



Systems Engineering Research Association

システムズエンジニアリング におけるモデルの役割

2019年10月28日

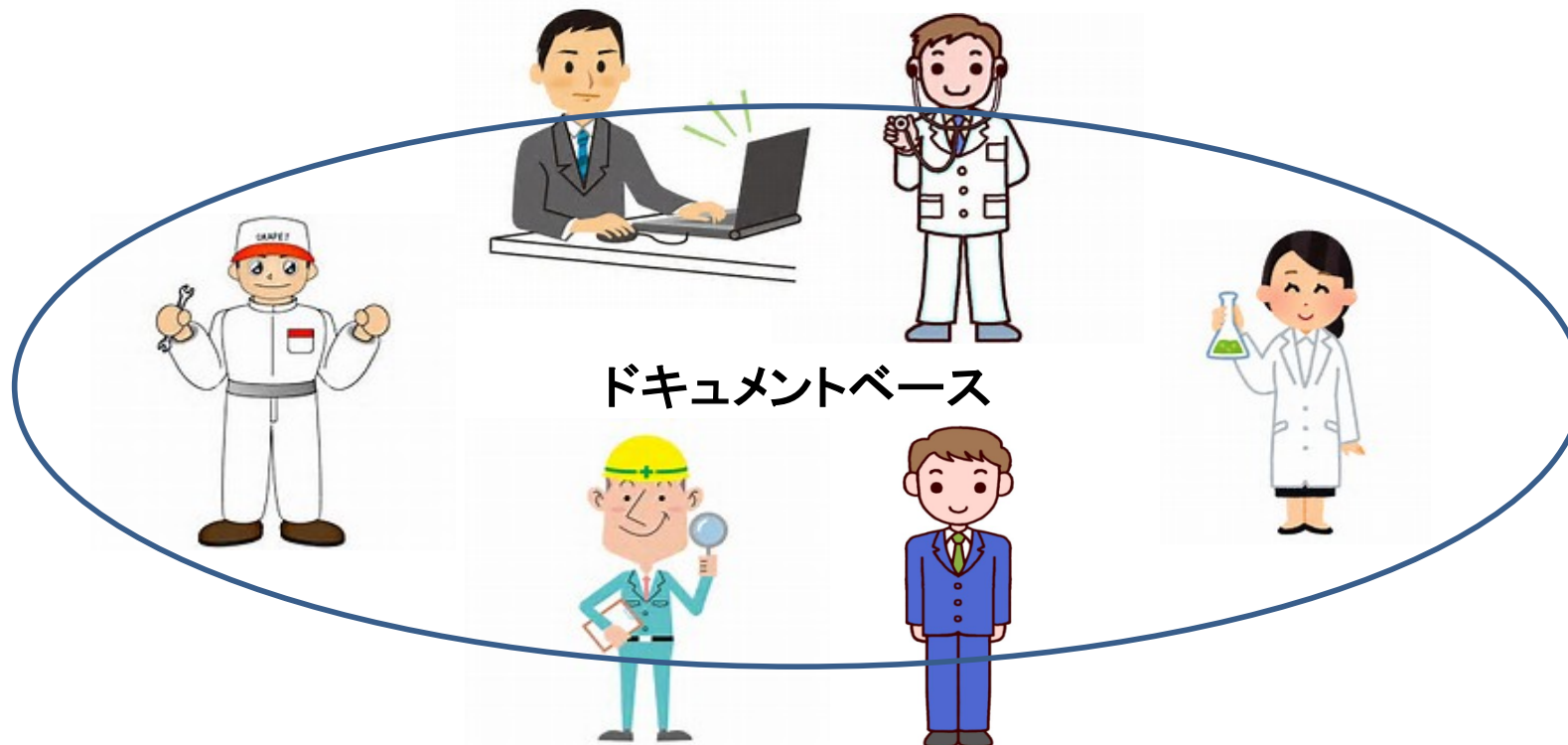
一般社団法人システムズエンジニアリング研究会

代表理事 内田功志

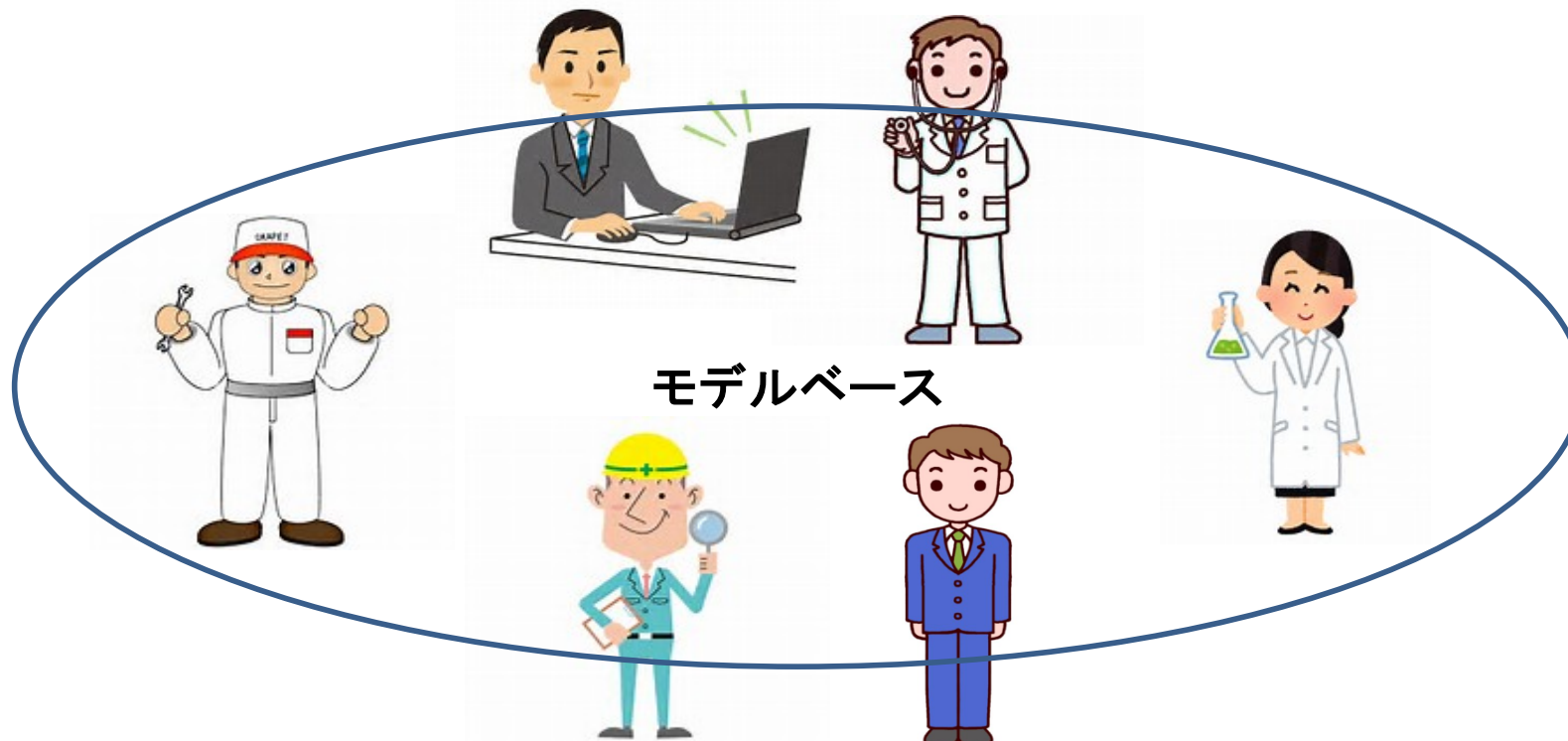
- システムズエンジニアリングに関する考え方は自動車業界を中心に普及しつつある
- MBSE(モデルベースシステムズエンジニアリング)に関しても、少しずつ認知され始め、導入を進めている組織も増えてきた
- 実際にはモデルに関して、敷居が高くMBSEに踏み切れない組織もある
- しかし、モデルに関するメリットは絶大で、使うことによる優位性は見逃せない

システムズエンジニアリングとは *SERA*

- システムの実現を成功させることができる複数の専門分野にまたがるアプローチおよび手段である



- モデルベースシステムズエンジニアリング (MBSE) は、System Development Life Cycle (SDLC) を通じてシステムズエンジニアリング活動に厳しいビジュアルモデリング原則とベストプラクティスを適用することを強調したシステムズエンジニアリングのパラダイムである



本日のテーマ

- MBSEの現状について
- MBSEの問題点
- モデルの役割

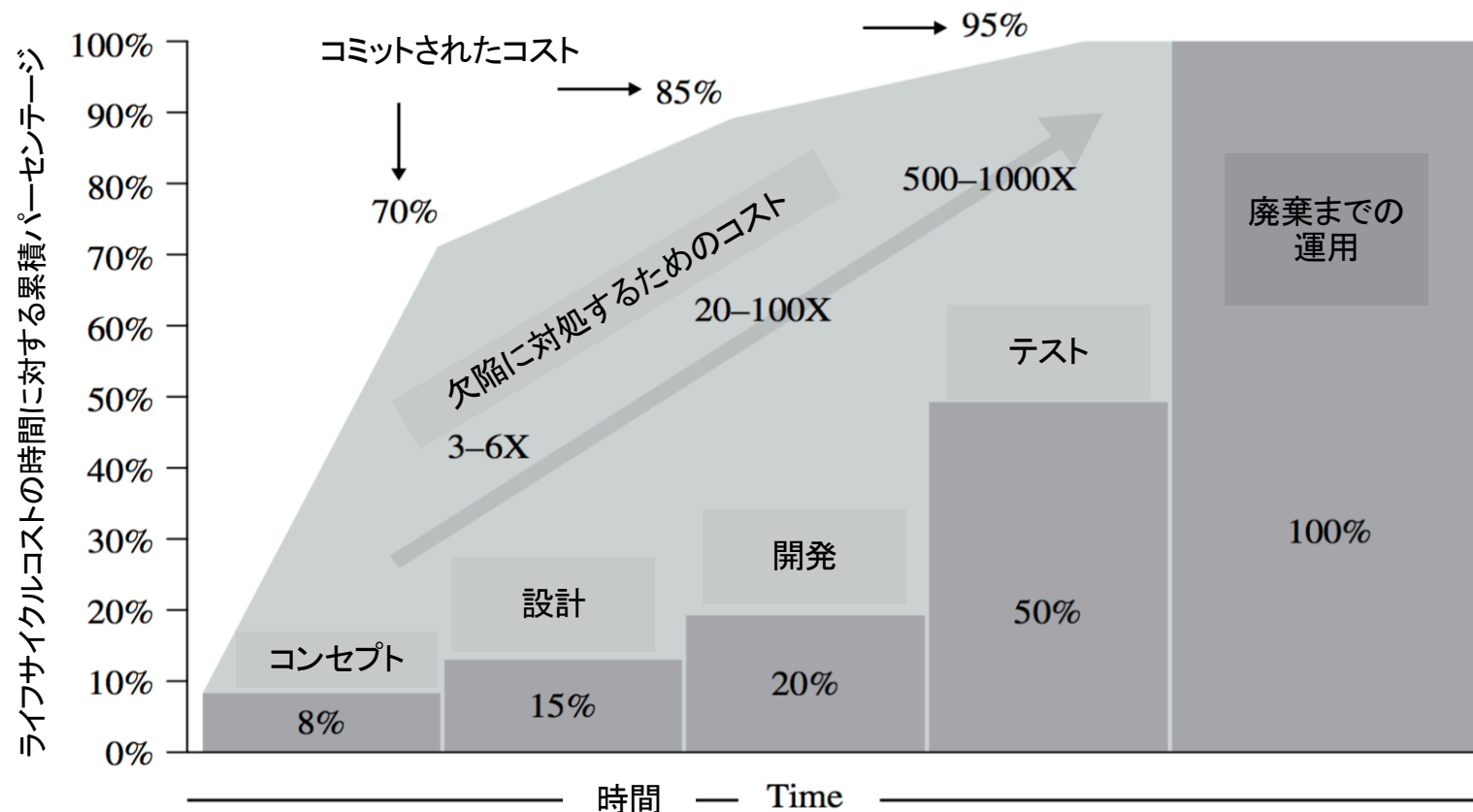
MBSEの現状について

- 自動車
- 航空宇宙・防衛
- スマートエネルギーシステム
- 鉄道
- ロボット
- 製造業
- その他

- システムズエンジニアリングは、複雑性および変化をマネジメントするための効果的な方法
- 適しているシステム
 - ◆ 大規模システム
 - ◆ 複雑なシステム
 - ◆ 複数のシステム連携
- 適さないシステム
 - ◆ 小規模システム
 - ◆ 単純なシステム
 - ◆ 連携しないシステム

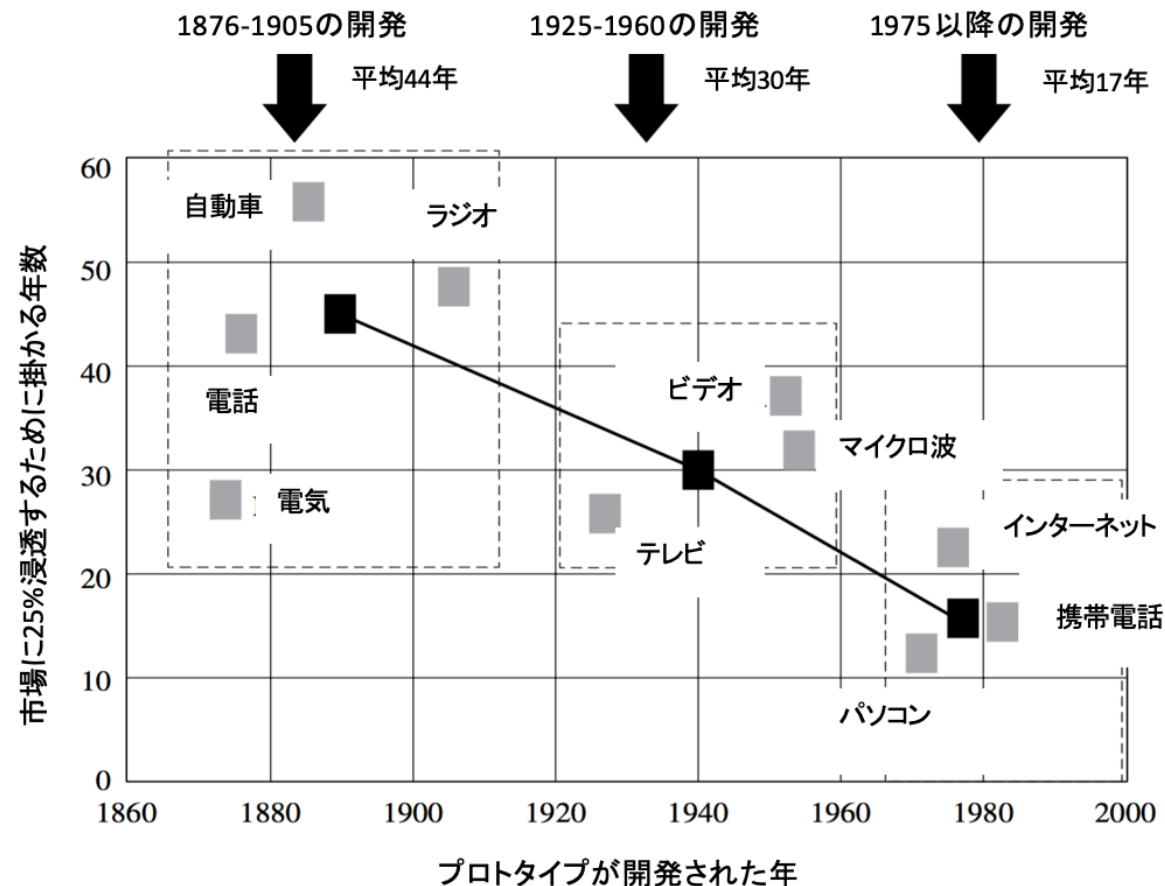
SEの必要性(1)

- 製品、サービス、および社会において複雑さと変化の両方が増大し続けているため、新しいシステムに関連するリスクの低減または複雑なシステムへの変更は、システムズエンジニアの主な目的であり続ける

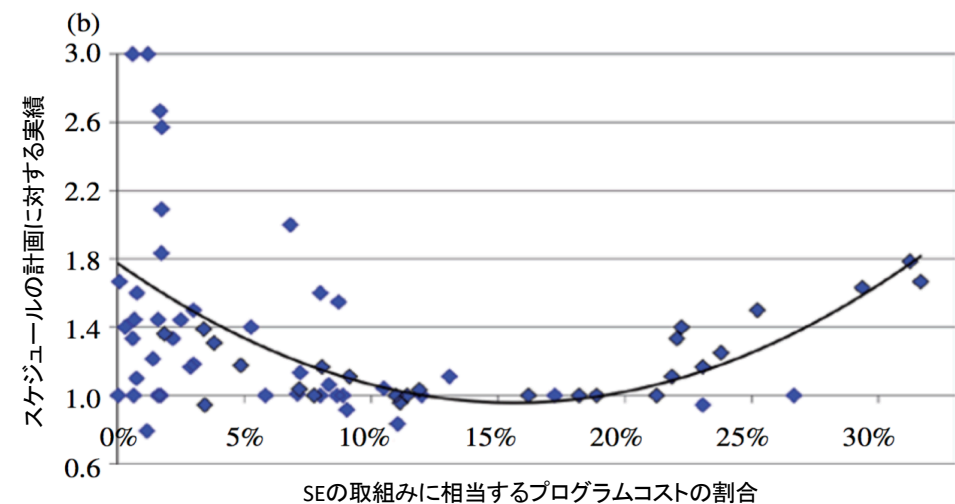
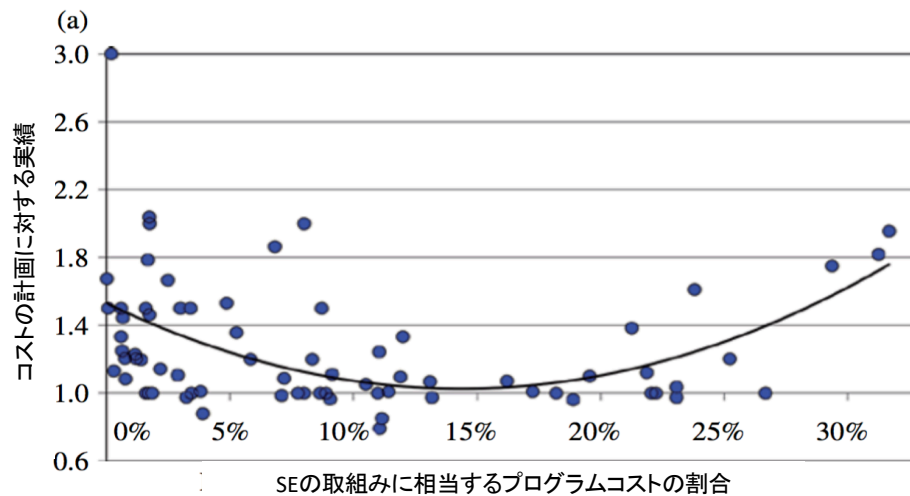


SEの必要性(2)

- 技術の開発が過去100年間で加速した
- 前世紀のこの製品サンプルでは、25%の市場浸透率を達成するのに要した時間が約50から12年未満に短縮された
- 平均して、プロトタイプから25%の市場浸透率までの開発は44年から17年になる



- 全体のSEの取組みをコスト(最高値)とスケジュール(最低値)と比較する
- 両グラフは、SEの取組みが、相互関係において80%の高さで、プログラムの成功に関し、定量化された効果があることを示している
- 実際には、ほんの7%(一般にSE努力のうちの最適なレベルの約半分)のSEの取組みでも、プログラムが動作することを示した
- プログラムコストの8%から19%のSEの取組みが最適である



MBSEの問題点

- 小規模システムのパイロットプロジェクトは意味がない
 - ◆ 小規模システムのMBSE適用は手間がかかるだけで効果は期待できない
 - ◆ 既存の開発手法で問題なく開発できるシステムをMBSEを適用すると、余計な時間と労力がかかってしまう
 - ◆ つまり、小規模システムでMBSEを評価すると最悪な値になる可能性がある
- パイロットプロジェクトでも大規模複雑なシステムをターゲットにすると、それなりの効果が期待できる

- 大規模システムや複雑なシステムの部品（一部分）の開発を行う場合、部品が担う構造と振る舞いを明確にして、全体のシナリオに適合する必要がある
- 部品が担うシナリオを切り出して、何をすべきかを明確にする必要がある
- 部品が担うシナリオが明確になれば、それ以降の作業はそれほど変わらない
- 部品に対してユースケースを検討することはあまり勧められない

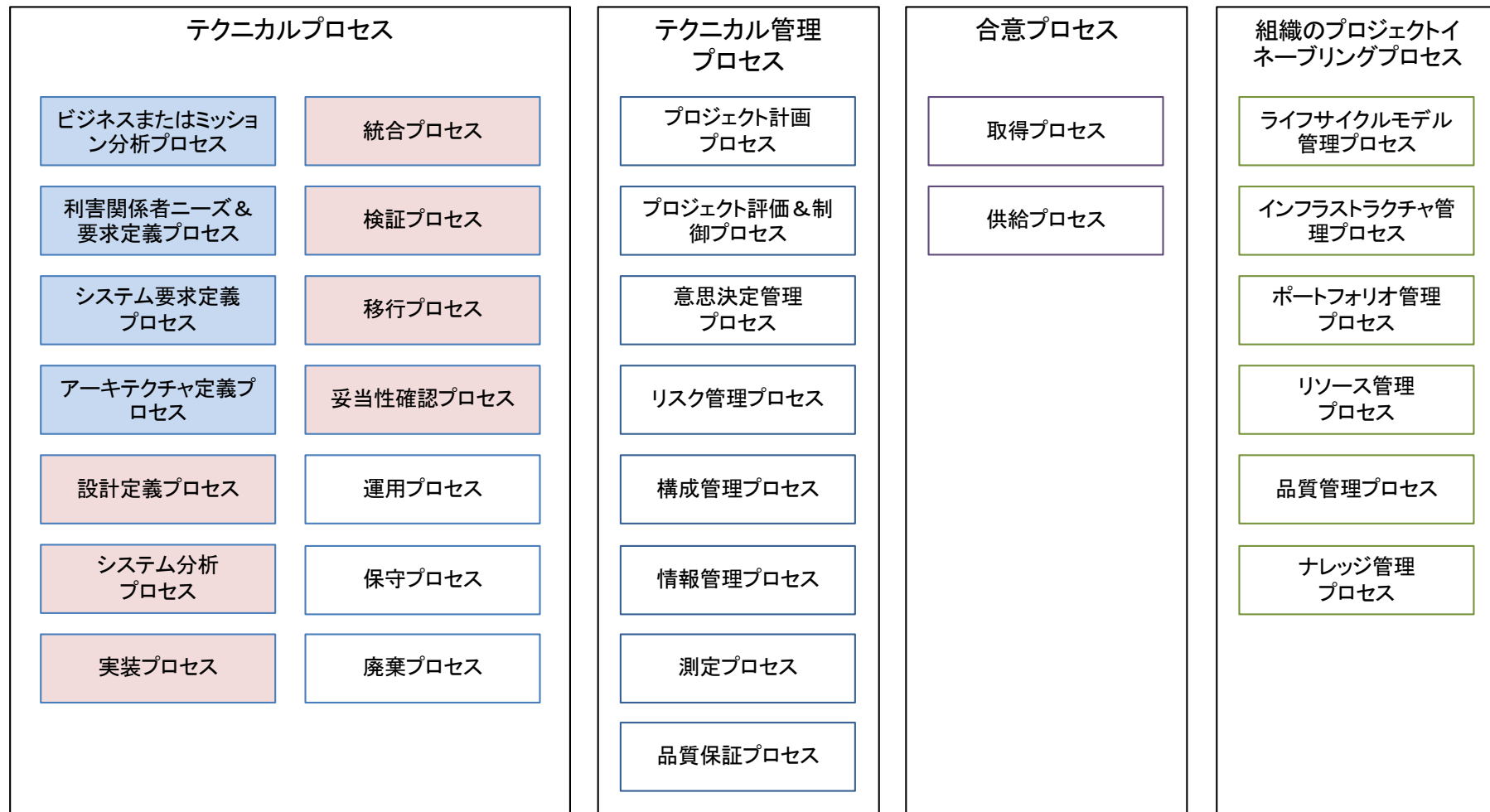
- MBSEを使いこなせる人材が圧倒的に不足している
- 実際に使いこなせるようになるのには1～2年程度かかることが足かせになっている
- ハンズオンのセミナーでは全体の流れや方法論は理解できても、修得までは程遠い
- システムズエンジニアリングは理解できても、モデルを使うと敷居はいっきに高くなる
- しかし、王道はない

- 経験者が0でMBSEを実践すると、失敗する可能性は95%以上になる
 - ◆ MBSEで失敗すると、損失は計り知れない
- UMLの失敗は繰り返さない
 - ◆ コンサルティングだけでは遅々として進まない
 - ◆ 全員が使いこなせなくても良い
 - ◆ 実際のプロジェクトでサンプルを示す必要がある
 - ◆ 資格制度の充実を図る必要がある

モデルの役割

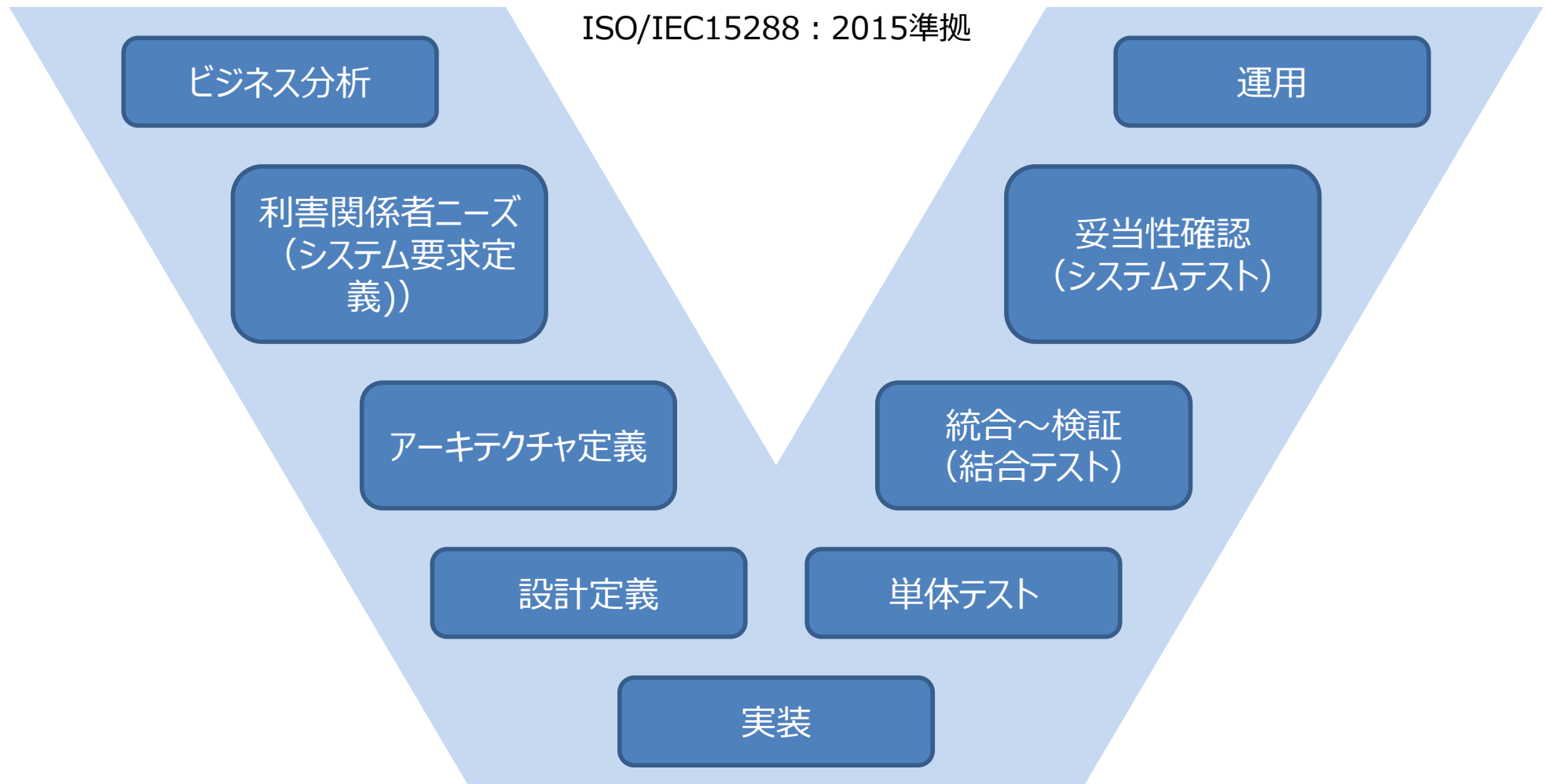
システムズエンジニアリングのプロセス *SERA*

ISO/IEC15288:2015

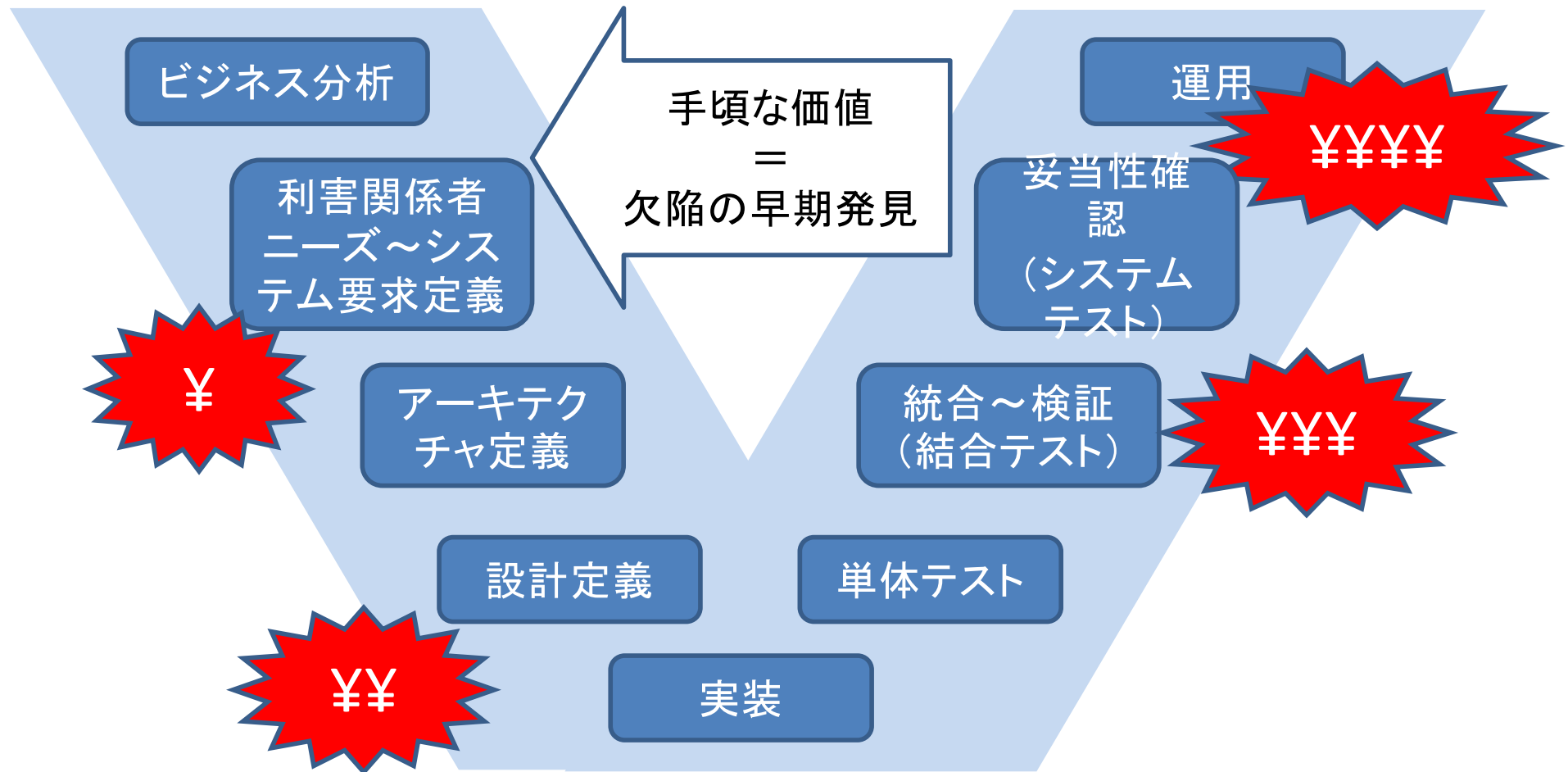


MBSEのプロセス

- システムのプロセスをV字モデルにしたものが以下である

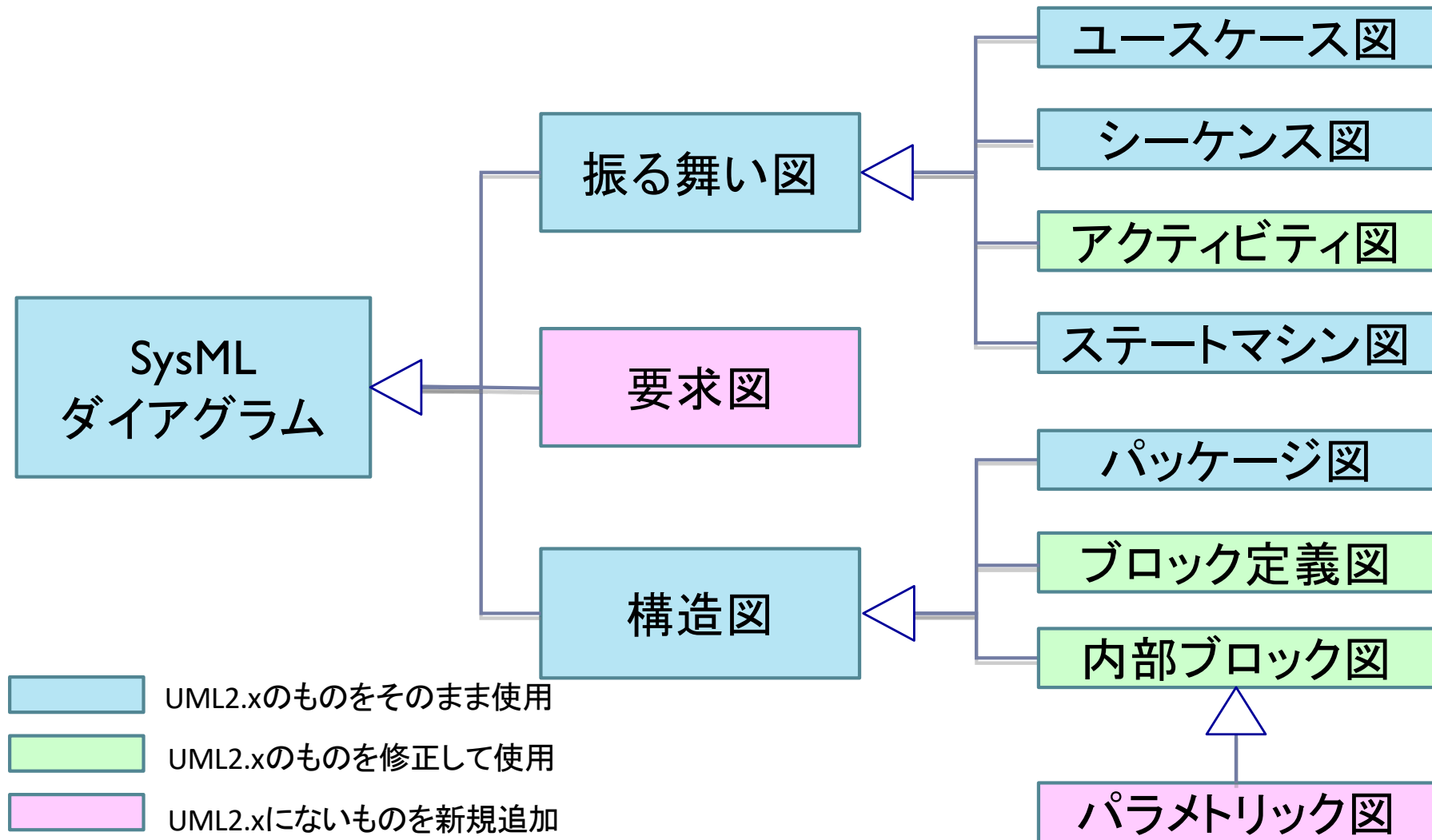


欠陥発見時のコスト



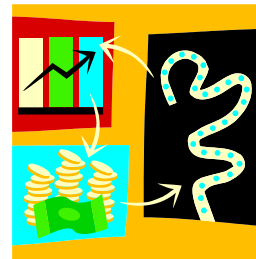
- System Modeling Languageの略で、システム開発のためのOMG主催の世界統一規格であり、現在はSysML1.3が標準となっている
 - ◆ 本テキストはSysML1.3に準拠する
- 要求を明確に定義し、分析し、さらに設計へ、そしてハードウェアとソフトウェアへと分割する
 - ◆ SysMLはそれらのモデルを表現する道具である
- OOSEM (Object Oriented Systems Engineering Modeling: INCOSE)、Harmony for SE(IBM社)などのシステムズエンジニアリング手法で使うことができる

SysML図式の分類



なぜSysMLなのか！？

- 自分ひとりでアイデアを形にするなら



自分だけが分かれば良い！

- アイデアを共有するなら



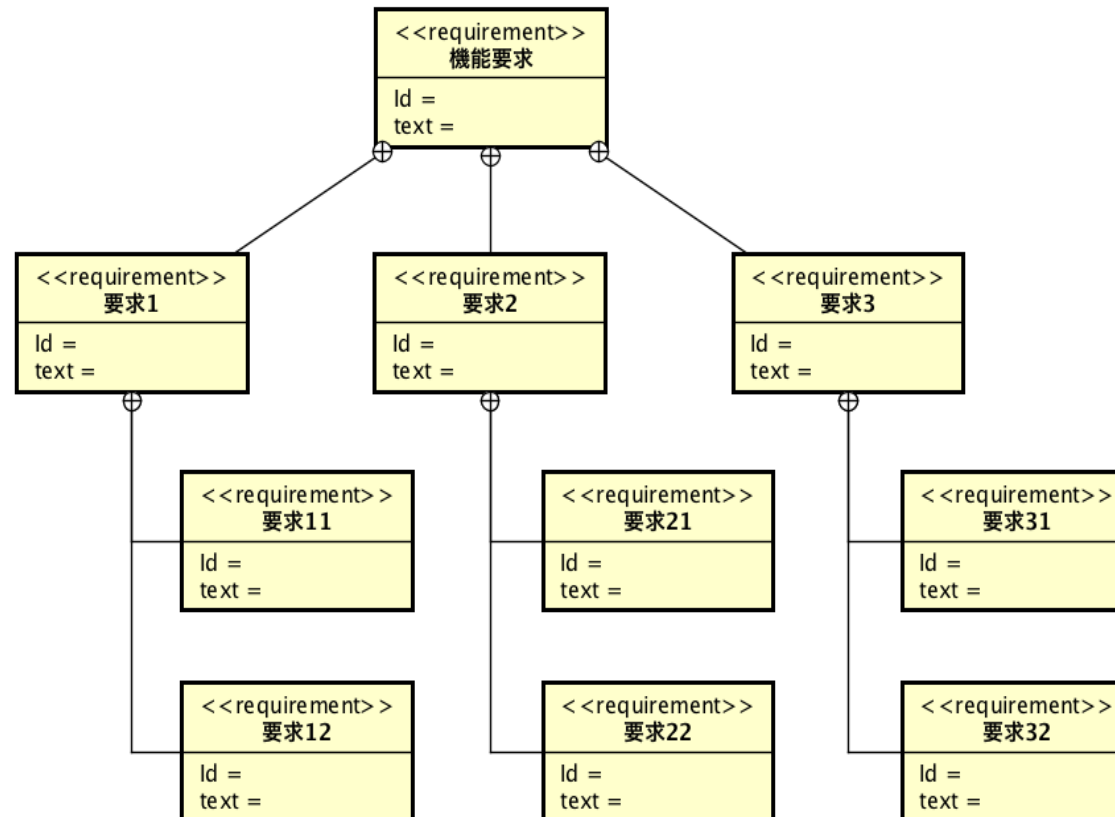
皆が分からなければダメ！

- UMLやSimulinkではダメか？

- ◆ UMLはソフトウェアのためのモデリング言語でシステムには向かない
- ◆ Simulinkは制御のためのモデリングツールなのでシステムには向かない

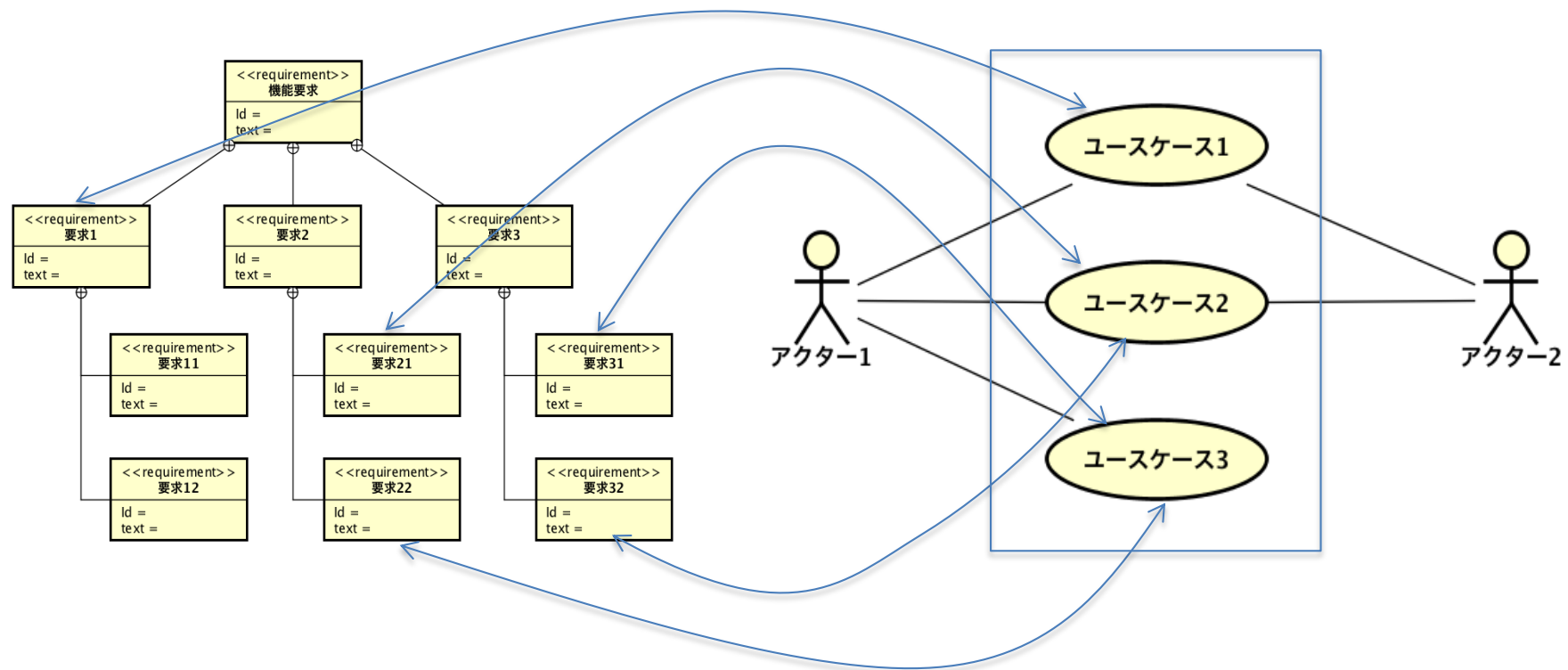
利害関係者の要求を明確に

- 利害関係者が何を求められているかを要求図を使用して、階層化することで明確にする



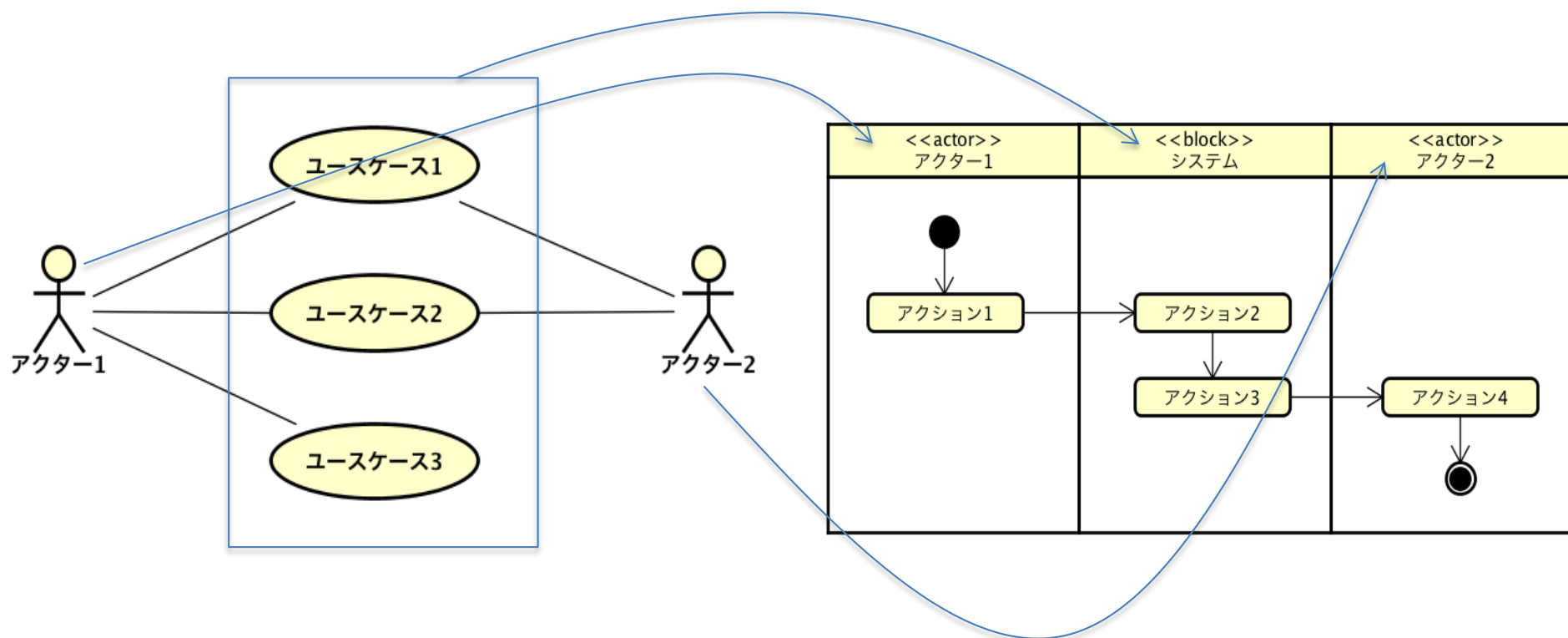
システムの要求を明確に

- 利害関係者の要求からシステムの要求へトレーサビリティを損なわないように移行する



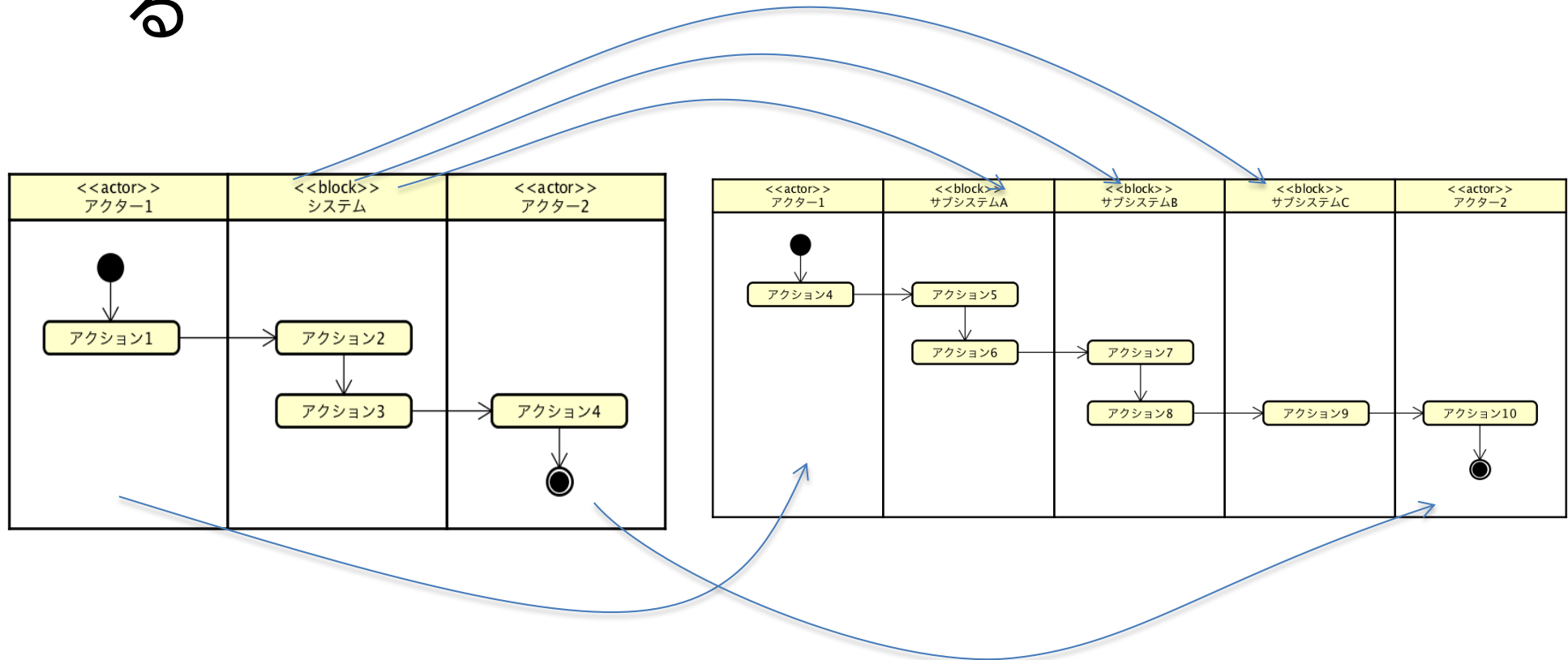
システム要求のシナリオ

- ユースケースの振る舞いをユースケース単位でシナリオとして定義する



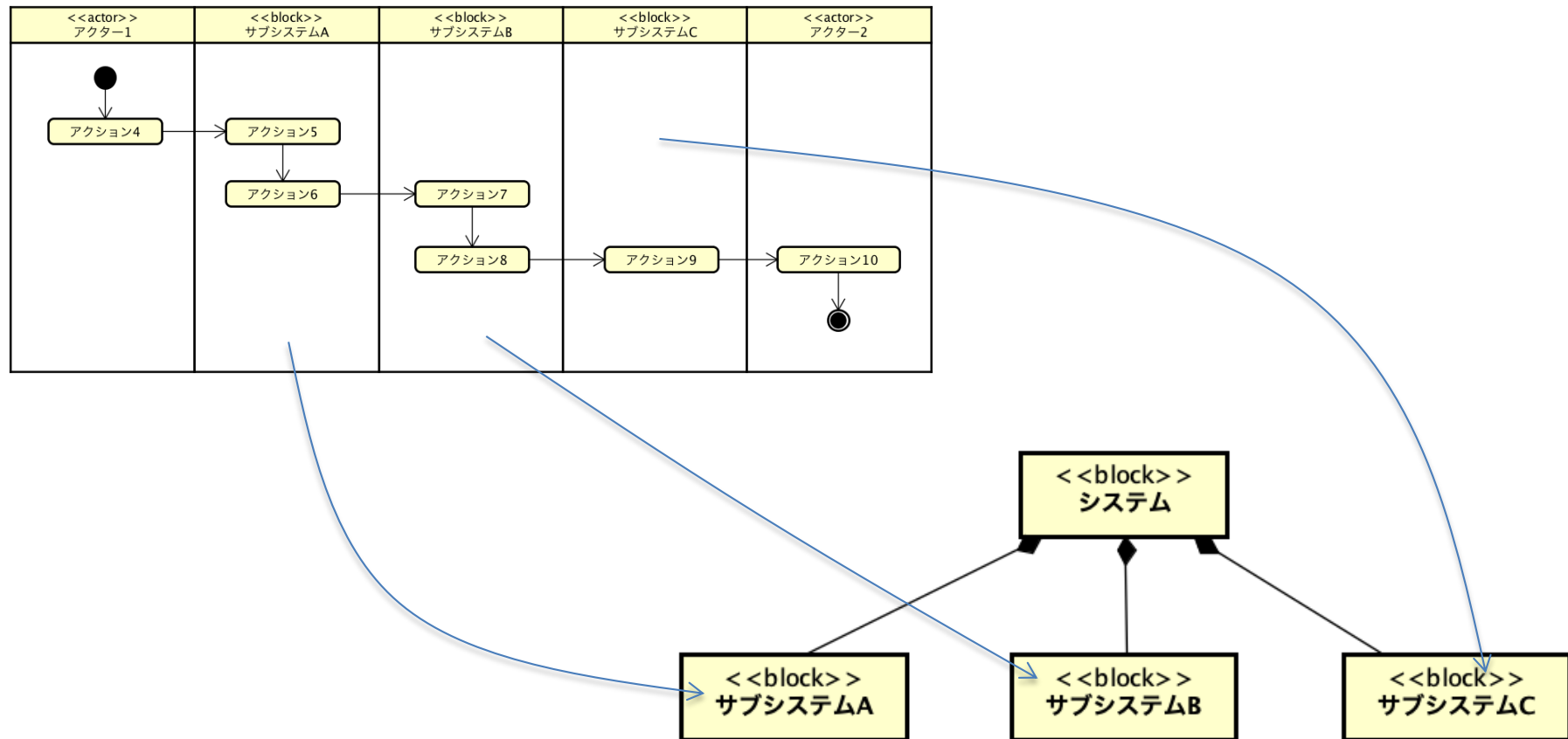
システムを明確に：振る舞い

■ シナリオからシステムの振る舞いを明確にする



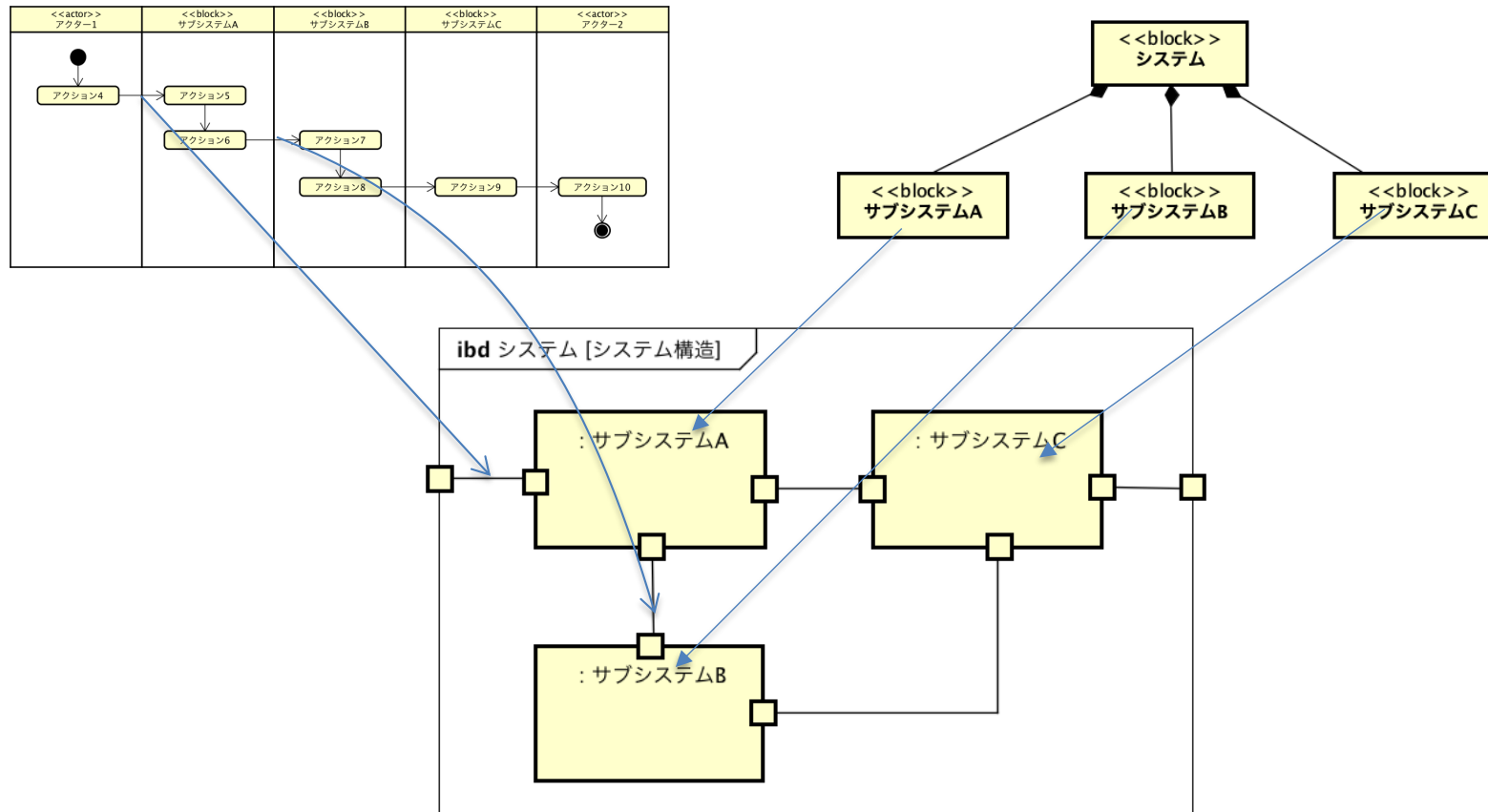
システムを明確に：構成

- システムの振る舞いからどのようなサブシステムが必要かを明確にする



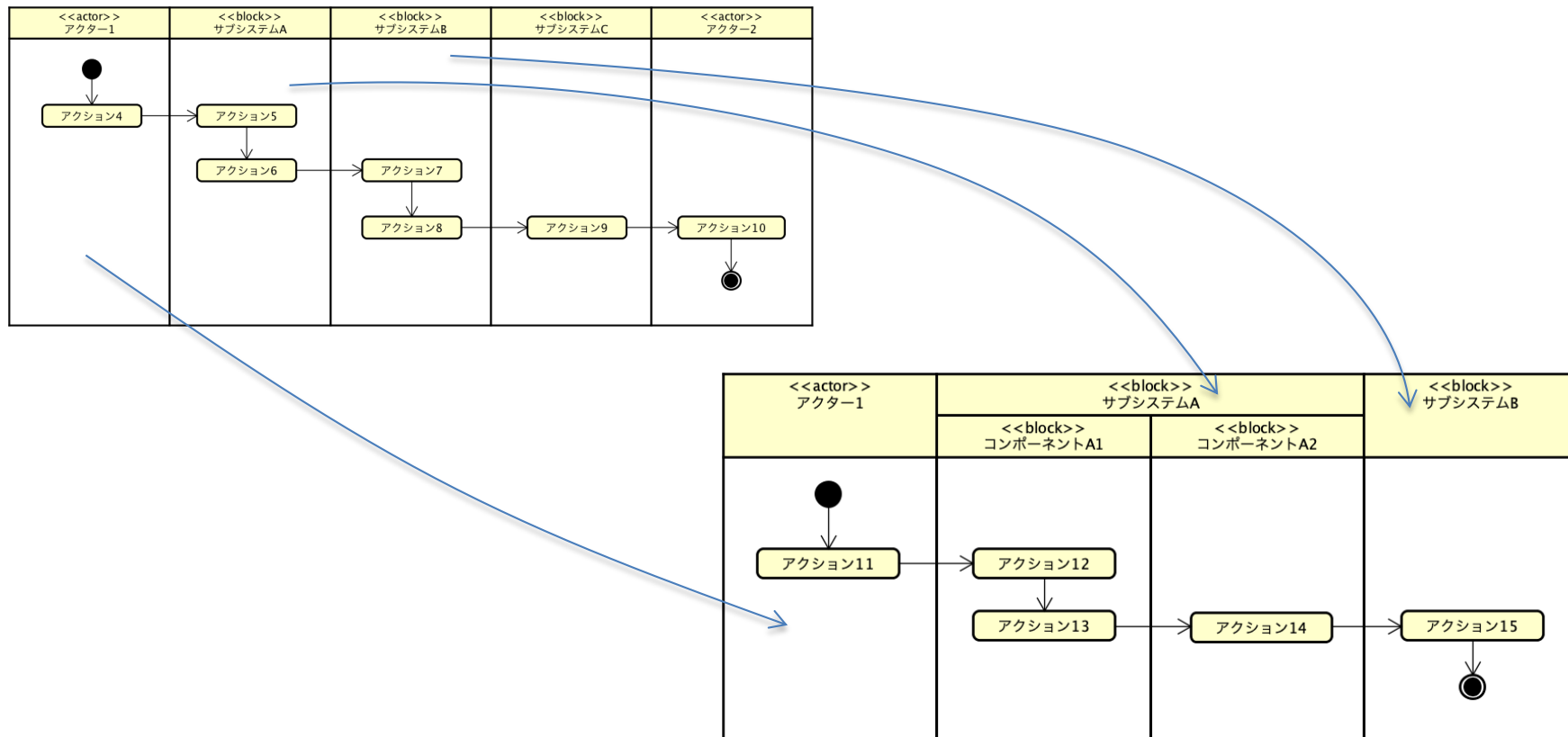
システムを明確に：構造

- システムの振る舞いと構成からシステムの内部構造を明確にする



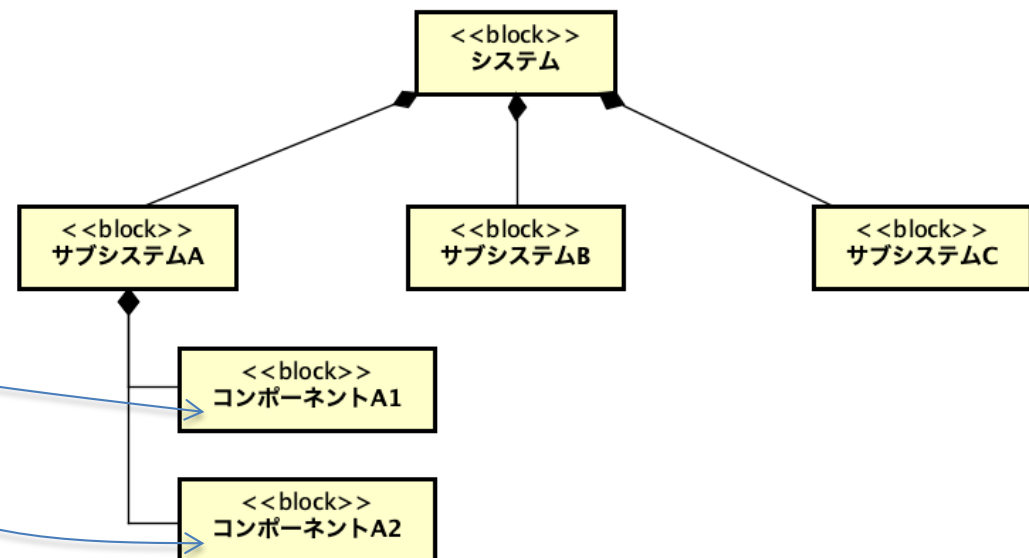
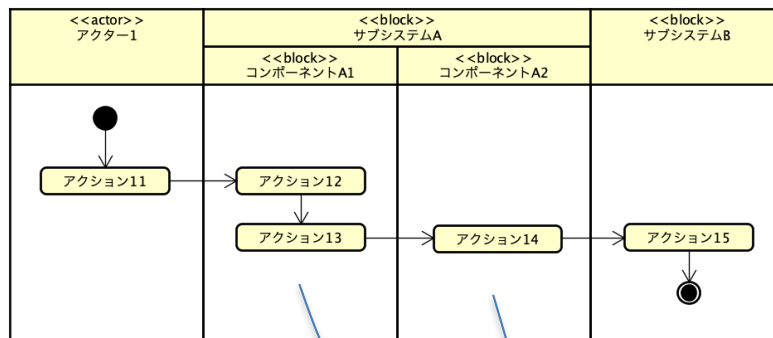
サブシステムを明確に: 振る舞い *SERA*

■ システムの振る舞いからサブシステムの振る舞いを明確にする



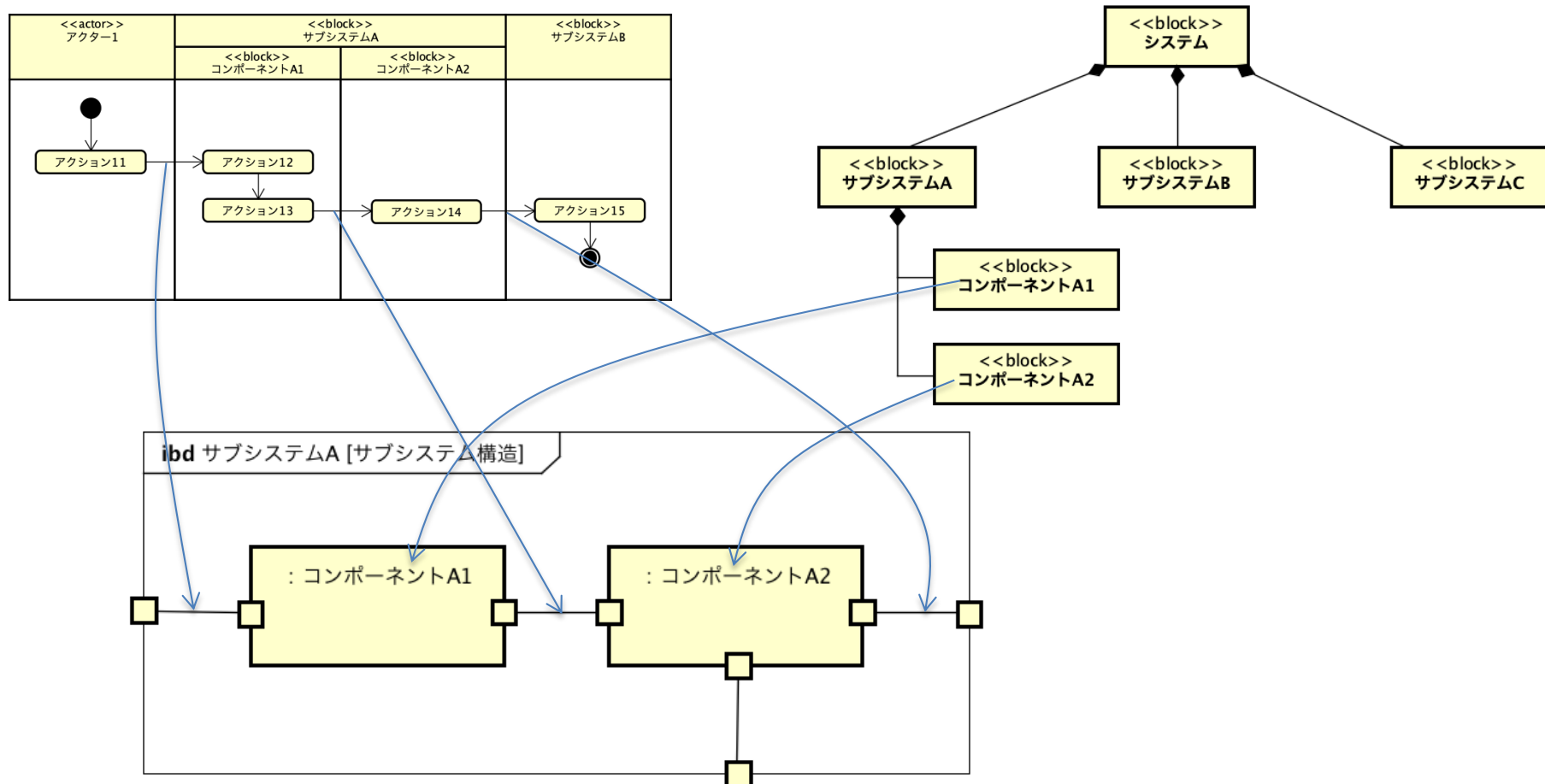
サブシステムを明確に：構成

■ サブシステムの振る舞いからサブシステムの構成を明確にする



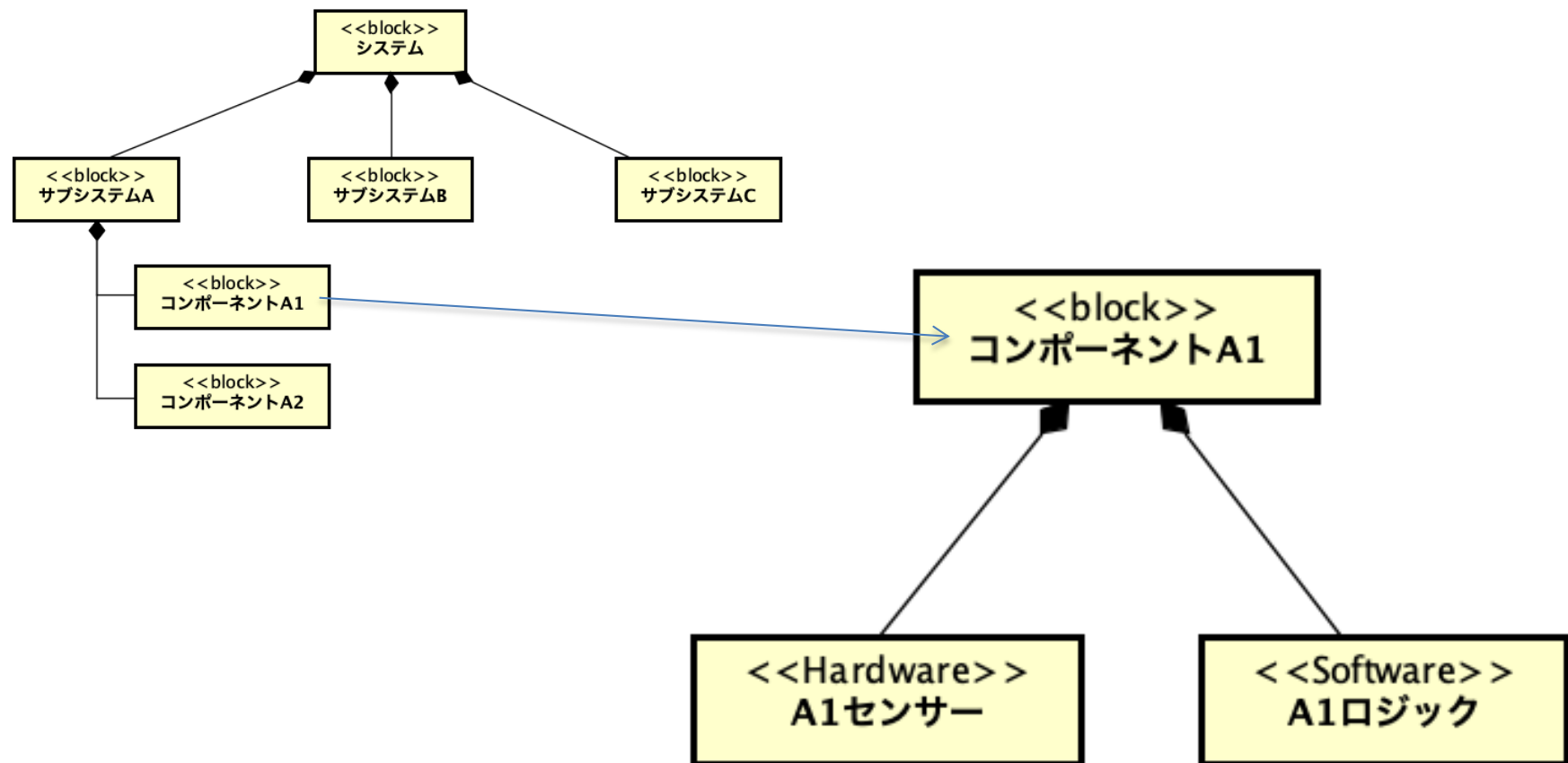
サブシステムを明確に: 構造

■ サブシステムの振る舞いと構成からサブシステムの構造を明確にする

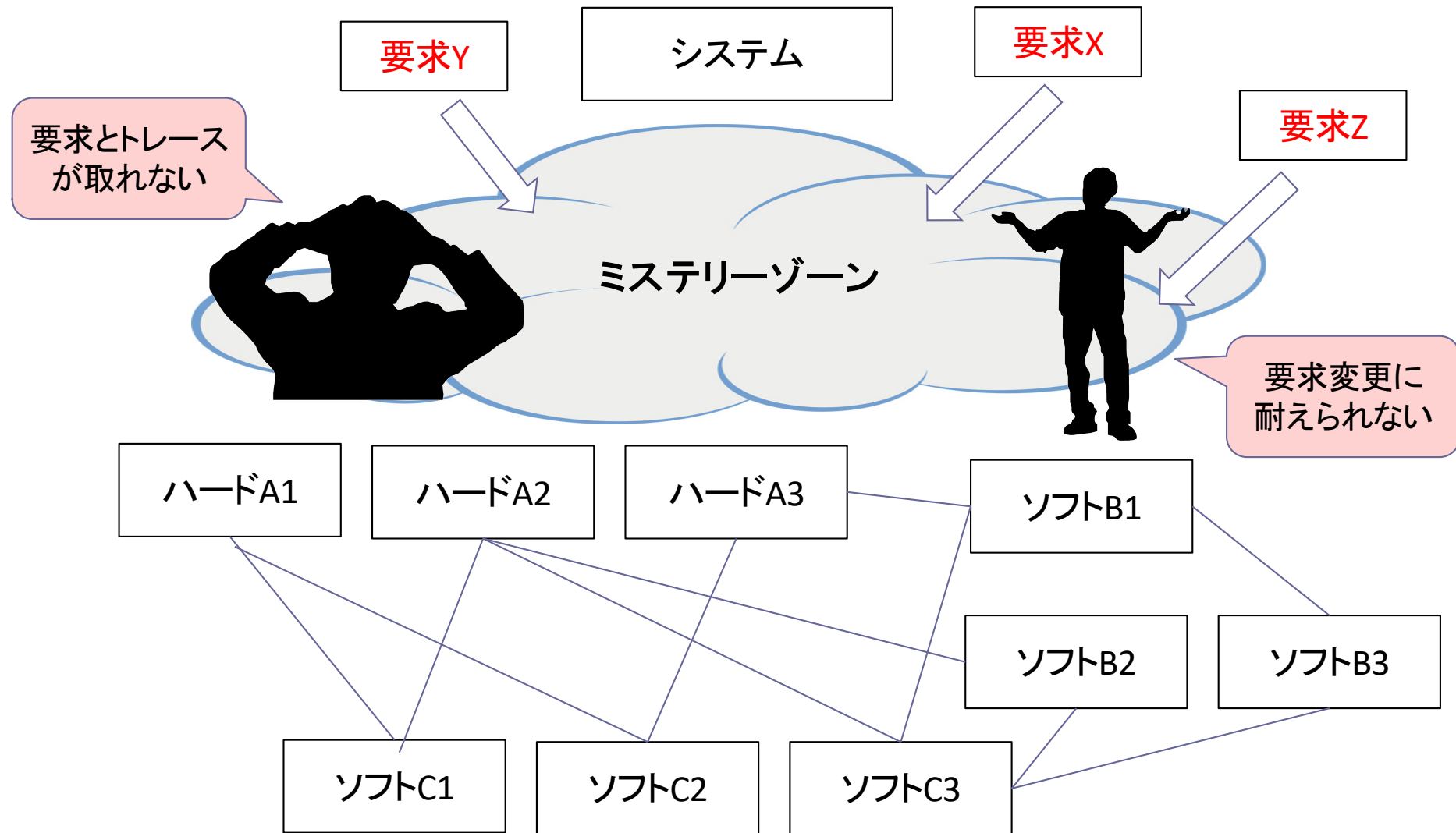


コンポーネントを明確に

- サブシステムからコンポーネントの詳細を明確にする



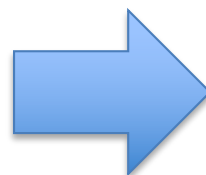
MBSEがなければ



- 複雑なシステムの開発が可能
 - ◆ 複雑なシステムも段階的に分割して扱うため、混乱なく開発を進めることができる
- 抽象度の高いモデルで検討することができる
 - ◆ システム全体を見渡すことができる
 - ◆ 関係者間の意思疎通が容易にできる
 - ◆ 今まで気付かなかったアイデアが生まれる
 - ◆ 全体最適のための新しい概念を検討できる
- フロントローディング開発
 - ◆ 不具合の早期発見が可能
 - ◆ 手戻りを防ぐことができる
- モデルによるトレーサビリティの確保
 - ◆ 要求の変更に容易に対応できる
 - ◆ 開発根拠の喪失を防ぐ

まとめ

- MBSEはモデルを使用することでトレーサビリティをより厳密にできる
- MBSEは利害関係者の要求をもとにソフトウェアやハードウェアにハンドオフできる
- 段階的に詳細化するので、どんなに複雑なシステムでも表現することができる



MBSEの実践方法 (MBSEハンドブック第2版解説)

ご清聴ありがとうございました *SERA*

MBSEで悩んだら
SERAのホームページへ
<http://www.sera.or.jp/>