戦略的創造研究推進事業　CREST
研究領域
「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」
研究課題「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」

研究終了報告書

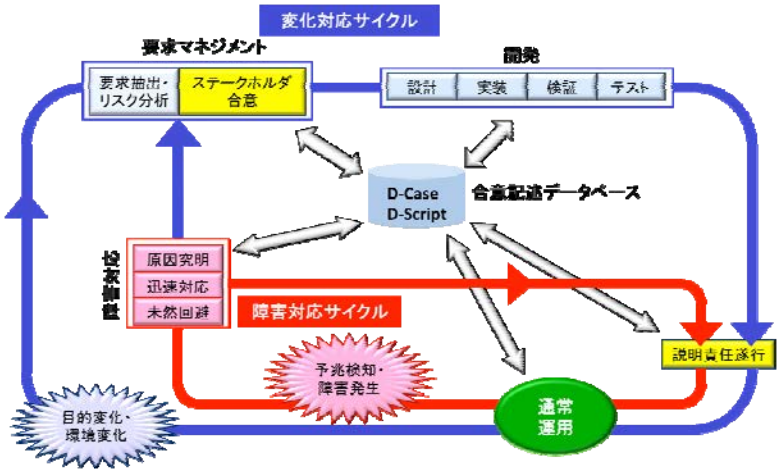
研究期間　平成18年10月～平成24年3月

研究代表者：石川　裕
（東京大学情報基盤センター　センター長）

§１　研究実施の概要

（1）実施概要

研究開始時には、Linux をベースに、高信頼単一システムイメージを提供する並列分散 OS を実現することをめざした。研究開始後、他の研究チームの若手研究者とともに DEOS コアチームを編成し、本研究領域で研究開発するディペンダブルシステムに必要とされる要件とそれを実現するための要素技術の検討を行ってきた。中間報告までに、「オープンシステムディペンダビリティ」の考えが本領域総括より提唱され、オープンシステムディペンダビリティを実現するための DEOS プロセス、DEOS プロセスを実現するための DEOS アーキテクチャが検討された。中間報告後も DEOS プロセスならびに DEOS アーキテクチャは改良が加えられている。



上記図に示すとおり、DEOS プロセスでは、変化対応サイクルと障害対応サイクルの同時存在を扱う。目的・環境変化対応サイクルにおいては、ステークホルダ間のディペンダビリティに関する合意は、D-Case と呼ばれるディペンダビリティに関する合意および証拠に基づく保証を記述可能な言語で記述される。障害対応サイクルにおいては、障害発生時の迅速対応や障害発生の未然回避を D-Case に基づいて遂行する。

本研究チームは、DEOS プロセスのなかで以下の研究開発を行った。

- DEOS プロセスの中心となる D-Case の研究開発
- D-Case が扱う証拠を提供するためのディペンダビリティベンチマークツール DS-Bench の研究開発
- D-Case が扱う証拠の一つとして、ソフトウェアの安全性検証を支援する P-Bus および P-Component の研究開発
- 合意形成されたディペンダビリティに対する性質が実行時にも満たされているかどうかを監視、記録、再構成可能する実行時環境 D-RE の研究開発

研究開発当初目標としていた高信頼単一システムイメージを提供する並列分散 OS は、DEOS プロセスにおける一つの適用例と位置づけて、研究開発を継続した。また、DEOS プロセス、DEOS アーキテクチャは、DEOS センターならびに DEOS コアチームと協業した。

（2）顕著な成果

1. 高性能・高可用・高信頼なインターネットサーバを実現するための単一 IP アドレスクラスタ機構を提案、開発した。ブロードキャスト型クラスタを基盤とし、その上に柔軟な TCP 接続分散機構、サーバプロセスのマイグレーション機構、サーバプロセスの耐故障機構を構築した。
2. システムのディペンダビリティのためには、ステークホルダ間の合意形成が重要であることを、領域内で広め、合意形成のための手法・ツールである D-Case を開発した。
3. 本研究領域で提唱している概念「オープンシステムディペンダビリティ」の達成のために、中

心となる DEOS Process の原案を提案し、DEOS Process の実現のために必要な DEOS Architecture の基本設計を行った。さらに、それらをサポートする、 D-Case Editor、DS-Bench などの開発を行った。

§2. 研究構想

(1)当初の研究構想

研究開始時には、Linux をベースに、高信頼単一システムイメージを提供する並列分散 OS を実現することをめざし、以下の 3 課題に取り組む方針を掲げた。

1) クラスタリングされたコンピュータ群で単一システムイメージを提供する機構の研究開発
2) アプリケーションの性質に応じたマルチコア利用機構の研究開発
3) 時間制約保障機構および静的時間制約検証の研究開発

また、研究開始 5 年後に本研究領域の他研究チームの成果物と統合することを念頭に、3 年後にプロトタイプシステムを開発することとした。

(2)新たに追加・修正など変更した研究構想

研究開始時に想定した研究課題のうち、「アプリケーションの性質に応じたマルチコア利用機構」については、その後の検討によりマルチコア CPU を利用して複数の OS を同時に動作させる「論理分割機構」の研究開発を行うこととした。

また、領域内の研究チーム間の連携を密にするため、平成 20 年度よりチームを横断する「コアチーム」が組織され、本研究領域の方向性について議論が行われた。その中から、ディペンダビリティを表現、評価するための枠組みの必要性が認識され、やがて、ディペンダブルなシステムを開発・運用するための「DEOS プロセス・アーキテクチャ」へと発展した。

このような流れをうけ、当チームでは当初の研究計画に加え、以下の研究開発を行うこととした。

1) ディペンダビリティ議論のための枠組み D-Case の研究
2) D-Case ダイアグラムの作図・編集を支援するツール D-Case Editor の開発
3) ディペンダブルシステムベンチマークフレームワーク DS-Bench の開発
4) アプリケーションのディペンダビリティ向上を支援する共通フレームワーク DEOS Runtime Environment の開発

§3 研究実施体制

(1)「石川」グループ

①研究参加者

氏名	所属	役職	参加時期
石川 裕	東京大学情報基盤センター	センター長	H18.10～
横手 靖彦	東京大学情報基盤センター	特任教授	H22.4～
松野 裕	東京大学情報基盤センター	特任講師	H22.4～
藤田 肇	東京大学情報基盤センター	特任助教	H18.10～
Balazs Gerofi	東京大学情報理工学系研究科	D3	H21.4～
廖 剣偉	東京大学情報理工学系研究科	D3	H21.4～
藤原 祐二	東京大学情報理工学系研究科	M1	H23.4～
高橋 弘美	東京大学情報基盤センター	事務補佐員	H18.12～
松葉 浩也	東京大学情報基盤センター	助教	H18.10～H21.3
平野 貴仁	東京大学情報理工学系研究科	M2	H19.4～H21.3

加藤真平	東京大学情報理工学系 研究科	PD	H21.4～H22.3
酒井将人	東京大学情報理工学系 研究科	D3	H18.10～H22.3
山本啓二	情報基盤センター	PD	H21.4～H22.4
下沢拓	東京大学情報理工学系 研究科	D2	H19.4～H23.3
加藤 純	東京大学情報理工学系 研究科	M2	H21.4～H23.3

②研究項目

・ 高信頼シングルシステムイメージ OS

(2)「恩田」グループ

①研究参加者

氏名	所属	役職	参加時期
島田 利郎	富士ゼロックス株式会社・ 研究技術開発本部 コミュニケーション 技術研究所	マネージャー	H22.1～ H23.3
恩田 昌徳	富士ゼロックス株式会社・ 研究技術開発本部 コミュニケーション 技術研究所	グループ長	H23.4～
伊東 敦	富士ゼロックス株式会社・ 研究技術開発本部 インキュベーションセンター		H22.1～
上野 肇	富士ゼロックス株式会社・ 研究技術開発本部 インキュベーションセンター		H22.1～

②研究項目

・ D-Case に基づくソフトウェア開発支援環境

§4 研究実施内容及び成果

4.1 シングルシステムイメージ機構(東京大学　石川グループ)

(1)研究実施内容及び成果

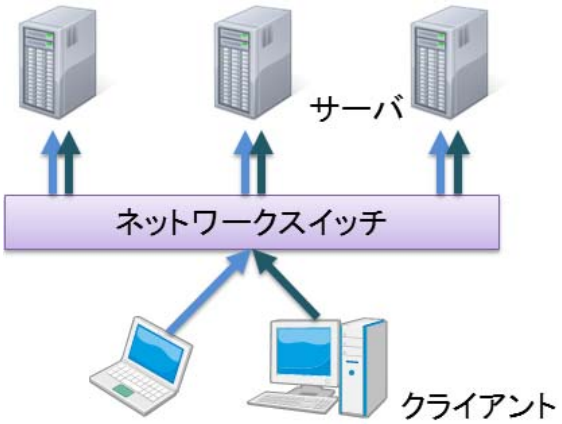


図 1 ブロードキャスト型単一 IP アドレスクラスタ

ディペンダブルなサーバを構築するための要素技術として、単一IPアドレスクラスタの研究開発を行った。単一IPアドレスクラスタは複数の計算機から構成されるサーバに 1 つのIPアドレスを割り当て、サーバの性能および可用性を向上させるための技術である。従来研究開発されてきた単一IPアドレスクラスタは、クラスタを代表する特殊な役割を持ったノード(代表ノード)が 1 つ存在し、代表ノードが負荷分散とIPパケットのルーティングを行う方式が主流であった。この方式は、代表ノードが単一障害点となるため、サーバの高可用化のためには代表ノードの冗長化が必須である。一方、代表ノードを用いずに、すべての入力パケットを全クラスタノードにブロードキャスト配送する方式のクラスタも存在する。我々はこのブロードキャスト型クラスタ(図 1)をもとに研究開発を行った。

1 つめは負荷分散機構の開発である。ブロードキャスト方式ではあるサーバノードが故障しても他のノードとクライアントとの通信には影響を及ぼさないという利点がある一方、柔軟な負荷分散が行えないという問題があった。そこで我々は前述の 2 種類の方式のうち、後者のブロードキャスト型クラスタを元に、柔軟な負荷分散を可能とする方式の研究開発を行った。具体的には、クライアントから到着した新規 TCP 接続をどのサーバノードで処理するかを判断する機構を加えた。判断の指標としては、各ノードの負荷状況や、クライアントから送信されたリクエストの内容を用いる。

2 つめはサーバの耐故障機能の開発である。単一 IP アドレスクラスタにおいては、複数のサーバノードを用意することで、あるノードが故障してもサーバ全体としてはサービスを継続できるものの、故障が発生したまさにその瞬間に故障ノードで行われていた処理の途中経過はすべて失われてしまう。そのため、まずサーバノード間でサーバプロセスを移送するプロセスマイグレーション機構を開発した。この機構は、クライアントとの間に確立しているTCP 接続を維持したままプロセスを移送可能であり、クライアントから透過にプロセス移送が可能である。また、全ノードが同じパケットを受け取るというブロードキャスト型の特性をいかし、プロセス移送実行中に受信したパケットをプロセス移送先で受信しておくことで、プロセス移送に伴う受信パケットの取りこぼしを防ぐ。さらに、上記プロセスマイグレーション機構を拡張し、実行中のプロセスイメージを他のノードに複製し、ノード故障時には複製側のノードに実行を切り替えることでクライアントに対して故障を隠ぺいする機構の開発も行った。プロセスイメージの複製は、サーバがクライアントに見える状態変化、すなわちクライアントに対するネットワーク送信を行う際に実行する必要がある点に着目し、クライアントに対するネットワーク送信をバッファに蓄えて遅延させることで、プロセスイメージ複製の頻度を下げ、耐故障機構の導入によるオーバーヘッドを削減した。

本研究の成果は以下のとおりである。
(a) 上記の通り、単一 IP アドレスクラスタの性能、可用性、信頼性を向上させる手法を考案し、提案手法を Linux カーネル上に実装し評価した。成果は論文の形で IEEE 国際会議や情報処理学会論文誌に採録されている。
原著論文発表: [1, 2, 5, 7, 8, 9, 10, 12, 17, 18, 19, 20 ,21]
国際学会発表及び主要な国内学会発表: 口頭発表 [1, 2, 6, 9, 10, 13, 16, 17, 26, 28]
(b) 提案している負荷分散機構を応用したデモとして、平成 21 年度中間成果報告会および ET2010 において、分散ファイルサーバを構築し出展した。これは、複数のサーバ機器を 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。

(2)研究成果の今後期待される効果
研究成果の一部をオープンソースソフトウェアとしてパッケージ化し、一般公開を準備中である。これによって、研究成果を社会に還元することが期待される。

4. 2 論理分割機構 (東京大学 石川グループ)
(1)研究実施内容及び成果
1 台の計算機を論理的に複数に分割し、複数の OS を同時に動作させる機構の研究開発を行った。同様の目的を達成する方法としては、計算機の仮想化を用いる方法や、論理分割のための特殊なハードウェア機構を用いる方法があるが、前者は仮想化のオーバーヘッドが無視できず、後者は一般的なコモディティ計算機で利用できないという問題がある。我々の提案した手法 SHIMOS (Single Hardware with Independent Multiple Operating Systems)は、ソフトウェアのみで論理分割を実現するものである。

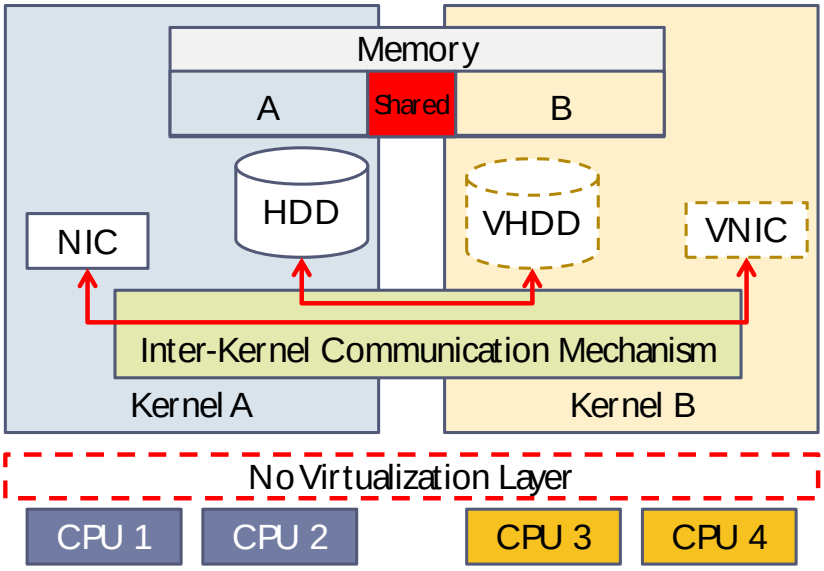


図 2 論理分割機構 SHIMOS の概要

提案手法SHIMOSの概念図を図 2に示す。SHIMOSでは、物理計算機の資源を静的に分割し、特定のハードウェア資源(CPUコア、メモリ領域、周辺機器など)をそれぞれどれか 1 つのOSに占有させる。あるOSカーネルは他のカーネルが使用している資源に干渉しないことで、計算機の分割が実現される。また、OSカーネル同士は共有メモリ領域を介して通信することが可能であり、これを利用して、あるカーネルが占有しているデバイスを別のカーネルから使用する機構も実現されている。
Linux カーネルを改造して本手法を実装し、仮想マシンとの比較を行ったところ、CPU 性能

を測定するベンチマーク(SPEC CPU)では、仮想マシンより 6%高速、I/O 性能を測るために行った Apache Benchmark では、仮想マシンよりも 3 倍高速となる結果が得られた。
本研究の主な成果は以下の通りである。
原著論文: [3, 4, 15]
国際学会及び主要な国内学会発表: 口頭発表 [8, 20]
(2)研究成果の今後期待される効果
本手法は、性能劣化なく計算機を分割できるという点において、高性能計算に適した性質をもっている。そのため本手法は、CREST 研究領域「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」の研究課題「メニーコア混在型並列計算機用基盤ソフトウェア」において開発されているメニーコア計算機向けOSの基盤技術として使用されている。

4．3 最悪実行時間予測ツール（東京大学 石川グループ）
(1)研究実施内容及び成果
リアルタイム分野の要素技術として、プログラムの最悪実行時間を予測するツールの研究開発を行った。機械制御のような領域においては、制御ソフトウェアが一定の時間周期内に実行を完了することが必要であり、あるプログラムが実行を完了するまでにどのくらいの時間を要するのかを事前に見積もることが重要である。
当グループでは、JST CREST 領域名「情報社会を支える新しい高性能情報処理技術」研究課題名「ヒューマノイドのための実時間分散情報処理」において最悪実行時間予測ツール RETAS の開発を行ったが、本研究領域においては RETAS を基に拡張を行った。
本領域においては、SH4 アーキテクチャへの対応、1 つのプロセッサを時分割で共有する複数プロセスを考慮した最悪実行時間予測を行った。
本研究の主な成果は以下の通りである。
原著論文: [6]
国際学会及び主要な国内学会発表: 口頭発表 [11, 18]

(2)研究成果の今後期待される効果
研究成果を、オープンソースソフトウェアとして研究室ホームページで一般公開している(http://www.il.is.s.u-tokyo.ac.jp/retas/)。これにより、研究成果の活用が期待される。

4．4 P-Bus, P-Component: ソフトウェア検証のためのシステムプログラミングインタフェース（東京大学 石川グループ）
(1)研究実施内容及び成果
当チームの単一 IP アドレスクラスタ機構を含め、本研究領域では OS カーネルの拡張によりディペンダビリティ向上のための機構を研究開発するテーマが存在する。しかし、一般に新しく開発されたソフトウェアには未発見の不具合が存在することが想定され、システム運用時に OS カーネル内で不具合が顕在化することはシステムの安定性・安全性にとって致命的な問題となる。C 言語で記述されたソフトウェアの安全性を証明するツールは本研究領域の東京大学前田チームで研究開発されている。同チームで研究開発されているモデル検査ツールを利用するためには、ソフトウェアが満たすべき仕様をCプログラム上に別途与える必要がある。そこで当チームでは、OS カーネル拡張モジュールをはじめとするシステムソフトウェアの安全性向上のためのプログラミングインタフェースの研究開発を行った。

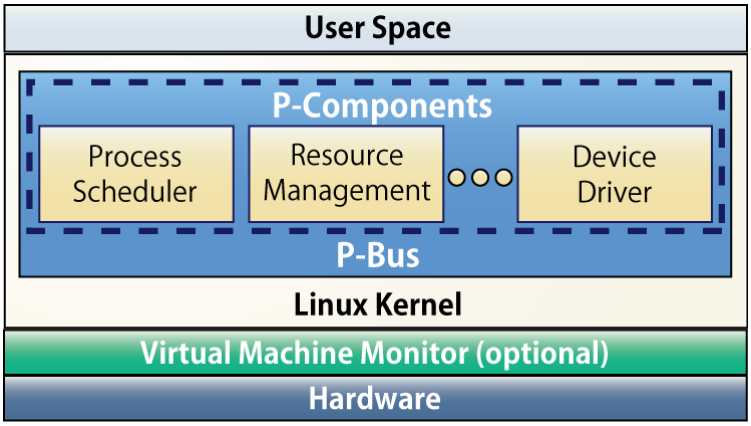


図 3 P-Bus 概念図

具体的には、図 3に示すように、Linuxカーネル内にP-Busとよぶ新たなプログラミングインタフェース層を構築し、この上にP-ComponentとよぶOSカーネル拡張モジュールを実装する。P-Busのプログラミングインタフェースには、インターフェイスの安全な利用のために満たすべき仕様が規定されており、それらの仕様が形式的に記述されている。記述される仕様の例としては、関数の引数として許される値の範囲、関数を呼ぶ前に満たされているべき条件(ロックが取得されている、等)、関数の戻り値とその後に成り立つ条件(関数が 0 を返したら処理は成功であり、ロック変数が未取得状態から取得済み状態に変わる)、などがある。記述した形式的仕様は前田チーム(東京大学)のモデル検査ツールで処理することを前提に、前田チームと共同でC言語向け使用記述言語ACSL[1]を拡張した言語を設計し使用した。

P-Bus の使用記述の手法自体は P-Bus に限らず、C 言語で記述されたシステムプログラムに利用できるものである。実際、P-Component 検証と同様の手法を用いて仮想マシンモニタ(早稲田大学 中島チーム)の検証が中島チームによって行われた。

本研究の成果:

国際学会発表及び主要な国内学会発表: 招待講演 [2, 3] 口頭発表 [2, 3, 4, 5, 7, 19, 29, 32] ポスター発表 [1]

(2)研究成果の今後期待される効果

前田チームのモデル検査ツールとともに、システムソフトウェアの検証手法が本研究領域内の他のソフトウェアプロダクトに応用されることが期待される。さらには前田チームのツールの展開次第ではより広く一般への応用も期待される。

4. 5 ディペンダビリティベンチマークフレームワーク (東京大学 石川グループ)

(1)研究実施内容及び成果

あるシステムが、必要なディペンダビリティを備えているかどうか、様々なベンチマークツールによって測定し評価を行う必要がある。また、システムのディペンダビリティを評価するためには、様々な異常状態においてシステムが想定通りに振る舞うかを評価できなければならない。

そこで、我々はディペンダブルシステムベンチマークフレームワーク DS-Bench を開発してきた。DS-Bench はそれ自体でディペンダビリティ指標を計測するものではなく、様々なベンチマークプログラムを統一的に実行し、結果を記録するための環境を提供する。ベンチマークプログラムとしては、既に存在するソフトウェアや、ユーザが自作したプログラムを用いる。ベンチマークプログラムにはプレーンテキストの表形式で結果を出力するものが多いが、

[1] http://frama-c.com/acsl.html

これは人間にとっては見やすいものの計算機が個々の数値をデータとして扱うには不便である。そこで DS-Bench では、ベンチマークプログラムからの出力を解析し、データを切り出したうえで XML 形式でデータベースに格納するようにした。この際のデータ切り出しルールを追加することで、新しいベンチマークプログラムを DS-Bench で扱うことができるようになる。また、DS-Bench では、システムの異常状態を模擬するため、Anomaly load と呼ぶ様々な異常をベンチマーク対象システムに加えることができる。Anomaly load には、ネットワークの切断やマシンの電源切断といった故障だけでなく、CPU やメモリを過剰に消費するプログラムといった、ソフトウェア的な異常状態も含まれる。

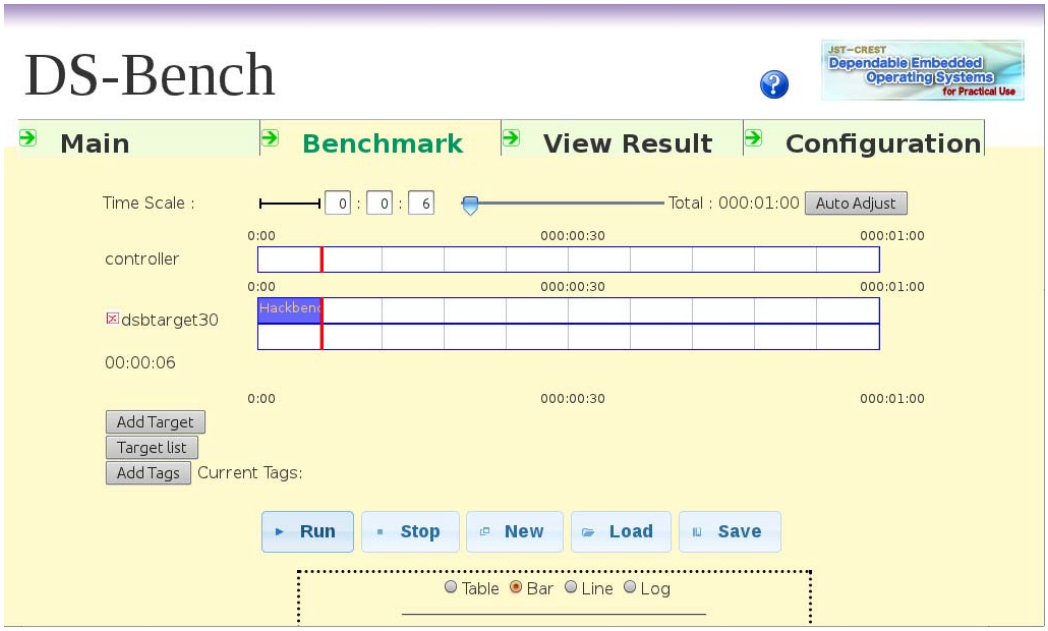


図 4 DS-Bench ベンチマークシナリオ編集・実行画面

DS-Benchでは、実行するベンチマークや加えるAnomaly loadのタイミングを記述したベンチマークシナリオを作成する。シナリオはXML形式で記述されるが、作成はWebインタフェースを用いて行うことができる(図 4)。また、シナリオの実行、結果の確認も同じくWebインターフェースを用いて行うことが可能である。

DS-Bench は、本 CREST 研究領域において、システムのディペンダビリティに関するエビデンスを生成し記録する仕組みの 1 つと位置づけられる。すなわち、D-Case を用いてシステムに求めるディペンダビリティをステークホルダ間で議論した後、実際のシステムが要求を満たしているかどうかを DS-Bench で計測し、結果を D-Case 上の議論に反映させる。このようなサイクルをサポートするため、DS-Bench は富士ゼロックス恩田グループで開発されている D-Case Editor と連携する機能を備えている。ステークホルダは、D-Case Editor 上でシステムの前提条件(たとえば、サーバに対する同時アクセス数)とディペンダビリティ要件(たとえば、応答時間や、障害発生時のダウンタイムなど)を D-Case Editor 上で設定し、ベンチマークを起動する。ベンチマークが終了すると、システムが実際にディペンダビリティ要求を満たしていたかどうかが D-Case のノードに表示され、また、そのベンチマーク実行に関する詳細情報をエビデンスとして呼び出せる。

DS-Benchは本研究領域の筑波大学 佐藤チームで開発されているTest-Envと連携して動作する(図 5)。Test-Envはベンチマークを実行するターゲット環境となる物理マシンや仮想マシン、さらにはネットワークスイッチや電源制御装置などの機器を管理・制御し、ベンチマークのために使用可能な状態とする。また、Test-Envはターゲットマシンに対してマシン外部から故障を注入する機能を提供する。DS-Benchは、ベンチマークシナリオに基づいて、指定したタイミングで故障を注入するよう、Test-Envに指示を行う。

ディペンダビリティに関するベンチマークという概念では、先行する研究にDBench[2]がある。DBenchでは、定義されたフレームワークに基づき、対象とする応用領域ごとにベンチマーク実行環境が作成された。一方、DS-Benchでは、対象領域によらない統一的な実行環境でベンチマーク実行を扱い、ベンチマークプログラムの再利用やベンチマーク結果のデータベース化を可能とする設計を行っている。

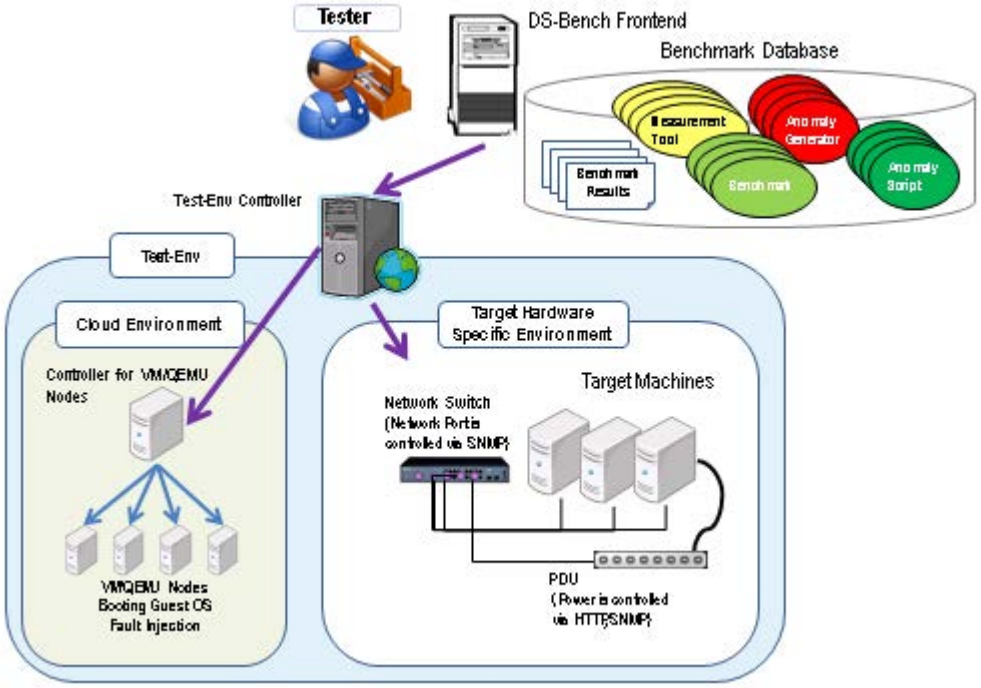


図 5 DS-Bench/Test-Env の概念図

本研究の主な成果は以下の通りである。
原著論文: [22]
国際学会及び主要な国内学会発表: 口頭発表 [15]
ポスター発表: [2]

(2)研究成果の今後期待される効果
DS-Bench および Test-Env は、実際のシステム開発において活用できるツールとして開発を進めてきた。開発された成果物はオープンソースプロダクトとして一般公開を予定しており、今後の本研究領域の研究開発に資するのみならず、一般の開発者や企業においても利用可能となる。

[2] K. Kanoun, L. Spainhower, “Dependability Benchmarking for Computer Systems”, IEEE COMPUTER SOCIETY

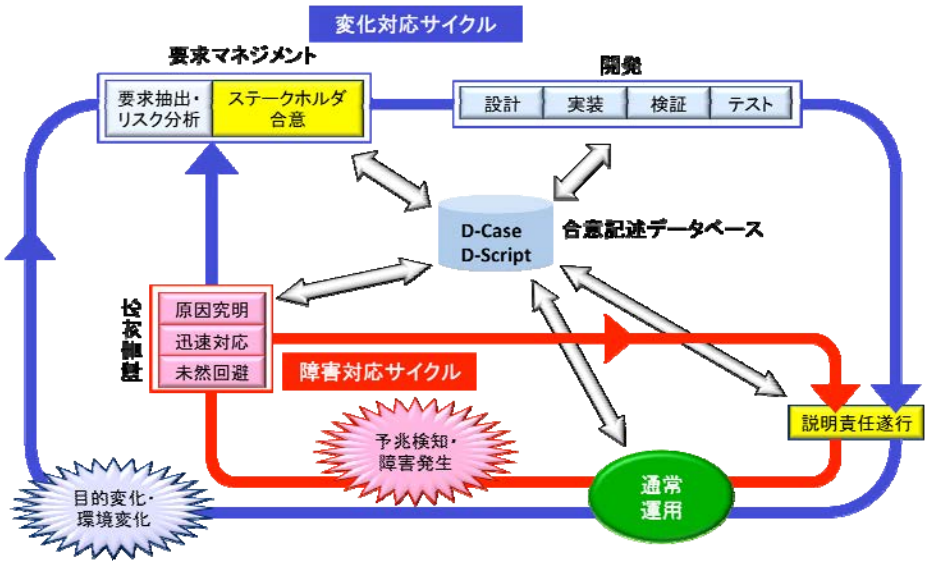


図 6 DEOS プロセス

4．6 D-Case（東京大学 石川グループ）

(1)研究実施内容及び成果

DEOS サブコアチームの一つである D-Case サブコアチームをとりまとめ、以下を行った。

D-Case は DEOS プロセスにおいて重要なステークホルダ間合意形成のための手法およびツールとして研究を進めた。D-Case は欧米で必要性が増している Safety Case という、システムの安全性を保証するための手法を元にしている。D-Case は、DEOS プロセスにおける合意記述データベースとして用いられ、図 6 に示すとおり、障害発生時の迅速対応や障害発生の未然回避に関する合意内容が格納され、D-RE の実行に用いられるよう Safety Case を拡張している。平成22，23年度では、D-Case を記述する手法やツール開発を、富士ゼロックスおよび D-Case サブコアチームとの共同で行った。

D-Case の特徴は以下の 4 つである。

(a) 議論とエビデンスに基づく合意のための、構造化された表記法
構造化された表記は、イギリスYork 大学で開発された、Safety Case の主要な記述法である Goal Structuring Notation (GSN) を元にしている。

(b) ステークホルダ間合意のマネージメントサポート
現時点の D-Case は、GSN のノードに加えて、ステークホルダと、合意内容の関係を表すノードなどを持っている。

(c) 通常運用時におけるステークホルダ間合意のモニタリングサポート
運用時の説明責任を支援するために、D-System Monitor もしくは D-Application Monitor(4.7 節参照)による、運用時のモニタリングにより得られるログなどを表すモニタノードが導入されている。

(d) 合意内容記述の整合性検査サポート
合意文書の作成は時間がかかる。作成を容易にするために、様々な適用分野における D-Case パターンを用意する(また D-Case における合意内容の整合性を確かめることは、面倒であるが、これについては D-Case チームに参加している木下チームにより定理証明器による整合性検査が開発されている)。

D-Case の主な部分は、以下のノードを含む、構造化された文書である。ステークホルダ間で議論する主張、命題を表すゴールノード、ゴールノードをサブゴールに分割するときの説明を表すストラテジノード、ゴールが満たされることを保証するエビデンスを表すエビデンスノード、システムの環境などに関する情報をあらわすコンテキストノード、そしてモニタリング

ノードがある。

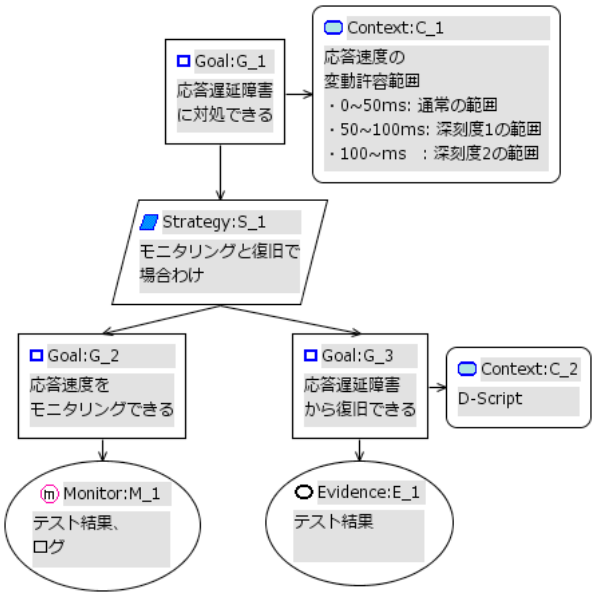


図 7 D-Case の例

図 7のD-Caseの例は、サーバの応答遅延障害に関する議論内容を表している。コンテキストノードC_1 に応答時間の変動許容範囲(0-50ミリ秒)と、変動許容範囲を超えた場合の深刻度(1および2)が定義されている。このD-Caseは、サーバの応答時間を常にモニタリングできることを、モニタリングノードM_1 により保証し、応答時間が変動許容範囲を超えた場合、深刻度に応じて、BPMNなどで記述された倉光チームで開発されているD-Scriptにより、応答時間を変動許容範囲に戻せることを、事前のテスト結果によるエビデンス(E_1)により保証している。

成果として以下をあげる。

(a) 従来主に自然言語のみで記述されていた Safety Case を、ツール、さらには D-RE など実システムで扱いやすいよう、型などプログラミング言語の技術を導入し、ディペンダビリティ合意内容を、ツールチェーンによる他の開発ツール(DS-Bench など)とのやりとりが容易になるようにし、IEEE 国際会議などで採択され、特許出願を行った。D-RE など、実システムとの連携の基礎を作った。
特許出願: 国内出願 [1]
原著論文発表: [11, 13, 14, 16]
国際学会発表及び主要な国内学会発表: 口頭発表 [14, 21, 22, 23, 24, 27, 30, 31, 33, 34, 35]

(b) 富士ゼロックスと共同で D-Case 記述支援ツール D-Case Editor を開発し、オープンソースとして公開した(詳しくは4．8参照のこと)。D-Case Editor に関して招待講演の依頼や、商用化の提案を受けている。公開後2ヶ月のダウンロード数は50件程度である。

(c) D-Case は2009年9月の DEOS 中間成果報告会で、本研究領域で開発中の要素技術や検証・ベンチマークツールの、システムのディペンダビリティにおける位置づけを明確にするために導入された。以来、DEOS の個々のチームで研究開発中の技術をつなげるものとして研究が進められ、DEOS プロセス、DEOS アーキテクチャへの、DEOS 技術の集約に貢献した。

(d) Object Management Group(OMG)や Open Group での国際標準化に積極的に参加し、D-Case, OSD 概念を広めることに貢献した。トヨタ、産総研、東大による、ディペンダビリティ、D-Case を用いたプロセス規格を OMG で共同提案し、規格化のロードマップに乗るなど、DEOS の国際標準化への足がかりを作った。

(2)研究成果の今後期待される効果

D-Case Editor は、現在商用化、オープンソース化による共同開発などを検討中であり、広く使われる可能性がある。DS-Bench など、他の DEOS ツールとツールチェーンを構築することにより、DEOS の研究成果が広く使われることにつながると期待される。また、D-Case の元となった Safety Case は、自動車の機能安全国際規格 ISO26262 の要求事項となっており、日本でも関心が高まっている。Safety Case の研究において、我々は City University London など、欧米の研究機関などとの交流を通じて日本企業の ISO26262 対応などに貢献できると考える。DEOS プロセス、DEOS アーキテクチャの考え方は、従来の Safety Case による安全性保証などを包含し発展させたものである。DEOS の実用化へ貢献し、社会へ研究開発の還元を行うことが期待される。

4．7 D-RE（東京大学 石川グループ）

D-RE は DEOS プロセスを実現するための DEOS アーキテクチャにおけるランタイム環境（DEOS Runtime Environment）として研究を進めた。DEOS プロセスの要諦は(1)変化対応サイクルと障害対応サイクルの同時存在であり、(2)目的・環境変化対応サイクルにおけるステークホルダー間のディペンダビリティに関する合意された要求を D-Case に記述している点であり、(3)障害対応サイクルにおける障害発生時の迅速対応や障害発生の未然回避を D-Case に基づいて遂行できる点である（図 6 参照）とした。D-Case を設計図とするなら、その実現形態が D-RE であり、そこにはステークホルダの合意された要求が反映されている。しかし、ステークホルダの要求は各種環境の変化を要因として変化するので、ある時点で合意された D-Case は別の時点では変化していることになる。この D-Case の変化に D-RE は対応する必要がある。

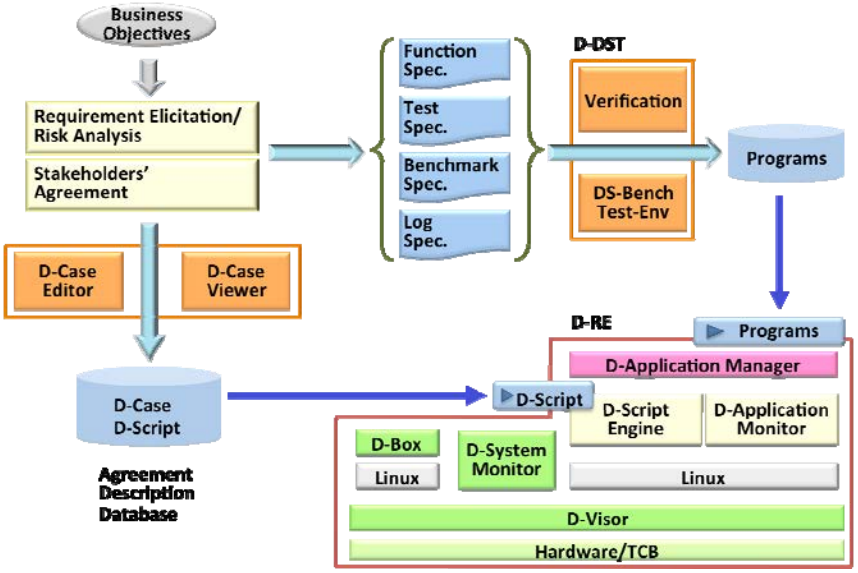


図 8 DEOS アーキテクチャ

D-RE の実装を複数のハードウェアプラットフォームに対応させるためにはアーキテクチャの定義が必要であるので当該アーキテクチャの研究を行った。当該アーキテクチャは DEOS プロセスを実現しなくてはいけない。そこで、我々は商品やサービスのライフサイクル全体に関わるアーキテクチャとして図 8 に示す DEOS アーキテクチャを開発した。当該アーキテクチャ図に於いては D-RE はランタイム環境を提供する。

図 9 に D-RE のソフトウェア構造を示す。

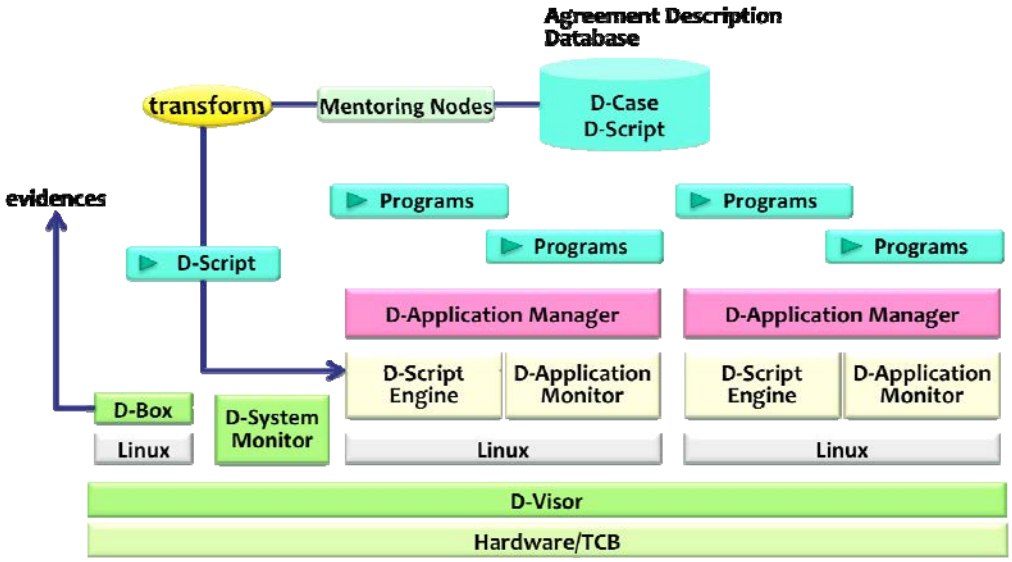


図 9 D-RE ソフトウェア構成

動作監視:D-Case に記述された対象システムがステークホルダによる合意通りに運用され稼働しているかを監視する。その際に「合意通り」を裏付けるデータを収集する必要がある。ステークホルダは D-Case モニタノードとして監視ポイントを記載している。すなわち、ステークホルダはそれら監視ポイントをオープンシステムズディペンダビリティ実現の為には重要であるとして合意している。本稿ではモニタノードで収集されたデータをエビデンスと呼び、ステークホルダの説明責任遂行の際のデータとする。

再構成:D-Case の変化に対応するために実行環境を再構成する。その為には、システム構成要素を隔離する機能が必要である。障害発生時には障害部位を隔離し、他の部位に影響を与えないようにする。隔離の仕方や再構成の仕方は対象部位により、また実行状況により異なり、実行時の柔軟性が要求されるので、次に述べるスクリプトが必要になる。

スクリプト:システム再構成時には対象毎に異なる再構成手順を実行し、動作監視時にはモニタノードに対応して監視手順とエビデンス収集手順を実行する。これらの実行手順を記述したプログラムをスクリプトと呼ぶ。実行環境にはこのスクリプトを安全・確実に実行する必要がある。

記録:動的監視により得られたエビデンス、再構成の記録、スクリプトの実行記録を始め、OSD 実現に有益な情報を記録する。これらの記録は、アクセス管理、暗号化、改竄検出を行い、情報の整合性を守る必要がある。

セキュリティ機能:動作監視、再構成、スクリプト、そして記録機能を安全に実行するためにセキュリティ機能が必要となる。

ライフサイクル管理:D-RE 上の動作するアプリケーションのライフサイクルを管理する。

D-RE のリファレンス実装では、前記機能を実現する構成要素を組み込むことになる。

上記D−REソフトウェア構成をSymphony社のKnowledgeLineアプリケーション[3]に組み込むことで、DEOSプロセスにおける変化対応サイクルと障害対応サイクルを支援するランタイム環境を構築した。変化対応サイクルにおいてはKnowledgeLineが備えるドキュメント管理、およびプロジェクト管理機構を利用し対象システムとしてのKnowledgeLineアプリケーションのディペンダビリティ要求に関する議論を進める。一方、障害対応サイクルにおいては、D−RE対応のために拡張した機構(D−KLを呼ぶ)を利用することで、KnowledgeLineアプリケーション自身のディペンダビリティ要求を満たすように上記D−RE基本動作

[3]http://www.symphonies.jp/cgi-bin/WebObjects/11863105199.woa/wa/read/129b53b99c9/

を実行する。本アプリケーションにより、DEOSプロセスが有効に機能することが示された。また、本アプリケーションは残りのDEOSプロジェクト推進におけるDEOSプロセス実験環境として利用される。図 10 にアプリケーションのスナップショットを示す。

図 10 D-KL スナップショット

本研究に関連する成果は以下の通り。
特許出願: 国内出願 [3] 海外出願 [2]

(2)研究成果の今後期待される効果
研究領域成果物の応用に際して商品・サービスライフサイクルにおける全体に関わって統合するランタイム環境として利用可能である。

4.8 D-Case に基づくソフトウェア開発支援環境(富士ゼロックス 恩田グループ)
(1)研究実施内容及び成果
東大石川チームのサブテーマとして D-Case 研究チームの松野リーダーの元で、以下の項目の研究開発を進めた。
1) D-Case エディタ開発
D-Case 作成ツールとして、平成 22 年 3 月に D-Case エディタのプロトタイプを開発。これをベースに、要件抽出と仕様策定、実装と試用による更なる要件抽出を繰り返すインクリメンタル開発を行い、4 半期毎にツールのリリースを行った。オープンソースの統合開発環境である Eclipse を対象とし、Eclipse 上のグラフィックアプリケーションフレームワークである GMF(Graphical Modeling Framework)を利用する構成を踏襲し、基本図形描画機能の実装と共に、エディタと外部ソフトウェア或は外部機能との連携に必要な拡張機能の実装を行った。
拡張機能としては、D-Case 手法の普及・展開を図る際に既存のソフトウェア開発プロセスへの導入を容易にすることを目的に、OSS プロジェクト管理ツールである redmine との連携機能を実装した。また、D-RE、DS-Bench/Test-Env、および、D-Case DB と連携してリアルタイムなディペンダビリティの不具合箇所の提示、ディペンダビリティ要求を満たしているかの確認、ソフトウェアの検証、蓄積された D-Case 資産の再利用等をエディタから実行出来るようにした。これらの拡張機能について学会・ワークショップ・展示会で、デモンストレーションを行い、手法やコンセプト

を実業務にどう適用するのかを紹介するための材料として活用した。D-Case エディタの外観を次の図に示す。

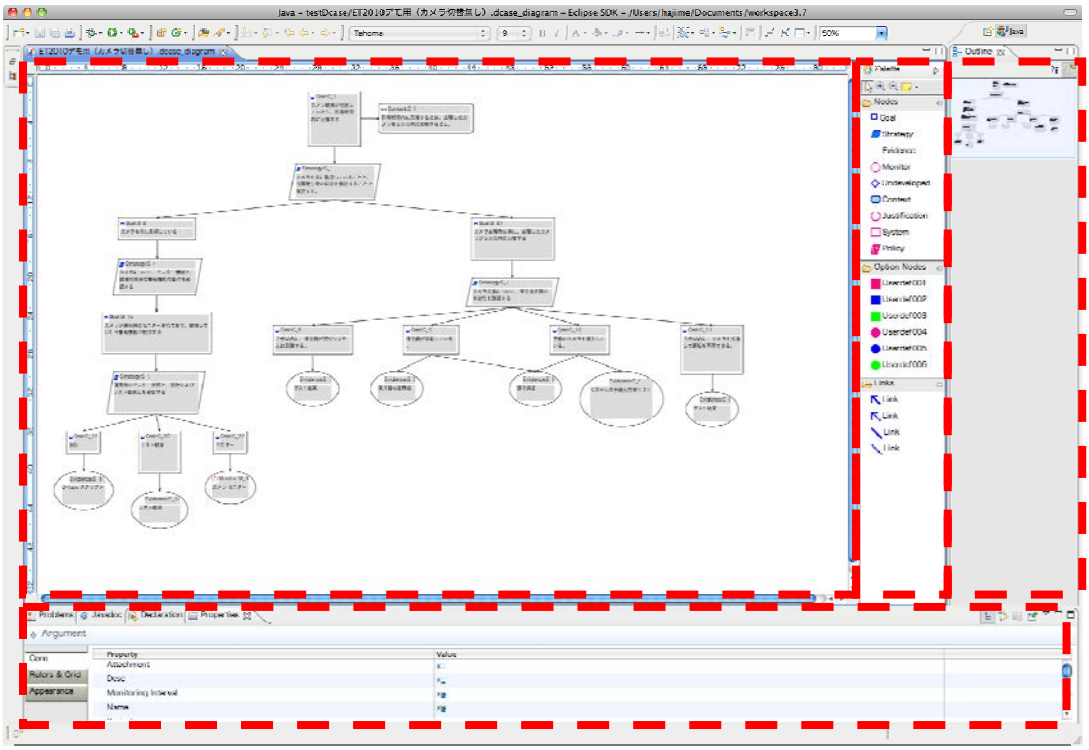


図 10 D-Case エディタ

２）D-Case Viewer 開発

ディペンダビリティを合意形成する際には、ステークホルダは必ずしも開発者とは限らない。開発者でないステークホルダが D-Case ダイアグラムの容易に拡大・縮小を簡単に操作し、議論を行うためのツールとして、iPad 上で動作する D-Case Viewer の提案・試作を行い、D-Case エディタと合わせてデモンストレーションを行った。D-Case Viewer の外観を図１１に示す。

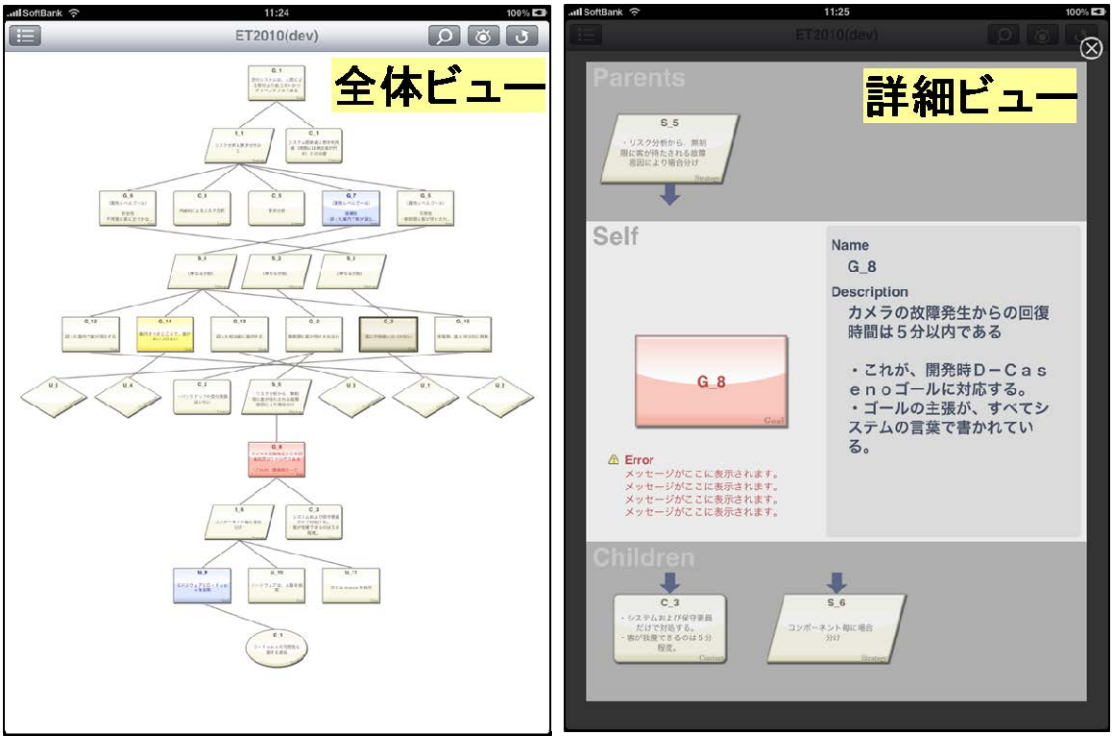


図 11　D-Case Viewer

３）D-Case DB の開発

D-Case の作成を効率よく実施するために、D-Case テンプレート機能、および、過去の D-Case や記述パターンのネットワーク経由による共有を可能にするデータベース機能を開発した。現状、D-Case DB から D-Case Editor への取り込み機能、および、外部ツールから D-Case DB へのテンプレート登録機能を実装済みである。

４）既存開発プロセスへの適用及び実事例検討

D-Case 手法の有効性と普及促進を図るために、組込システム商品の企画・開発プロセスに対して、D-Case 手法をどのような形で適用し統合するのかを明らかにすると共に、D-Case 手法導入コストやそのメリットが感じられる事例獲得を試みた。具体的には、富士ゼロックスの開発部門および品質保証部門に紹介するとともに、複数の開発活動への適用トライアルを実施し、D-Case 手法の利点および改善点の抽出を行った。さらに、D-Case 手法適用時に有用なドキュメントとして、D-Case 作成ガイドライン、他の開発ツールとの関連図の作成を行った。また、普及促進のため、Object Management Group への提案の中や、展示会で来場者に配布する冊子の中の参考事例として紹介し、ET2010/ET2011 において他ツール（D-RE, DS-Bench/Test-Env）との連携機能を含めたデモ展示を実施した。

本研究の主要成果一覧:

特許出願: 国内出願［1］海外出願［1］

国際学会発表および主要な国内学会発表: 招待講演［5, 6］

(2)研究成果の今後期待される効果

ディペンダブルシステムの開発、運用・保守、改良を支援する D-Case 手法に基づいた開発支援環境の構築出来たため、ディペンダブルシステムの開発プロセスにおいて効率的にステークスホルダーの合意形成が可能となる。

D-Case ツールの要件の抽出も出来たため、今後ツールベンダーにおいて D-Case ツ

ール開発を加速することが出来る。
D-Case 作成ツールおよびガイドラインを整備・公開することにより、一般の開発者・企業が D-Case を作成することが可能となった。今後設立予定の企業によるコンソーシアムでは D-Case の DB 化を予定しており、これにより D-Case 適用のノウハウを蓄積することが可能となるため D-Case 作成の効率化に繋がる。

§5 成果物等

(1)ソフトウェア

1. 単一 IP アドレスクラスタ機構を用いたサーバロードバランシング機構
2. ディペンダビリティベンチマークフレームワーク DS-Bench
3. D-Case 編集支援ツール D-Case Editor
4. 最悪実行時間予測ツール RETAS

(2)規格

1. Dependability of Consumer Devices Request For Information (RFI), White Paper, トヨタ、産総研、東大の共同提案、2011 年 12 月に RFI が正式採択される予定, Object Management Group (OMG)

(3)知財出願

①国内出願 (3 件)

1. 発明の名称:要件構造表示装置及びプログラム、発明者: 伊東敦、上野肇、島田利郎、山浦一郎、出願人：富士ゼロックス株式会社、出願日：2010/11/26、出願番号：特願 2010-263681.
2. ディペンダビリティ維持装置、ディペンダビリティ維持装置の制御方法、障害対応システム、制御プログラムおよびそれを記録したコンピュータ読み取り可能な記録媒体、松野裕、横手 靖彦、独立行政法人科学技術振興機構、2010 年 11 月 30 日、特願 2010-26746
3. ディペンダビリティ維持装置、ディペンダビリティ維持システム、ディペンダビリティ維持装置の制御方法、制御プログラムおよびそれを記録したコンピュータ読み取り可能な記録媒体、横手 靖彦、所 眞理雄、山本 修一郎、独立行政法人科学技術振興機構、2011 年 08 月 12 日、特願 2011-177322

②海外出願 (2 件)

1. 日本出願の「要件構造表示装置及びプログラム」を、タイトル「REQUIREMENT STRUCTURE DISPLAY APPARATUS AND COMPUTER READABLE MEDIUM」として、米国(出願日:2011/4/15、出願番号:13/087964)、中国(出願日:2011/5/18、出願番号:201110129221.5)、オーストラリア(出願日:2011/4/21、出願番号:2011201866)へ出願
2. ディペンダビリティ維持装置、ディペンダビリティ維持システム、ディペンダビリティ維持装置の制御方法、制御プログラムおよびそれを記録したコンピュータ読み取り可能な記録媒体、横手 靖彦、所 眞理雄、山本 修一郎、独立行政法人科学技術振興機構、2011 年 11 月 14 日 PCT 出願済(JP2011-076219)

③その他の知的財産権

なし。

(4)原著論文発表 (国内(和文)誌 4 件、国際 (欧文) 誌 18 件)

1. Masato Sakai, Hiroya Matsuba and Yutaka Ishikawa ``Fault Detection System

Activated by Failure Information,'' Proceedings of the 13th Pacific Rim International Symposium on Dependable Computing (PRDC'07), pp. 19 -- 26, Melbourne, Australia, December 2007
2. Hajime Fujita, Hiroya Matsuba, Yutaka Ishikawa, ″TCP Connection Scheduler in Single IP Cluster″, 8th IEEE International Symposium on Cluster Computing and the Grid (CCGRID'08), pp. 366-375, Lyon, France, May 2008
3. Taku Shimosawa, Hiroya Matsuba, Yutaka Ishikawa, ″Logical Partitioning without Architectural Supports″, 32nd IEEE International Computer Software and Applications Conference (COMPSAC 2008), pp. 355-364, Truku, Finland, 2008
4. 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
5. 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
6. 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
7. Balazs Gerofi, Yutaka Ishikawa, A Multi-core Approach to Providing Fault Tolerance for Non-deterministic Services, 2010 Ninth IEEE International Symposium on Network Computing and Applications, pp. 233-238, Jul 2010 (DOI: 10.1109/NCA.2010.42)
8. Jianwei Liao, Yutaka Ishikawa, A New Concurrent Checkpoint Mechanism for Real-Time and Interactive Processes, 2010 IEEE 34th Annual Computer Software and Applications Conference 2010, pp. 47-52, Jul 2010 (DOI: 10.1109/COMPSAC.2010.12) (Short Paper)
9. Balazs Gerofi, Hajime Fujita, Yutaka Ishikawa, An Efficient Process Live Migration Mechanism for Load Balanced Distributed Virtual Environments, 2010 IEEE International Conference on Cluster Computing (Cluster 2010), pp. 197-206, Sep 2010 (DOI:10.1109/CLUSTER.2010.25)
10. Jun Kato, Hajime Fujita, Yutaka Ishikawa, Design and Implementation of a Fault Tolerant Single IP Address Cluster, The 16th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'10), pp. 175--183, Dec 2010 (DOI: 10.1109/PRDC.2010.39)
11. Yutaka Matsuno, Jin Nakazawa, Makoto Takeyama, Midori Sugaya, Yutaka Ishikawa, Toward a Language for Communication among Stakeholders, Proc. of the 16th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'10), pp. 93-100, Dec 2010 (DOI: 10.1109/PRDC.2010.47)
12. Hajime Fujita, Yutaka Ishikawa, DTS: Broadcast-based Content-aware TCP Connection Handover, IPSJ Transactions on Advanced Computing Systems (ACS 33) , Vol. 4 No. 2 pp.59-69, Mar 2011
13. Jin Nakazawa, Yutaka Matsuno, Hideyuki Tokuda, Evaluating Degree of Systems' with Semi-Structured Assurance Case, Proc. 13th European Workshop on Dependable Computing (EWDC 2011), 2 pages, May 2011 (DOI: 10.1145/1978582.1978607) (Short Paper)
14. Yutaka Matsuno, Kenji Taguchi, Parameterised Argument Structure for GSN patterns, Proc. IEEE 11th International Conference on Quality Software (QSIC 2011), 6 pages, July 2011 (DOI: 10.1109/QSIC.2011.35) (Short Paper)

15. Jianwei Liao, Taku Shimosawa, Yutaka Ishikawa, Configurable Reliability in Multi-core Operating Systems, IEEE 14th International Conference on Computational Science and Engineering, Aug 2011
16. Yutaka Matsuno, Makoto Takeyama, Jin Nakazawa, Dependability Case for Open Systems Lifecycle, Proceedings of Inconsistency Robustness, Aug 2011, Stanford University, California USA
17. Balazs Gerofi, Yutaka Ishikawa, RDMA based Replication of Multiprocessor Virtual Machines over High-Performance Interconnects, IEEE International Conference on Cluster Computing (CLUSTER), Sep 2011, Austin Texas USA
18. Hajime Fujita, Yutaka Ishikawa, Anytime Available Single IP Address Cluster, The 13th IEEE International High Assurance Systems Engineering Symposium (HASE 2011), Nov 2011 (Short Paper; to appear)
19. Balazs Gerofi, Zoltan Vass, Yutaka Ishikawa, Utilizing Memory Content Similarity for Improving the Performance of Replicated Virtual Machines, ACM/IEEE International Conference on Utility and Cloud Computing (UCC), Dec 2011, Melbourne Australia
20. Balazs Gerofi, Yutaka Ishikawa, Workload Adaptive Checkpoint Scheduling of Virtual Machine Replication, IEEE Pacific Rim International Symposium on Dependable Computing (PRDC), Dec 2011, Pasadena California USA
21. Balazs Gerofi, Yutaka Ishikawa, Enhancing TCP Throughput of Highly Available Virtual Machines via Speculative Communication, ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE), Mar 2012, London, UK
22. Hajime Fujita, Yutaka Matsuno, Toshihiro Hanawa, Hajime Ueno, Mitsuhisa Sato, Shinpei Kato, Yutaka Ishikawa, DS-Bench Toolset: Tools for Dependability Benchmarking with Simulation and Assurance, The 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012), Jun 2012, Bonston, USA (to appear)

(5)その他の著作物(総説、書籍など)

1. 藤田肇、「ディペンダブル・シングルシステムイメージOS」、ソフトウェアデザイン2010年2月号、pp.89-92、技術評論社
2. 松野裕, D-Case:ステークホルダとシステムをつなぐドキュメント, Bulletin JASA, 2010, available from http://www.jasa.or.jp/top/activity/bulletin/bulletin034-03.html
3. 山本 啓二, A Study on Execution Time Analysis of Real-Time Tasks, 東京大学学位論文, 博士(情報理工学), 平成22年3月
4. 藤田 肇, OS Supported Dependable Single IP Address Cluster, 東京大学学位論文, 博士(情報理工学), 平成24年3月
5. Balazs Gerofi, Efficient Replication Mechanisms for Highly Available Virtual Machines, 東京大学学位論文, 博士(情報理工学), 平成24年3月
6. 廖 剣偉, Replication Mechanisms of Process and File System’s Metadata for
7. Fault Tolerance, 東京大学学位論文, 博士(情報理工学), 平成24年3月
8. 下沢 拓, Operating System Organization for Manycore Systems, 東京大学学位論文, 博士(情報理工学), 平成24年3月

(6)国際学会発表及び主要な国内学会発表

① 招待講演　　（国内会議 3 件、国際会議 3 件）

1. 石川裕、「ディペンダブルシステムソフトウェアにおける挑戦」、 組込みシステムシンポジウム 2007、情報処理学会、東京、H19 年 10 月
2. 石川裕、「システムソフトウェアにおける検証技術への期待」、第 4 回システム検証の科学技術シンポジウム、 日本ソフトウェア科学会ディペンダブルソフトウェア研究会、名古屋、H19 年 11 月
3. 石川 裕, 「ディペンダブルシステムソフトウェア開発環境構築に向けて」, 第 20 回 コンピュータシステム・シンポジウム (ComSys2008), 11 月
4. Yutaka Ishikawa, Hajime Fujita, Toshiyuki Maeda, Motohiko Matsuda, Midori Sugaya, Mitsuhisa Sato, Toshihiro Hanawa, Shinichi Miura, Taisuke Boku, Yuki Kinebuchi, Lei Sun, Tatsuo Nakajima, Jin Nakazawa, and Hideyuki Tokuda, "Towards an Open Dependable Operating System", IEEE 12th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC 2009), pp. 20-27, Tokyo, Japan, March 2009
5. Yutaka Matsuno, D-Case Editor: A Typed Assurance Case Editor, OSADL 13th Real-Time Linux Workshop, Czech Technical University in Prague, Czech, Oct 2011
6. Yutaka Matsuno, D-Case Editor: A Free Assurance Case Editor, presented in OSADL session of Embedded World, March, 2012, Nuremberg, Germany

② 口頭発表　　（国内会議 25 件、国際会議 10 件）

1. 酒井将人、松葉浩也、石川裕　「エラー情報から原因を特定する障害検知システム」、 第 5 回 ディペンダブルシステムワークショップ、 日本ソフトウェア科学会ディペンダブルソフトウェア研究会、函館、H19 年 7 月
2. 藤田肇、松葉浩也、石川裕、「柔軟な負荷分散を可能にする分散型シングル IP クラスタ」、情報処理学会 研究報告 2007-OS-106、pp. 1 - 8、旭川、H19 年 8 月
3. 平野貴仁、藤田肇、松葉浩也、石川裕、「PBus：柔軟なカーネル機能拡張のためのインターフェイス」、情報処理学会 研究報告 2007-OS-106、pp. 63 - 70、旭川、H19 年 8 月
4. 平野貴仁, 藤田肇, 松葉浩也, 石川裕「カーネル機能拡張のための抽象化レイヤ P-Bus の実装」 情報処理学会研究報告 (OS-108), pp. 17-24, 沖縄, April 2008
5. 藤田肇, 平野貴仁, 松葉浩也, 前田俊行, 菅谷みどり, 石川裕, "安全かつ拡張可能な OS 開発基盤実現に向けて", 第 6 回ディペンダブルシステムワークショップ (DSW2008), pp. 29-32, 函館, July 2008
6. 酒井将人, 藤田肇, 松葉浩也, 石川裕, "ネットワーク管理システムにおける状態収集機構", 第 6 回ディペンダブルシステムワークショップ (DSW2008), pp. 83-86, 函館, July 2008
7. 藤田肇, 平野貴仁, 山本和典, 松葉浩也, 石川裕, 「P-Bus における OS カーネル間通信機構の設計と実装」, 情報処理学会研究報告 (OS-109 SWoPP2008), pp. 33-38, 佐賀, August 2008
8. 下沢拓, 藤田肇, 石川裕, 「マルチコア SH における複数カーネル実行機構の設計と実装」, 情報処理学会研究報告 (OS-109 SWoPP2008), pp. 25--32, 佐賀, August 2008
9. Jun Kato, Hajime Fujita, Yutaka Ishikawa, "Evaluation of Energy-Efficient Cluster Server using Embedded Processors", First International Workshop on Software Technologies for Future Dependable Distributed Systems (STFSSD 2009), 東京, March 2009
10. 藤田肇, 石川裕,「レイヤー7 負荷分散のための TCP 接続移送機構」, 情報処理学会研究報告 (OS-111), 沖縄, Apr 2009

11. 山本啓二，石川裕，松井俊浩，「移植性の高い最悪実行時間予測ツール RETAS の設計と実装」,情報処理学会研究報告 (OS-111)，沖縄，Apr 2009
12. 下沢 拓，石川 裕，「マルチコア向け複数カーネル実行機構におけるデバイス共有」，情報処理学会研究報告 (OS-111)，沖縄，Apr 2009
13. Balazs Gerofi，Hajime Fujita，Yutaka Ishikawa
"TCPmig: Migration of processes with TCP sockets in Single IP Address Cluster"
情報処理学会研究報告 (OS-111)，沖縄，Apr 2009
14. 中澤仁，松野裕，菅谷みどり，塙敏博，前田俊行，藤田肇，石綿陽一，杵渕雄樹，高村博紀，三浦信一，山田浩史，松田元彦，倉光君郎，石川裕，"オペレーティングシステムおよび実システムにおけるディペンダビリティの評価と見積り"，第 7 回ディペンダブルシステムワークショップ (DSW2009)，pp. 27-41，函館，July 2009
15. 加藤真平，藤田肇，中澤仁，松田元彦，前田俊行，杵渕雄樹，塙敏博，三浦信一，石綿陽一，松野裕，高村博紀，山田浩史，吉田哲也，倉光君郎，菅谷みどり，石川裕，"ディペンダブルシステム向けベンチマークフレームワークの提案"，第 7 回ディペンダブルシステムワークショップ (DSW2009)，pp. 171-178，函館，July 2009
16. 加藤純，藤田肇，石川裕，「単一 IP アドレスクラスタにおける耐故障機構の設計と実装」，情報処理学会研究報告 (OS-112，SWoPP 2009)，仙台，Aug 2009
17. Balazs Gerofi, Hajime Fujita, Yutaka Ishikawa, Live Migration of Processes Maintaining Multiple Network Connections, コンピュータシステム・シンポジウム(ComSys2009), pp. 1-10, November 2009
18. 加藤真平，石川裕，「完全モジュール型リアルタイム Linux の開発」，コンピュータシステム・シンポジウム(ComSys2009)，pp. 129-138，November 2009
19. Yutaka Ishikawa，“D-Core and P-Bus”，57th IFIP WG 10.4 Meeting，January 2010
20. 下沢拓，石川裕，複数カーネル実行機構を利用したアプリケーション実行環境の設計と実装,情報処理学会研究報告 (OS-114)、伊東、 Apr 2010
21. Toshinori Takai, Atsushi Ito, Makoto Takeyama, Hajime Ueno, Kenji Taguchi, Hiroki Takamura, Jin Nakazawa, Yutaka Matsuno, A White Paper on Assurance Case Process Metamodel, Presented in System Assurance Task Force, OMG Technical Meeting, June, 2010. Available from http://sysa.omg.org/sysa_presentations.htm
22. 高井利憲，伊東敦，武山誠，上野肇，高村博紀，松野裕、D-Case を用いた保証プロセスについて，第 8 回ディペンダブルシステムワークショップ (DSW2010)，函館，Jul 2010
Available from sysa.omg.org/docs/Sep10/Assurance_Case_Pattern.omg20100921.pdf.
23. 松野裕，組み込みシステムのディペンダビリティ評価，日本応用数理学会 2010 年度年会講演予稿集，pp. 237-238，東京，Sep 2010
24. Yutaka Matsuno, Kenji Taguchi, Hiroki Takamura, Standardization of GSN Patterns in OMG sysA PTF, Presented in System Assurance Task Force, OMG Technical Meeting, September 2010.
25. Yutaka Matsuno, Hiroki Takamura, Yutaka Ishikawa, A Dependability Case Editor with Pattern Library, Procs. IEEE 12th International Symposium on High-Assurance Systems Engineering (HASE), pp. 170-171, Tokyo, Japan, Nov 2010 (Fast Abstract)
26. Hajime Fujita, Yutaka Ishikawa, DTS: Broadcast-based Content-aware TCP Connection Handover, 第 22 回 コンピュータシステム・シンポジウム (ComSys 2010), pp. 21--30, Nov 2010
27. 松野裕，ディペンダビリティ維持システムの試作，第 11 回計測自動制御学会システム

インテグレーション部門講演会講演集, 仙台, Dec 2010

28. 藤田肇, シングルIPアドレスクラスタとOS検証, オペレーティングシステムと形式手法・形式検証に関するワークショップ in 金沢, 金沢, Dec 2010
29. Hajime Fujita, Motohiko Matsuda, Toshiyuki Maeda, Shin'ichi Miura, Yutaka Ishikawa, P-Bus: Programming Interface Layer for Safe OS Kernel Extensions, The 16th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'10), pp. 235-236, Tokyo, Japan, Dec 2010 (Fast Abstract)
30. Yutaka Matsuno, Jin Nakazawa, Makoto Takeyama, Toshinori Takai, Takeo Matsuzaki, Kenji Taguchi, Atsushi Ito, Hajime Ueno, Dependability Case for Open Systems Lifecycle, Presented in System Assurance Task Force, OMG Technical Meeting, December 2010.
    Available from http://www.omg.org/cgi-bin/doc?sysa/2010-12-04
31. Yutaka Matsuno, Jin Nakazawa, Makoto Takeyama, Toshinori Takai, Takeo Matsuzaki and Kenji Taguchi and Atsushi Ito, Hajime Ueno, Dependability Case for Open Systems Lifecycle, Presented in Real-Time & Embedded Systems forum, Open Group Conference, February 2011.
32. 藤原祐二, 藤田肇, 石川裕, 耐故障分散ロック機構の設計と検証, 情報処理学会研究報告 (OS-118 SWoPP 2011), 鹿児島, Jul 2011
33. Makoto Takeyama, Kenji Taguchi, Yutaka Matsuno, Mechanizing Assurance Cases, Presented in System Assurance Task Force, OMG Technical Meeting, Sep 2011
34. Luke Emmet, Yutaka Matsuno, Assurance Case Repository Whitepaper (Draft), Presented in System Assurance Task Force, OMG Technical Meeting, Sep 2011
35. Yutaka Matsuno, Kenji Taguchi, Yoshihiro Nakabo, Akira Ohata, Iterative and Simultaneous Development of Embedded Control Software and Dependability Case, Workshop on Dependable Systems of Systems, Sep 2011, University of York, York UK

③ ポスター発表　　(国内会議　2件、国際会議　0件)

1. 菅谷 みどり, 藤田 肇, 塙 敏博, 中澤 仁, 松田 元彦, 前田 俊行, "ディペンダブルな組込みOSの提案", 第10回 組込みシステム技術に関するサマーワークショップ(SWEST10) 予稿集, pp. 35-38, 浜松, September 2008
2. 藤田 肇, 松野 裕, 塙 敏博, 佐藤 三久, 加藤 真平, 石川 裕, DS-Bench ツールセット: ディペンダビリティベンチマークのための支援ツール, ディペンダブルシステムワークショップ&シンポジウム (DSW & DSS 2011), 京都, 平成23年12月

(7)受賞・報道等

① 受賞

1. 藤田肇、平成20年度情報処理学会コンピュータサイエンス領域奨励賞
2. 下沢拓、第109回OS研究会(平成20年8月) 最優秀学生発表賞
3. Balazs Gerofi、情報処理学会コンピュータシステム・シンポジウム若手優秀論文賞(平成21年11月)
4. *下沢拓、平成21年度情報処理学会山下記念賞

② マスコミ(新聞・TV等)報道(プレス発表をした場合にはその概要もお書き下さい。)

なし

③ その他

1. ポスター展示および P-Bus デモ発表、CREST 公開シンポジウム、東京、平成 19 年 12 月

2. P-Bus API 仕様書を公開
(URL: http://www.il.is.s.u-tokyo.ac.jp/deos/pbus/spec/)、平成 20 年 4 月
3. 研究内容紹介および P-Bus デモンストレーション、組み込みシステム開発技術展(ESEC)、東京、平成 20 年 5 月
4. 研究内容紹介および、筑波大佐藤グループによって開発されている RI2N を P-Bus 上で動作させるデモ、組み込み総合技術展(ET2008)、横浜、平成 20 年 11 月
5. シングル IP アドレス機構を用いたディペンダブルファイルサーバのデモ展示、CREST 中間成果報告会、東京、平成 21 年 9 月
6. シングル IP アドレス機構についてのポスター展示、ET2009、横浜、平成 21 年 11 月
7. シングル IP アドレス機構を用いたディペンダブルファイルサーバのデモ展示, ET2010, 横浜, 平成 22 年 11 月
8. DS-Bench2 のデモ展示, ET2010, 横浜, 平成 22 年 11 月
9. D-fops と D-Case 統合デモ, ET2010, 横浜、平成 22 年11月
10. 松野裕、説明責任を支援する技術(1), ET2010 カンファレンス C6, オープンシステムが世界を変えるにおける講演, 平成 22 年 11 月
11. 横手靖彦、説明責任を支援する技術(2), ET2010 カンファレンス C6, オープンシステムが世界を変えるにおける講演, 平成 22 年 11 月
12. シングル IP アドレス機構を用いたディペンダブルファイルサーバのデモ展示, PRDC2010, 東京, 平成 22 年 12 月
13. DS-Bench2 のデモ展示, PRDC2010, 東京, 平成 22 年 12 月
14. D-fops と D-Case 統合デモ、ET2010、横浜、平成 22 年年 11 月
15. DS-Bench/Test-Env/D-Case 統合デモ、D-RE デモ、D-Case デモ、ET2011、横浜、平成 23 年 11 月
16. 松野裕, 変化し続けるシステムのディペンダビリティ合意形成の方法とツール, ET2011 カンファレンス C7-2, オープンシステムが世界を変えるにおける講演、平成 23 年 11 月
17. 横手靖彦、ディペンダブルシステムの実行環境 ET2011 カンファレンス C7-2, オープンシステムが世界を変えるにおける講演、平成 23 年 11 月

(8)成果展開事例

①実用化に向けての展開

- 富士ゼロックスにおいて、開発部門および品質保証部門へ展開を実施(D-Case 手法およびツールの紹介と適用トライアルを実施した)。
- D-Case Editor の商用化についての提案をある企業より受けている。
- DEOS センターにて D-RE に関するリファレンス実装の開発は継続されている。

② 社会還元的な展開活動

当研究チームは、他の研究チームを取りまとめて、DEOS アーキテクチャ、DEOS プロセスの全体像を決めていった。また、ET2010 では、オープンシステムにおけるディペンダビリティに関する講演会を開催した。さらに、先に述べているとおり、研究成果である開発ソフトウェアはインターネット上で公開している。

§6 研究期間中の主なワークショップ、シンポジウム、アウトリーチ等の活動

年月日	名称	場所	参加人数	概要
H22 年 12 月 16ー17 日	DEOS 国際シンポジウム	慶応大学三田キャンパス		招待講演者 3 名の招聘事務作業

§7 結び

研究開発当初の目標に加えて新たなる目標が設定された。当初の目標である「高信頼単一システムイメージを提供する並列分散 OS」に関しては、原著論文発表で示しているとおり、情報処理学会論文誌や IEEE 主催の Cluster、CCGrid、PRDC 等国際会議で多くの成果発表を行った。開発したシステムも公開する予定であり、当初の目標を達成していると考える。プロジェクトの途中で新たに加わった研究開発である DEOS プロセス関連では研究論文発表数が多くないが特許出願も行っており一定の成果は上がっていると考える。DEOS プロセスのさらなる進化は後 2 年間研究開発が残っている 3 期研究チーム群に期待する。研究チーム代表者のミッション以外に、本研究領域の全研究チームの若手研究者から構成される DEOS コアチームを編成し取りまとめ、本研究領域で研究開発するディペンダブルシステムに必要とされる要件とそれを実現するための要素技術の検討を行ってきた。このような研究体制は、本来であれば、公募時にルール化しておくべきであったろう。