組込みシステムのアーキテクトとモデリング

2016年11月11日

パナソニック（株）AVCネットワークス社
四反田秀樹

自己紹介（経歴）　四反田秀樹

- 現業
  - オートモーティブ開発センター
- それ以前
  - ブルーレイ/DVDレコーダ(DIGA)
- 技術経歴
  - 組込みLinuxの専門家
  - ソフトウェアアーキテクト
- Title
  - ソフトウェアプラットホーム総括担当
  - 上級ソフトウェアアーキテクト

本日の内容

- ソフトウェアアーキテクチャとは
- モデリングとは
- アーキテクトと組織

本日の内容

- ソフトウェアアーキテクチャとは
- モデリングとは
- アーキテクトと組織

ソフトウェアアーキテクチャって何？

- IEEE Std 1471-2000の定義
  - The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution
  - コンポーネントとそれらの関連（I/F）から構成されるシステムの基本構造
  - 設計上の重要な決定事項・方針

http://www.sei.cmu.edu/architecture/definitions.html　には多くのアーキテクチャの定義が載っている。

理想的なアーキテクチャとは？

- 第３者が理解しやすい
  - １つを理解すれば、残り9個が推測できそれが正しい
  - 部分的理解で変更しても不具合が入らない
  - 再利用しやすい

- 差分開発と規模増大

「全体を理解してから差分開発できる時代は終わった」
「現場は、部分的理解で差分開発をせざるを得ない」

「部分的理解で不具合が入らないようにすること」

一人のプロジェクトでは要らない？

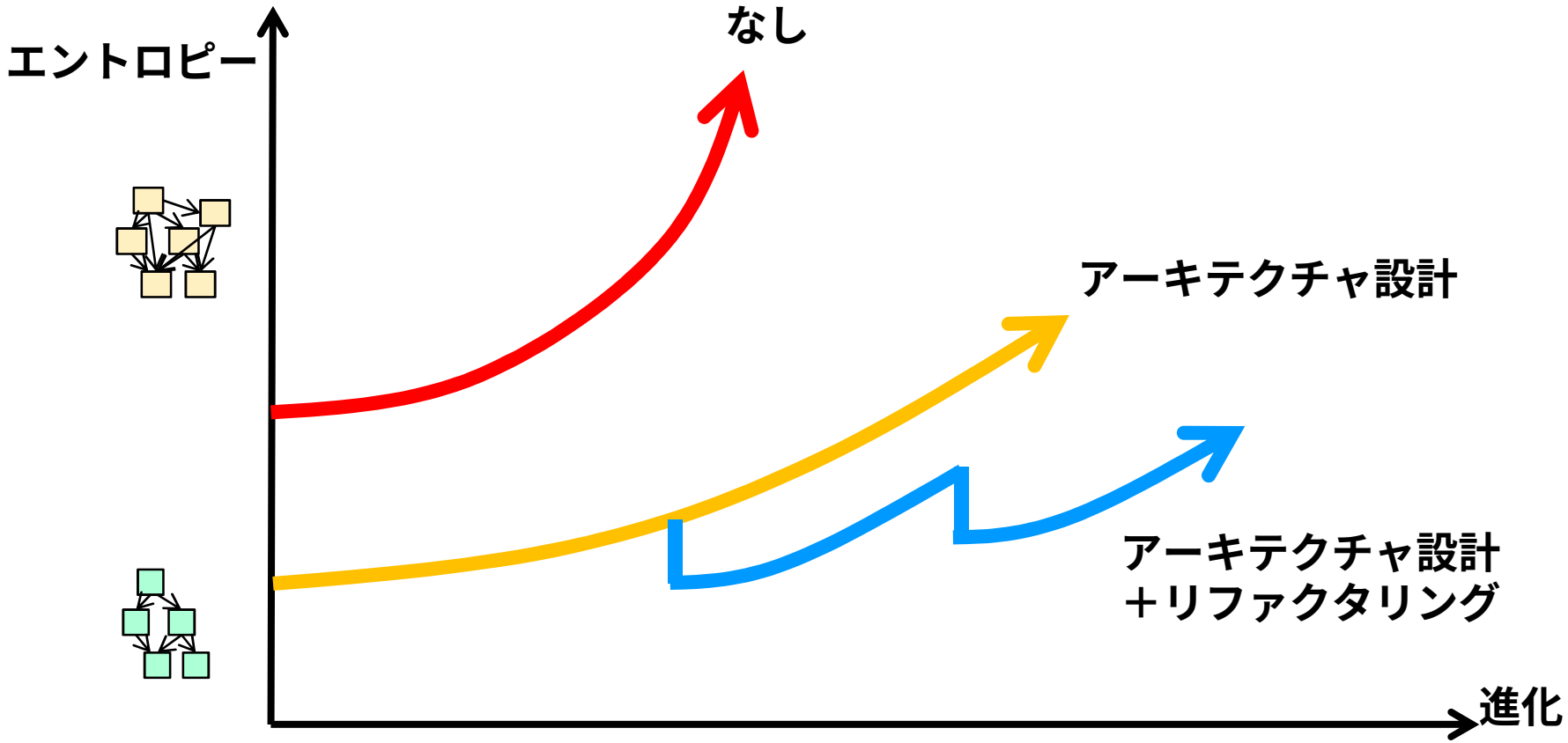
- プロジェクトを第３者に引き継がない？
- でも…３年後のあなたは「第３者」です

何故必要なのか？

- 開発者の支援
  - 作り方に制約を与えることで
    - 複雑さの管理（同時に考える規模を小さくする）
    - 整合性の維持（類似コンポーネントの類推を可能にする）
    - コンポーネント間の揉め事を防止
- 再利用・拡張の基礎
  - 機能拡張するときの設計の基礎
    - どのコンポーネントまで再利用し、新規追加したコンポーネントにどこまでやらせるか

ソフトウェアエントロピーは常に増大

エントロピー（乱雑さ）増大を少なくするのが
アーキテクチャ設計とリファクタリング



どこまでアーキテクチャ設計で行うか？

- ある本の回答
  - 「アーキテクト次第」、アーキテクトが必要と思うところまでがアーキテクチャ設計である

- 私の考え
  - システム設計の一貫性を維持するのに必要な方針/設計
  - 主要なユースケースのシステム設計事例
    - アーキテクチャ設計に従ったシステム設計の事例
  - 矛盾する要件の判断基準
    - 「性能」と「保守性（わかりやすさ）」のトレードオフ境界
    - 外部から導入したソフトの修正方針　等

本日の内容

- ソフトウェアアーキテクチャとは
- モデリングとは
- アーキテクトと組織

モデリングとは？

モデリングというキーワードから出てくるもの

- モデル駆動開発(MDD)
- モデルベース開発(MBD)
- 統一モデリング言語 (UML)

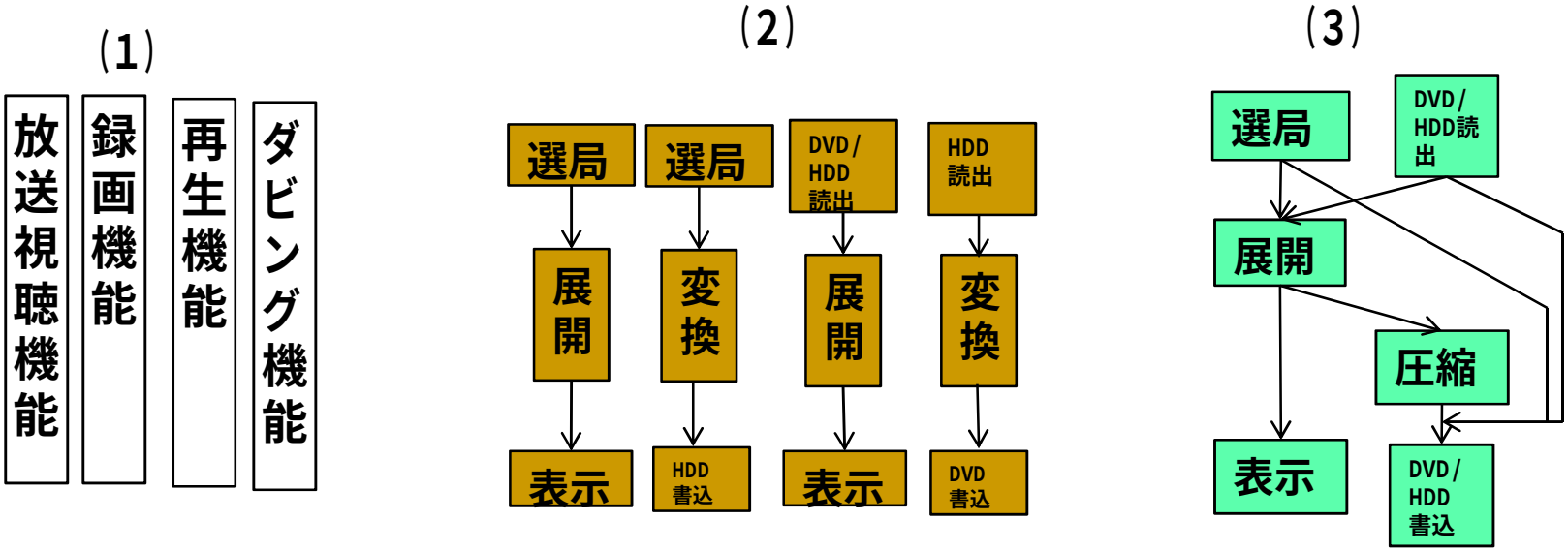
ツール、コード生成、・・・・・

ここで述べるモデリングとは？

アーキテクチャ設計時の「対象の抽象化」の作業

アーキテクチャ設計の中のモデリング

(1) 機能要件を列挙する
(2) 機能を実現するためのブロック構成を考える
(3) (2) で作ったブロックの中で共通的なものを統合する

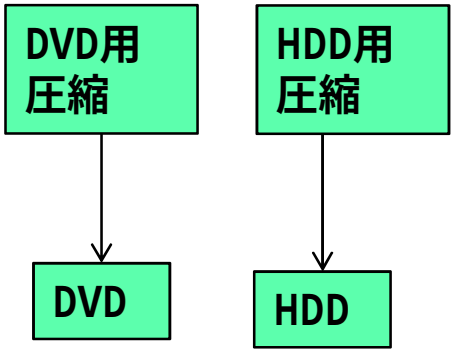


(3) の工程でまとめるときに考えるのがモデリング

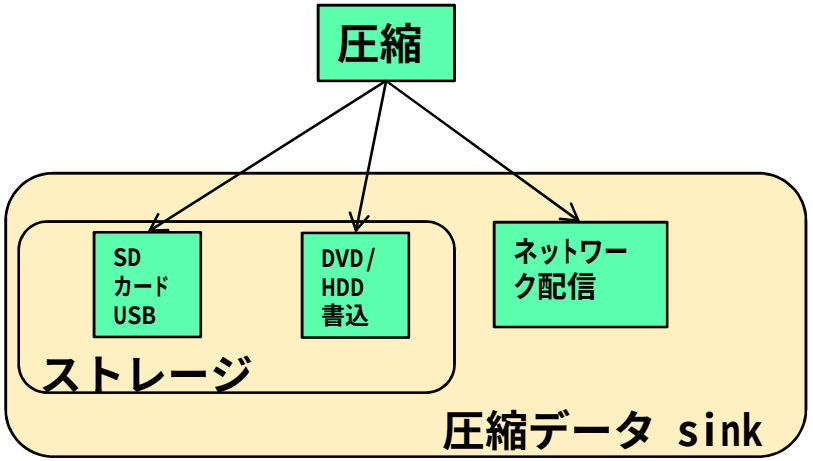
アーキテクチャ設計の中のモデリング

- 工程(3)でうまくまとめられないのは、工程(2)の分割の問題
  圧縮処理で、後段がHDD/DVDという知識を使うと共通化できない
  いいモデル化なら、他のストレージやネットワーク配信に進化できる

- 現実物のものを抽象化したモデルは理解しやすい。



この状態では共通化できない



下段の知識を使わず圧縮が出来ると展開性大

モデリングとは

- モデリングとは
  - 「共通なもの」と「個別なもの」を分離する作業で使う
  - 仮想的なもの（モデル）が持つ特性を定義し、個別部分をパラメータ化し、各ブロックを共通化する

- いいモデルとは
  - 進化方向を予測し、それをモデルに内包している

差分開発でのモデリング

- 前提
  - 既存のアーキテクチャ設計、モデリングが存在
  - 既存のアーキテクチャで（少しの修正で）設計したい

- アーキテクチャ設計
  - 新規要件を既存のアーキテクチャ設計、モデリングにどのようにマッピングするか？
  - どのモデルをより抽象化する必要があるのか？
  - コンポーネントの責務が巨大になりすぎないか？

差分開発のモデリングのコツ

- 手段（仕様）ではなく目的（なにがしたいか）に拘る
  - 既存のモデルに合わせるには「仕様調整」が必須
  - 目的がわからないと「仕様調整」は難しい
- 仕様が制御できること
  - 仕様提案が出来ると差分開発はやりやすい
- 見た目の仕様を直接モデリングしない
  - 仕様開発者は「見た目」に拘る。ここは調整しにくい
- 性能要件変更もアーキテクチャに影響する

差分開発アーキテクチャに影響を与える要素

- 外部から導入したソフト資産
- 開発の順番（過去の経緯）

外部から導入したソフトの影響

- 独立コンポーネントとすることが多い
- 変更ポリシーの選択肢
  - 原則変更しない
    - 不自然な基本構造の一因
  - 必要なら変更する
    - 今後も進化する場合は追従性に課題

開発順番の影響

- 過去は別だったが今は類似商品のものに見られる
  - デジカメとムービー
  - コピー、FAX、プリンタ
  - TVとDVDレコーダ
- アーキテクチャは全然違う

歴史的経緯を理解しないとなぜこのような構造になっているのか理解できない

何故影響をうけるか？

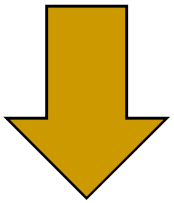
- アーキテクトは新しい機能を既存機能の概念の応用と捉える特性をもつ

動画は静止画の連続記録、再生

静止画は動画のスナップショット

差分開発でのアーキテクチャ設計書の必須項目

アーキテクチャに影響を与えた項目とその理由
特に仕様以外の項目



第3者のアーキテクチャ理解を助ける

本日の内容

- ソフトウェアアーキテクチャとは
- モデリングとは
- アーキテクトと組織

アーキテクトとは何をする人？

ソフトリーダと呼ばれている人の実態は？

リーダの名称	責務
プロジェクトリーダ	プロジェクトの開発進捗を管理
チームリーダ	プロジェクトへの人材割り当てと人材育成
アーキテクト	設計方針を決めコンポーネントの責務を定義、 コンポーネントの設計がアーキテクチャ設計に合致しているかをレビュー

アーキテクトはプロジェクトリーダ/チームリーダとは別の人材

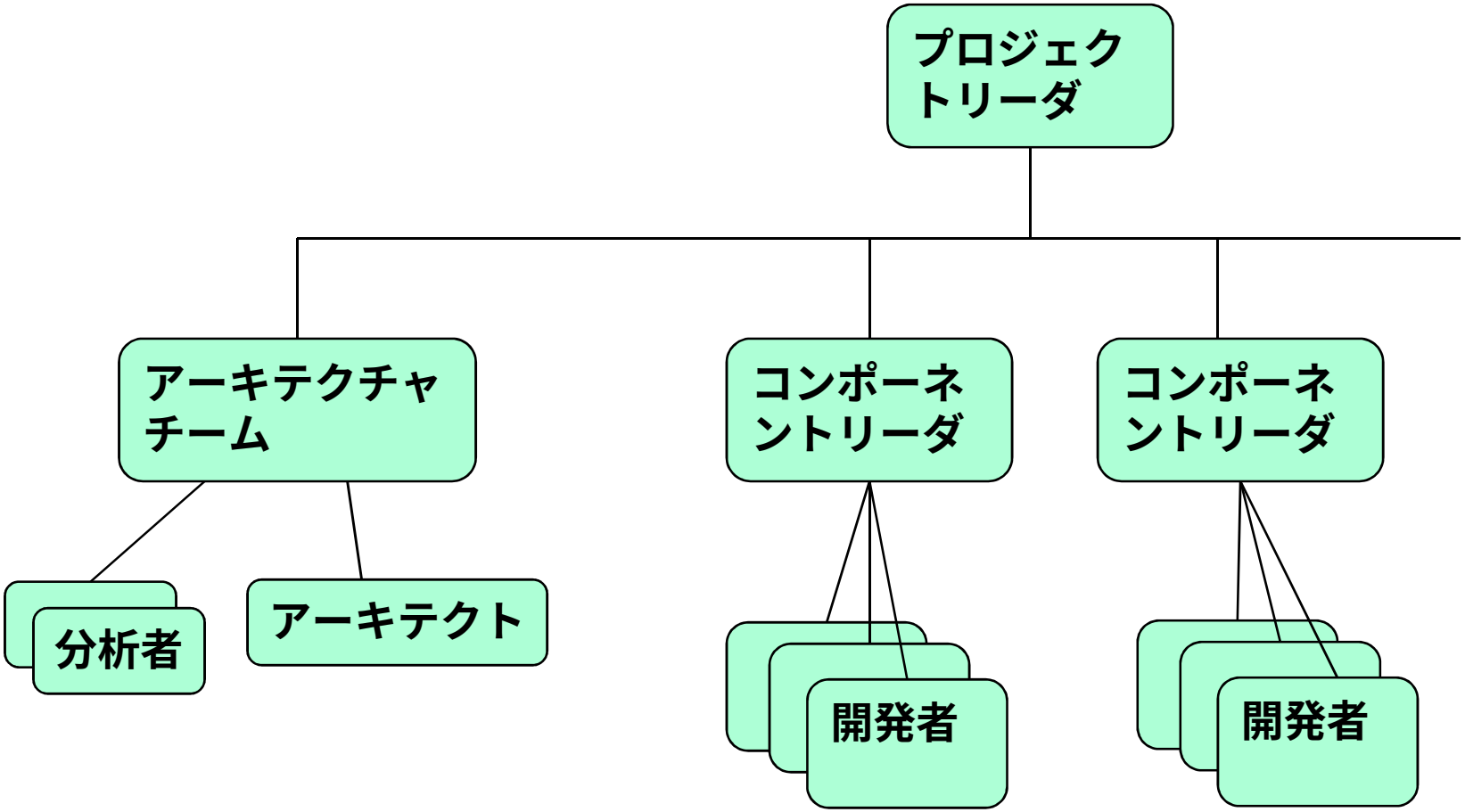
プロジェクトリーダがアーキテクトを兼任すると

- 規模が大きくなるとプロジェクトリーダ優先

  - プロジェクトリーダ不在 →
    進捗管理が出来ない
    プロジェクトが止まる

  - アーキテクト不在　→
    設計をすすめても物は(なんとか）できる
    困るのはあとになってから

プロジェクト構成


プロジェクトリーダ
アーキテクチャチーム
コンポーネントリーダ
コンポーネントリーダ
分析者
アーキテクト
開発者
開発者

アーキテクトに必要な能力

- ドメイン知識
  - 要求を理解し、ソフトウェア構造に分解できること
  - 進化方向を予測できること
- ソフトウェア開発力
  - 複数の機能/処理から、共通と固有を分離する力（モデル化、抽象化）
- 信頼感
  - 重要な判断/決定を行ない、メンバーに浸透させることが出来ること
- コミュニケーション力
  - 決定に対する根拠をうまく説明できる能力（コンポーネント開発者間で利害が対立する事案の決定が多い）
  - 深入りしすぎない

組織に求める事(1)
～アーキテクトが仕事をやりやすくするために～

- プロジェクトリーダとアーキテクトの分離
  - 兼任すると進捗管理しか出来なくなる
- アーキテクトに権限を与える
  - システム設計レビュー、再設計指示権限
    - アーキテクチャ設計を守らなくてもシステムは出来る
    - しかし、継続的にシステムを発展できない
  - アーキテクトが信頼されているのが条件

組織に求める事(2)
～アーキテクトが仕事をやりやすくするために～

- 糸口のつかめない問題の解決を割り当てない。
  - アーキテクトがパンクする
- システム設計をスケープゴートにしない
  - アーキテクトのなり手がいなくなる

組織に求める事(3)
～アーキテクトが仕事をやりやすくするために～

- 適度なコンポーネント間の開発メンバー入替
  - コンポーネント間の押し付け合いの緩和

ご静聴有難うございました