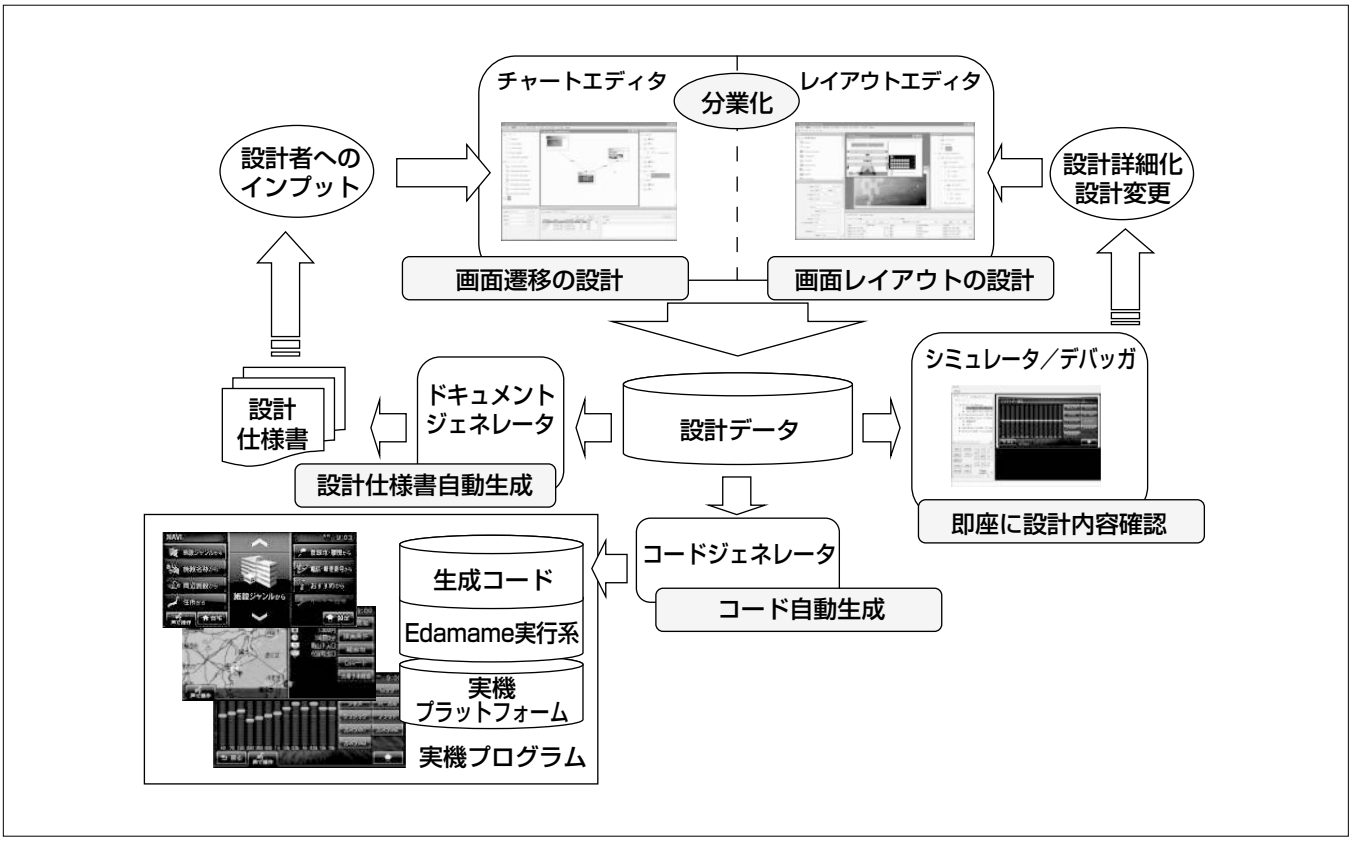
近年，カーナビや家電機器など様々な組み込み機器におけるUser Interface（UI）の高度化，開発期間の短縮に伴い，組み込みソフトウェア開発におけるUI開発の生産性向上が重要な課題となっている。三菱電機では，この課題解決をねらった開発環境として，UI設計ツール“Edamame（Embedded system Design Architecture with Model-based Approach and Middle-ware Environment）”の開発を行っている。

Edamameは，その前身である“NINA（Navigator for Interface of Application）”から，SCO（State Chart Object）というUI設計データの部品化による再利用を促進するための機構を引き継ぎ，さらに，年々肥大化するUI設計仕様に対応するため，分業によって同時並行的な開発を可能とする設計モデルへの拡張，及び連続的な動きや透明から徐々に浮かび上がるといった高度な表現機能を実現したものである。

Edamameによって大規模なUI設計の再利用性を高めるためには，UI設計データの部品化の粒度や，様々なスキルを持つ作業者間での分業範囲，成果物の受渡しの流れといった開発プロセスをどのように定めるかがポイントとなる。

現在Edamameはカーナビ開発に適用されているが，この開発における部品の再利用回数を測定したところ，部品の再利用が促進されていることが確認できた。

本稿では，UI設計ツールEdamameの機能の概要を述べるとともに，大規模なUI設計での設計モデルと設計の部品化の方針，及び開発プロセスの例について述べる。



UI設計ツールEdamameの構成とUI設計開発の流れ

UI設計ツールEdamameは複数の機能モジュールから構成される。レイアウトエディタではUI画面のレイアウトを設計し，チャートエディタではUI画面間の遷移やUI部品の制御を設計する。次にシミュレータ／デバッガによって，設計したUIの動作を確認し，修正が必要な場合にはエディタに戻り修正する。設計時には，ドキュメントジェネレータによる文書自動生成によって，最新の文書を閲覧することが可能である。コードジェネレータによって設計データから実機用ソースコードが自動生成され，Edamame実行系と結合することで，実機用のUIソフトウェアが完成する。

*先端技術総合研究所（エネルギー科学博士）　**同研究所　***三田製作所（工博）

1. まえがき

近年，カーナビや家電機器など様々な組み込み機器におけるUIの高度化，開発期間の短縮に伴い，組み込みソフトウェア開発におけるUI開発の生産性向上が重要な課題となっている。当社では，この課題解決をねらった開発環境として，UI設計ツールEdamameの開発を行っている(1)(2)。

Edamameは，その前身であるNINA(3)から，UI設計データの部品化による再利用を促進するための機構を引き継ぎ，さらに，年々肥大化するUI設計仕様に対応するため分業によって同時並行的な開発を可能とする設計モデルの拡張，及び連続的な動きや透明から徐々に浮かび上がるといった高度な表現機能を実現したものである。

本稿では，UI設計ツールEdamameの概要と，大規模なUI設計での設計モデルと開発プロセスの例，及びこれらの適用効果について述べる。

2. UI設計ツール Edamame

2.1 Edamameの特長

Edamameでは，設計データの再利用の促進及び，開発の効率化を実現するために次の特長を備えている(**扉図**)。

(1) 設計データの部品化

巨大なひとかたまりの設計データを，小さな独立した設計データのかたまり(部品)の組合せとして構築する。また，この部品をユーザーが自在に作ることを可能とする。

個々の部品が持つ機能や振る舞いを単純で明確なものにすることで，部品単位での再利用を図ることが容易となる。これらの部品は，チャートエディタ，レイアウトエディタによって構築される。

(2) モデル設計と設計文書の自動生成

部品はステートチャートに基づく図や，レイアウト図として設計する。このように設計データを図的に定義することで，その部品の振る舞いを，設計者本人以外でも容易に把握することができる。Edamameではドキュメント生成機能によって，設計データから設計仕様書のような設計文書を自動生成することができる。この機能は，設計者にとっては，従来の手書きによる文書作成の手間が省けるというメリットだけでなく，設計者本人以外にとって，実際の設計と完全に一致した設計文書がいつでも参照可能となり，部品の再利用に大きな効果を上げることができる。

(3) 設計即実行

シミュレータによって，設計データは部品単位で設計直後にその場で実行し動作を確認できる。これによって設計の誤りをいち早く発見することができる。Edamameでは，コンパイルやリンクといった従来のプログラミングで必要であったプロセスを経ることなく，即座にシミュレータで実行することが可能なため，設計－実装－デバッグといった繰り返しを効率よく実施できる。また，デバッガを使うと，あとで述べるSCO内のイベントハンドラを対象に，ブレイクポイントの配置やステップ実行，変数の参照・編集も可能である。

(4) コード生成とEdamameランタイム

Edamameでは，設計情報から，実機上で動作可能なコンパクトなソースコードを生成できる。生成したコードは，C++言語で記述されたEdamameランタイムライブラリと結合して，Edamame上のシミュレータの動作と完全に一致した挙動を再現できる。

このEdamameランタイムライブラリは，Windows(注1)，Linux(注2)など多様なプラットフォームに対応している。

(注1) Windowsは，Microsoft Corp.の登録商標である。
(注2) Linuxは，Linus Torvalds氏の登録商標である。

3. 設計モデルと開発プロセス

UI設計は，設計作業の面から見ると，画面デザインと制御ロジック設計の二つに大きく分類できるため，Edamameではこのような画面デザインと制御ロジック設計の成果物を，レイアウト部品とSCOの二つのタイプの部品として構築する。

次に，レイアウト部品，SCOのそれぞれの特性と，大規模なシステムにおける開発プロセスについて述べる。

なお，画面デザインの設計者を画面デザイナ，制御ロジックの設計者をプログラマと呼ぶ。

3.1 レイアウト部品

画面デザインを定義した部品をレイアウト部品という。レイアウト部品には，テキストやピクチャーといったEdamameにあらかじめ組み込まれている数種類の基本部品や，他のレイアウト部品，あとで述べるUI-SCOを配置することができる。**図1**にレイアウトエディタを使ったレイアウト部品の設計例を示す。

レイアウト部品は，部品サイズに応じた矩形(くけい)の上に，直接他の部品を配置することによって，その部品の見映えを直感的に，かつ，特別なプログラミングスキルを持つことなく定義することができる。

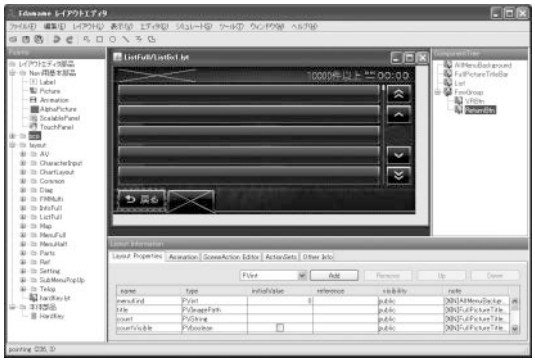


図1．レイアウトエディタの画面例

Edamameでは，頻繁に利用される定型的な表示効果を，プログラミングすることなく実現できる機構（アニメーション）を設けている。レイアウト部品には，外部から設定可能なプロパティを設けることができるが，このプロパティ値に応じて，そのレイアウト内部に配置されている部品を指定の位置に移動させたり，色を変えたり，変形させたり，透過度を変えるといった，プロパティ値とその値における制御内容をアニメーションとして複数定義可能である。

さらに，プロパティ値を時々刻々変えるタイムライン定義が可能である。このアニメーションとタイムライン定義を組み合わせることによって，画面の下部から上りながらスライドインしてくるメニューや，徐々に透明になって消えていくボタンなどの高度な表現を，レイアウト部品単体で実現可能である。

3.2 SCO

複数の表示状態と，それらの間の遷移を設計可能なUI部品である，UI-SCOの概念を**図2**の上部に示す。

UI-SCOは，ユーザーが定義可能な動きのある表示部品であり，複数の表示状態を持つことができる。例えば，**図2**で，UISco1は，選曲，再生の二つの表示状態を持っている。選曲，再生のそれぞれの表示状態には，Layout3やLayout4といったレイアウト部品を保持している。UISco1は，ユーザーの操作やシステム内部の状態の変化などのイベントを受け，レイアウト部品を切り替えることによって，部品の見え方を制御している。このようなイベントを受けて表示状態を切り替えるものをイベントハンドラといい，このイベントハンドラには，簡単なスクリプト（プログラム）を定義することも可能である。SCOを定義するためには，チャートエディタを用いる（**図3**）。

一方，UI-SCOと異なり，各状態における表示部品を保持せず，SCO全体で一つのレイアウト部品の制御を行う制御部品としてControl-SCOがある。**図2**の下部にControl-SCOの概念図を示す。Control-SCOは，先のUI-SCOと同様にイベントハンドラを保持できることから，イベントを受けて，制御対象であるレイアウト部品に対して様々な表示制御を行うことが可能となる。**図2**におけるControlSco1は，Layout1に対する制御を行っており，停止中，再生中のそれぞれの状態が変わる際に，イベントハンドラに定義されたLayout1の画像の変更や文字列の変更といった制御を行う。

SCOやレイアウト部品の特長は，設計者が作成した部品を，他の部品に張り込むことができる。例えば，**図2**では，Layout1がLayout4の一部に張り込まれている。このように，画面中の小さな部品から，より大きな表示領域を持つ部品，アプリケーション全体にまで，様々な階層の設計を部品化し，組み合わせることを可能としている。また，複数の画面に配置した部品の機能拡張や障害修正を行うと，その効果はすべての配置先に水平展開される。

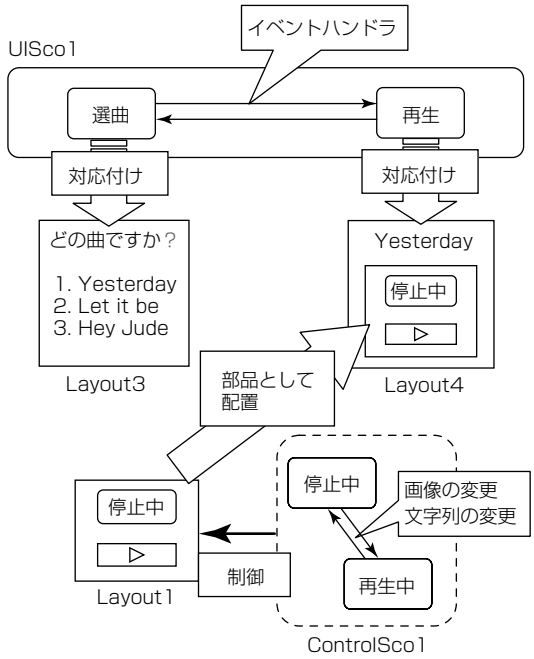


図2．モデル構造の例

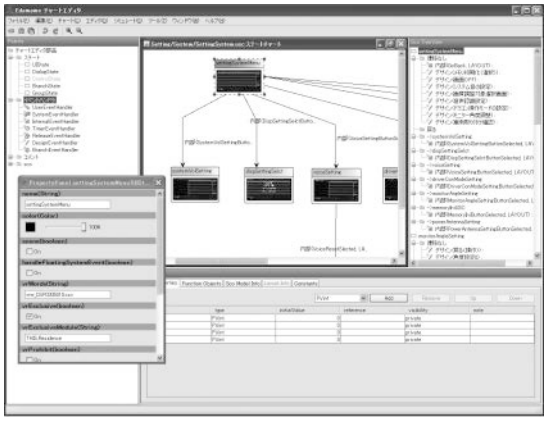


図3．チャートエディタの画面例

3.3 分業のための開発プロセス

UI設計の再利用性を高めるために，UI部品として適切な粒度を定めることと，それぞれの出現頻度に応じた作業担当を定める必要がある。まず，UI部品の粒度に応じて次の4種類の分類を行った。

(1) 小部品

ボタンのように基本的な機能を提供するUI部品。アプリケーション全体で頻繁に出現する。

(2) 中部品

ボタンが縦や横に並ぶリストのような，アプリケーション中のある範囲で共通的に利用されるUI部品。小部品ほどではないものの，しばしば出現する。

(3) 画面部品

待ち受け画面，メニュー画面といった，アプリケーションレベルでの画面を表すUI部品。出現頻度は低い。

⑷ アプリケーション

アプリケーション全体における画面フロー。画面部品がどのような順序で表示されるかを設計する。ナビや，オーディオ，設定といった単位で作成される。

このように粒度から分類されたUI部品を，次のように画面デザイナとプログラマは分担して設計を行う。ここでは**図4**の例に基づき，開発プロセスについて述べる。

⑴ 第1段階

小部品であるボタン部品は，レイアウト部品としては画像とテキストなど数点の部品を保持する単純な部品であるが，これを制御するControl-SCOは，配置箇所に応じて多くの表示バリエーションを実現するなどの汎用(はんよう)性を考慮した設計が必要である。そのため，小部品については，スキルの高いプログラマが開発初期に集中して担当することが望ましい。この段階で，部品単位の挙動をシミュレータで確認する。

⑵ 第2段階

第1段階の成果を利用して，中部品，画面部品を画面デザイナが作成する。中部品，画面部品になると，画面の数に比例して大量に必要になることや，仕様が頻繁に変更されることを踏まえ，あまり汎用性を重視した高度な部品を設計するのではなく，各画面の表示内容に対応したレイアウト部品を量産することが求められる。また，定型的な表示効果については，アニメーションとタイムライン定義を組み合わせて実現する。

⑶ 第3段階

第2段階の成果をプログラマが引き受け，それぞれのレイアウト部品に対応した制御ロジックを，Control-SCOとして定義してゆく。この段階で，画面内の挙動をシミュレータで再現することが可能となる。

⑷ 第4段階

第3段階で作成された画面を，プログラマがUI-SCOの画面間遷移として取り込む。このとき，画面間の遷移時に必要な様々なイベントハンドラの定義も一緒に行い，このUI-SCOをアプリケーションとして作り上げていく。

3.4 Edamame適用による効果

Edamameを大規模なUI設計を必要とするカーナビの開発に複数機種にわたって適用し，再利用性，開発効率について評価した。

この結果，平均1部品当たり3.7回，また，ボタン部品のような汎用性の高い小部品については，100回以上再利用が行われていた。この再利用の回数は各機種でもおおむね同じであった。比較的小さな部品を中心に，効率良く再利用されていることがわかる。

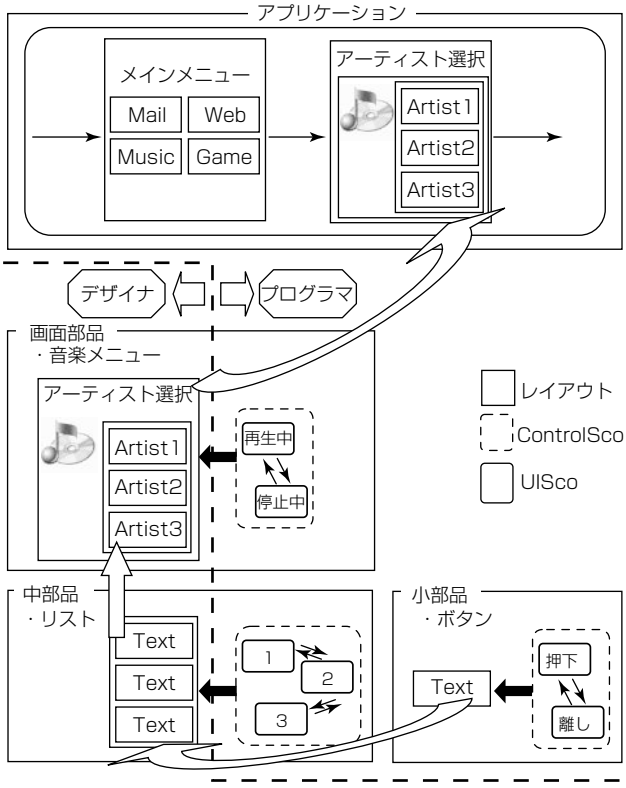


図4．部品の分割と役割分担

さらに，従来のソースコードによる開発と比べると，シミュレータを活用した問題の早期発見による手戻り防止効果も確認できた。

4. むすび

UI設計ツールEdamameの概要と，大規模なUI設計での設計モデルと開発プロセスの例について述べた。そして，カーナビのUI開発にこれらを適用したところ，小さな部品を中心に，効率良くUI設計が再利用されていることを確認した。今後は，カーナビに加え，他のUI設計が高度化するような組み込み機器へ展開を図る計画である。

参考文献

⑴ 岡本啓嗣，ほか：スキルに応じて作業分担可能なUI設計ツール，電気学会全国大会(3)，154～155（2007）
⑵ 岡本啓嗣，ほか：UI設計ツールを用いた開発における効率的運用，情報処理学会FIT2008 第7回情報科学技術フォーラム（2008）
⑶ 小中裕喜，ほか：階層的部品定義に基づく組み込み用UI設計ツール，組み込みソフトウェア工学シンポジウム2002，情報処理学会研究報告，2002－SE－139，7～8（2002）