

デジタルテレビの事例に基づくモデルベース開発導入の検討

Study of Installing Model Based Development Supported by the Case of Digital TV

田代 沙希子[†] 浅田 幸則[†] 大條 成人[†] 松本 紀子[†]

Sakiko TASHIRO, Yukinori ASADA, Shigeto OEDA, Toshiko MATSUMOTO

[†] (株) 日立製作所 コンシューマエレクトロニクス研究所

[†] Hitachi, Ltd., Consumer Electronics Laboratory

1. 緒言

近年の組込みソフトウェアは、高機能化に伴い急速に規模が増大している。一方で開発期間の短縮と開発コストの削減が求められ、開発効率の向上が急務となっている。この状況に対し、ソフトウェアの構造や処理をモデルで表現して設計・検証を行うモデルベース開発(MBD)が注目を集めている^[1]。

本研究では、代表的な組込み機器の一つであるデジタルテレビを対象に、その中でも、機能拡張の繰り返しにより大規模かつ複雑となった状態管理部に MBD を導入する。これにより、今後の状態管理部、さらにはテレビのソフトウェア全体の工数削減を図る。また、組込みソフトウェアの既存資産に対する MBD の導入指針を示す。

2. 概要

2.1. モデルベース開発(MBD)の概要

MBD とは、設計工程でソフトウェアの構成や振舞いをモデルで記述し、そのモデルをコーディング工程、テスト工程で一貫して活用する開発手法である(図 1)。

設計工程では、統一モデリング言語であるクラス図や状態遷移図で仕様を表現する。そのため、設計ミスや設計情報の伝達ミスが減少する。また、コーディング工程では、コードをモデルから自動生成するため、コード作成の工数やコードに混入するエラーが減少する。さらに、テスト工程では、不具合が見つかったとモデルを修正し、修正が終わると再度コードを自動生成する。そのため、設計書であるモデルとコードの整合性を常に維持できる。

上述したように MBD では、モデルを用いた設計がトータルでの開発効率向上に寄与するといえる。

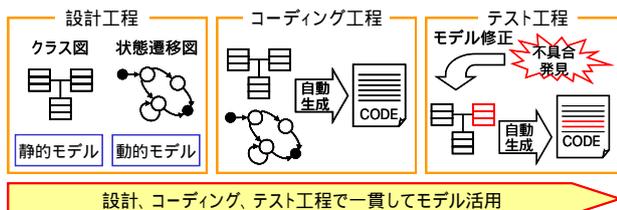


図1 MBDにおける設計、コーディング、テスト工程

2.2. 状態管理部の概要

状態管理部とは、ユーザ操作や予約機能などのトリガ

に基づいて、アプリケーション(アプリ)の起動・終了を制御するミドルウェア層のモジュールである。

たとえばユーザが録画番組一覧を表示させる操作を行った場合、図 2 に示すように、リモコンキーのコード(キーコード)を受信し、起動対象アプリを決定する。次に、内部で管理しているシステム状態を確認し、必要に応じてシステムモジュールに対して、番組再生中の HDD を停止させるなど、システム状態の変更を要求する。そして、最後にアプリモジュールに対してアプリの起動、すなわち録画番組一覧の表示を要求する。状態管理部は他にも番組表、メニューなど数多くのアプリを制御しており、大規模かつ複雑なモジュールと化している。

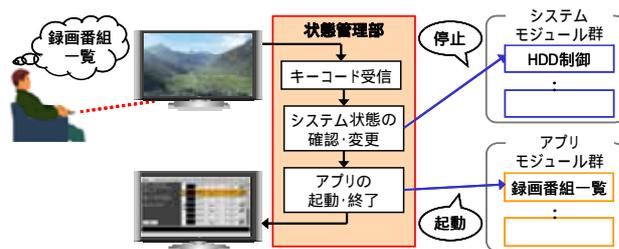


図2 録画番組一覧の表示手順

3. 課題

本研究では、MBD の設計工程に入る前の段階で、MBD の導入準備を行う。これは、現状の状態管理部が複雑で、モデルで表現することが困難なためである。

状態管理部の複雑さは、モジュール内で多様な機能を担っていることと、複数の制御フローが存在することに起因する。よって、モデル化を容易にするには、モジュール構成の最適化と、制御フローの統一が課題となる。

4. モデルベース開発の導入準備

4.1. モジュール構成の最適化

状態管理部では、多様な機能の混在が複雑化の要因となっている。これらの整理と分類を行った結果、性質が異なる 6 種類の機能があることが明らかになった。

複雑度を軽減するには機能分担が有効であると考え、キーコード制御部、アプリ制御部、システム状態制御部の 3 つのモジュールを新設した。また、アプリ制御部とシステム状態制御部が、それぞれアプリとシステムの状

態を一元管理できる独立性の高いモジュールとなるように、6種類の機能を振り分けた(図3)。分割後の新しい構成では、3つのモジュールによって従来の状態管理部と同等の機能を果たす。

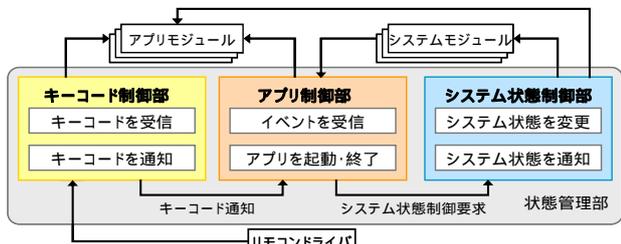


図3 モジュール分割後の状態管理部

4.2. 制御フローの統一

従来のアプリ制御フロー(図4)では、まず、キーコードを受信して、制御対象となるアプリを決定する。次に、現在のシステム状態の確認、システム状態の変更、変更結果の確認を行う。最後にアプリ制御を実行する。

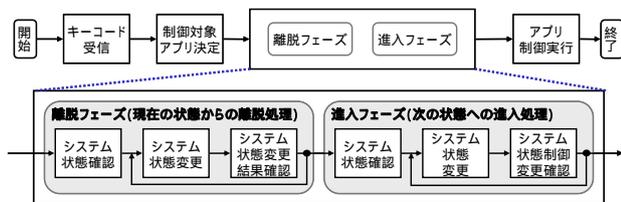


図4 従来のアプリ制御フロー

図中の離脱フェーズと進入フェーズは、それぞれ現在の状態から離脱する処理、次の状態へ進入する処理を目的としたものであり、アプリに応じていずれか一方のフェーズを選択する実装となっていた。また、システム状態を変更するフローもアプリに依存していた。変更が必要となるシステム状態の数はアプリによって異なるが、その数に応じて繰り返し処理することが必要であった。

制御フローを統一するためにはアプリに依存した処理をなくす必要があり、まず、離脱フェーズと進入フェーズの区別を廃止した。次に、フェーズの区別を廃止したフロー中の各処理を、分割した3つのモジュールで分担した。ここで、アプリ制御部からシステム状態制御部に対してシステム状態の変更を要求する際には、変更要求リストを送信するようにした。これにより、複数のシステム状態を変更する場合でも、統一されたフローで処理することが可能となった(図5)。

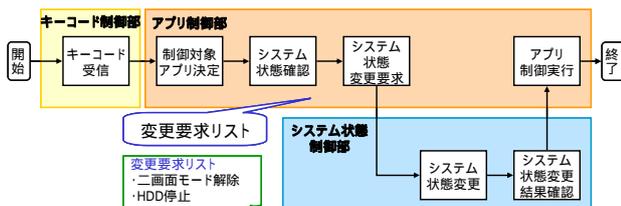


図5 再編後のアプリ制御フロー

5. モデルベース開発の導入

MBDでは、静的モデルのみを使用する場合(図6a)と、それに加えて動的モデルを使用する場合(図6b)とがある。前者では、静的モデルからデータ定義文や関数の宣言文のコードを自動生成し、処理ロジックは開発者が記述する。この場合、既存の処理ロジックを移植できる利点がある。後者では、処理ロジックを動的モデルで記述する。この場合、処理フローのシミュレーションが可能になる。本研究では、一部の処理ロジックが従来と同様であり、既存の処理ロジックを移植することで工数を削減できる。限られた開発期間内で効果を得るため、静的モデルであるクラス図を用いて状態管理部にMBDを導入した。

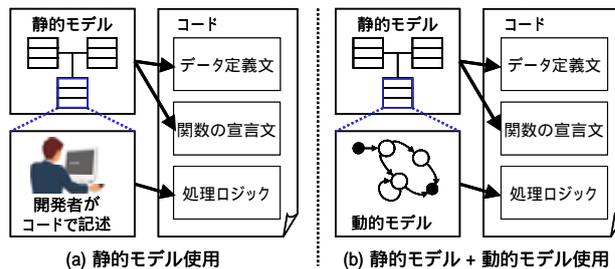


図6 MBDの導入方法

6. 結果と考察

モジュール構成の最適化と制御フローの統一を実現したことで状態管理部の複雑度が軽減され、静的モデルを用いた既存資産へのMBD導入が容易になった。

モジュール構成の最適化は主に不具合解決のスピード向上、制御フローの統一は主に設計ミスの減少に寄与した。この他にもコーディングの作業量が減少するなど、MBD導入により、各過程で工数削減の効果が得られた。

7. 結言

本研究では状態管理部の開発工数削減を目標とし、モジュール構成と制御フローを再編した上で、状態管理部にMBDを導入した。その結果、状態管理部の開発工数を削減し、デジタルテレビ全体のソフトウェア開発工数もMBDの導入により削減できる見通しを得た。

デジタルテレビの事例で示した構成および処理フローを再編してMBDを導入する手法は、既存資産を有する他の組込み機器に対しても応用可能であると考えられる。

参考文献 [1] モデル駆動型開発手法によりソフトウェア開発の効率化と品質の向上を実現: IBM PROVISION, No. 57, pp. 18-24 (Spring 2008)

†(株)日立製作所 コンシューマエレクトロニクス研究所
〒244-0817 神奈川県横浜市戸塚区吉田町 292 番地
E-mail: sakiko.tashiro.bo@hitachi.com