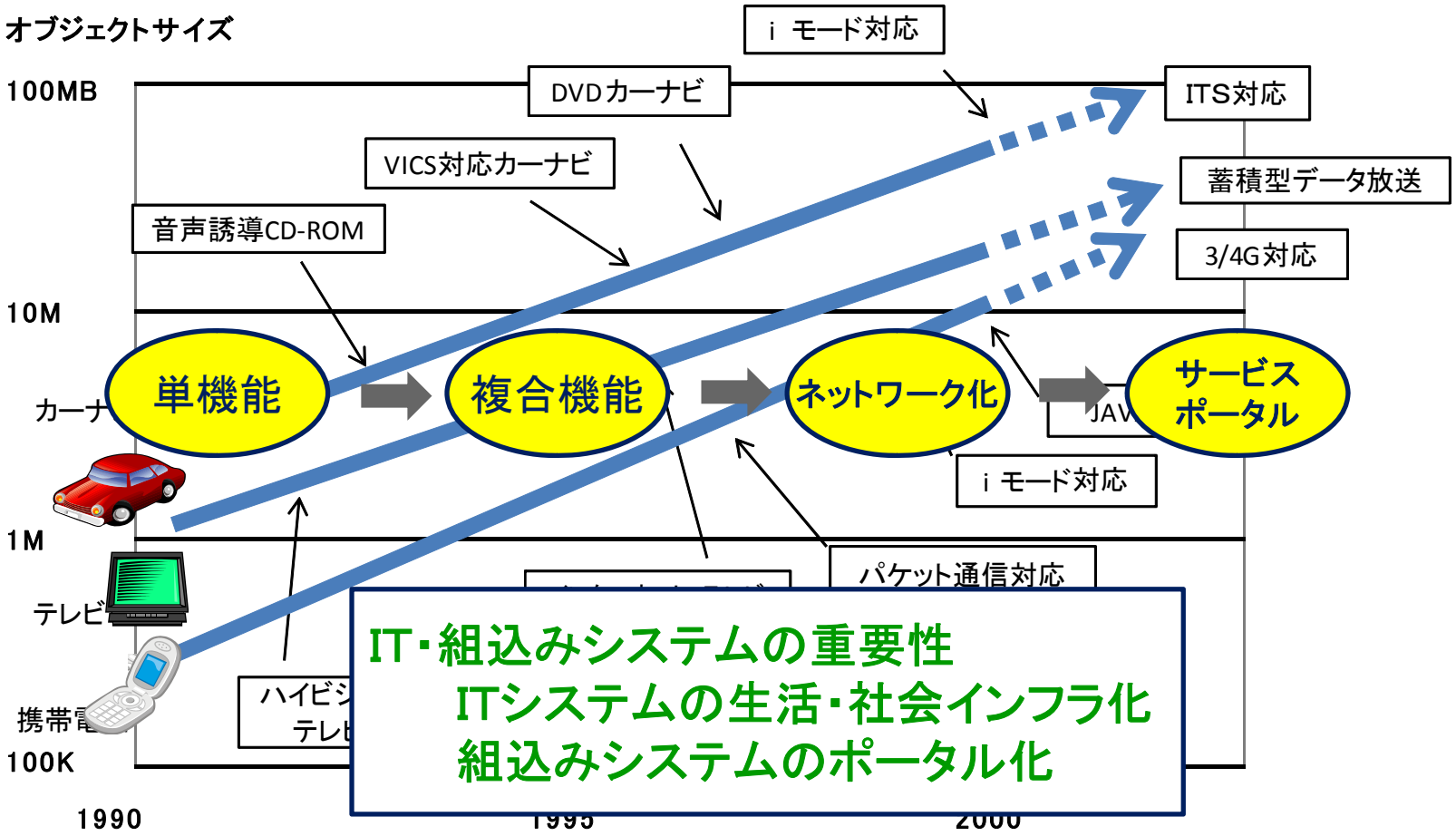
オープンシステムディペンダビリティとDEOSプロジェクト

Open Systems Dependability and the DEOS Project

2009年11月20日

所　眞理雄
JST/CREST　DEOS プロジェクト研究総括
（株）ソニーコンピュータサイエンス研究所 代表取締役社長

組込みソフトウェア規模の拡大とネットワーク化



出展：日経エレクトロニクス2009.9-11（no.778）をベースに追加、修正。
経済産業省「組込みソフト産業の課題と政策展開」平成19年11月14日より

大規模なシステム障害

システムの大規模化
- プログラムサイズ
- 多機能化
- ブラックボックス化したコンポーネント
- 複雑化

環境の変化
- 技術進化のスピード
- 接続システム

利害関係者の変化
- 要求の頻繁な変更
- 要求や合意に対する考え方

組込みシステムの現状

①組込みソフトウェアの規模と領域の拡大

②ネットワークの広がり

③常に要求・環境が変化する

- 開発プロセス・要件定義手法など従来技術は重要

であるが

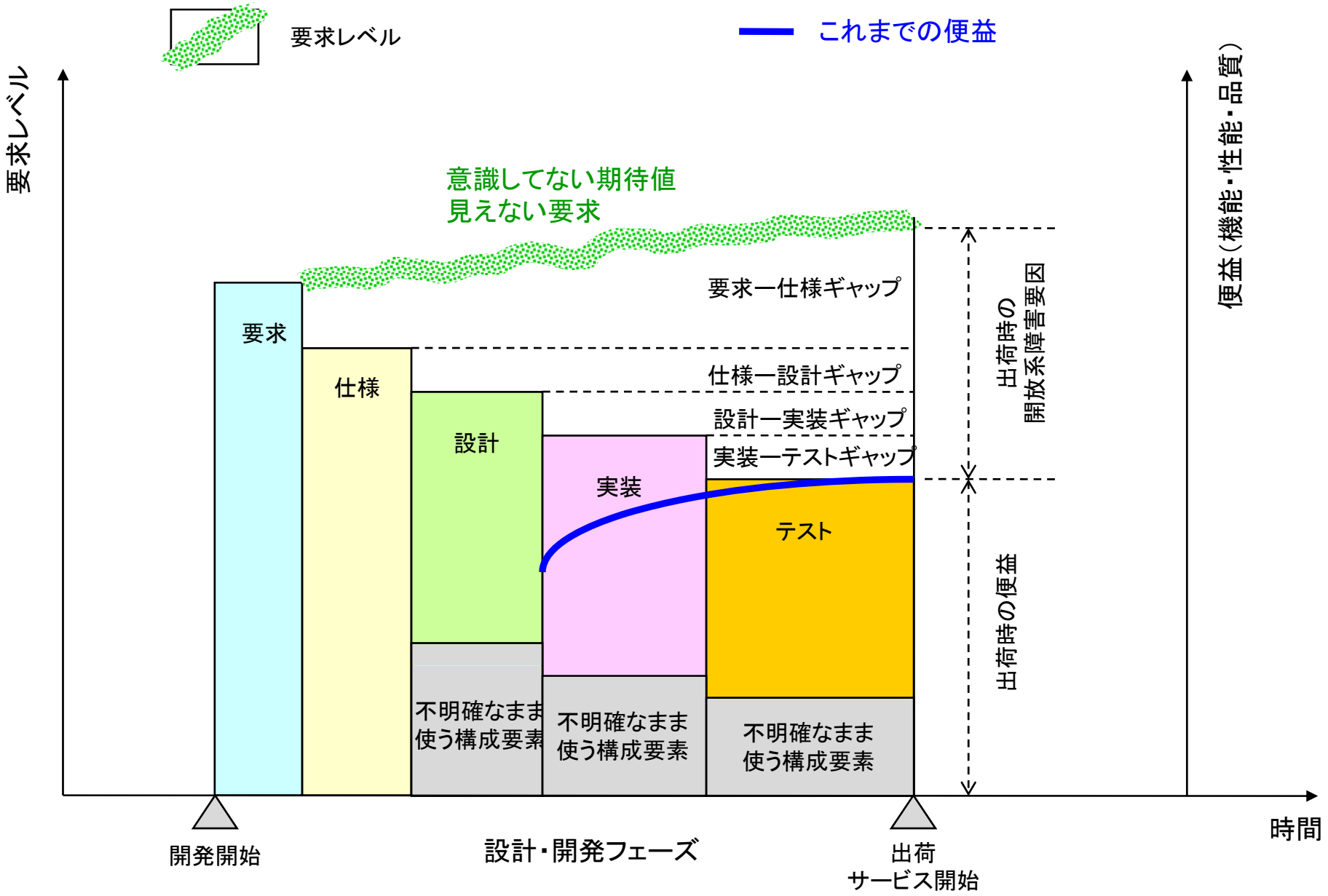
- 従来技術だけでは解決できない障害の解決の重要性

新たに対応の必要な分野の特徴

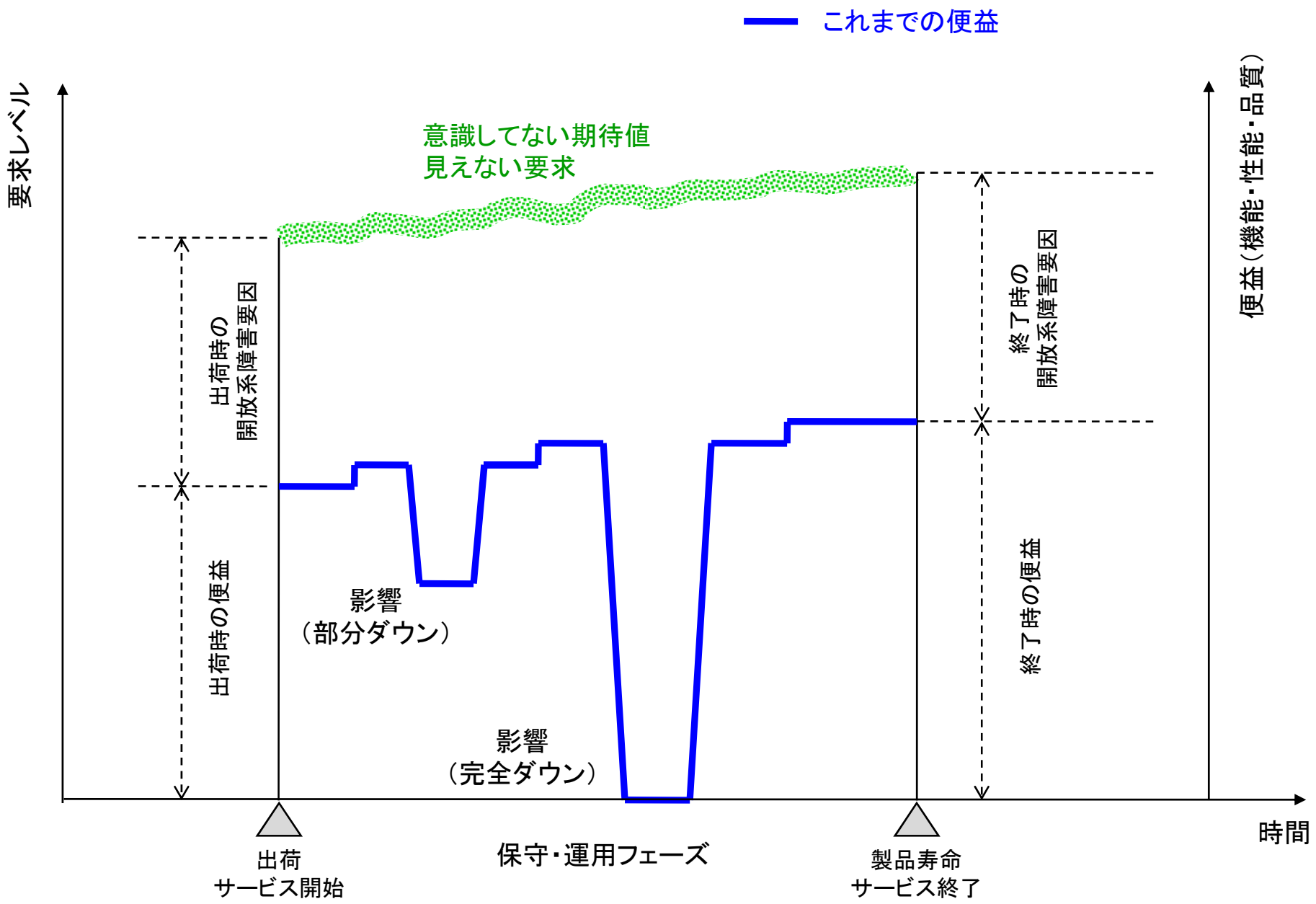
- **仕様/実装の不完全さ、システムの完全理解の困難さ**
  - 要求、仕様、設計、実装、テスト仕様、テスト実施の不完全さ
  - 構成要素の論理的不透明さ(複雑化、巨大化、ブラックボックス化、既存のコードの利用、他)
- **使用環境の変化に伴う不確実さ、システムの挙動予測の困難さ**
  - 要求事項・レベルの変化、想定外の使われ方
  - ネットワークを介しての構成要素の変化、想定外の接続
  - ネットワークを介した外部からの意図的な攻撃

⇒ オープンシステム(開放系)の問題

仕様/実装の不完全さ、システムの完全理解の困難さ



使用環境の変化に伴う不確実さ、システムの挙動予測の困難さ

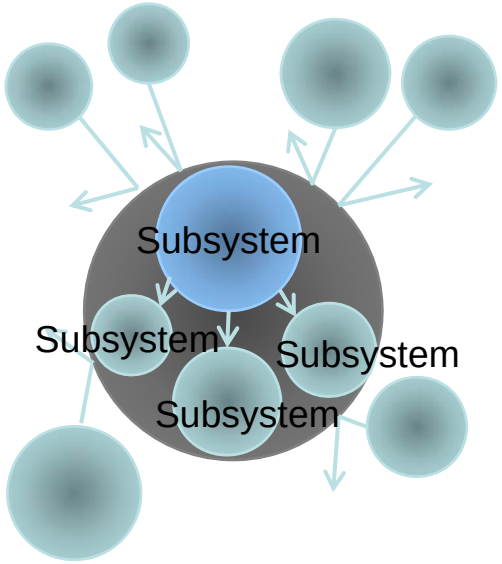


これからのディペンダビリティ

オープンシステムディペンダビリティ

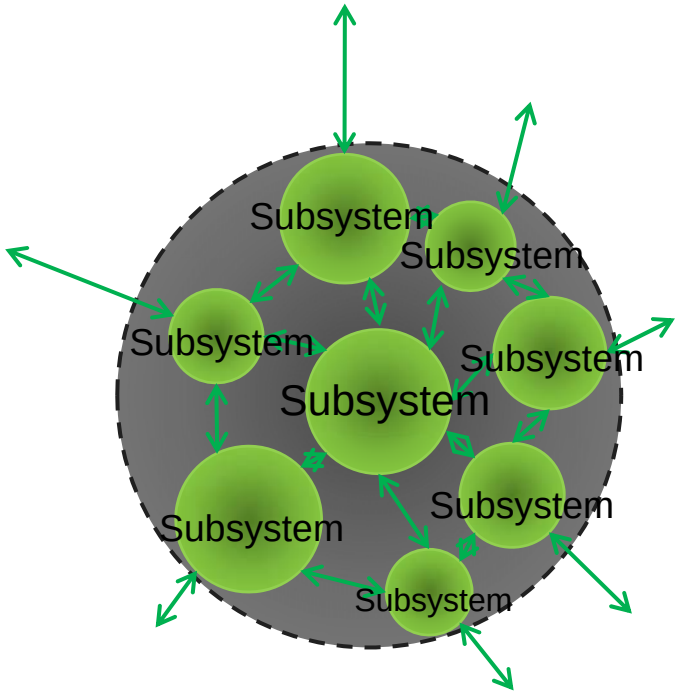
クローズドシステムとオープンシステム

クローズドシステム（閉鎖系）



- 分割して部分を理解すれば全体の問題が解決する
- 要素還元主義が成り立つ

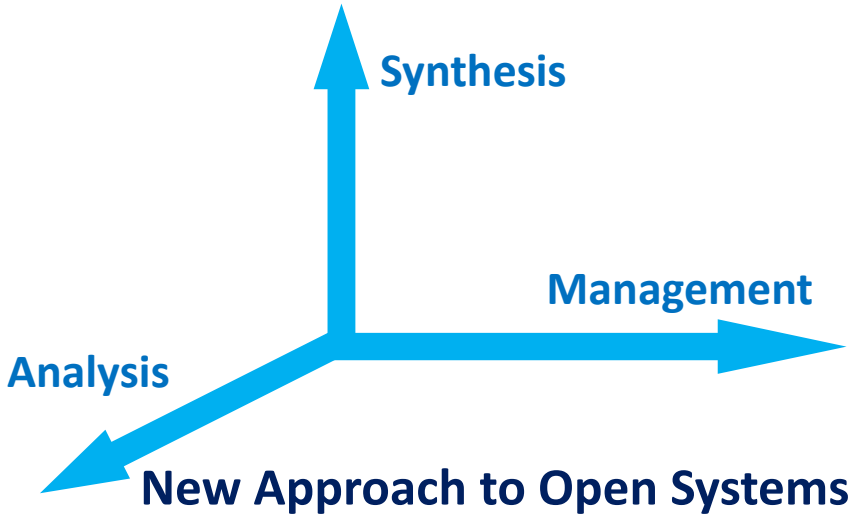
オープンシステム（開放系）



- サブシステム間の相互作用が動的に変化
- 境界領域や境界条件が動的に変化
- 定義や仕様が時間と伴に変わる
- 要素還元主義が成り立たない

オープンシステムに対処するには

- 外界に接する複雑なシステムを、生かしたまま、あるいは稼働させたまま（止めずに）問題を解決する
- サブシステムに分割しても相互依存性を保存する
- 抽象化しても捨象しない
- 「内部観測者視点」しか取りえないので、システムのモデルを現実（観測値）に一致するように常に保守する
- 「解析（Analysis）」と「合成（Synthesis」に加えて「マネージメント（Management)を統合・融合した新しいアプローチ



これまでのアプローチとの比較

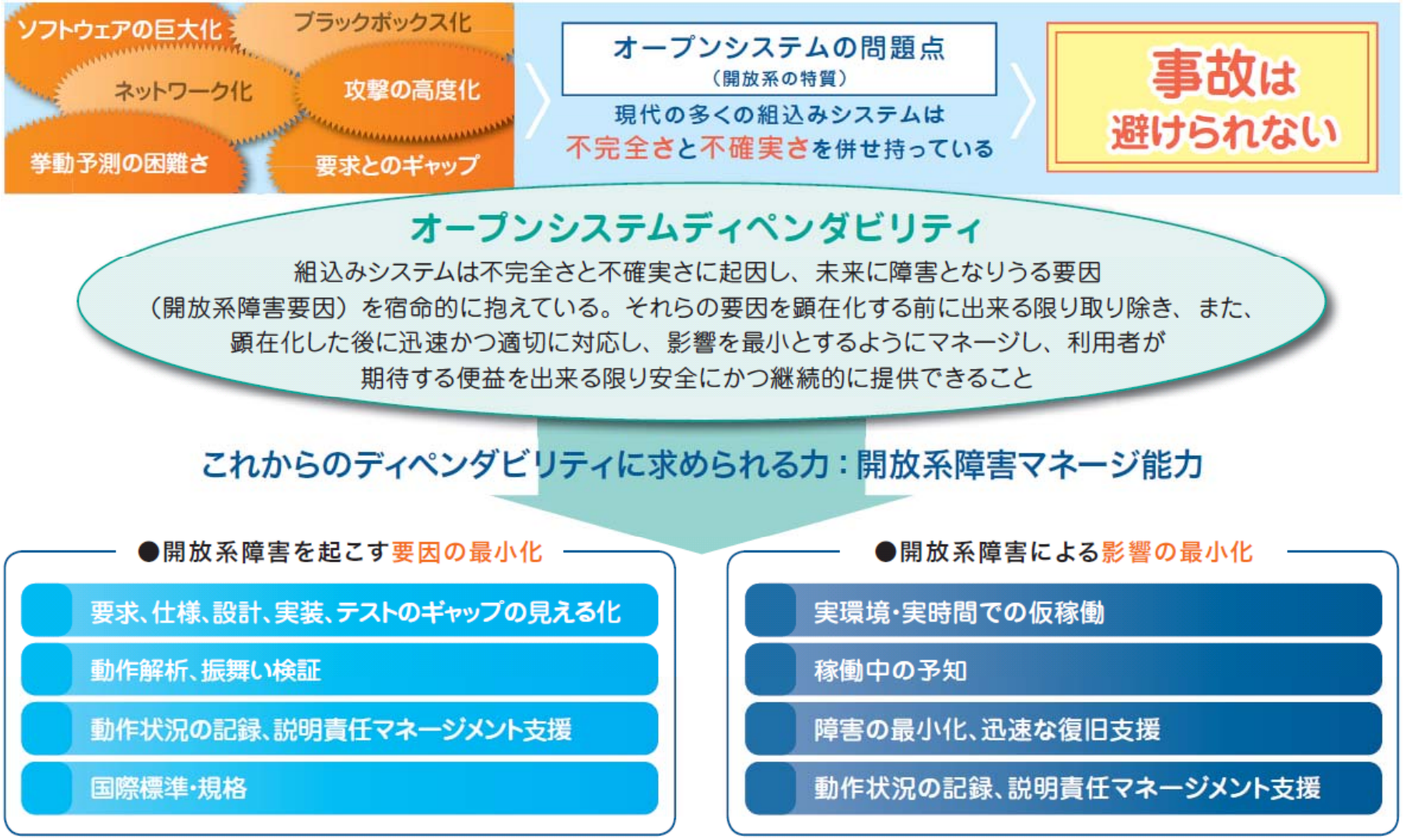
クローズドシステムアプローチ

- 単純なシステム
- 一旦止めてもよい
- 主に平衡系
- 基本原理の解明
- 外部観測者視点
- 要素還元・抽象化が可能
- 強力な解

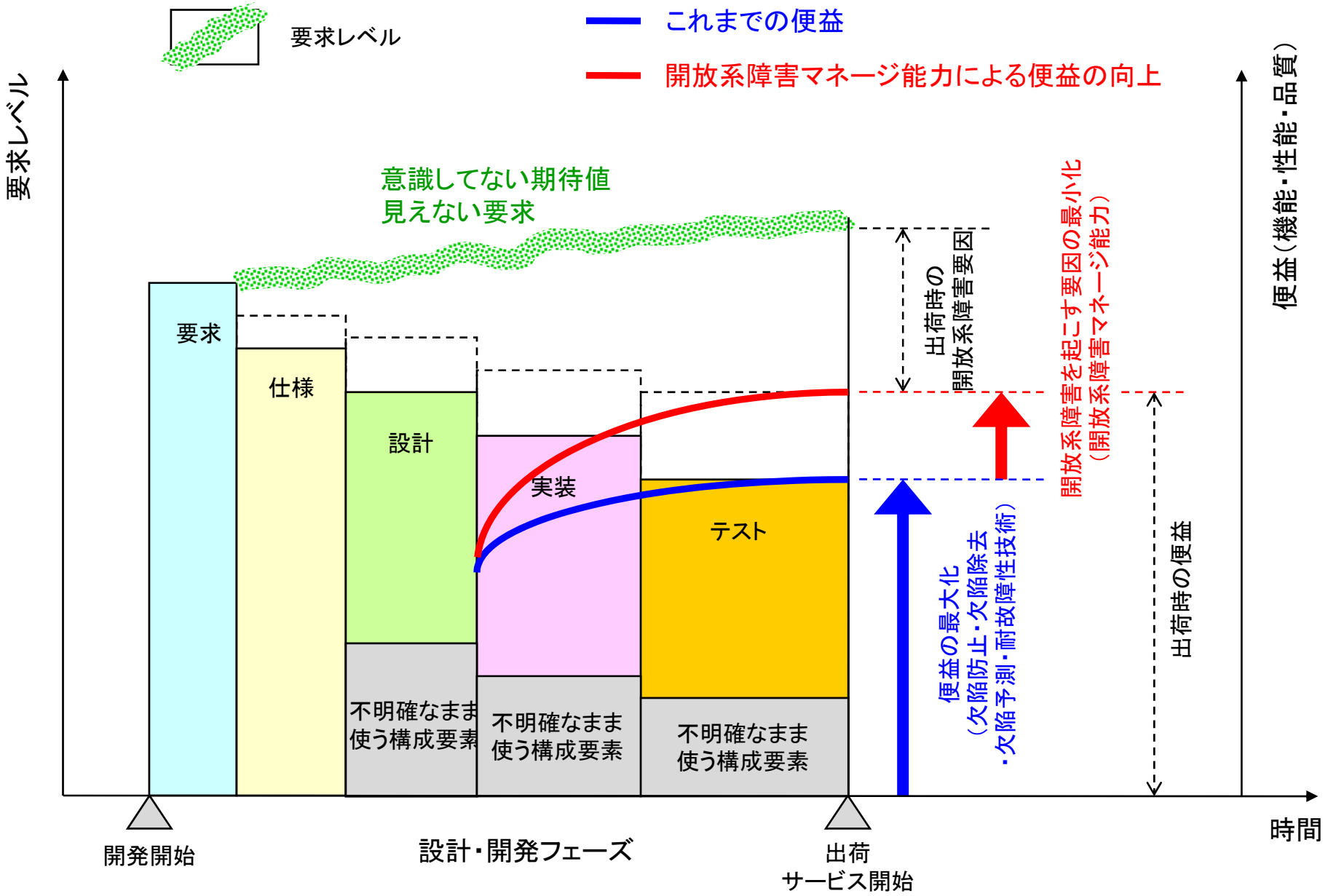
オープンシステムアプローチ

- 外に開いた複雑なシステム
- 生きたまま、稼働したまま
- 時間発展系
- システムの持続性・継続性と問題の解決（多様性・一回性の理解と解決）
- 内部観測者視点
- 要素還元・抽象化が不可能、常に全体視点
- ベストエフォートによる「運営」

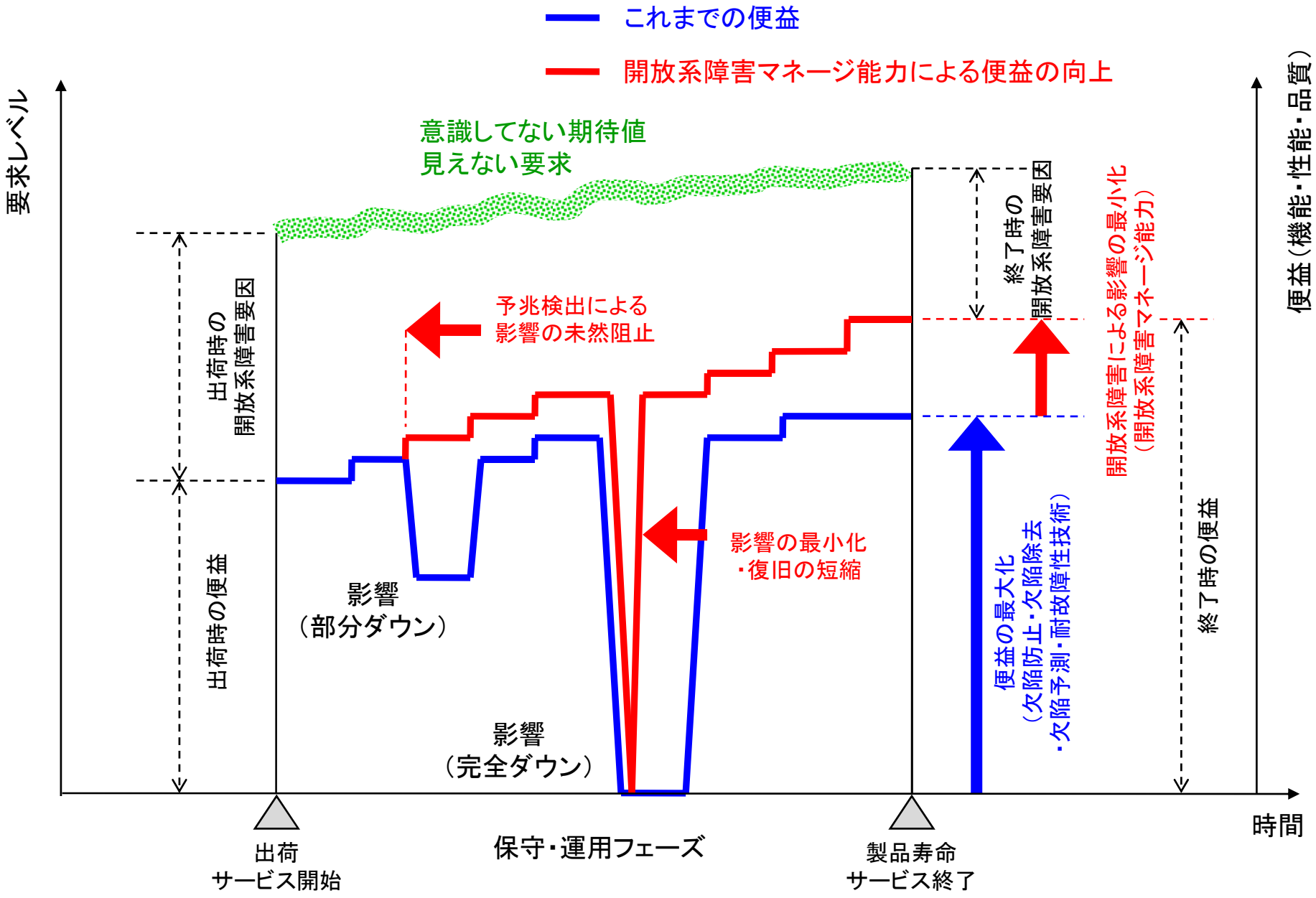
オープンシステムディペンダビリティ



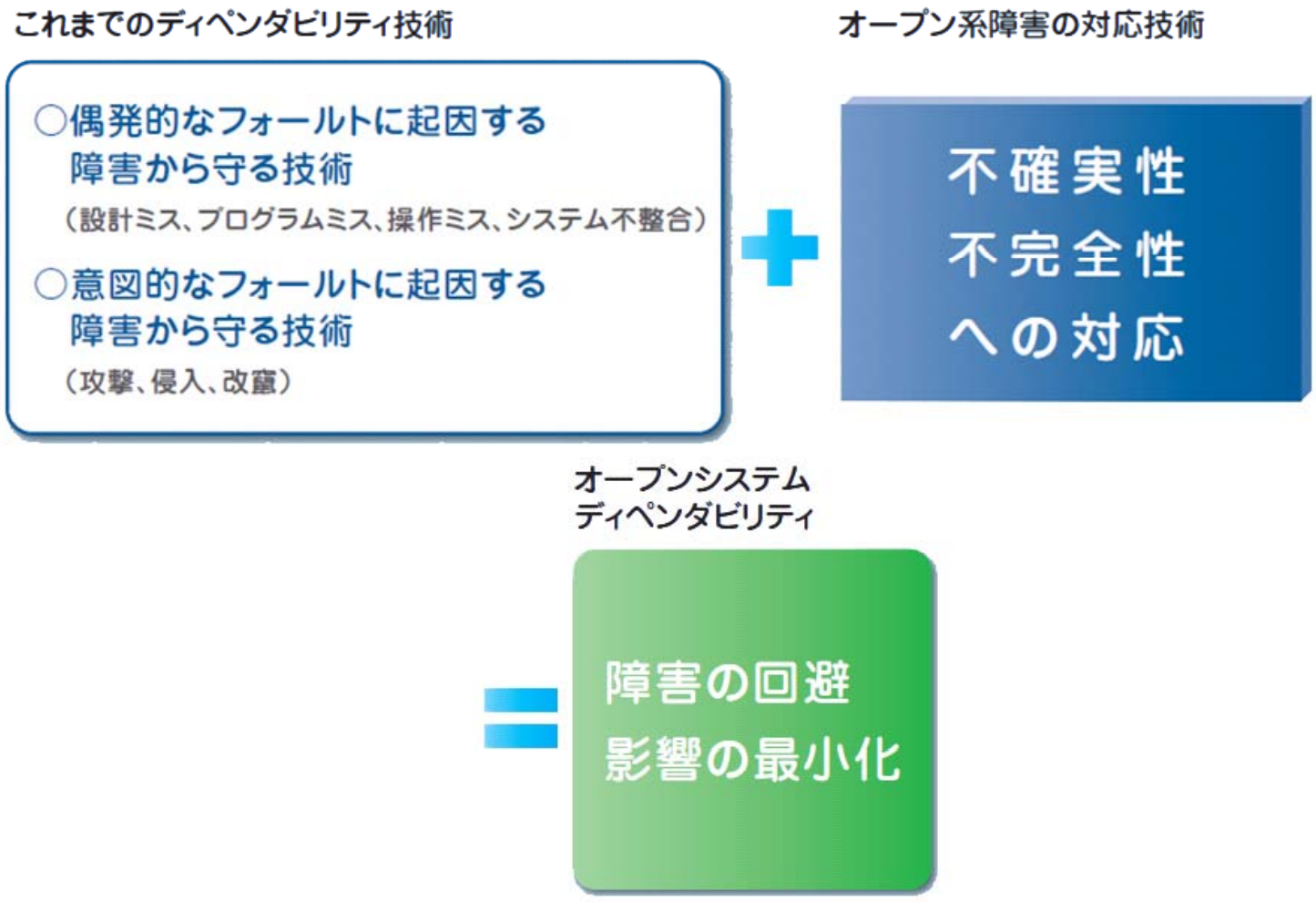
開放系障害を起こす要因の最小化



開放系障害による影響の最小化



ディペンダブル・システムとメーカーの責任



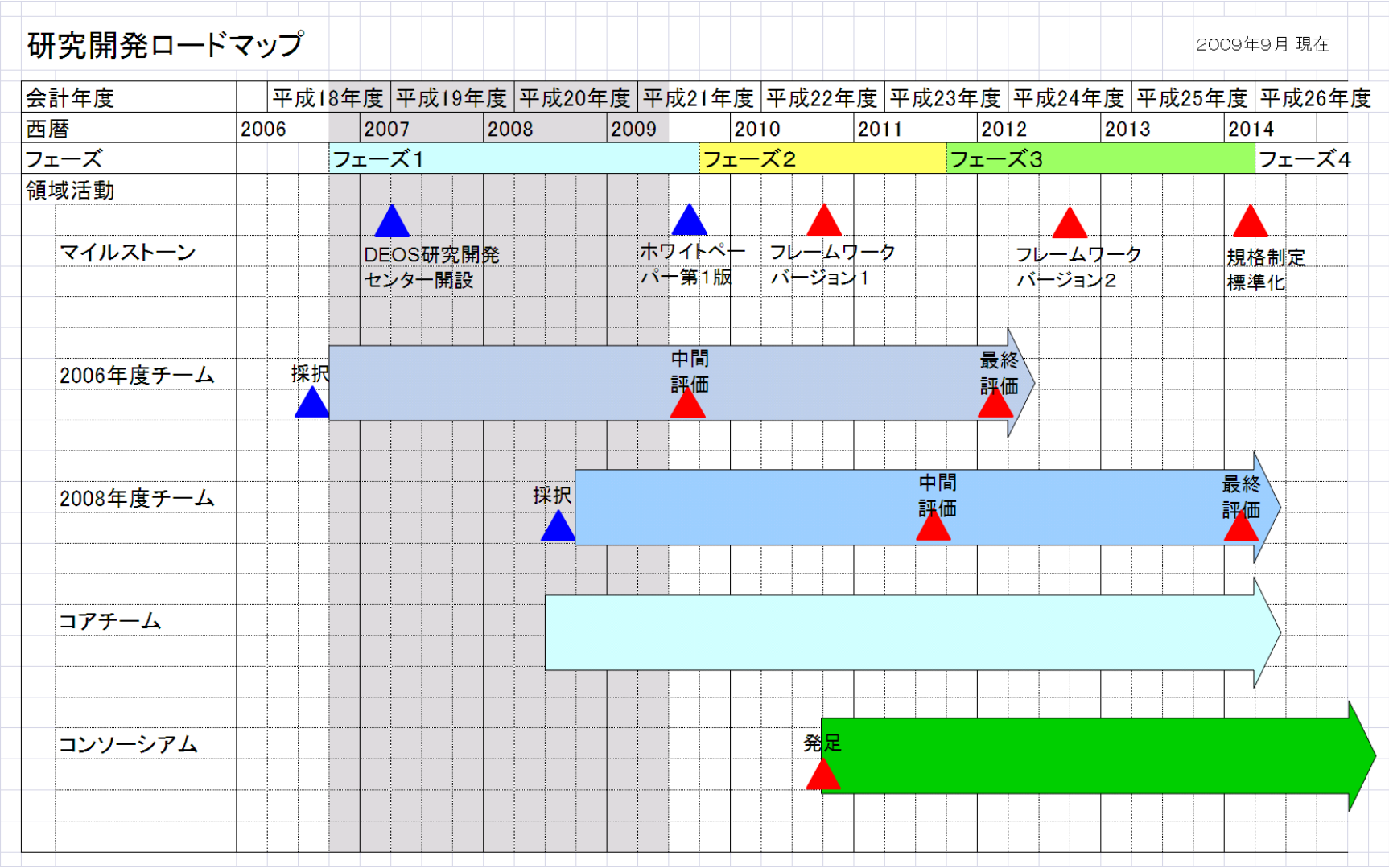
DEOS プロジェクト

本プロジェクトの目的

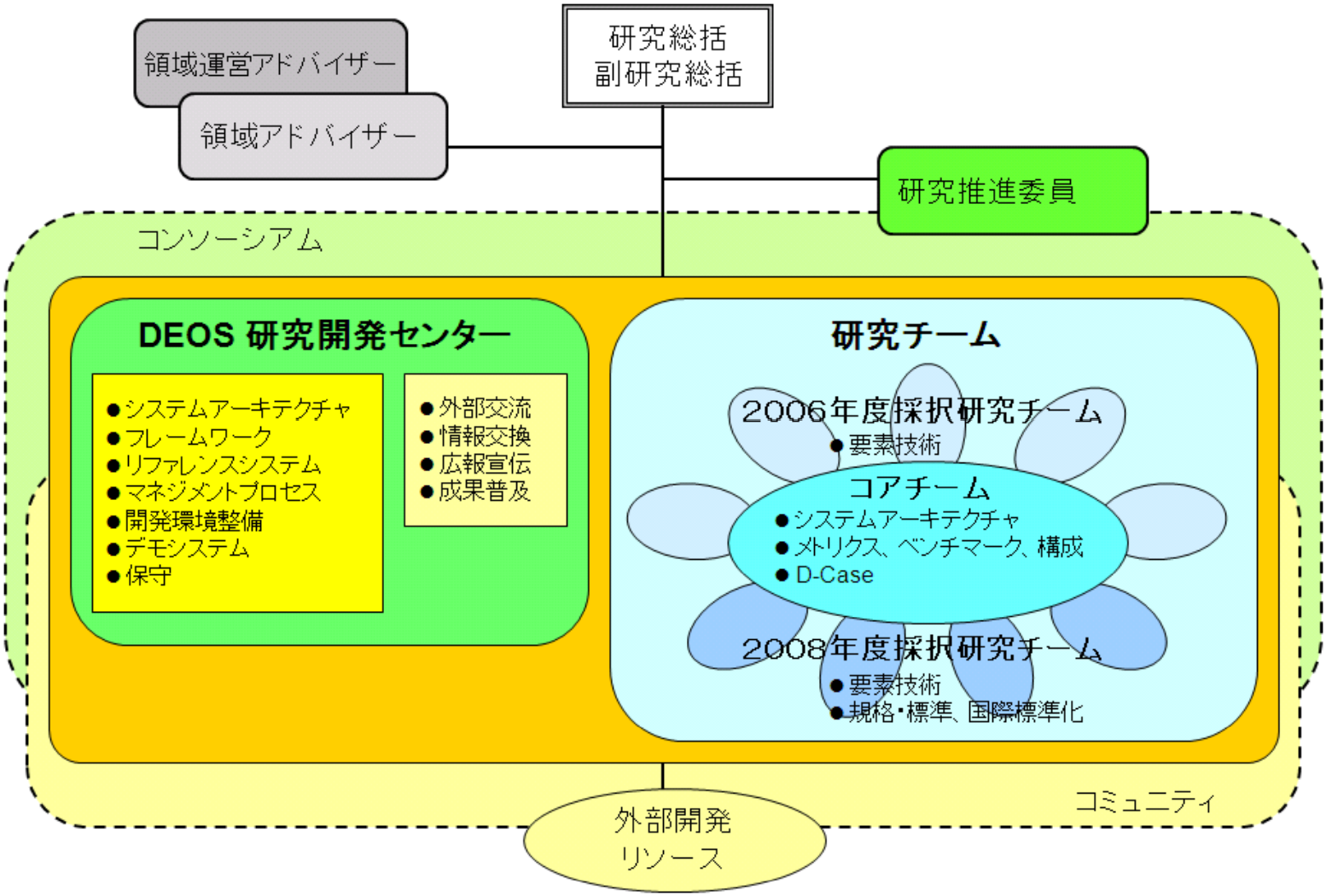
- オープンシステムディペンダビリティの観点からの組込みシステムにおける信頼性、安全性、使いやすさなどの要件の再検討

- 実用化を目指し、ディペンダブルシステムの実現に必要な概念、アーキテクチャ、仕様書・実装ガイドライン、マネジメントプロセス、開発環境・ツールなどの関連基盤技術も含めた開発・評価基準作成・見える化・標準化

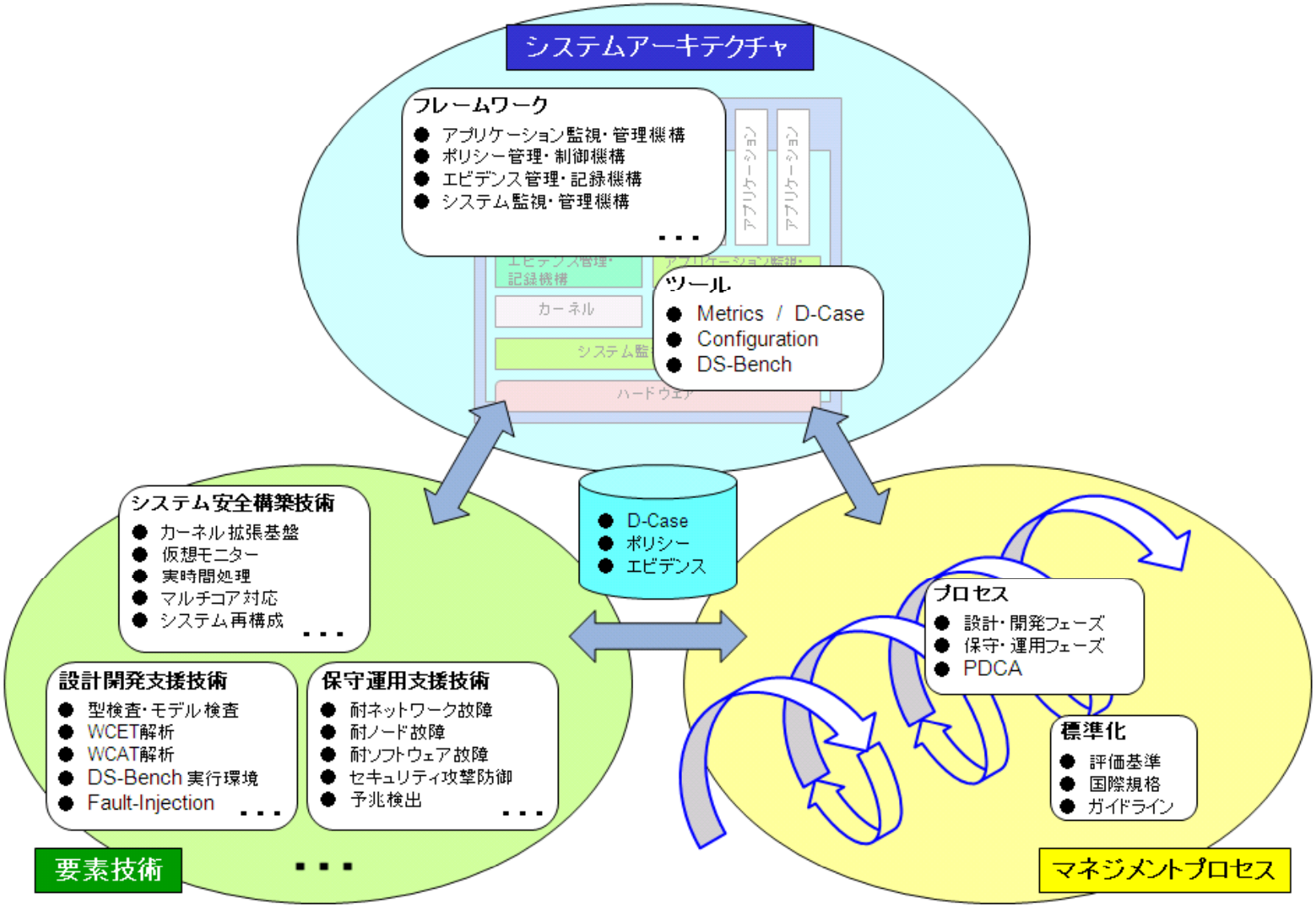
プロジェクトの経緯


研究開発ロードマップ
2009年9月 現在
会計年度
平成18年度
平成19年度
平成20年度
平成21年度
平成22年度
平成23年度
平成24年度
平成25年度
平成26年度
西暦
2006
2007
2008
2009
2010
2011
2012
2013
2014
フェーズ
フェーズ1
フェーズ2
フェーズ3
フェーズ4
領域活動
マイルストーン
DEOS研究開発センター開設
ホワイトペーパー第1版
フレームワークバージョン1
フレームワークバージョン2
規格制定標準化
2006年度チーム
採択
中間評価
最終評価
2008年度チーム
採択
中間評価
最終評価
コアチーム
コンソーシアム
発足

研究開発体制


領域運営アドバイザー
領域アドバイザー
研究総括
副研究総括
研究推進委員
コンソーシアム
DEOS 研究開発センター
●システムアーキテクチャ
●フレームワーク
●リファレンスシステム
●マネジメントプロセス
●開発環境整備
●デモシステム
●保守
●外部交流
●情報交換
●広報宣伝
●成果普及
研究チーム
2006年度採択研究チーム
●要素技術
コアチーム
●システムアーキテクチャ
●メトリクス、ベンチマーク、構成
●D-Case
2008年度採択研究チーム
●要素技術
●規格・標準、国際標準化
コミュニティ
外部開発
リソース

オープンシステムディペンダビリティ実現の仕組み



フレームワーク：D-Framework for Open Systems（D-fops）

■目標

- ◆ 開放系障害マネージ能力を発揮させる仕組み・構造を提供する
- ◆ マネジメントプロセス（企画・実施・監視・分析サイクル）を支援する仕組み・構造を提供する
- ◆ システムの進化を支援し、生きたままの変化を可能とする仕組み・構造を提供する

■検証すべき観点 – AMPI（安否）

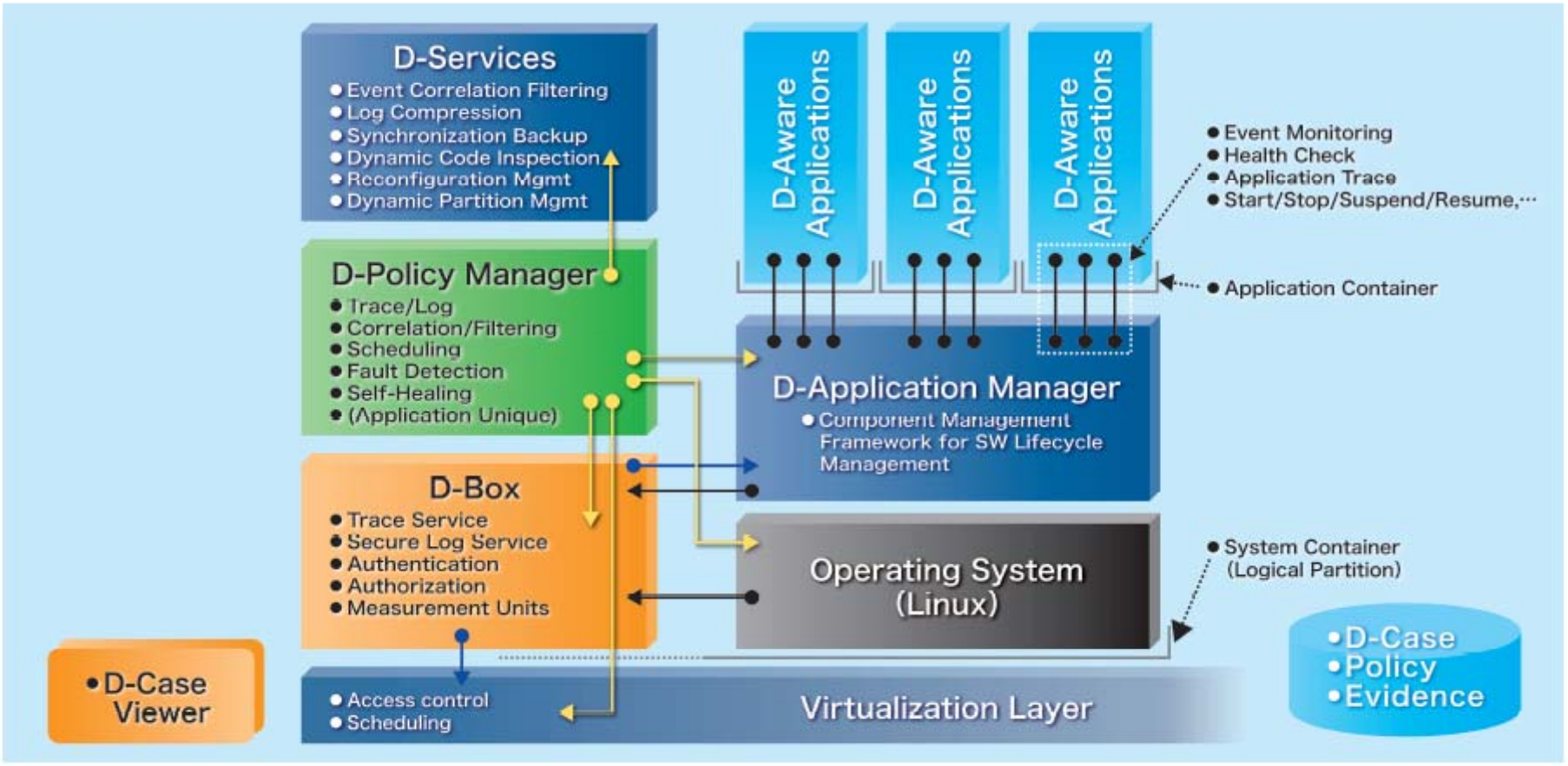
説明責任性	障害対応性	未然回避性	継続向上性
Accountability	Manageability	Preventability	Improvability

■実装が計画されている機能

- ● システムに起きるさまざまな事象を監視・記録する（Evidence Logging）
- ● 障害予兆を検出する（Failure Prediction）
- ● ポリシーに従って障害部位を切り離す（Failure Isolation）
- ● ポリシーに従って構成を変える（Reconfiguration）
- ● ポリシーに従って構成を元に戻す（Undo）
- ● 未来時間で試験する（Time-shift Rehearsal）
- ● 時間経過によって非効率になったシステムを若化させる（Software Anti-aging）
- ● システムの資源を制限したり、システムの動きを制御する（Quota）
- ● ポリシーの記述を助け、ライフサイクルを通して管理する（Policy Management）
- ● 開放系ディペンダビリティをステークホルダーと合意し、ライフサイクルを通して向上する（D-Case）

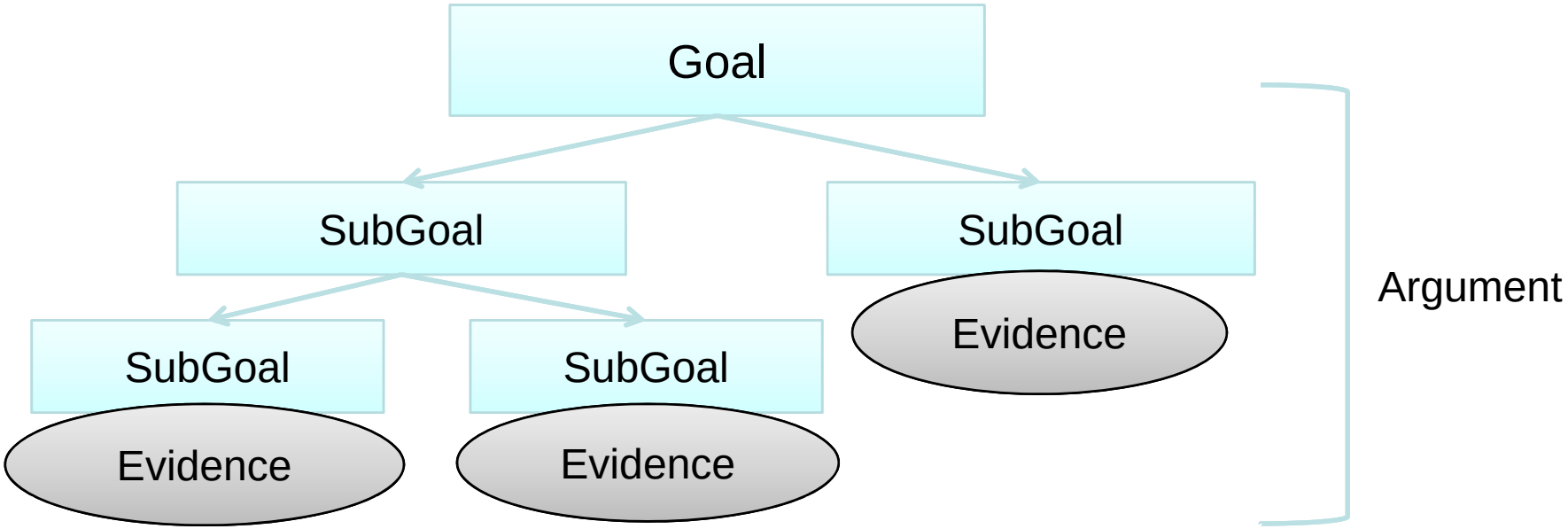
フレームワーク：D-Framework for Open Systems（D-fops）

■システム概念図

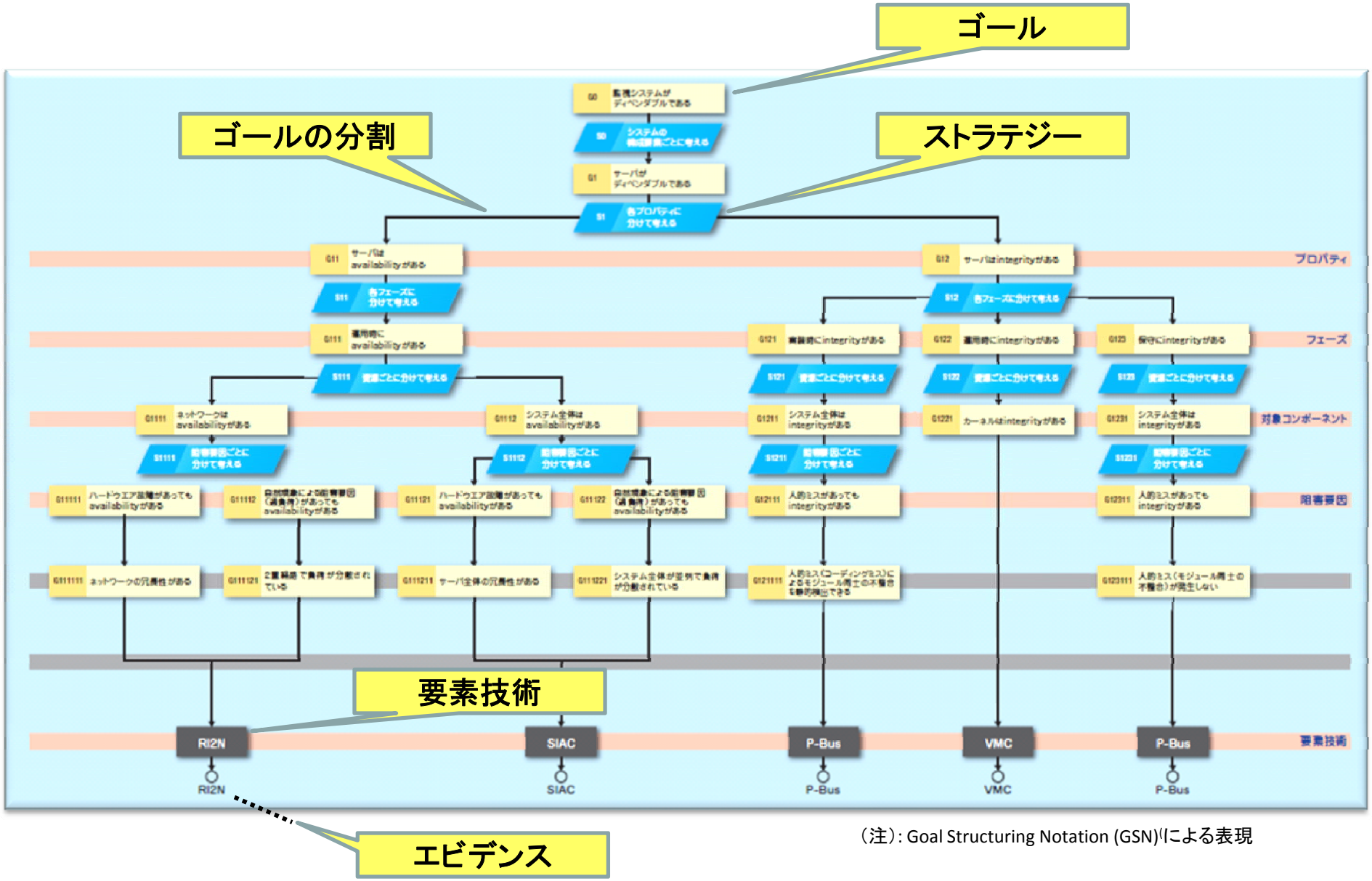


D-Case：ディペンダビリティ要件合意手法

- ステークホルダーとシステムのディペンダビリティ 要件(Goal)を議論 (Argument)しながら詳細化し、詳細化されたSubGoalをベンチマークなどの根拠 (Evidence)を示すことにより保証し、ステークホルダーどうしでGoalを合意し、ライフサイクルを通して継続向上保守するための手法



D-Case表記例



（注）: Goal Structuring Notation (GSN)による表現

ディペンダビリティの継続向上：マネジメントプロセス

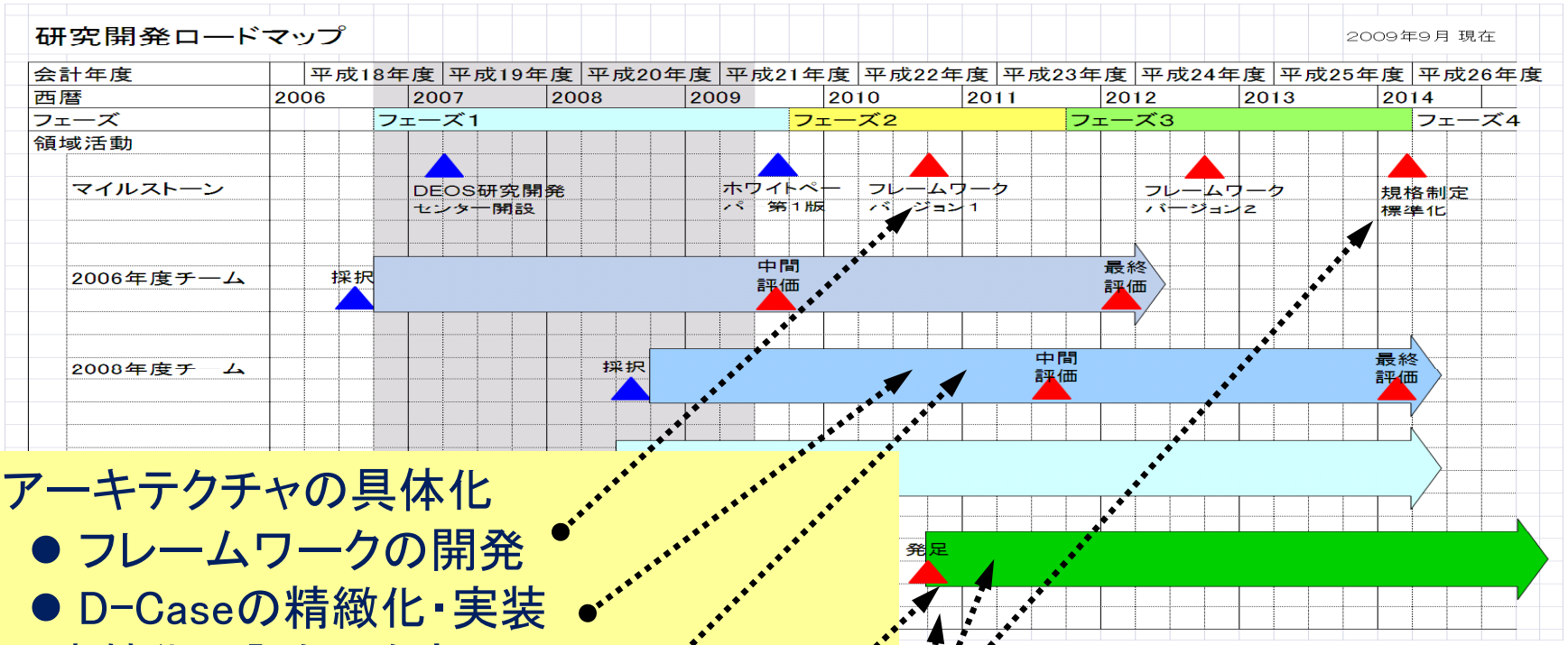
- オープンシステムディペンダビリティ評価基準の規格化・国際標準化
- オープンシステム向上サイクル・マネジメントプロセスの規格化・国際標準化
- ライフサイクルを通して継続向上保守するための手法・ツール
  - D-fops/D-Case
- プロセスガイドラインの発行、改訂、普及活動
- コンソーシアムによるベストプラクティスの共有

関連規格：IEC 61508 Functional Safety
IEC 60300-1 Dependability Management
ISO/IEC 12207 Software Life Cycle Processes
ISO/IEC 15026 Systems and Software Assurance
ISO/IEC 15288 System Life Cycle Processes

オープンシステムディペンダビリティの価値

- ディペンダビリティ確保のためのライフタイムコストの低減
  - システム安全技術、設計支援技術、保守運用技術などの要素技術、システムアーキテクチャ、ならびにマネジメントプロセスにより、統合的にディペンダビリティを確保し、日々改良・改善を行うことができる
- 企業の社会的責任の全うを支援
  - 障害原因ならびに回復状況の迅速な説明支援
  - 設計開発・保守運用過程の分かりやすい説明支援

今後の計画



- ◆ アーキテクチャの具体化
  - ● フレームワークの開発
  - ● D-Caseの精緻化・実装
- ◆ 要素技術の評価・改良
  - ● セキュリティ技術の融合
- ◆ マネジメントプロセスの確立
  - ● 実システムへの適用・評価
  - ● 規格の制定ならびに国際標準化
- ◆ 実用化
  - ● オープンソース化
  - ● コンソーシアム設立による継続的支援