

「実用化を目指した組込みシステム用  
ディペンダブル・オペレーティングシステム」  
平成 18 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

## § 1. 研究実施の概要

当研究領域では、組込み機器を含む情報システム的设计・開発・運用・保守のそれぞれの段階にわたってシステムのディペンダビリティを向上させていくための手法や技術を研究している。平成 21 年度当研究課題では、以下の研究課題に取り組んだ。

### 石川グループ

システムのディペンダビリティを向上させるための要素技術として、OS カーネル拡張インターフェース P-Bus、シングルシステムイメージ機構、論理分割機構、静的最悪実行時間予測ツールの研究開発を進めた。さらに、システムのディペンダビリティを評価するための手段として、ディペンダビリティベンチマークフレームワークを設計し、プロトタイプを開発した。

### 島田グループ

研究領域横断で議論を重ねて提案されている、ディペンダビリティの形式記述法である D-Case について、具体的で実用的な検討を加速するための研究用のツールとして開発環境の中で動作する D-Case ツールを開発する必要があると研究領域全体で判断された。本開発では、公開に向けた特許や保守などの課題を解決し、普及させることを視野に入れている。このために、大学における研究開発よりも企業との共同研究による研究開発を行うことが重要であると判断し、新たに富士ゼロックス株式会社の島田グループを追加し、D-Case ツールの研究開発を進めた。

D-Case 手法がまだ確立できていない現状の中で、実際に動作するツールと、そのツールを用いたサンプルデータの作成を実施する。手法の課題を抽出すると共に、ツールへの要求を同時並行で抽出し、手法とツールの改良を進めることで、手法の確立とそれをサポートするツールの確立、そして関連するさまざまな応用アイデアを実験する環境を提供し、手法の研究・開発を促進する。H21 年度は 12 月から活動を開始し、D-Case ツールのプロトタイプ第 1 版を開発。H22 年 3 月に

内部関係者へのリリースを行った。またツールのテスト及び D-Case 作成時の課題抽出としてサンプルデータ作成を実施。実際のシステムに基づいたデータ作成は H22 年度に実施する。

## § 2. 研究実施体制

### (1) 石川グループ

① 研究分担グループ長: 石川 裕 (東京大学、教授)

② 研究項目

高信頼組込シングルシステムイメージ OS

### (2) 島田グループ

① 研究分担グループ長: 島田 利郎 (富士ゼロックス(株)、グループ長)

② 研究項目

D-Case に基づくソフトウェア開発支援環境

## § 3. 研究実施内容

(文中に番号がある場合は(4-1)に対応する)

### 石川グループ

石川グループでは平成 21 年度、以下の要素技術に関する研究開発を実施した。

#### 1) P-Bus

OS カーネル拡張のためのインターフェース P-Bus に関して、以下の開発を進めた。

i) P-Bus/P-Component の開発時に必要となる各種開発支援ツールの整備

- P-Bus/P-Component の自動テストツール
- ソースコードから P-Bus API リファレンスを自動作成するツール

ii) 基盤 Linux カーネルのアップグレード

P-Bus の実行基盤となる Linux カーネルのバージョンを、従来の 2.6.24 から 2.6.29 へ移行した。

#### 2) ディペンダビリティベンチマークフレームワーク

様々な既存ベンチマークソフトウェアを自動実行し、出力結果をデータベース化しやすいよう整形するフレームワークのプロトタイプを開発した。

#### 3) シングルシステムイメージ

i) 要素技術の研究開発

シングル IP アドレスクラスタを実現するための要素技術に関して、クライアントから送られてくるデータの内容を考慮したうえで担当サーバを決定できるような手法(Layer-7 スイッチング)を開発した。また、サーバ上のプログラムがクライアントとの TCP 接続を保持したまま他のサーバ機器に移動(プロセスマイグレーション)できる機構の研究開発を行った[2]。

#### ii) 実証デモシステムの開発

シングル IP アドレス機構を応用した実証デモとして、分散ファイルサーバを開発した。これは、複数のサーバ機器をシングル IP アドレス機構によって 1 つのファイルサーバとして見せるもので、利用者はクライアントに変更を加えることなくサーバ性能の向上やサーバ機器故障への自動的対処といった利点を享受することができる。

#### iii) P-Component 化

これまでシングル IP アドレス機構は Linux カーネルモジュールとしてプロトタイプの開発を行ってきたが、これらを P-Component の形に直した上で P-Bus 上に移植した。

#### 4) 論理分割機構(マルチコア・マルチプロセッサ管理機構)

平成 20 年度に開発した複数カーネル間の通信機能について、追加で評価を行い論文を発表した[1]。そのうえで、x86\_64 アーキテクチャへの移植対応を含めた再実装を行った。この再実装によって、実行する Linux カーネルのイメージの共通化が実現でき、また、メモリマップの見直しにより Linux カーネル以外のプログラムの実行を簡単に行える機能が追加された。この機能を利用し、論理分割機構によるカーネルの性能改善の実証のために、計算プログラムの実行環境としての単純なカーネルの実装を行った。

#### 5) 静的 WCET(Worst Case Execution Time)検証ツール

Pentium, XScale, SH アーキテクチャへの移植性と実行時間の予測精度を評価し論文としてまとめた[3]。加えて、21 年度はプリエンプションが発生する状況下での最悪実行時間の予測手法について検討を行った。プリエンプション発生時刻を静的に計算し、その時刻での実行中タスクのキャッシュエントリ数を予測することでプリエンプションを考慮した最悪実行時間を見積もる。本手法を RETAS に実装し評価した結果、プリエンプションが発生する状況下で最悪実行時間を見積もることができることを確認した。

### 島田グループ

本年度は 12 月末から 3 月末までの 3 ヶ月間で、H21 年度段階でのツールに対する要求仕様の明確化と H22 年度末目標の機能仕様を明確化すると共に、研究領域内の展開を目的としたプロトタイプ第 1 版の開発を以下の方針に従い実施することを計画した。(1) ツールは IDE として普及している Eclipse 上のプラグインとして実装する。すなわち OS 非依存で標準的な操作作法を実現する。(2) Eclipse 上の Graphical Modeling Framework (GMF) を採用する。Eclipse の UML ツールのフレームワークとして実績のある GMF を採用することで実現・検証すべき機能に集中して取り組む。集中議論を行い、機能仕様案を作成。本年度 1 月のプロジェクト合宿における研究領域メンバーへのレビューと議論を経て、プロトタイプ第 1 版の機能仕様を確定し実装を行い、3 月末に内部関係者へのリリースを実施した。D-Case ツールを図 1 に示す。

並行してサンプルデータ作成にも着手。作成過程で課題抽出を実施しており、次年度のツール開発及び手法研究に反映させる。また実際のシステムに基づいたサンプルデータ作成を H22 年度に実施する。

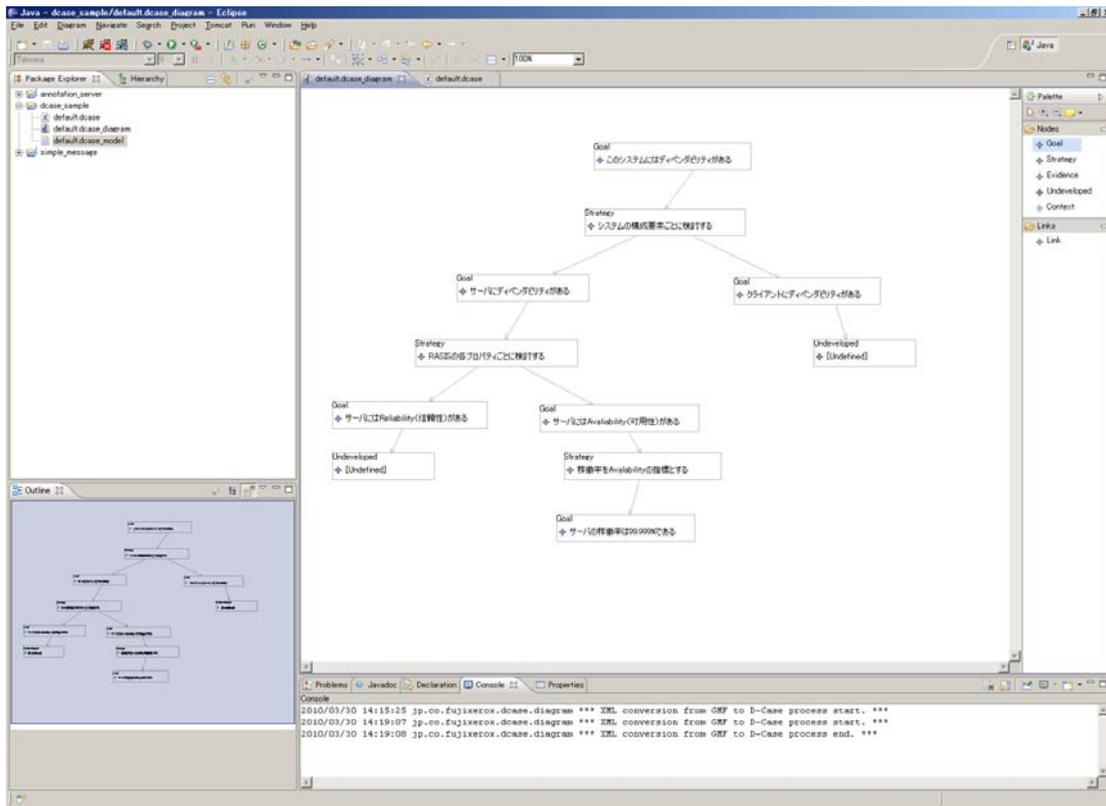


図 1 D-Case ツール

## § 4. 成果発表等

### 原著論文発表

- 論文詳細情報

[1] Taku Shimosawa, Yutaka Ishikawa, "Inter-kernel Communication between Multiple Kernels on Multicore Machines", IPSJ Transactions on Advanced Computing Systems Vol.2 No.4 (ACS 28), pp. 64-82, December 2009

[2] Balazs Gerofi, Hajime Fujita, Yutaka Ishikawa, "Live Migration of Processes Maintaining Multiple Network Connections", IPSJ Transactions on Advanced Computing Systems Vol.3 No.1 (ACS 29), pp. 1-12, March 2010

[3] 山本 啓二, 石川 裕, 松井 俊浩, 「高い移植性をもつ最悪実行時間解析手法」, IPSJ Transactions on Advanced Computing Systems Vol.3 No.1 (ACS 29), pp. 77-87, March 2010

[4] Jianwei Liao, Yutaka Ishikawa, "A New Concurrent Checkpoint Mechanism for Real-time and Interactive Processes", 34<sup>th</sup> Annual IEEE Computer Software and Applications Conference (COMPSAC 2010), July 2010 (accepted, short paper)