

グラスレス3Dレグザ (REGZA) の開発効率と品質を向上させるソフトウェアプラットフォームの構築

Construction of Software Platform to Improve Development Efficiency and Quality of Glasses-Free 3D REGZA

西岡 竜大 三宅 達也 池田 信之
 ■ NISHIOKA Tatsuhiro ■ MIYAKE Tatsuya ■ IKEDA Nobuyuki

世界初^(注1)のデジタルテレビ (TV) による専用眼鏡なし3D (立体視) 表示機能を実現するために、グラスレス3Dレグザでは、従来のデジタルTVと大きく異なる処理の追加が必要であった。更に、ハードウェア構成の異なる12GL1と20GL1の2機種^(注2)の同時開発や後続の大型化が控えており、機種ごとの開発効率を向上させることが重要な課題であった。

そこで東芝は、従来の2D TVソフトウェアプラットフォームを継承し、後続機種の開発を加速させるグラスレス3D TV向けソフトウェアプラットフォームを構築した。あらかじめ構築したプラットフォームをコア資産とするプロダクトライン型開発方法を取り入れることで、ソフトウェアの個別開発に比べ、開発効率と品質を大幅に向上させることができる。

To offer comprehensive three-dimensional (3D) capabilities without the need for dedicated glasses, the glasses-free 3D REGZA series, the world's first glasses-free liquid crystal display (LCD) TVs, require dramatically different types of additional processing compared with conventional two-dimensional (2D) TVs. Another important issue is improvement of development efficiency for concurrent design of the 20GL1 and 12GL1 models with different hardware engines, and for acceleration of the design of the subsequent lineup of glasses-free 3D REGZA models with new features and large LCDs.

In order to overcome these issues, Toshiba has constructed a new software platform, which is based on that for conventional 2D TVs, using a software product line engineering approach. This software platform makes it possible to improve the development efficiency and quality of glasses-free 3D REGZA models by responding to the various features of different models.

1 まえがき

近年、ソフトウェア開発規模は増大し、その仕様は多様かつ複雑になっている。このため、個々のソフトウェアをつど個別に開発したり、過去の類似製品を流用したりする開発スタイルでは、開発効率が低下するだけでなく品質も低下してしまう。

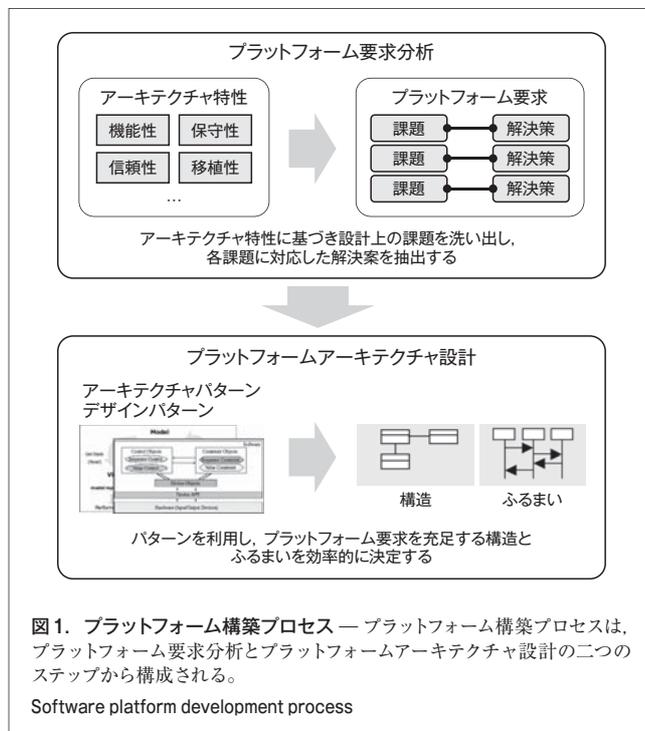
このような問題を解決するにはプロダクトライン型開発⁽¹⁾と呼ばれる方法が効果的である。プロダクトライン型開発では、製品の共通基盤となるソフトウェアプラットフォーム (以下プラットフォームと略記) をコア資産と定義し、プラットフォームをベースとした個別製品開発を行う。プラットフォームをあらかじめ構築することで、個別にソフトウェアを開発する場合に比べ、開発効率と品質を大きく向上させることができる。

東芝は、プロダクトライン型開発手法を適用してデジタルTVのプラットフォームを構築している。ここでは、プラットフォーム構築の概要とグラスレス3Dレグザでの適用事例について述べる。

2 プラットフォーム構築

プラットフォームは要求分析から実装・試験工程までの成

(注1) 2010年10月時点、民生用液晶デジタルテレビにおいて、当社調べ。



果物や開発環境、及び開発ノウハウに関する技術資料を含むコア資産である。プラットフォームを多様な製品ラインアップに適用可能にし、長く利用できるコア資産にするためには、製

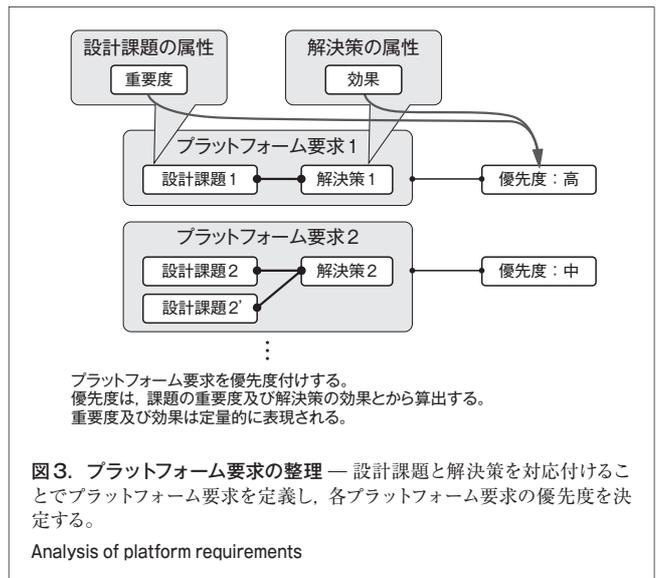
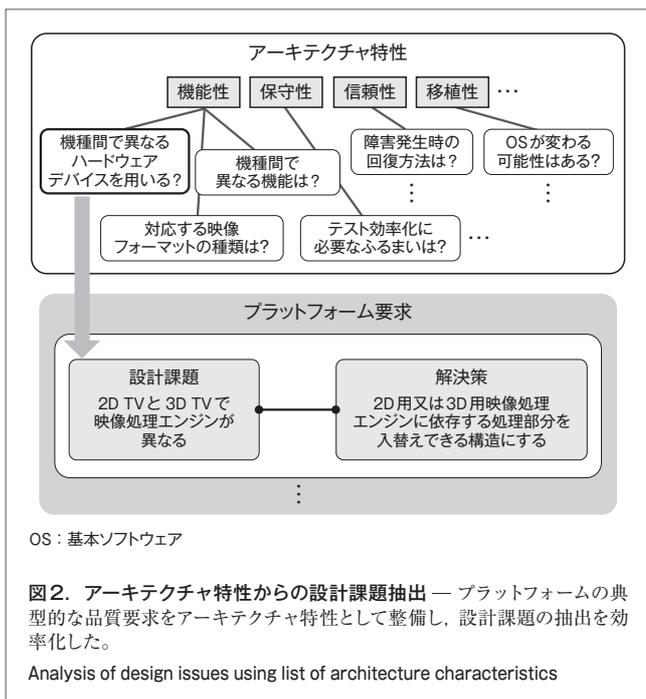
品ロードマップや開発部門のビジネス戦略を考慮してソフトウェアを作り込む必要がある。具体的には、将来発生しうる製品仕様の変更や必要とされる品質に対する要求を予測したうえで、それらの要求を満たす構造を持つプラットフォームを構築する。当社では、体系的かつ効率的にプラットフォームを構築するために、プラットフォーム構築プロセスを開発している。このプラットフォーム構築プロセスは、図1に示すプラットフォーム要求分析とプラットフォームアーキテクチャ設計の二つのステップから構成される。

2.1 プラットフォーム要求分析

プラットフォーム要求分析では、プラットフォームで解決したい設計課題を抽出し、各課題に対する解決策をシステム全体として相互に矛盾なく決定する。

ここで、設計課題とは、プラットフォームで吸収するハードウェア及び機能仕様のバリエーション、あるいは機能性や保守性といったプラットフォームで達成したい品質要求を示す。長期に利用できるプラットフォームを構築するためには、様々な観点から設計課題を洗い出すことが必要であるが、漏れなく抽出するのは困難を伴う。これを解決する手段として、プラットフォームに必要とされる一般的な品質要求をアーキテクチャ特性⁽²⁾として定義し、各特性を洗い出すためのチェックポイントを質問リストとして整備した(図2)。

抽出された各設計課題には、設計する際にどのようなソフトウェアの構造やふるまいが必要かという視点で定義された解決策を関連付ける。解決策は、ある機能を実現する構造やふるまいの選択肢を限定するように記述する。例えば、“2D TVと3D TVで映像処理エンジンが異なる”という設計課題に対

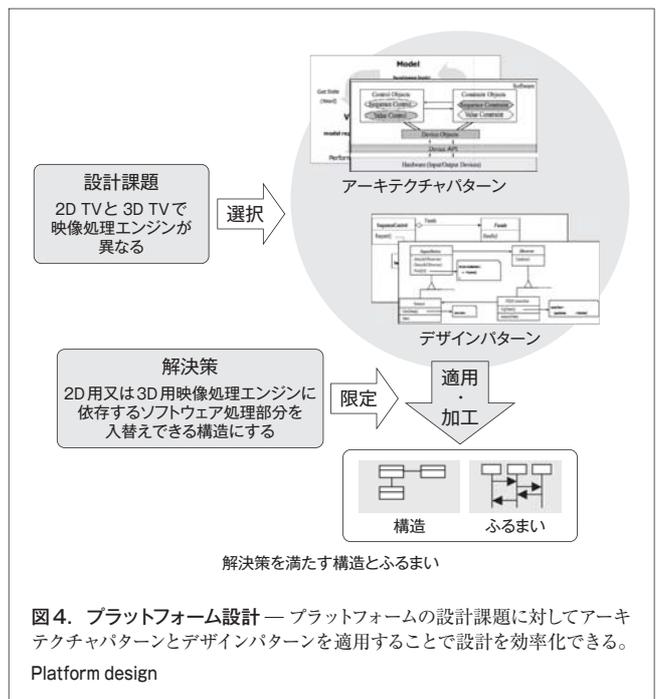


し、解決策は“映像処理エンジンに依存するソフトウェアの処理部分を入替えできる構造にする”という表現になる。

設計課題と解決策のペアをプラットフォーム要求と呼ぶ。全てのプラットフォーム要求は、設計課題の重要度や解決策の効果などから優先度付けされる(図3)。優先度付けの評価尺度を統一することで、システム全体として矛盾なくプラットフォーム要求の優先度を決定することができる。

2.2 プラットフォームアーキテクチャ設計

プラットフォームアーキテクチャ設計では、プラットフォーム要求を実現するための構造やふるまいをソフトウェアの設計として具体化する。



ソフトウェアを設計する際には、機能仕様を満たす構造やふるまいを定義するが、これらの定義には多くの意思決定や取舍選択が必要となり、時間が掛かってしまう。一方で、一般的な設計課題については、アーキテクチャパターンやデザインパターンといったソフトウェアの構造やふるまいの典型例(パターン)を利用することができる。アーキテクチャパターンとは、ある設計課題に対して最適とされるソフトウェアの大局的な設計の典型例、同様にデザインパターンはより局所的な設計の典型例である。

プラットフォームのソフトウェアを設計する際にも、アーキテクチャパターンやデザインパターンを利用することで効率化を図ることができる(図4)。この際、設計課題に適したパターンを選択し、パターンにより提供される設計の例を解決策に基づいて加工し、設計課題に特化した設計として具体化する。

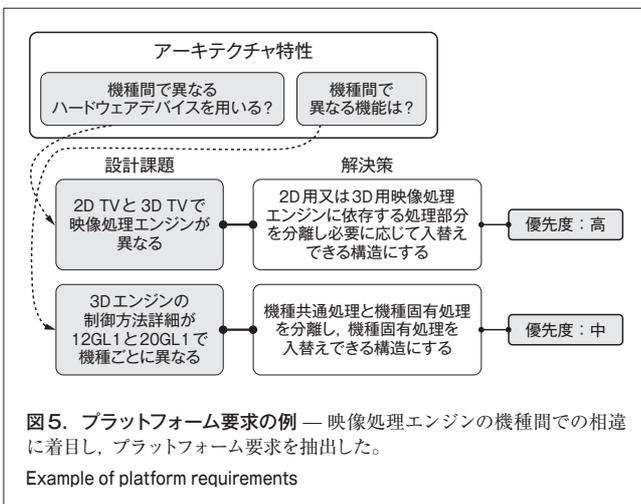
3 グラスレス3Dレグザのプラットフォーム構築

この章ではグラスレス3Dレグザのプラットフォーム構築を例に、プラットフォーム要求分析とプラットフォームアーキテクチャ設計について述べる。

3.1 グラスレス3Dレグザのプラットフォーム要求分析

グラスレス3D TVでは、大項目で30件の設計課題を抽出し、それに対する解決策を関連付けてプラットフォーム要求を抽出した。この節以降、特に重要な設計課題であった“3D化に伴う映像処理エンジンの変更”を例に、作業のステップについて述べる。

グラスレス3Dエンジン制御に関するプラットフォーム要求の例を図5に示す。ここで、アーキテクチャ特性を考慮し、機種間でのハードウェアデバイスの相違から“2D TVと3D TVで映像処理エンジンが異なる”という設計課題を、また、機能の相違から“3Dエンジンの制御方法詳細が機種ごとに異なる”という設計課題を抽出した。



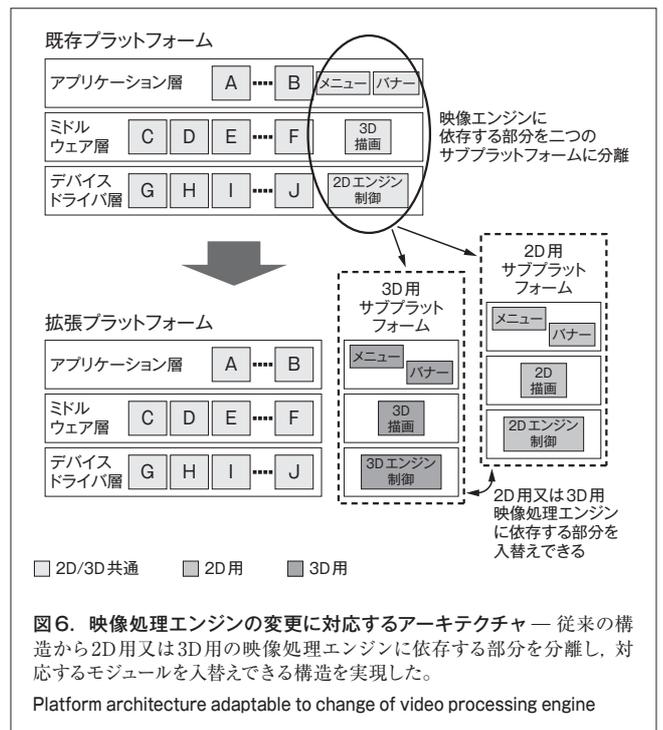
前者に対応する解決策として、“2D用又は3D用の映像処理エンジンに依存する処理部分を分離し、必要に応じて入替えできる構造にする”ことを考案した。また、後者に対応する解決策として“機種共通処理と機種固有処理を分離し、機種固有処理を入替えできる構造にする”ことを考案した。二つの解決策の優先度は、機種間のハードウェアの違いがソフトウェア開発に与える影響を抑制することが機能への対応より重要、と判断し前者をより高く設定した。

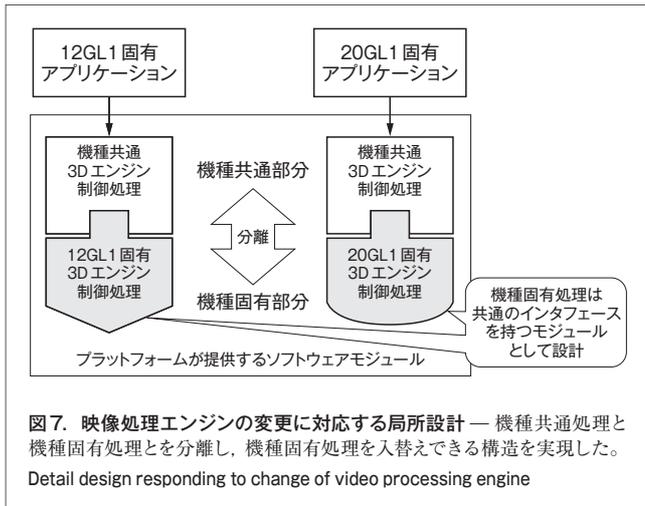
3.2 グラスレス3Dレグザのプラットフォームアーキテクチャ設計

ここでは、前節で示した映像処理エンジンの変更に関するプラットフォーム要求分析を基に行うプラットフォームのアーキテクチャ設計について述べる。

解決策の“映像処理エンジンに依存する処理部分を入替えできる構造にする”について、アーキテクチャの構造を工夫することで対応した。具体的には、まず、既存プラットフォームから2D用又は3D用の映像処理エンジンに依存するモジュールを分離した。次に、2D処理、3D処理それぞれに関係するモジュールを二つのサブプラットフォームとしてまとめ、サブプラットフォームのインタフェースを共通化した。これにより、サブプラットフォームを入替えできる構造を実現した(図6)。

更に、解決策“機種共通処理と機種固有処理とを分離し、機種固有処理を入替えできる構造にする”については、局所的なモジュール構成を工夫することで対応した。具体的には、まず開発対象の2機種(12GL1及び20GL1)の仕様から、共通の映像制御処理を“機種共通3Dエンジン制御処理モジュール”





として設計した。次に、機種固有の3Dエンジン制御処理について、インターフェースを共通化した二つのソフトウェアモジュールをそれぞれの機種に対応付けて設計した(図7)。この構成はデザインパターンの一つであるStrategyパターン^(注2)の応用となっている。これによって機種共通処理と固有処理とを分離し、機種固有の3Dエンジン制御処理を入替えるようにした。機種固有の3Dエンジン制御処理に共通インターフェースを定義する際には、将来の大型グラスレス3Dレグザにも適用できるレベルで抽象化した。

4 プラットフォームの効果

プラットフォームの構築により、グラスレス3Dレグザのソフトウェア開発では様々な効果が得られている。

3章の事例で述べた範囲では、まず、12GL1と20GL1で共通化を実施したことで実装期間を短縮することができた。更に、安定性と応答性の作り込みを共通モジュールに局所化したことで、品質の安定化が図れた。これらの効果により、2機種同時開発の加速を実現した。

また、将来の大型グラスレス3Dレグザの開発で想定される変更を設計課題として抽出したことで、差分開発箇所の削減が期待できる。3Dエンジン制御部分に関しては、機種固有のハードウェア制御部分だけを追加したり変更したりすればよい構造になっている。試算により、3Dエンジン制御全体を改良することに比べて30%の実装行数の変更を削減できると見積もっている。

(注2) コンピュータプログラムの領域において、アルゴリズムを交換可能にすることを目的としたデザインパターン。

5 あとがき

ここではグラスレス3Dレグザに適用したソフトウェアプラットフォーム構築プロセスのうち、プラットフォーム要求分析とプラットフォームアーキテクチャ設計を中心に述べた。今後は、グラスレス3Dレグザの更なる多様化や高機能化に対応するために、プラットフォームの評価と進化を繰り返していく。また、今回のプラットフォーム構築により得られた知見を基に、プラットフォーム構築技術の改善も進める。

文献

- (1) Clements, P.; Northrop, L. ソフトウェアプロダクトライン. 前田卓雄訳. 東京, 日刊工業新聞社, 2003. 652p.
- (2) 西岡竜大 他, “アーキテクチャ指向設計手法ACEの紹介”. 情報処理学会第136回ソフトウェア工学研究会予稿集. 東京, 2002-03, 情報処理学会. p.211-218.



西岡 竜大 NISHIOKA Tatsuhiro

ソフトウェア技術センター プロダクツ開発部主務。
ソフトウェア上流設計, ソフトウェアアーキテクチャ設計の技術開発と製品開発適用に従事。ACM会員。
Corporate Software Engineering Center



三宅 達也 MIYAKE Tatsuya

ソフトウェア技術センター プロダクツ開発部。
ソフトウェア上流設計, ソフトウェアアーキテクチャ設計の技術開発と製品開発適用に従事。
Corporate Software Engineering Center



池田 信之 IKEDA Nobuyuki

ソフトウェア技術センター プロダクツ開発部グループ長。
ソフトウェア上流設計, ソフトウェアアーキテクチャ設計の技術開発と製品開発適用に従事。情報処理学会会員。
Corporate Software Engineering Center