ディペンダブルシステム技術とその実用化

- Dependable Systems Technology and its Application -

2010年6月17日

Embedded Technology West 2010 @インテックス大阪

独立行政法人　科学技術振興機構
ディペンダブル組込みOS研究開発センター
屋代　眞

目次

- 科学技術振興機構（JST）
- 組込みシステムソフトウェアの展望
- ディペンダブルシステムとメーカーの責任
- ディペンダブルシステム構築技術
- デモ
- プロジェクト概要

JST ― 独立行政法人 科学技術振興機構

JST = Japan Science and Technology Agency

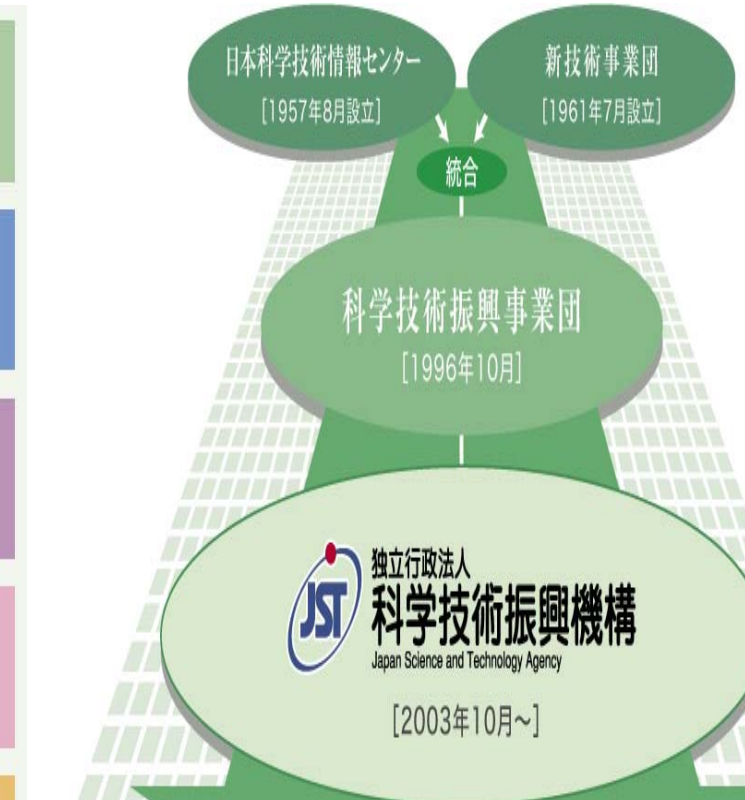
Creating advanced technology
新技術の創出に資する研究

Promoting technology transfer and innovation
新技術の企業化開発

Promoting dissemination of scientific and technological information
科学技術情報の流通促進

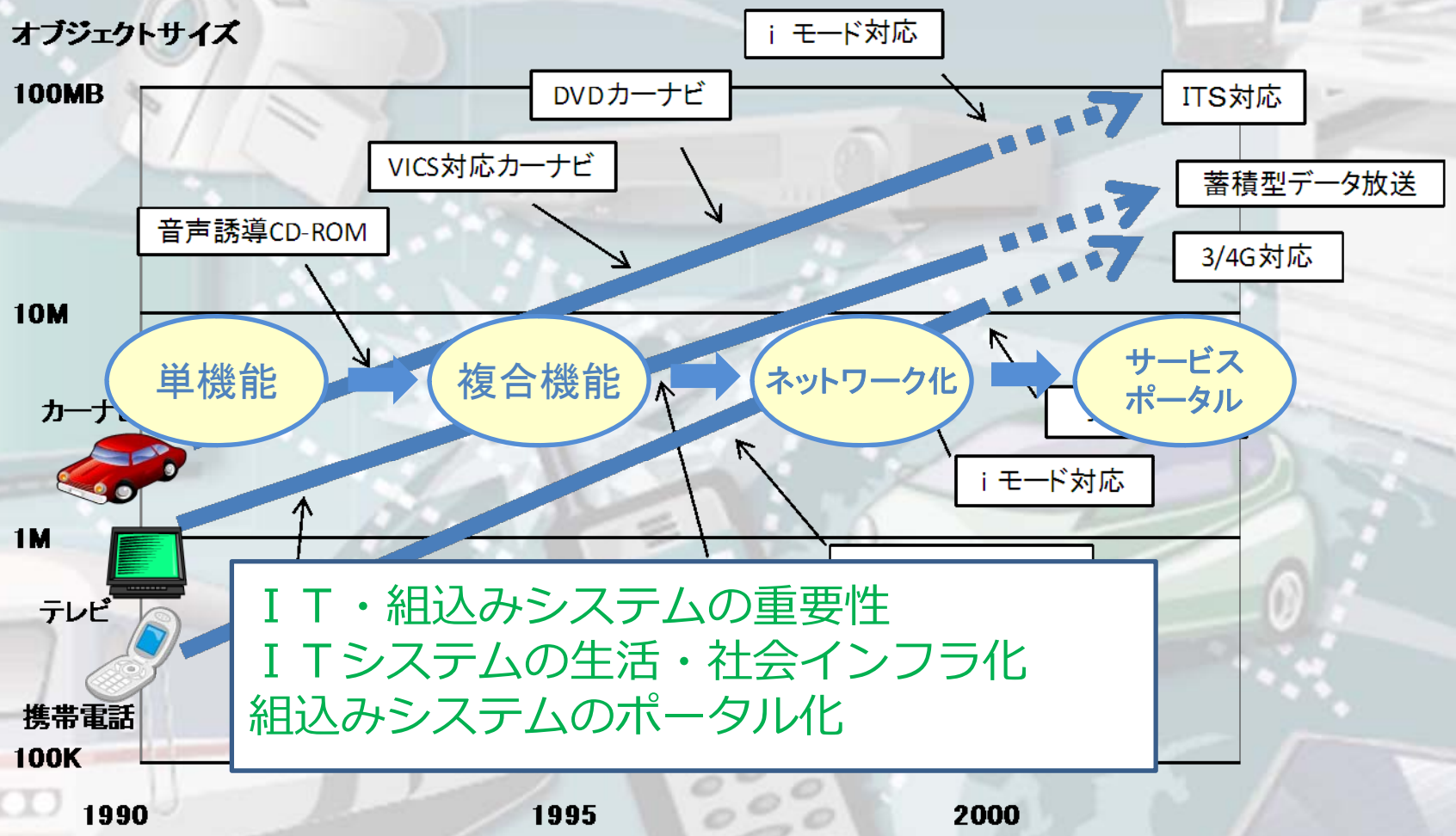
Researcher exchange and research support
科学技術に関する研究開発に係る交流・支援（研究開発に係る交流・支援）

Promoting public understanding of science and technology
科学技術に関する知識の普及、国民の関心・理解の増進（科学技術理解増進）



JST Home Page: http://www.jst.go.jp/gaiyou.html

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

障害は避けられない

- 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）
  - 人間の能力の限界
- 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

サービスの継続（Sustainability）

説明責任（Accountability）

サービスの継続 （Sustainability）

➢ 開放系障害を起こす要因の最小化

- 要求、仕様、設計、実装、テストのギャップの見える化
- 動作解析、振舞い検証
- 動作状況の記録、説明責任マネジメント支援
- 国際標準・規格

➢ 開放系障害による影響の最小化

- 実環境・実時間での仮稼働
- 稼働中の予知
- 障害の最小化、迅速な復旧支援
- 動作状況の記録、説明責任マネジメント支援

説明責任 （Accountability）

➢ サービスの継続のための最大限の努力と適切なリスクマネジメント

- 法令・標準・社内プロセスなどの遵守
- 証拠・事実に基づいた判断と記録
- ステークホルダーとディシジョンプロセスの明確化
- 被害を最小限にするためのマネジメント
- 同じ失敗を繰り返さない仕組み

➢ 証拠の提示に基づく社会と顧客の理解

- 事象や意思決定・合意形成の記録
- 顧客・ユーザーの不満の解消
- 信頼の維持回復
- 顧客離れの阻止
- ビジネス便益の確保

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
- 主に平衡系
- 要素還元・抽象化が可能
- 分析的視点
- 再現可能
- 基本原理の解明

- 一旦止めてもよい
- 外部観測者視点
- 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
- 時間発展系
- 要素還元・抽象化が不可能
- 常に全体視点
- 再現が不可能
- システムマネジメントの視点

- 生きたまま、稼働したまま
- 内部観測者視点
- ベストエフォートによる「運営」

「組込みシステムは不完全さと不確実さに起因し、未来に障害となりうる要因（開放系障害要因）を宿命的に抱えている。それらの要因を顕在化する前に出来る限り取り除き、また、顕在化した後に迅速かつ適切に対応し、影響を最小とするようにマネージし、利用者が期待する便益を出来る限り安全にかつ継続的に提供できること」

オープンシステムディペンダビリティ実現に必要な技術は？

新しいコンピューティングパラダイム（**Evidence Based Computing**）

- ステークホルダー視点でライフサイクルを通じて製品やサービスを向上させる。
- ステークホルダーの合意に基づいてサービス継続や向上のための最適なアクションをとる。
- “Evidence”に基づいて製品やサービスの最適な復旧を行う。

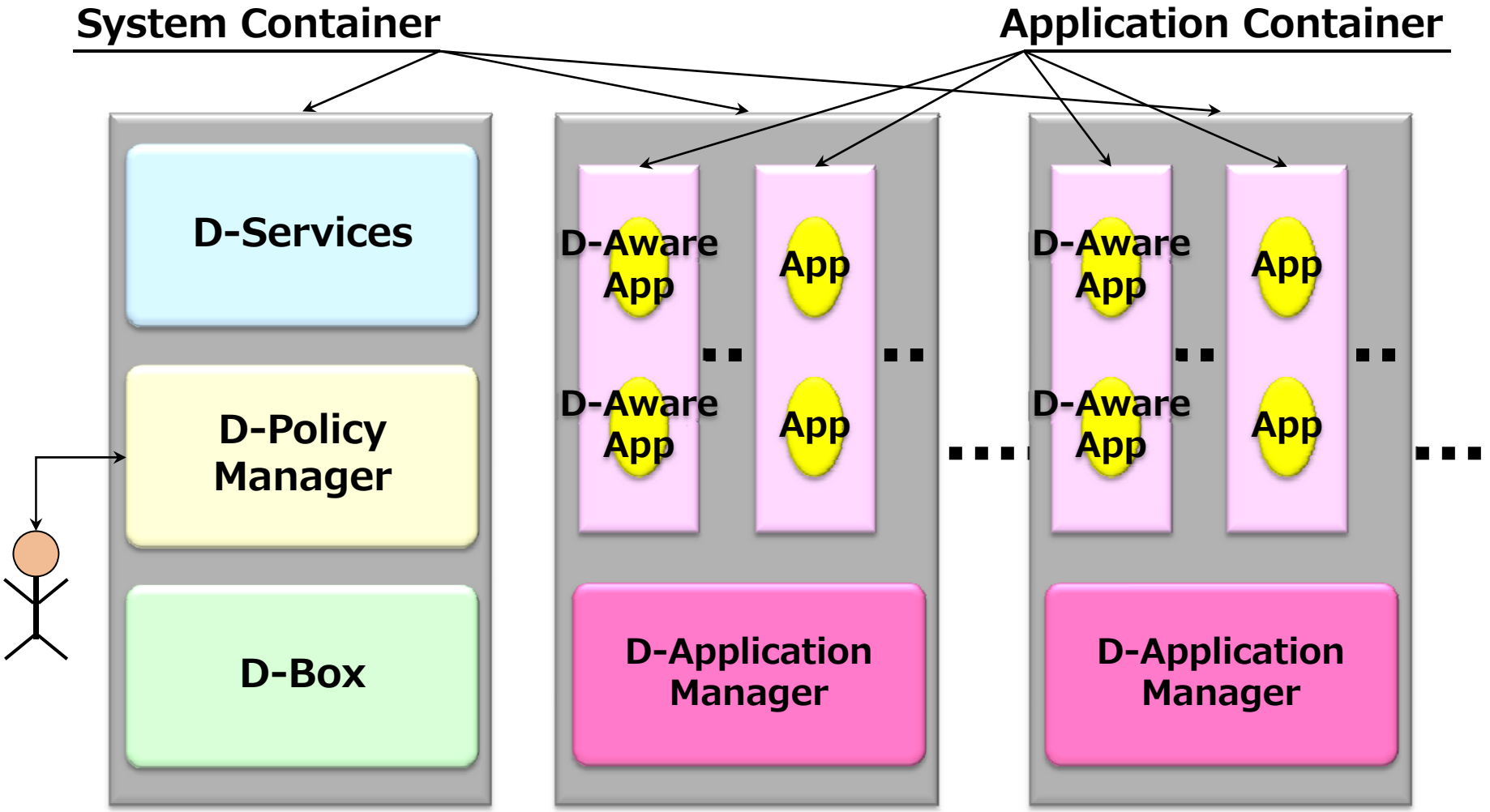
オープンシステムディペンダビリティーのフレームワーク

- 開放系のシステムにおいてそれに対応した
  - デザイン・運用ができる
  - 障害が起こった時に原因の把握・修復が迅速にできる
  - 障害を含むシステムの挙動やシステムの構成を証拠をもって説明できる
  - プロセスや標準に準拠している
  - サポートツールがある
- 開放系のシステムをサポートするシステムサービス
  - 証拠の提示（Presentation of evidences）
  - 分離（Isolation）
  - 時間シフトテスト（Time-shift testing）
  - ソフトウェアアンチエイジング（Software anti-aging）
  - 再実行（Undo）
  - 予測可能性（Proactive management）
  - 資源配分（Quota）
- ライフサイクルを通じたサポートのためのシステムサービス
  - 構成・変更管理（Configuration management）
  - ポリシー管理（Policy management）
  - D-Case管理（D-Case management）

D-fops: *D*ependability *F*ramework for *Op*en *S*ystems

- D-fopsは製品やサービスにおいて、オープンシステムディペンダビリティーを実現するための実装例
- D-fopsは製品やサービスのライフサイクルを通じてオープンシステムディペンダビリティーを実現するための評価システム
- D-fopsは製品やサービスのライフサイクルを通じてオープンシステムディペンダビリティーを実現するためのツール群を提供

Architecture Overview

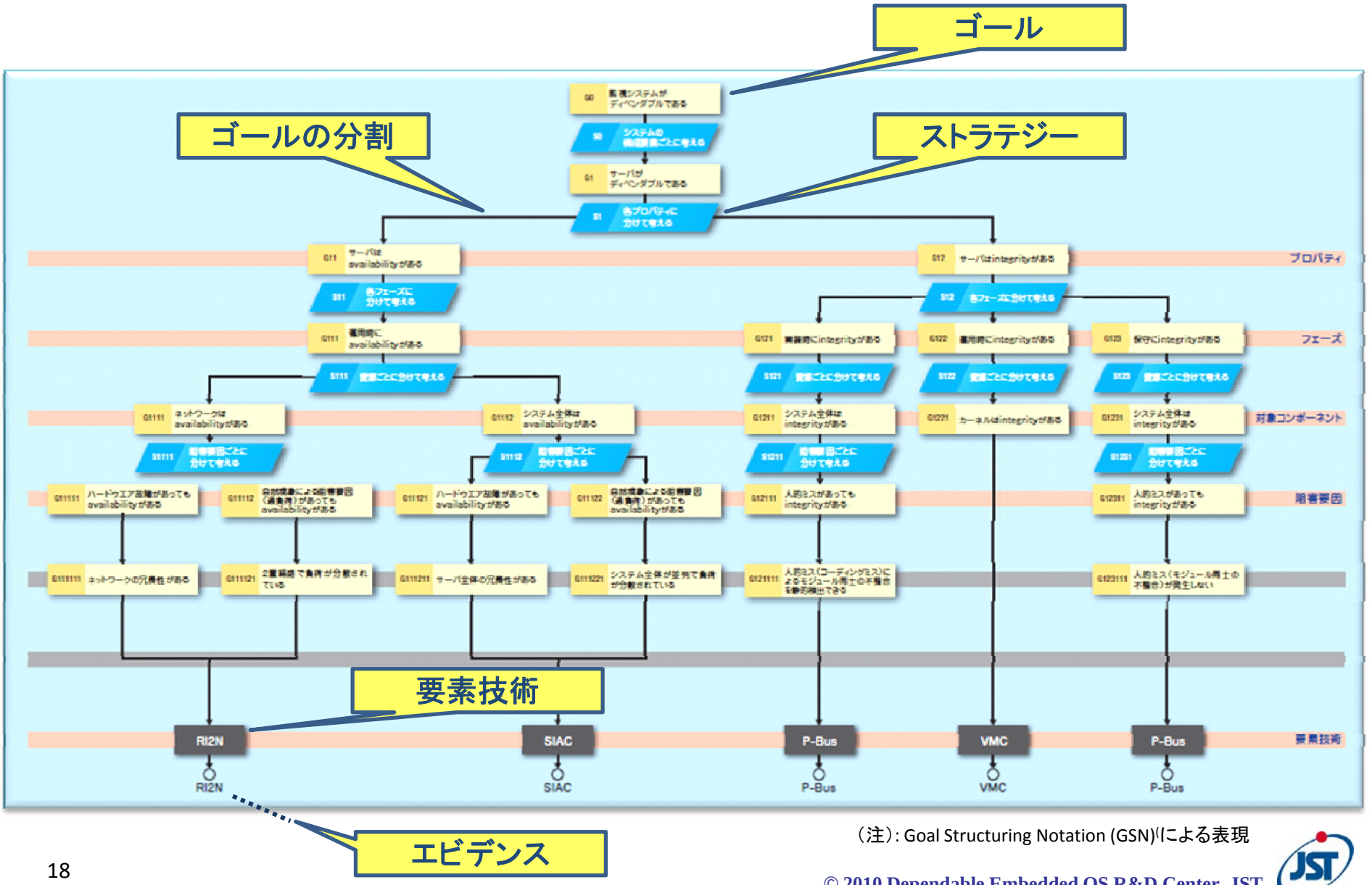


オープンシステムディペンダビリティーのライフサイクルサポート

- システムのライフサイクルにおける製品・環境の変化に対応できるディペンダビリティー特性の表現
- システム構成とディペンダビリティー指標との関連付け
- ディペンダビリティーに関するシステムデザインの選択、ステークホルダーの合意、ステークホルダー間の合意のための議論とその論拠の明示
- 上記の実現のために機器あるいはそれに連結したDBへの内蔵

D-Case

D-Case表現の例（監視システムの部分）



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

D-Case文法（案）

ゴール

G0 : DEOS has availability

前提付きゴール

G9 : All real-time processes are available.

where : G2, G10

ゴールの参照

G2

コンテクスト

C0 : Definition of availability is in [2]

エビデンス

E0 : DSBench A

未達成ゴール

要素技術

ACT

G9 : All real-time ...

where : G2, G10

ストラテジ

S0 : Consider ...

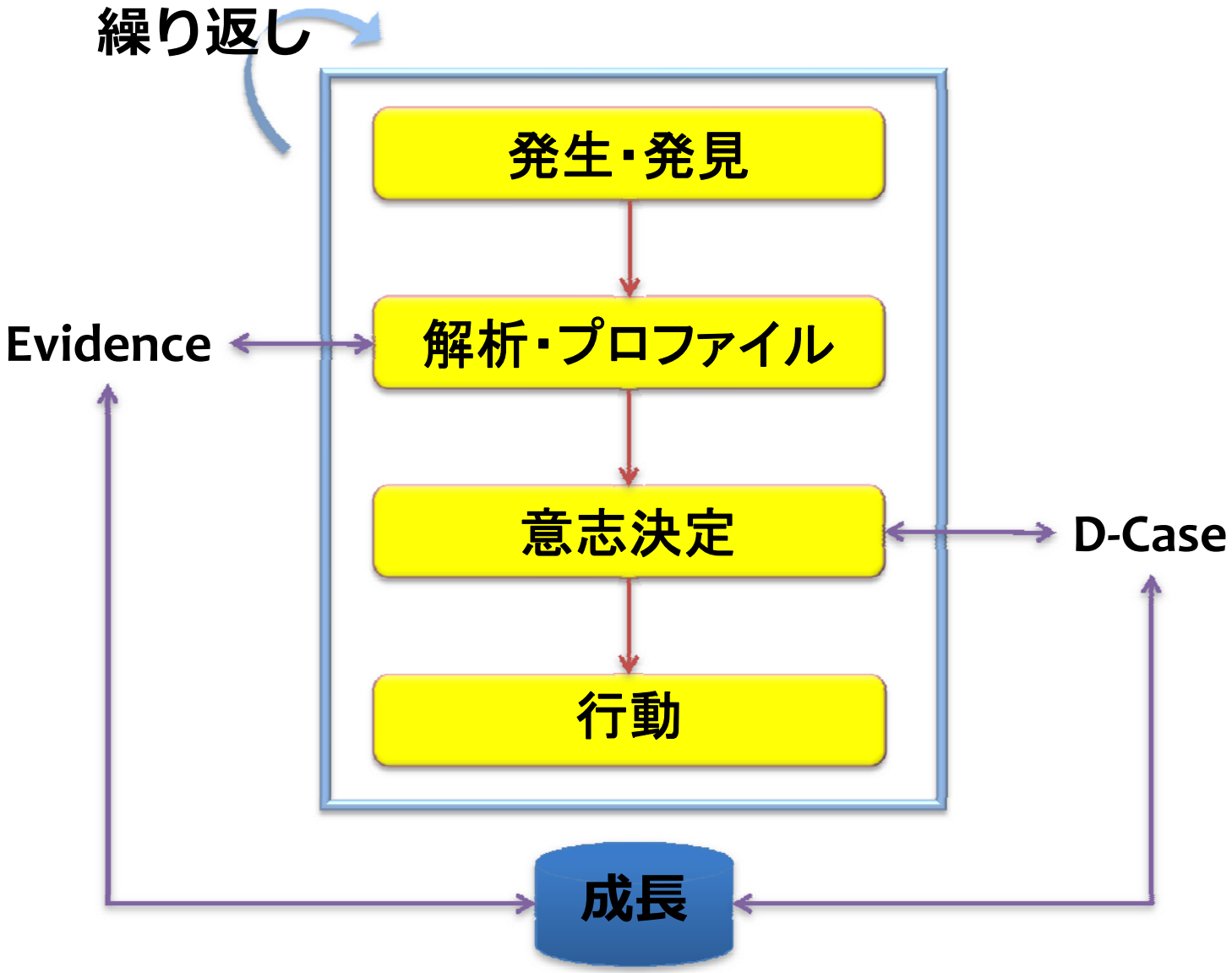
ゴールの分割

G0 : DEOS has availability

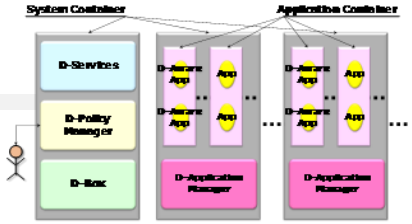
G1 : ...

G2 : ...

Basic Flow of Control



Software Structure



D-Aware Applications

Legacy Applications

D-Tools

Application Container Provider

D-Application Manager
[providing applications with D-awareness such as lifecycle management & containment service]

D-Box

D-Policy Manager

D-Services

Operating Systems (several instances such as Linux)

System Container Provider (virtualization/micro-kernel)
[trusted, hardware-assisted, & time constrained hardware resource partitioning]

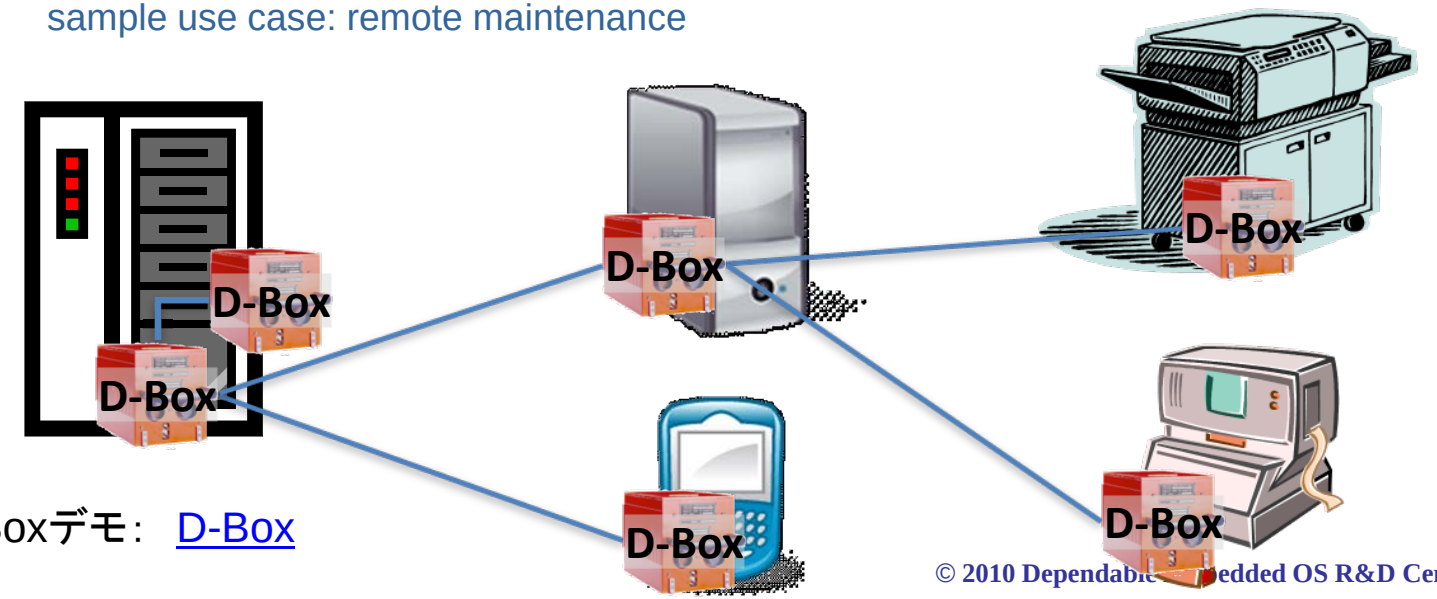
Hardware

D-Box – 証拠の保全

- D-BoxによるEvidence-Based Computingの実現
  - “証拠”を安全に保管
  - 各機器に一つのD-Box
  - D-Box の連鎖を構築可能
- D-Boxへの安全なアクセス
  - Authenticated
  - 暗号化による通信
- D-Boxに格納する内容の例
  - System Configuration
  - Event/Log 記録
  - Time-stamp
  - 暗号・認証 Key
- D-Boxはフライトレコーダーのようなもの

D-Box Network

- D-Box Networkはディペンダビリティーの連鎖を表現する。オープンシステムディペンダビィリティーはD-Boxをリンクさせることにより実現できる。
- それぞれのD-Box はその機器のディペンダビリティーに関連した“証拠”を安全に保持する。
- D-Box間で“証拠”に関する情報を交換することによりディペンダビリティーの状況を把握することができる。
  - sample use case: remote maintenance



D-Boxデモ： D-Box

デモ

- Video Captureと顔認識プログラムを同時に走らせたときに予期せずCPUの使用率が限度を超えてしまった

  ⇒　ポリシーに従った対応

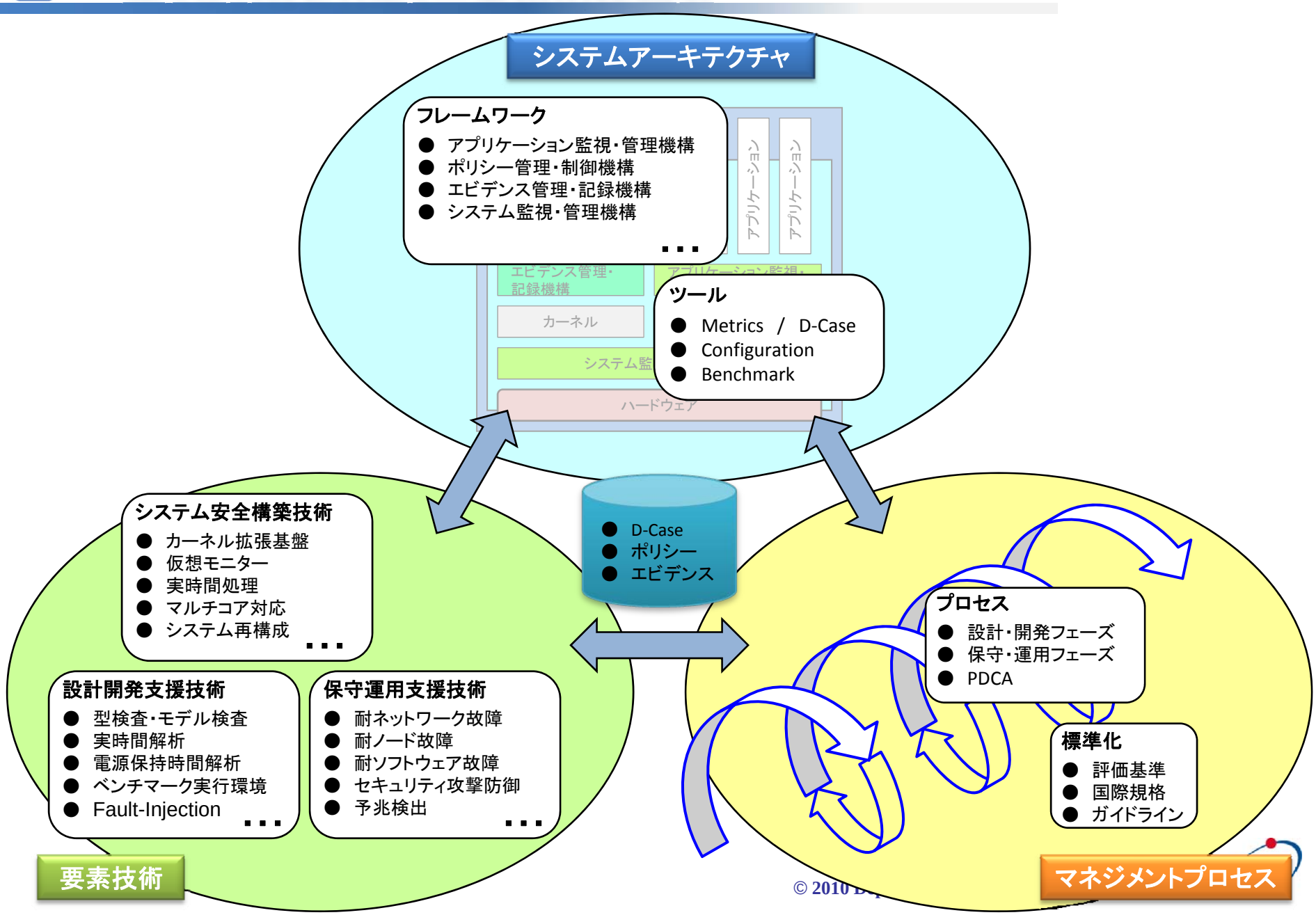
  - 松：　CPU割り当て制限
  - 竹：　Program Undo
  - 梅：　Program終了

- システムツールの例
  - System Container Creation
  - Application Container Creation
  - D-Box
  - System Migration

まとめ:オープンシステムディペンダビリティの価値

- サービス継続による**ユーザー便益**の維持確保
  - 予兆を検出、もしくは事前に障害を検出して障害回避
  - 障害回復時間の短縮
- サービス継続による**ビジネス便益**の維持確保
  （**収益機会**の維持、**ライフサイクルコスト**の低減）
  - 予兆を検出、もしくは事前に障害を検出して障害回避
  - 障害回復時間の短縮
  - 起きてしまった障害の以後の再発を防止
- 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

ディペンダビリティー実現の仕組み



研究チーム概要

H18年度採択	研究代表者		
石川 裕	東京大学 情報基盤センター	教授	並列・分散型組込みシステムのためのディペンダブルシングルシステムイメージOS
佐藤 三久	筑波大学 計算科学研究センター	センター長	省電力でディペンダブルな組込み並列システム向け計算プラットフォーム
徳田 英幸	慶應義塾大学 環境情報学部	教授	マイクロエビキタスノード用ディペンダブルOS
中島 達夫	早稲田大学 理工学術院	教授	高機能情報家電のためのディペンダブルオペレーティングシステム
前田 俊行	東京大学 大学院情報理工学系研究科	助教	ディペンダブルシステムソフトウェア構築技術に関する研究

H20年度採択	研究代表者		
加賀美 聡	(独)産業技術総合研究所デジタルヒューマン工学研究センター	副センター長	実時間並列ディペンダブルOSとその分散ネットワークの研究
大下 佳樹	(独)産業技術総合研究所 産学官連携推進部門 関西産学官連携センター 組込システム技術連携研究体	主幹研究員	利用者指向ディペンダビリティの研究
倉光 君郎	横浜国立大学 大学院工学研究院	准教授	Security Weaver とPスクリプトによる実行中の継続的な安全確保に関する研究
河野 健二	慶應義塾大学 理工学部	准教授	耐攻撃性を強化した高度にセキュアなOSの創出

D-fops　D-Case　検証　ベンチマーク　ヒビテンス　VM

プロジェクトの計画

研究開発ロードマップ

2010年5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

フェーズ　1　（2006/10 – 2009/9）
ディペンダビリティコンセプト確立・システムアーキテクチャの提示

フェーズ　2　（2009/10 – 2011/9）
フレームワーク・レファレンスシステムの実装とユーザーによる評価
オープンコミュニティー・コンソーシアム立ち上げ

フェーズ　3　（2011/10 – 2014/3）
フレームワーク・レファランスシステムの試用と評価

フェーズ　4　（2014/4 – ）
オープンコミュニティー・コンソーシアムによる活用・維持・発展

展示会(組込み総合技術展)

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

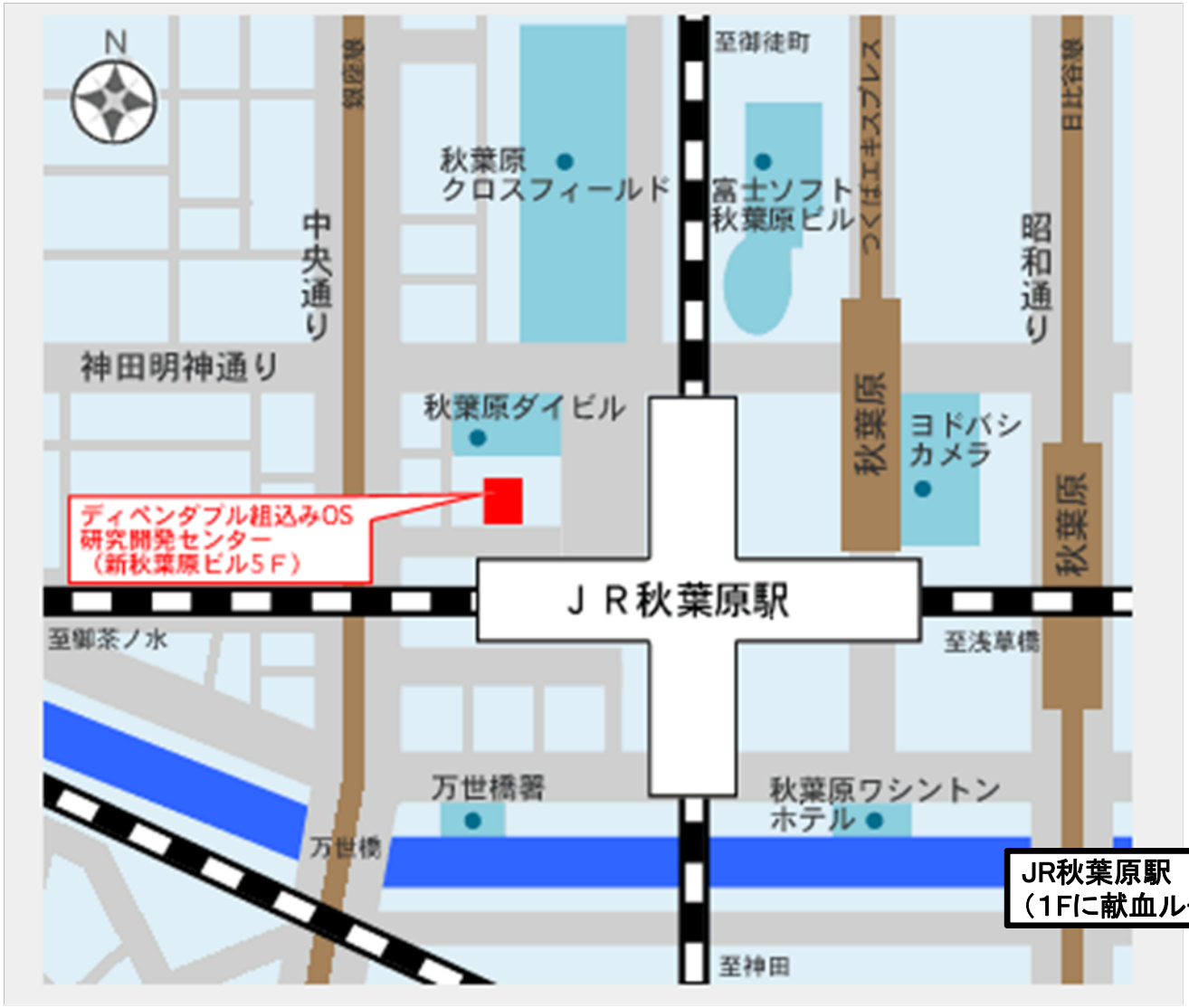
コンソーシアムに向けて

- ディペンダビリティー技術のためには企業の枠を超えた協力が必要
  - プロジェクトへのフィードバック
  - 企業・ユーザー間の情報の共有とそのしくみ
  - 共同でのディペンダビリティー技術の開発
  - PDCAを通じた実践とマネジメントプロセス進化の継続

- コンソーシアムの会員を募集します
  - 連絡先： center@dependable-os.net
  - 情報： www.dependable-os.net

DEOSセンター案内図

DEOSセンター　検索



www.dependable-os.net
center@dependable-os.net

JR秋葉原駅　電気街口を出てすぐ
（1Fに献血ルームあり）