JEITA組込み系ソフトウェア・ワークショップ2012
2012年11月7日

EPSON
EXCEED YOUR VISION

アーキテクチャの定義から読み解く

アーキテクトに大切な3つのこと

～ソフトウェアの過去と未来、そして現在を作る～

セイコーエプソン株式会社

IT推進本部 機器ソフトウェア品質・生産技術部
ソフトウェア生産技術トレーニンググループ

萩原 豊隆

自己紹介

EPSON
EXCEED YOUR VISION

担当業務:ソフトウェア工学の導入推進

- 名前: 萩原 豊隆
- 所属部門
  - IT推進本部 機器ソフトウェア品質・生産技術部
  - ソフトウェア生産技術トレーニンググループ
- 経歴
  - IEEE1394 WDMドライバ開発
  - 業務用小型プリンタのアーキテクチャ設計
- 社外活動
  - 一般社団法人 電子情報技術産業協会(JEITA)
    情報システム・ディスラプティブ技術調査委員会
    ソフトウェアエンジニアリング技術専門委員会 委員

アーキテクトに大切な3つのこと

EPSON
EXCEED YOUR VISION

ソフトウェアの過去と未来、そして現在を作る

1. アーキテクチャとは何か
2. コンポーネントと環境との関係
3. 設計と進化を導く方針
4. コンポーネント相互の関係
5. まとめ

Ⅰ．アーキテクチャとは何か

Ⅰ．アーキテクチャとは何か

Ⅱ．コンポーネントと環境との関係

Ⅲ．設計と進化を導く方針

Ⅳ．コンポーネント相互の関係

Ⅴ．まとめ

Ⅰ．アーキテクトとは何か？

EPSON
EXCEED YOUR VISION

アーキテクチャの

定義にはアーキテクトに大切なことが含まれる

➔ **IEEE1471のアーキテクチャ定義から考える**

1. アーキテクチャの定義
2. コンポーネントと環境との関係
3. 設計と進化を導く方針
4. コンポーネント相互の関係
5. アーキテクチャを作る

1. アーキテクチャの定義

EPSON
EXCEED YOUR VISION

アーキテクチャはシステムの基本構造である

➔ この構造は3つの要素で規定される

アーキテクチャの定義

- 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.
  - コンポーネントを配置または分類することで組織化したシステムの基本的な構造 ③
  - コンポーネント相互の関係 および コンポーネントと環境との関係 ①
  - 設計と進化を導く方針 ②

IEEE Std.1471-2000 Recommended Practice for Architectural Description of Software Intensive Systems

2. コンポーネントと環境との関係

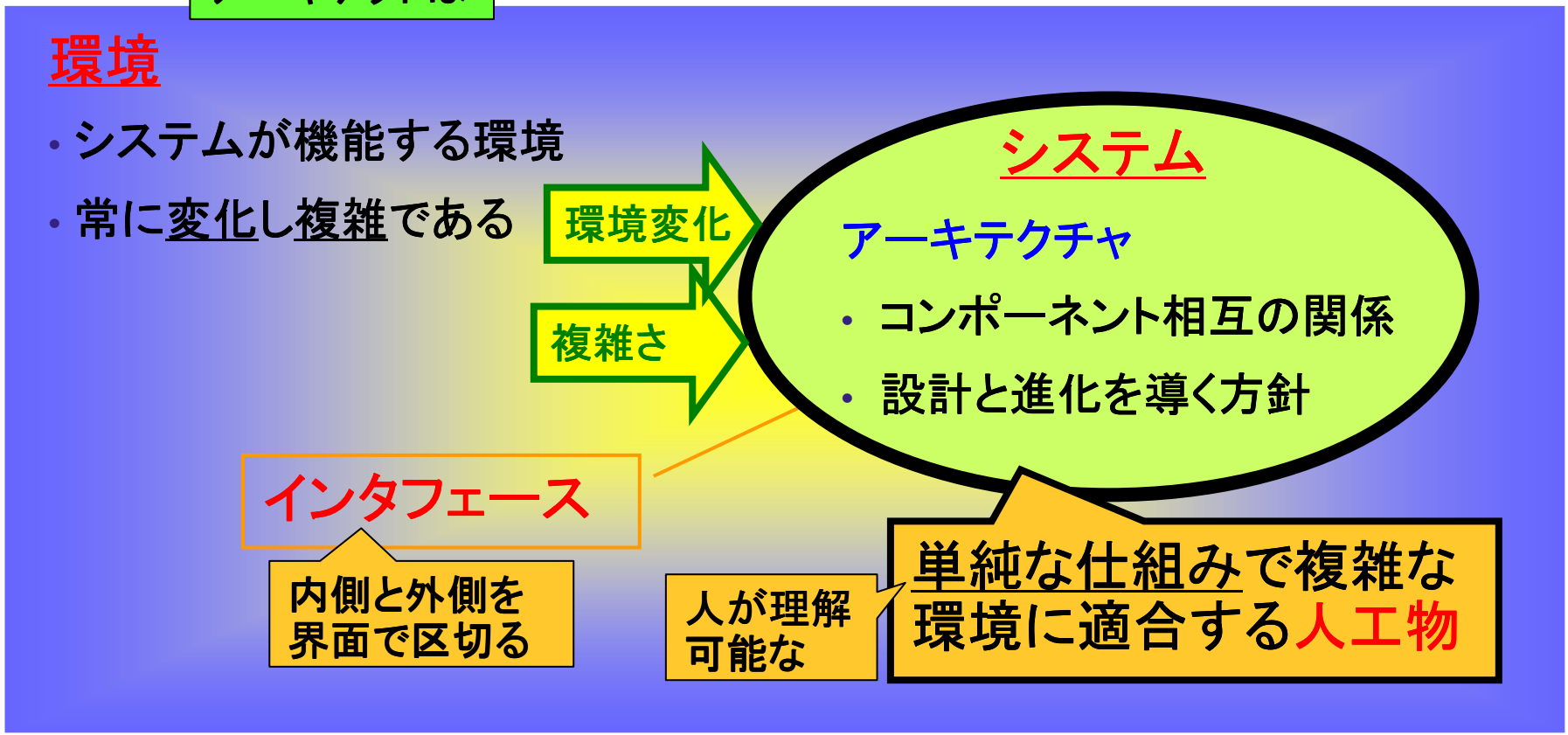
EPSON
EXCEED YOUR VISION

システムの

何になぜ適合したか目論見で語り過去を示す

➔ システムを変化と複雑さに適合させる

アーキテクトは



3. 設計と進化を導く方針

EPSON
EXCEED YOUR VISION

方針を徹底しシステムを未来まで存続させる

➔ システムの原則と方針を徹底させる

アーキテクトは

単純な仕組みで複雑な環境に適合する人工物

システム

アーキテクチャ

- 設計と進化を導く方針

単純な仕組みを作るための設計原則

将来に渡り単純さを保つための再利用方針

アーキテクチャだけでは設計は完結しない。細部までやり切る

原則と方針を共有し全員で実践する

4. コンポーネント相互の関係

EPSON
EXCEED YOUR VISION

ソフトウェアの現在を設計方式で語る

➔ システムを単純な仕組みに抽象化する

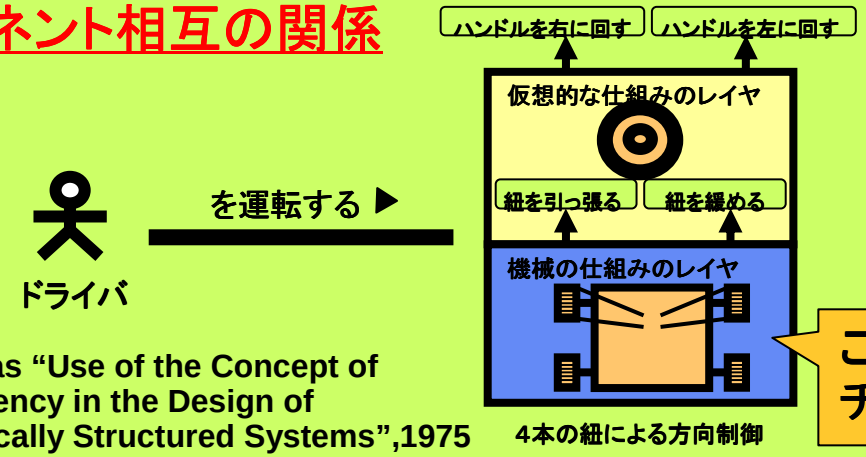
アーキテクトは

単純な仕組みで複雑な環境に適合する人工物

システム

アーキテクチャ

- コンポーネント相互の関係



D.L.Parnas “Use of the Concept of Transparency in the Design of Hierarchically Structured Systems”,1975

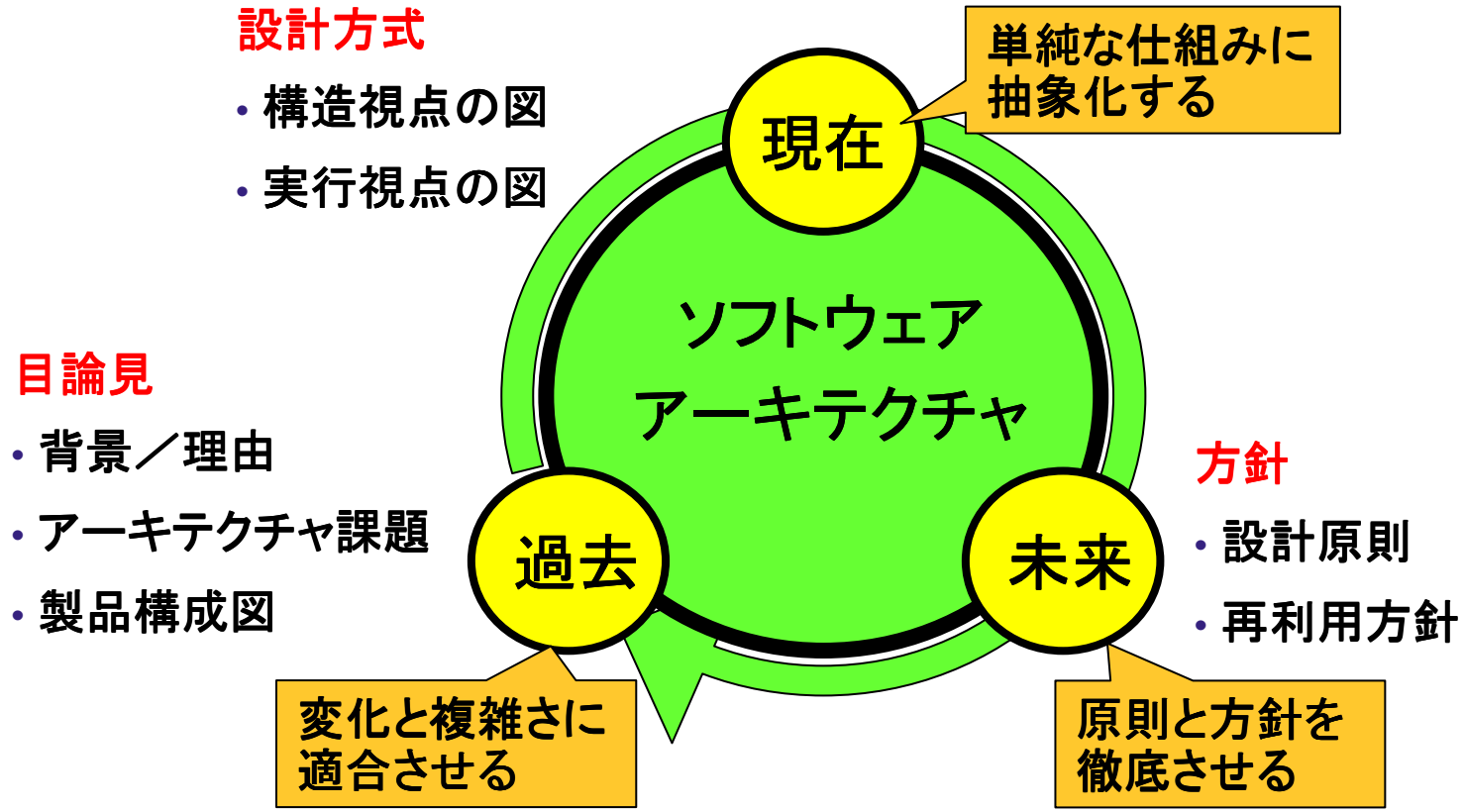
この図がアーキテクチャ設計の肝になる

5. アーキテクチャを作る

EPSON
EXCEED YOUR VISION

アーキテクトが過去から未来へ橋渡しする

➔ ソフトウェアの過去と未来、そして現在を作る



6. まとめ

EPSON
EXCEED YOUR VISION

定義にはアーキテクトに大切なことが含まれる

ポイント

1. アーキテクチャはシステムの基本構造である
2. 何になぜ適合したか目論見で語り過去を示す
3. 方針を徹底しシステムを未来まで存続させる
4. ソフトウェアの現在を設計方式で語る
5. アーキテクトが過去から未来へ橋渡しする

Ⅱ. コンポーネントと環境との関係

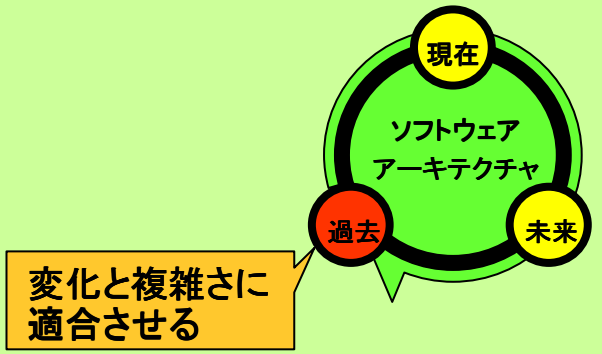
Ⅰ. アーキテクチャとは何か

Ⅱ. コンポーネントと環境との関係

Ⅲ. 設計と進化を導く方針

Ⅳ. コンポーネント相互の関係

Ⅴ. まとめ



Ⅱ. コンポーネントと環境との関係

EPSON
EXCEED YOUR VISION

アーキテクトは変化と複雑さに適合させる

➔ 変化を先取りし、関係者を巻き込み行動する

評論家でなく
リーダである

1. 小型情報機器ファームの事例
2. ビジネス環境の変化
3. 適合を阻む要因
4. アーキテクチャの再構築

1. 小型情報機器ファームの事例

EPSON
EXCEED YOUR VISION

変化する環境にアーキテクチャを適合させる

➔ 腕に装着し、電池で動く環境で機能させる

健康向け：生体情報のセンシングで世の中にないデバイス

ポイント

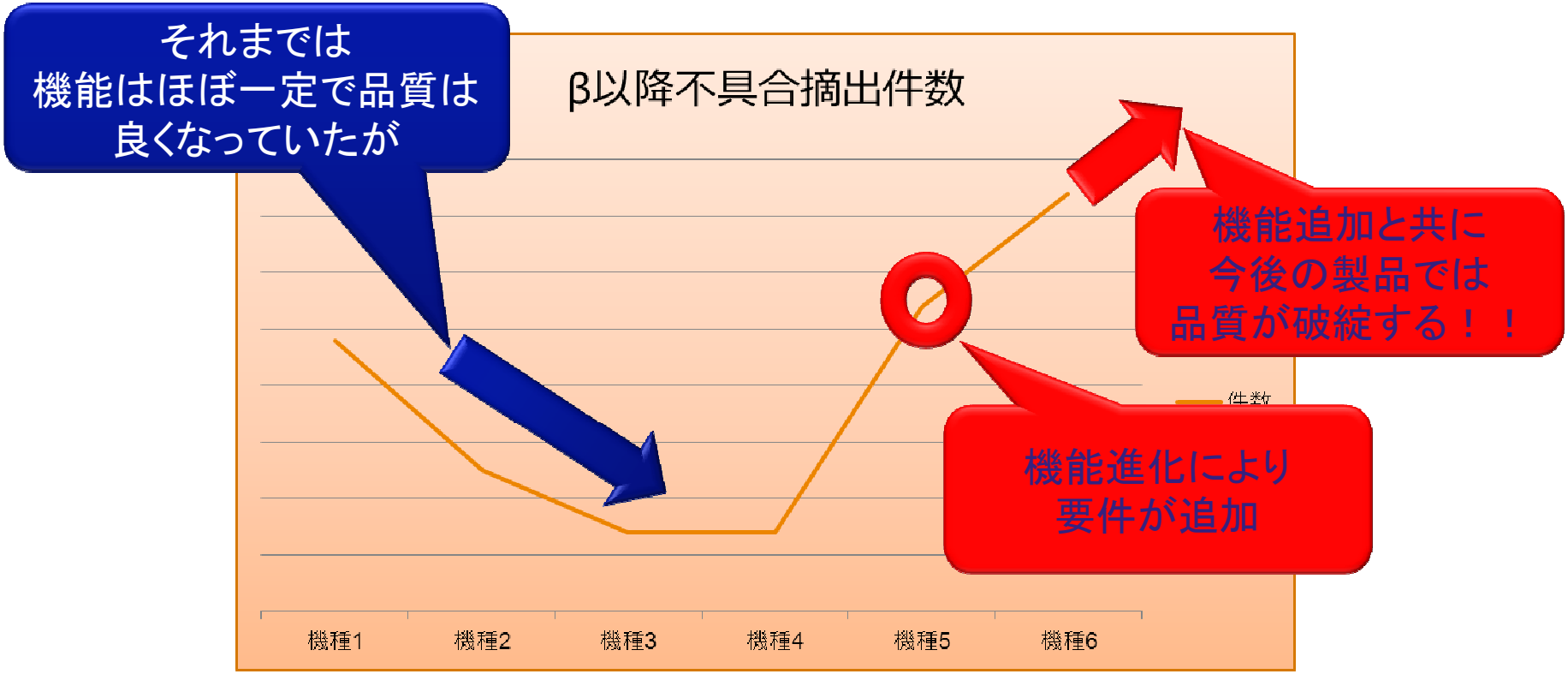
- 電源が電池である
- CPU4/8/16ビット
- RAM 数Kbyte/ROM128KByte
- アセンブラ/C言語のみの開発環境

2. ビジネス環境の変化

EPSON
EXCEED YOUR VISION

機能の進化にアーキテクチャが追いつかない

➔ アーキテクトは変化の兆候を見逃さない

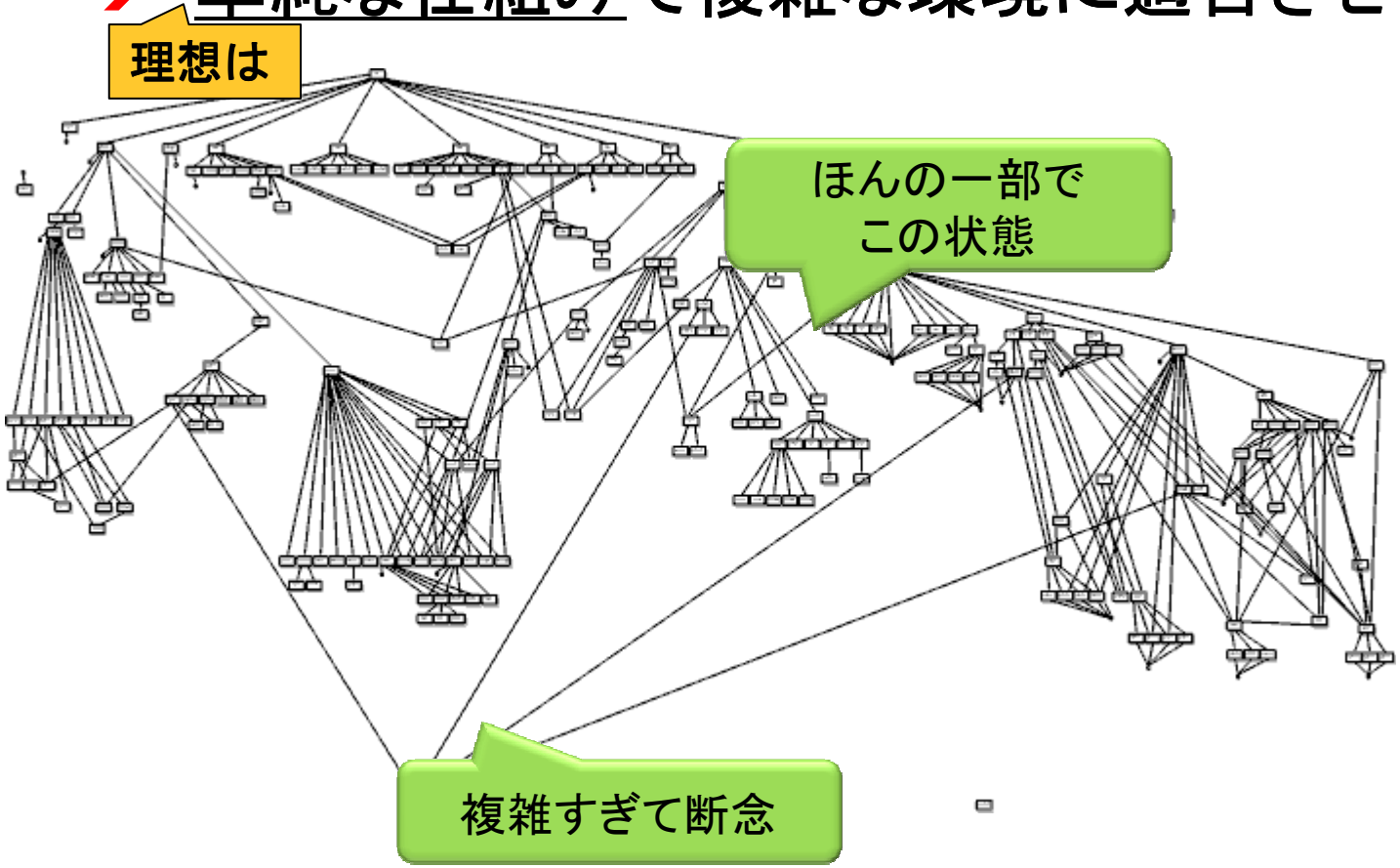


3. 適合を阻む要因

EPSON
EXCEED YOUR VISION

アーキテクチャが崩れ単純さを保てなくなった

➔ 単純な仕組みで複雑な環境に適合させる



4. アーキテクチャの再構築

EPSON
EXCEED YOUR VISION

動くプロトタイプを作って関係者を巻き込む

➔ アーキテクトは技術に立脚するリーダである

対設計者 — 製品系列で適用し維持するため

1. コンパイルして実機で動作をさせる
2. 設計図を公開して、良い設計構造を示す

対マネージャ — 必要なリソースを確保するため

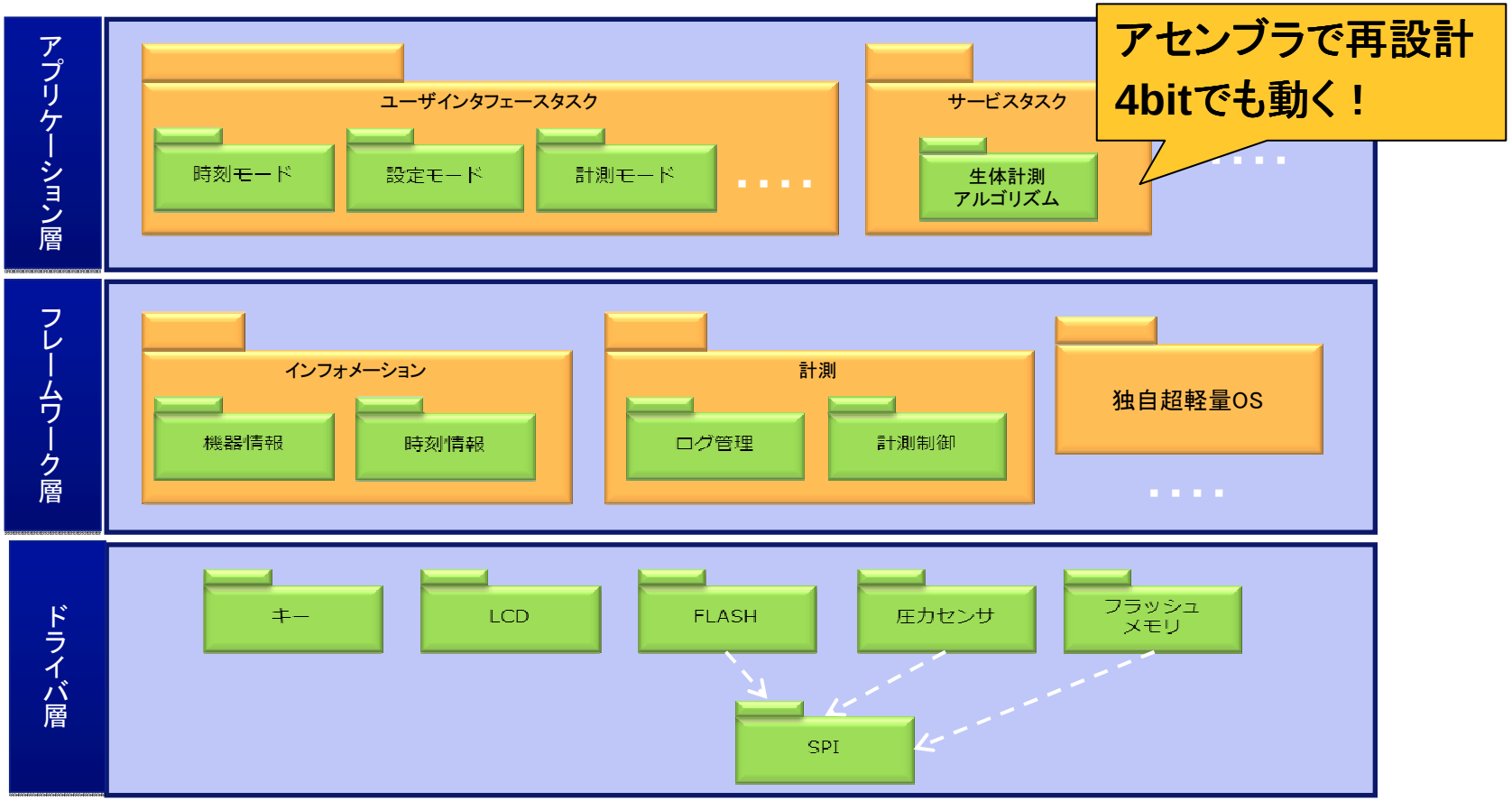
1. 製品群へ適用した際の効果を示す
2. 不具合の原因が解消したことを示す

4.1 納得を得る

EPSON
EXCEED YOUR VISION

机上の空論でないことを示し目論見を達成する

➔ 納得させた上でアーキテクチャを完成させる



5. まとめ

EPSON
EXCEED YOUR VISION

アーキテクトは変化と複雑さに適合させる

ポイント

1. 変化する環境にアーキテクチャを適合させる
2. 機能の進化にアーキテクチャが追いつかない
3. アーキテクチャが崩れ単純さを保てなくなった
4. 動くプロトタイプを作って関係者を巻き込む

Ⅲ. 設計と進化を導く方針

Ⅰ. アーキテクチャとは何か

Ⅱ. コンポーネントと環境との関係

Ⅲ. 設計と進化を導く方針

Ⅳ. コンポーネント相互の関係

Ⅴ. まとめ

Ⅲ. 設計と進化を導く方針

EPSON
EXCEED YOUR VISION

アーキテクトは原則と方針を徹底させる

➔ 全員で共有しやり切りことで設計を完結させる

1. プリンタファームの事例
2. 設計の原則を定める
3. データ抽象を活用する
4. MDD※を導入する

※ MDD (Model Driven Development) : モデル駆動開発

1. プリンタファームの事例

EPSON
EXCEED YOUR VISION

アーキテクチャで束縛タイミングを決める

例えば

➔ 言語や手法で自動的に決まるものではない

ビジネス向けプリンタ

ポイント

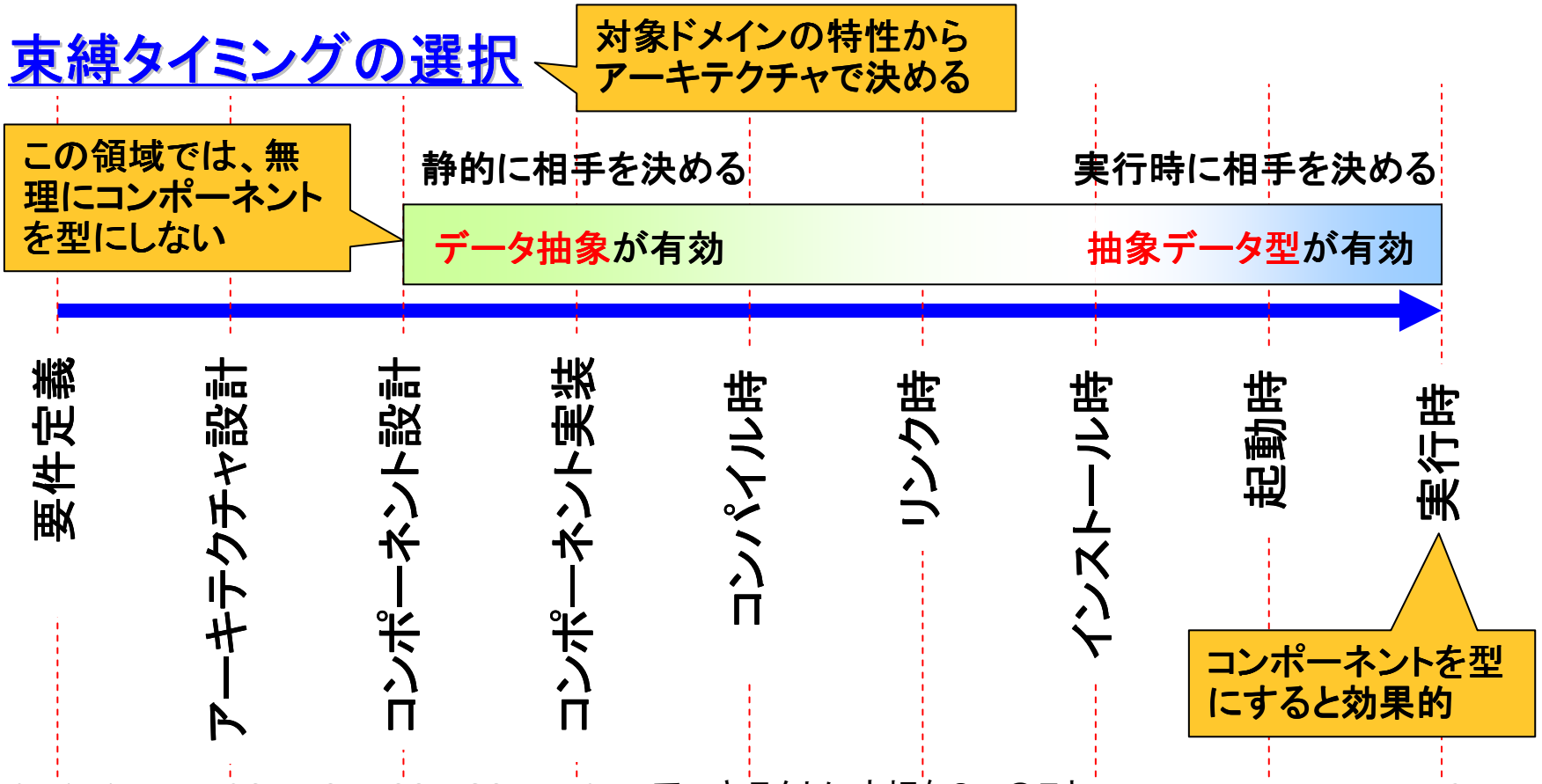
- ビジネス向けプリンタ
- 製品の販売期間が長く、派生開発も多い
- 課題
  - 属人性が高く保守が人に依存する
  - お客様のご要望にもっと早く応えたい

2. 設計の原則を定める

EPSON
EXCEED YOUR VISION

あえて型にしないことで単純な仕組みを導く

➔ アーキテクトは、設計手法に踊らされない

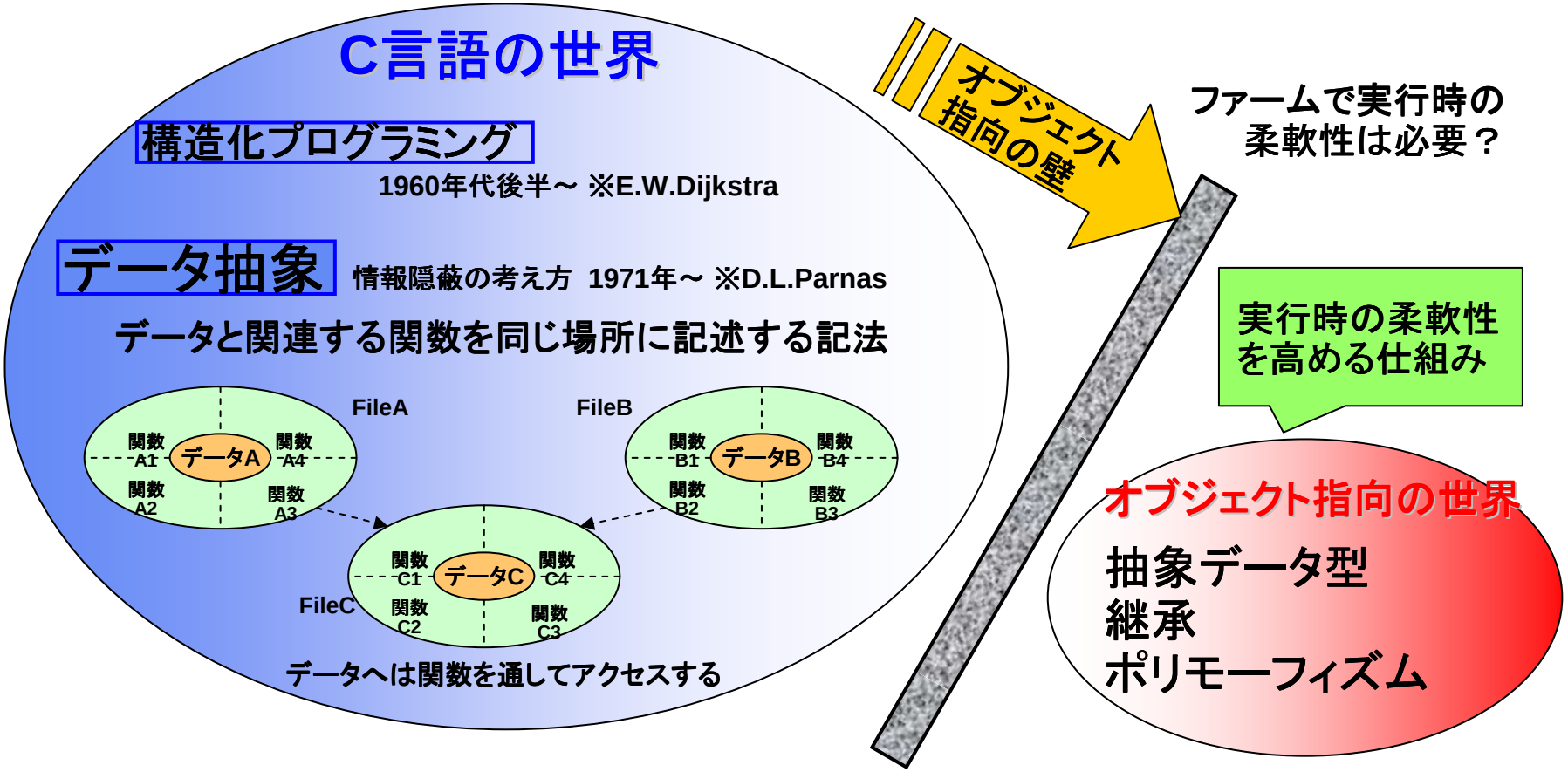


3. データ抽象を活用する

EPSON
EXCEED YOUR VISION

データと関係する関数は同じ場所に記述する

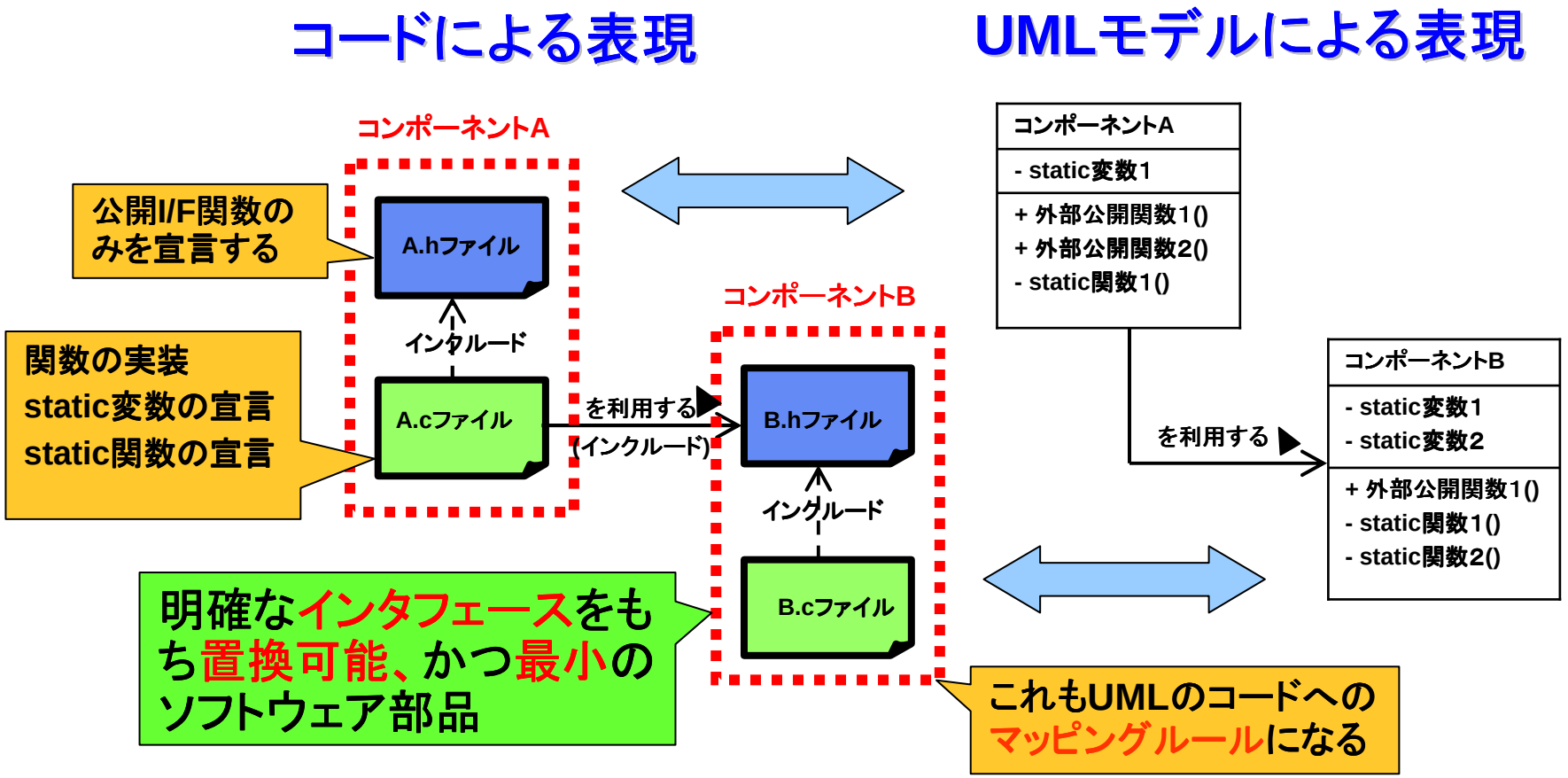
➔ 凝集度を高く、結合度を低くする



3.1 データ抽象の最小単位

EPSON
EXCEED YOUR VISION

データ抽象の最小単位は .hと.cのペアである

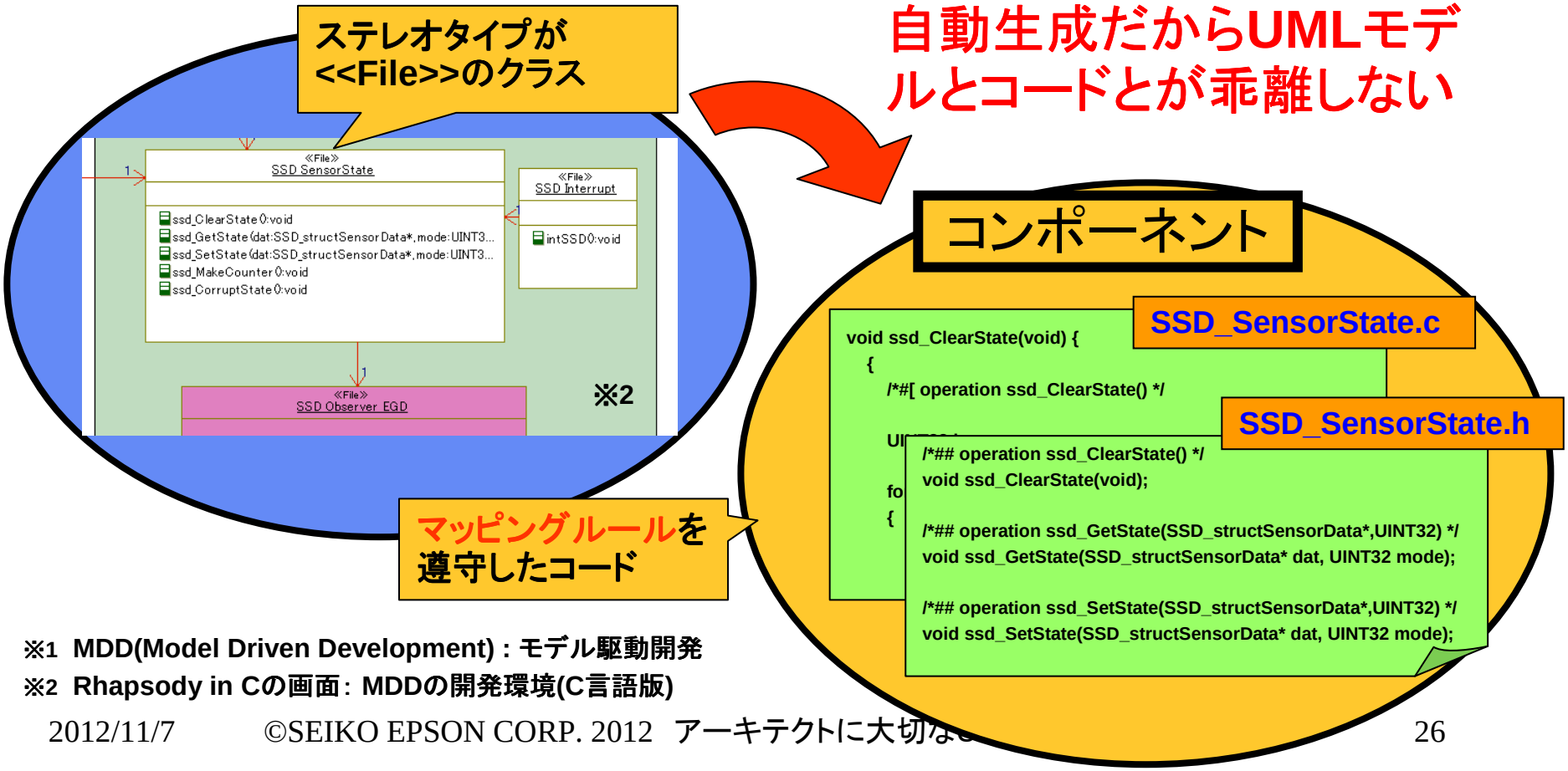


4. MDD※1を導入する

EPSON
EXCEED YOUR VISION

原則と方針を満たすためにツールで制約する

➔ アーキテクトは、必要ならツールも活用する



※1 MDD(Model Driven Development) : モデル駆動開発

※2 Rhapsody in Cの画面: MDDの開発環境(C言語版)

5. まとめ

EPSON
EXCEED YOUR VISION

アーキテクトは原則と方針を徹底させる

ポイント

1. アーキテクチャで束縛タイミングを決める
2. あえて型にしないことで単純な仕組みを導く
3. データと関係する関数は同じ場所に記述する
4. 原則と方針を満たすためにツールで制約する

Ⅳ. コンポーネント相互の関係

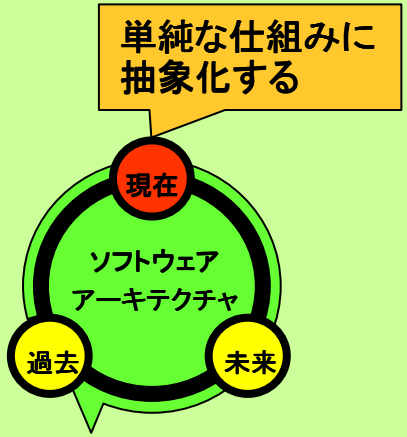
Ⅰ. アーキテクチャとは何か

Ⅱ. コンポーネントと環境との関係

Ⅲ. 設計と進化を導く方針

Ⅳ. コンポーネント相互の関係

Ⅴ. まとめ



Ⅳ. コンポーネント相互の関係

EPSON
EXCEED YOUR VISION

アーキテクトは単純な仕組みに抽象化する

➔ 基礎となる抽象化の力をトレーニングする

1. モデリングトレーニングの事例
2. 仮想性と透過性を理解する
3. 常に関連から考える
4. モデリングで抽象化力をつける

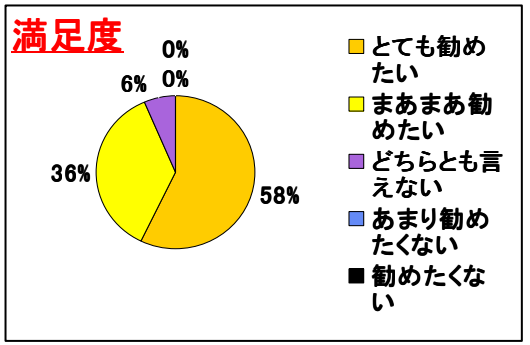
1. モデリングトレーニングの事例

EPSON
EXCEED YOUR VISION

仕組みを抽象化する力をトレーニングする ※1

➔ 実力は仕組みを抽象化する力に裏打ちされる

ソフトウェア技術者の



《質問》本トレーニングの受講を他の人に勧めたいと思いますか？ ※2

ポイント

- 透過性と仮想性を理解する
- 常に関連から考える
- モデリングで抽象化力をつける

※1「ソフトウェア工学実践の基礎」(落水浩一郎 著、1993年)
※2 調査期間:2011年3月～2012年7月、回答数:110名

2. 仮想性と透過性を理解する

EPSON
EXCEED YOUR VISION

仕組みの抽象化には仮想性と透過性を使う

ポイント

1. 仮想性を与える

   無いものを、あるように見せること

   ➔ 仮想的な仕組み（ソフトウェア）を実現する

2. 透過性をなくす

   あるものを、無いように見せること

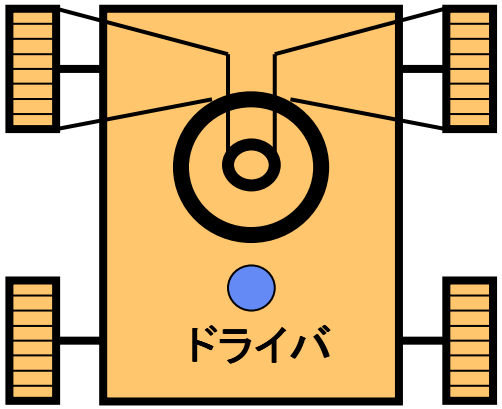
   ➔ 必要なものだけを見せる仕組みを実現する

2.1 仮想的な仕組みを作る

EPSON
EXCEED YOUR VISION

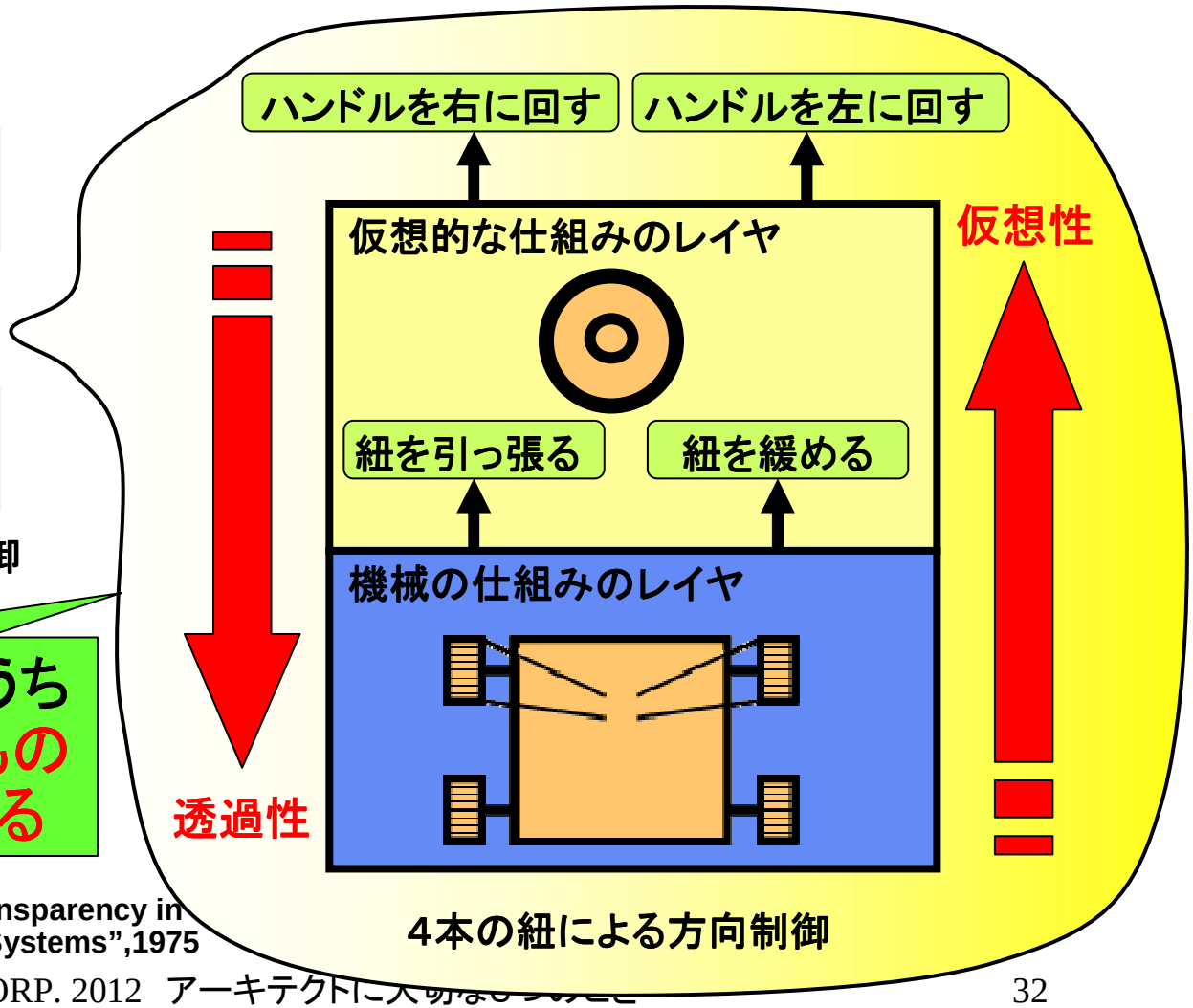
抽象化で単純で扱いやすい仕組みを作る

仕組みの



ハンドルを利用した方向制御

機械がもつ可能性のうち
目的に応じて必要なもの
を強調し、他を捨て去る



4本の紐による方向制御

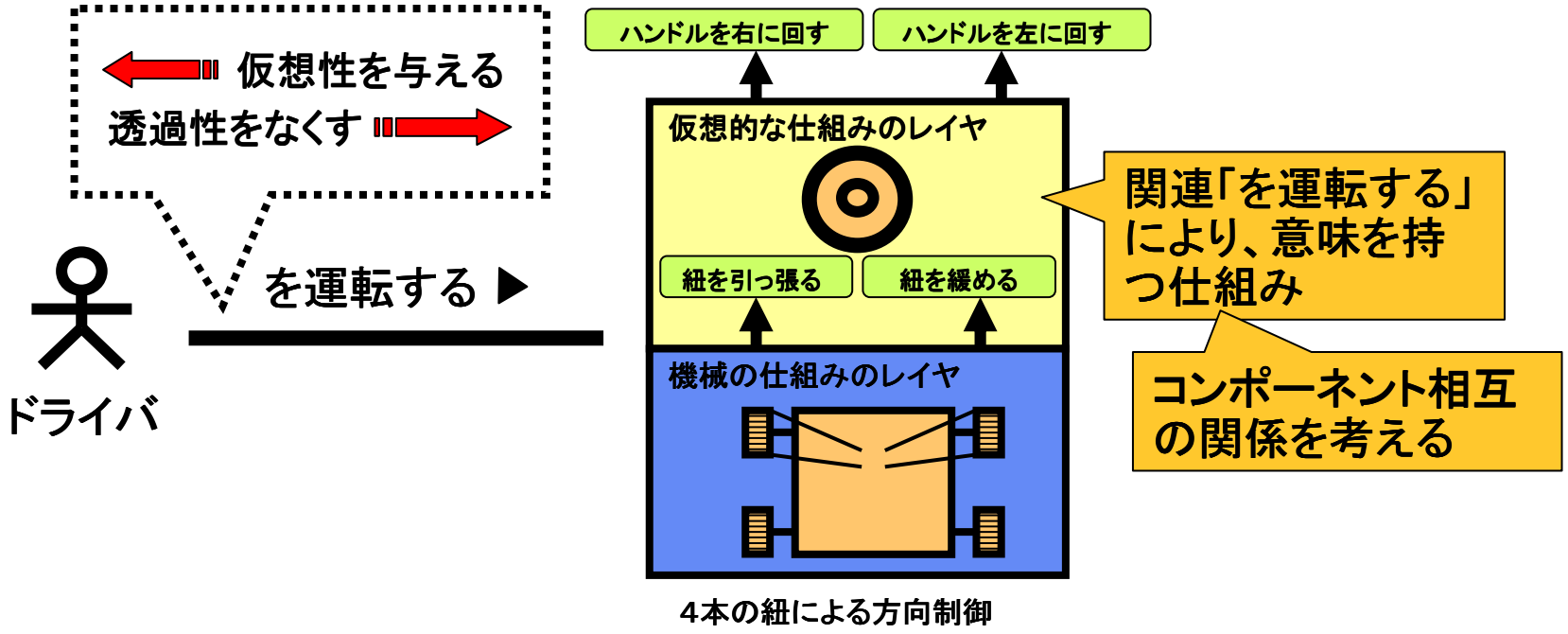
※D.L.Parnas “Use of the Concept of Transparency in the Design of Hierarchically Structured Systems”,1975

3. 常に関連から考える

EPSON
EXCEED YOUR VISION

透過性と仮想性は「誰から見て」が重要となる

➔ 誰から見て＝対象との関連が重要になる

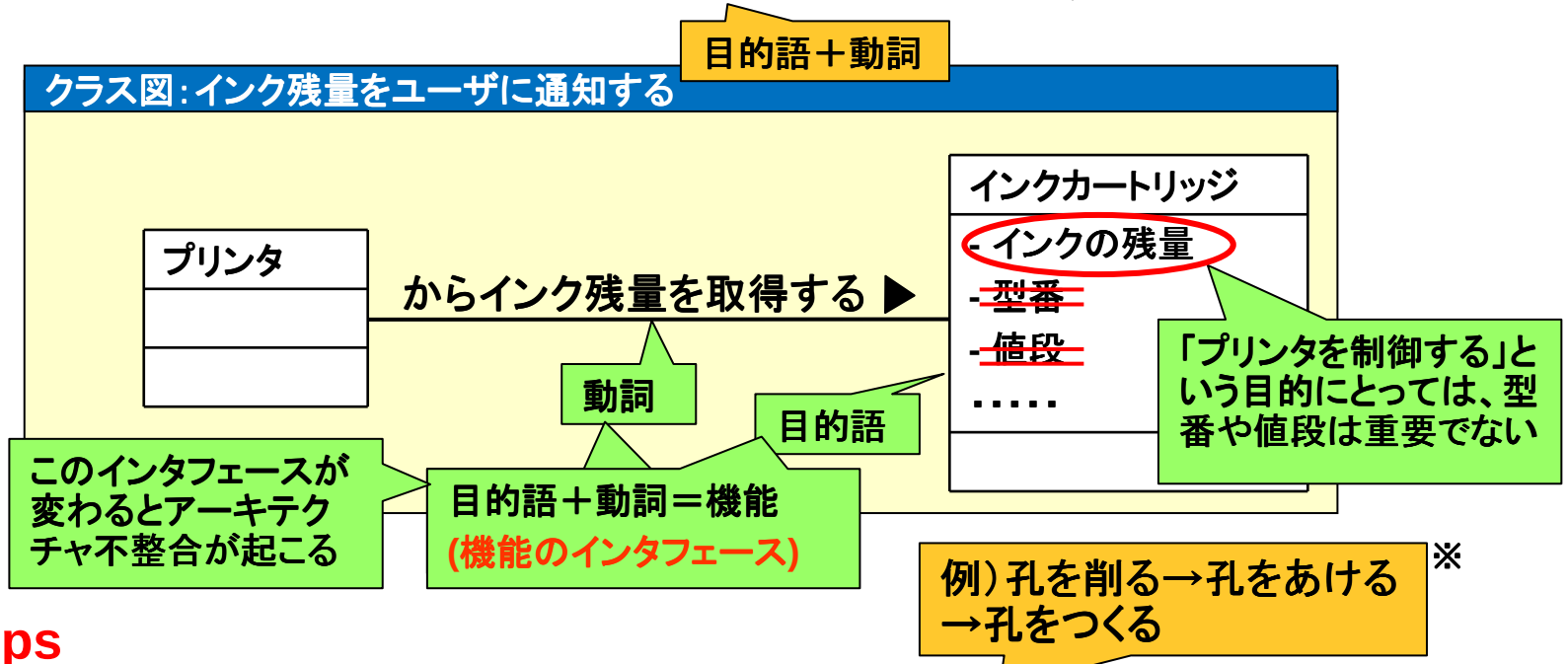


4. モデリングで抽象化力をつける

EPSON
EXCEED YOUR VISION

関連名を書くことからモデリングは始まる

➔ 関連（＝目的となる機能）によって見え方が異なる



Tips

「何のために？」と問うと、より抽象度の高い動詞が導け関連が安定する

➔ コンポーネント同士の関係＝アーキテクチャを安定化できる

※「新・VEの基本 価値分析の考え方とプロセス」（土屋裕 監修、産能大学VE研究グループ 著、1998年）

5. まとめ

EPSON
EXCEED YOUR VISION

アーキテクトは単純な仕組みに抽象化する

ポイント

1. 仕組みを抽象化する力をトレーニングする
2. 仕組みの抽象化には仮想性と透過性を使う
3. 透過性と仮想性は「誰から見て」が重要となる
4. 関連名を書くことからモデリングは始まる

Ⅴ. まとめ

Ⅰ. アーキテクチャとは何か

Ⅱ. コンポーネントと環境との関係

Ⅲ. 設計と進化を導く方針

Ⅳ. コンポーネント相互の関係

Ⅴ. まとめ

Ⅴ. まとめ

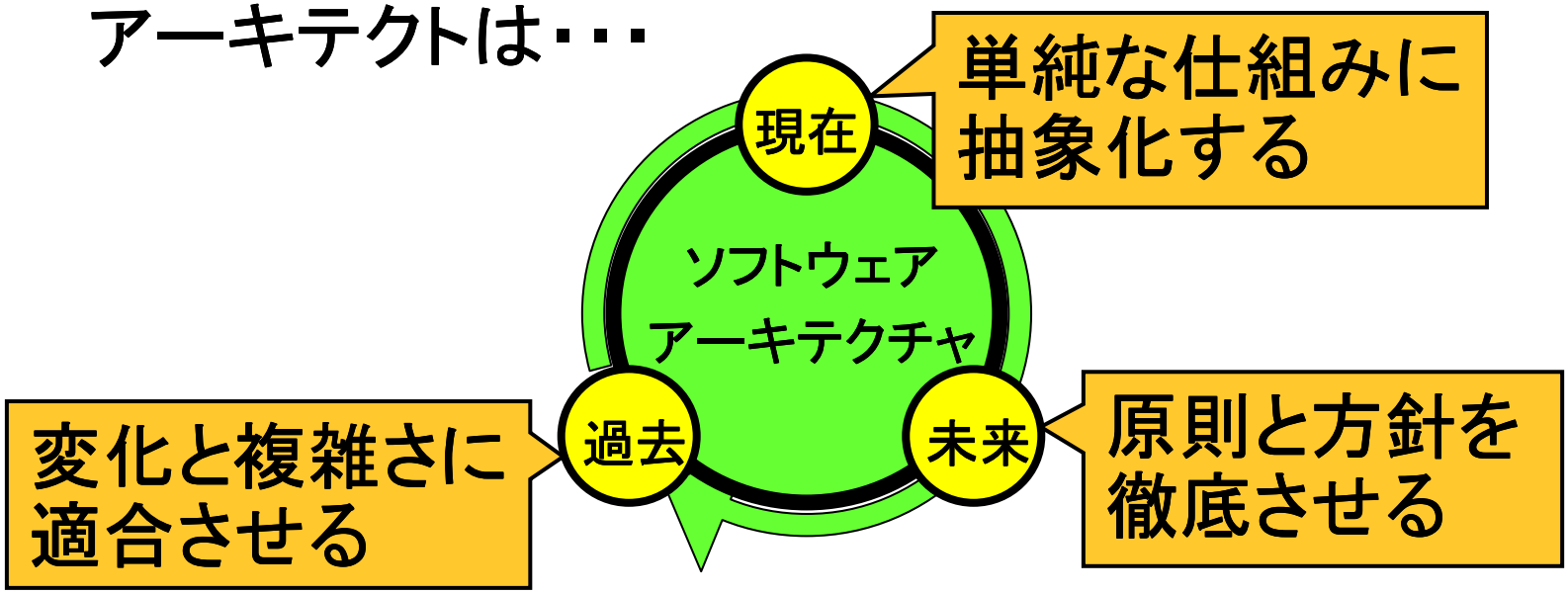
EPSON
EXCEED YOUR VISION

ソフトウェアの過去と未来、そして現在を作る

ポイント

定義にはアーキテクトに大切なことが含まれる

アーキテクトは・・・



EPSON

EXCEED YOUR VISION