

平成 22 年度 成果報告書

「PaaS-CAE 基盤技術に関する研究開発」

目 次

1	研究開発課題の背景	3
2	研究開発の全体計画	
2-1	研究開発課題の概要	4
2-2	研究開発の最終目標	6
2-3	研究開発の年度別計画	9
3	研究開発体制	
3-1	研究開発実施体制	10
4	研究開発実施状況	
4-1	HTTPS ベースでリモートアプリケーションサーバの画面を高速に伝送する機構の研究開発	11
4-1-1	研究開発内容	11
4-1-2	実施状況	11
4-1-3	達成状況及び今後の課題	13
4-2	RCM システムの負荷分散、冗長機構の研究開発	13
4-2-1	研究開発内容	13
4-2-2	実施状況	13
4-2-3	達成状況及び今後の課題	17
4-3	データベースの高機密化（排他的記録）機構の研究開発	17
4-3-1	研究開発内容	17
4-3-2	実施状況	17
4-3-3	達成状況及び今後の課題	18
4-4	Workflow や高品位な UI を GUI で設定、変更を可能にする機構の研究開発	19
4-4-1	研究開発内容	19
4-4-2	実施状況	19
4-4-3	達成状況及び今後の課題	25
4-5	RCM システム間（WebServer-WebServer）の連携機構の研究開発	26
4-5-1	研究開発内容	26
4-5-2	実施状況	26
4-5-3	達成状況及び今後の課題	28
4-6	既存（非 RCM）社内 R&D システムとの連携機構の研究開発	28
4-6-1	研究開発内容	28
4-6-2	実施状況	28
4-6-3	達成状況及び今後の課題	30
4-7	実証システムの構築と運用	30
4-7-1	研究開発内容	30

4-7-2	実施状況	31
4-7-3	達成状況及び今後の課題	37
4-8	総括	38
5	参考資料・参考文献	
5-1	研究発表・講演等一覧	40
5-2	産業財産権	40

1 研究開発課題の背景

- (1) データ構造やプロセスの変化が激しく、またリモートアプリケーションサーバとのフレキシブルな連動が必要な R&D 系業務に効率化、高品質化、継承容易性を備えさせるためには、R&D 系業務のシステム化が必要である。しかしながら、従来のシステム化手法（Web プログラミングと DB 設計が必要な特注開発）では、R&D 系業務の非定常、非定型性のため十分なシステム化が容易ではない。
- (2) 研究開発者には、できる限り研究開発に直接つながる業務、つまりクリエイティブな業務に集中できる環境を準備することが、会社として、また、研究開発部門として重要な課題である。しかしながら、実験計測器やコンピュータの性能が向上し、研究開発内の様々な雑務（ファイル探査やグラフ化処理等）が多く、研究開発者がクリエイティブな業務に割ける時間が少なくなっている。
- (3) いかなるクリエイティブな業務においても、適切な批判の目に晒されなければ、その正確性および洞察力に曇りが生じる。R&D 系業務に関してもその内容および進捗に関し、様々な人（特に、経営層や研究開発責任者）へ意見を求めたり、説明を行うことが重要である。そのためには、研究開発の進展状況等を研究者本人以外も簡単にトレースできる必要がある。しかしながら、現状では、研究開発が蛸壺型で、個人に委ねられており、適切な批判の目に晒された R&D 系業務にはなっていない。
- (4) CAE、実験データ解析に上記要求を満たす機能を付加したシステムを社内で構築するには多くの問題が存在する。
 - ・閑散期にあわせると必要なときに使えない。
 - ・繁盛期にあわせて設備投資すると過剰投資になる。
 - ・新しいソフトを試したいのだが、気軽に試せない。
 - ・高度なシステム管理者が必要である。
 - ・システム構築自体が特注開発であり、コストが膨大になるだけでなく、特注システムは納品をしてしまうと開発プロジェクトが終了してしまうため、製品システムに比べソフトウェア品質、保守性の点で遥かに劣ってしまう。つまり、システムを使う側としては、ハイリスク・ローリターンになってしまう。
- (5) SaaS 化は、上記問題を解決しうるが、CAE は解析課題ごとに異なる自由度が要求され、GUI 画面や DB 化は、利用者ごと解析課題ごとに異なるものにしなければならない。したがって、従来の SaaS 化手法では、対応できなく、PaaS である必要がある。
- (6) CAE などのシミュレーション系業務は、社外のシステム（Web の向こう側のシステム）でも良く、商用 PaaS サービスを活用することができる。しかし、実験データなどは、装置のある社内側にシステムを構築しなければならない、商用 PaaS サービスで業務を行うことはできない。社外の商用 PaaS サービスと社内側システムの双方を FireWall を超えてシームレスに連携できる機構が実現されなければならない。

2 研究開発の全体計画

2-1 研究開発課題の概要

PaaS-CAE 基盤技術に関する研究開発

経理や営業等の事務業務系システムは、特注開発システムから SaaS、PaaS 等のクラウドコンピューティング化が進んでいる。一方、CAE や実験データ解析等の R&D 系業務は、取り扱うべき対象や処理プロセスの変化が激しく、特注開発でさえシステム化が容易ではない。RCM は、上記困難を克服し、R&D 系業務のシステム化を可能にした。

本開発は、RCM を拡張し、R&D 系業務の PaaS 化を可能にすること（FireWall 透過性、信頼性、機密性、利便性）を目指すものである。ただし、R&D 業務では社外 PaaS 化になじまない実験装置等が存在するため、PaaS と社内 R&D システムとを連携する仕組みの開発を行う。これらの目標を達成するために具体的には、以下の 6 つの機構を開発し、実証システムの構築及び運用を行う。

- (1) HTTPS ベースでリモートアプリケーションサーバの画面を高速に伝送する機構
- (2) RCM システムの負荷分散、冗長機構
- (3) データベースの高機密化（排他的記録）機構
- (4) Workflow や高品位な UI を GUI で設定、変更を可能にする機構
- (5) RCM システム間（WebServer-WebServer）の連携機構
- (6) 既存（非 RCM）社内 R&D システムとの連携機構
- (7) 実証システムの構築と運用

- (1) HTTPS ベースでリモートアプリケーションサーバの画面を高速に伝送する機構

現状の RCM システムソフトウェアは、Workflow 機能によりリモートのアプリケーションサーバの画面をクライアントに表示させ、インタラクティブな操作を可能にすることを実現できている。この方式では、SSH トンネリング機能及び VNC サーバ機能を利用しているため、クライアントとリモートアプリケーションサーバ間が直接 SSH で接続できる必要がある。

社内システムの場合は、上記接続方式で大きな問題は生じないが、SaaS や PaaS 環境のように社外にシステムがある場合は、上記接続方式は、Firewall 等に阻まれることになる。したがって、リモートアプリケーションサーバは、コントロールサーバに SSH のみで接続し、クライアントはシステムの Web サーバに HTTPS のみで接続することができる機構を開発する必要がある。

本開発は、RCM からリモートアプリケーションサーバのインタラクティブアプリケーションを利用する場合における FW 透過性を獲得することを目指すものである。

本開発では、現状方式に比べ通信ルートが複雑となるため単純な実装ではインタラクティブ性の性能劣化が予想される。本開発では、機構開発とともに性能向上も大きな課題になるため、この機構開発に限って、本研究開発の方針を破り、リモートアプリケーションサーバに通信性能向上のための専用モジュールを追加する可能性もある。

- (2) RCM システムの負荷分散、冗長機構

現状の RCM は、リモートアプリケーションサーバの負荷分散、冗長機構を有しているが、システムコアである WebServer、ControlServer、DataBaseServer は、それぞれ、1 システム 1 サーバとなっており、負荷分散、冗長機構を持っていな

い。既存の Web システムの負荷分散や冗長化関連製品は、セッション情報の冗長化しか行わないが、RCM は、Workflow 機能等の複雑な状態遷移処理機構により、連続的な処理やユーザインタフェース構築を自動的に行っているためセッション情報の冗長化だけでは、システム全体の冗長化を達成できない。したがって、既存の負荷分散、冗長化製品では RCM の連続稼働性を保証できない。

本開発は、RCM を拡張し、WebServer、ControlServer、DataBaseServer の負荷分散、冗長機構を付加し、商用 PaaS サービスに耐えうる信頼性を獲得することを目指すものである。

冗長化機能を考えると、データベース部は、永続的なデータを格納しているため完全な多重化システム（全ての状態が同一）が必要となるが、Web、Control 部は、Workflow 単位での永続性のみが必要であることから、Workflow 単位での多重化で十分である。したがって、本機能の負荷軽減のため Master-Slave ペアを確立し、どちらか一方に障害があった場合は、自らがマスターになり動的に Slave を割り当てる動的 Master-Slave ペア方式を採用する。

(3) データベースの高機密化（排他的記録）機構

現状の RCM は、データベース部の並列化が可能であるが、この並列化はあくまで高速化としての側面である。しかしながら、データの機密性を重んじるシステムの場合は、データの機密度ごとに異なるハードウェアを準備し、そこに格納することが条件づけられる。例えば、PaaS サービスの場合、より高い価格を支払うことである情報部分のデータのみを別ハードウェアへ割り当てることを要求する場合も考えられる。事務業務系 PaaS サービスであっても、この条件をクリアできないがために PaaS サービスが採用されない場合も多い。

本開発は、データベースの排他的記録機構を開発し、任意のデータおよびメタデータを別ハードウェアに格納することを可能にする機能の獲得を目指すものである。

(4) Workflow や高品位な UI を GUI で設定、変更を可能にする機構

現状の RCM システムソフトウェアは、XML-Workflow 内に変数化の設定や UI の空間配置等のデザインを XML で記述している。入力系 UI を簡易 UI と呼び、出力系 UI を XML-Viewer と呼んでいる。これら UI は、汎用的な XML を表示するための強力なエンジンではあるが、特注開発や商用アプリケーションと比べると、装飾性が非力であり、画面表現力が乏しい。RCM では、入力系 UI、出力系 UI とともに外部で記述した HTML と差し替えることができるように実装されているが、PaaS 化を考えると GUI で UI のデザインを設計し、変更できる程度にまで利便性を向上させる必要がある。また、XML-Workflow 全体に関しても、現状は、特別なツールがないため通常の XML エディターで入力・編集を行っているが、これも GUI インタフェースで入力・編集できるようにする必要がある。

本開発は、GUI で XML-Workflow および入力系 UI、出力系 UI のデザインを設計し、変更できる機能の獲得を目指すものである。

(5) RCM システム間（WebServer-WebServer）の連携機構

現状の RCM は、1 システムで完全に閉じた形となっており、システム同士は連携できない。商用 PaaS サービスと社内システムのように 2 つのシステム配下のリモートアプリケーションサーバの双方を使った処理プロセスは、研究開発者が手動で 2 つのシステムを切り替えながら実行するしかない。つまり、双方跨って利

用するサービスなど強く連携したサービス構築は不可能である。

したがって、装置のある社内側に存在する実験データは、社内 R&D システムで処理し、シミュレーションは、商用 PaaS サービスを使いそれらと比較するような体系的な処理を実現する場合、社内 R&D システムと商用 PaaS サービスの双方のシステムがシームレスに連携できる必要がある。

本開発は、WebServer-WebServer の連携機構を開発し、RCM を拡張し、複数の (RCM を使った) システムをシームレスに連携することができる機能の獲得を目指すものである。

(6) 既存 (非 RCM) 社内 R&D システムとの連携機構

現状の RCM システムソフトウェアは、Workflow 機能に含まれている SSH 接続機能により、リモートアプリケーションサーバへのコマンド実行、つまりアプリケーション利用を可能にしている。その他、REST によるコマンド発行も可能である。しかしながら、独自に作られた既存システム (ほとんどのシステムは、特注開発であり、仕様で決められた機能を実現するための独自実装である) と連携することは、困難である。

本開発は、既存システムとの連携を行うために、任意の通信プロトコルを受信し、応答するためのブリッジモジュール機能の獲得を目指すものである。また、拡張性の高い汎用通信プロトコルである SOAP や REST 等は、標準装備する予定である。

(7) 実証システムの構築と運用

本研究開発で作成した機構をインプリメントした RCM を使って、実証システムを構築し、CAE における仮想商用 PaaS サービス+社内 R&D システムの連携システムが有効に機能することを確認する。また、実際のシステム運用によって運用モデルを明確化するとともに、問題点等を改善し、より使いやすいシステムへと改良を行う。

2-2 研究開発の最終目標 (平成 23 年 10 月末)

(1) HTTPS ベースでリモートアプリケーションサーバの画面を高速に伝送する機構

- 1.1 クライアントと別ネットワークにあるリモートアプリケーションサーバに関して、クライアントはシステムの RCM-Web サーバに HTTPS のみで接続するものとし、リモートアプリケーションサーバは、RCM コントロールサーバに SSH のみで接続することで、クライアントからリモートアプリケーションサーバのインタラクティブアプリケーションを操作できること。
- 1.2 現状の接続方式 (クライアントと別ネットワークにあるリモートアプリケーションサーバが direct に SSH でつながっている) に比べ、中継が 2 段になり、伝送路が 3 倍になるが、現状の 1/5 以内のインタラクティブ性能 (FPS) を確保する。

(2) RCM システムの負荷分散、冗長機構

- 2.1 RCM-Web サーバの分散化ができ、ロードバランス機能により負荷を分散できること。
- 2.2 RCM-Web サーバの一部に障害が発生した場合は、他の RCM-Web サーバが処理を継続できること。

- 2.3 RCM- Control サーバの分散化ができ、ロードバランス機能により負荷を分散できること。
 - 2.4 RCM- Control サーバの一部に障害が発生した場合は、他の RCM- Control サーバが処理を継続できること。
 - 2.5 RCM-DB サーバの分散化ができ、ロードバランス機能により負荷を分散できること。
 - 2.6 RCM-DB サーバの一部に障害が発生した場合は、他の RCM- DB サーバが処理を継続できること。
- (3) データベースの高機密化（排他的記録）機構
- 3.1 任意のデータおよびメタデータを別ハードウェアに格納するための DB リクエストを開発し、それ以外のデータとハードウェア的に分離した管理が可能であること。
- (4) Workflow や高品位な UI を GUI で設定、変更を可能にする機構
- 4.1 XML-Workflow（入力系 UI、出力系 UI を含む）を GUI 画面で設定できること。
 - 4.2 XML-Workflow 設定 GUI 画面では、XML のタグ名入力は不要にすること。
 - 4.3 XML-Workflow 設定 GUI 画面では、固定的な複数選択しは、ドロップダウンで選べるようにすること。
 - 4.4 XML-Workflow 設定 GUI 画面では、数値、日付など明らかなフォーマット指定がある場合は、入力時にそのフォーマットを誘導するとともにチェック機構を有すること。
 - 4.5 XML-Workflow 設定 GUI 画面では、job 間の相関性（参照、重複不可等）を意識させながら入力できること。
 - 4.6 入力系 UI、出力系 UI 設定は、レイアウト設定が簡単なように HomePage 作成ツールのようなレイアウト画面で設定できること。
- (5) RCM システム間（WebServer-WebServer）の連携機構
- 5.1 RCM の Workflow 記述において、異なる RCM システムを跨いだ記述が可能であり、Workflow 内で異なる RCM システムで実行される job（Workflow 内の 1 つの作業単位で最小記述レベルでもある）は、RCM システムの WebServer 間連携機構より、他方の RCM システムに処理を依頼し、その戻値を受け取ることができること。
 - 5.2 RCM の Workflow 記述において、1 つの job 内で複数の異なるサーバにアクセスを行うものに関しても、異なる RCM システムを跨いだ記述が可能であり、当該 job 内で異なる RCM システム支配下のサーバを利用する場合、RCM システムの WebServer 間連携機構により、他方の RCM システムと連携して 1 つの job 機能を果たすこと。
- (6) 既存（非 RCM）社内 R&D システムとの連携機構
- 6.1 任意の通信プロトコルを受信し、応答するためのブリッジモジュール機構を開発し、既存システムの通信プロトコルに合わせ専用ブリッジモジュールを開発でき、Controller に簡単に（システム全体のリコンパイルなしに）追加できること。
 - 6.2 ブリッジモジュールにおいて SOAP プロトコルを使う場合、ESB を使った通信を

サポートできるようにすること。

- 6.3 ブリッジモジュールの稼働・停止状態を RCM システムのメンテナンスモードから監視、制御できること。

(7) 実証システムの構築と運用

- 7.1 2つ以上の RCM システムを別ネットワークで構築、運用し、(1)～(6)の機能が正しく動作するかを1ヶ月以上検証すること。
- 7.2 (1)～(6)の機能の性能を評価し、性能、機能面で利用上問題がある部分に関して、ボトルネックポイントを同定し、改善プランを策定すること。
- 7.3 7.1の検証時に運用モデルを明確化し、運用マニュアル、運用における注意点をドキュメントにまとめること。また、そのマニュアルに従って、運用を実際に行って、問題点がないかを確認すること。

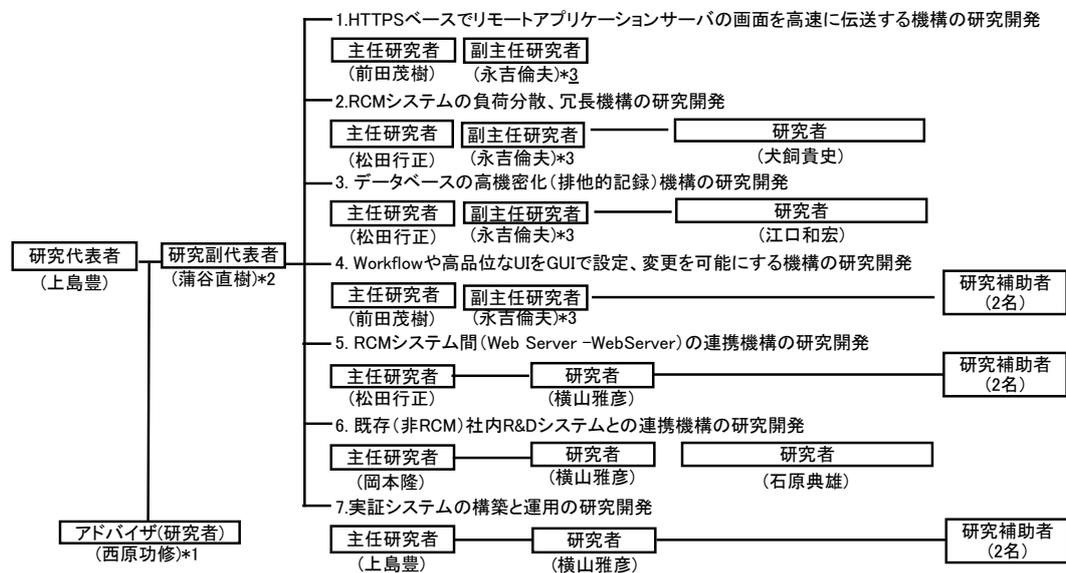
2-3 研究開発の年度別計画

金額は非公表

研究開発項目	21年度	22年度	23年度	計	備考
PaaS-CAE 基盤技術に関する研究開発					
(1)HTTPS ベースでリモートアプリケーションサーバの画面を高速に伝送する機構の研究開発	—	—	—	—	
(2)RCM システムの負荷分散、冗長機構の研究開発	—	—	—	—	
(3)データベースの高機密化（使用領域指定）機構の研究開発	—	—	—	—	
(4)Workflow や高品位な UI を GUI で設定、変更を可能にする機構の研究開発	—	—	—	—	
(5)RCM システム間(WebServer-WebServer)連携機構の研究開発	—	—	—	—	
(6)既存システムとの連携機構の研究開発	—	—	—	—	
(7)実証システムの構築と運用の研究開発	—	—	—	—	
間接経費	—	—	—	—	
合計	—	—	—	—	

3 研究開発体制

3-1 研究開発実施体制



*1) 1～7の研究課題全般にわたり、横断的な観点でシステム全体が最良な進展をするように、研究代表者、各課題の主任研究者(主任研究者)と合議を行い、研究方針の決定を行う。

*2) 1～7の研究課題全般にわたり、研究代表者である上島の補佐的役割を果たす。

*3) 副主任研究者は、研究開発実務を行い、且つ、主任研究者を補佐する。

4 研究開発実施状況

4-1 HTTPS ベースでリモートアプリケーションサーバの画面を高速に伝送する機構の研究開発

4-1-1 研究開発内容

1.1 RCM に、HTTPS ベースによる、リモートアプリケーションサーバの画面伝送機構を開発した。これにより、HTTPS 通信路上で、クライアントからリモートアプリケーションサーバのインタラクティブアプリケーションの操作を行うことが可能となった。

1.2 HTTPS ベースの画面伝送のインタラクティブ性能のチューニングを行い、実用上問題のない速度を確保した。具体的には、SSH 直接接続時の速度に比べ、1/5 以内の FPS (Frame Per Second) を確保した

4-1-2 実施状況

(1) HTTPS ベースでリモートアプリケーションサーバの画面を高速に伝送する機構の開発を実施した。

実施内容 1

図 1に示す通り、アプリケーションサーバ上の GUI を、HTTPS ベースでクライアント PC に伝送する機構を開発した。

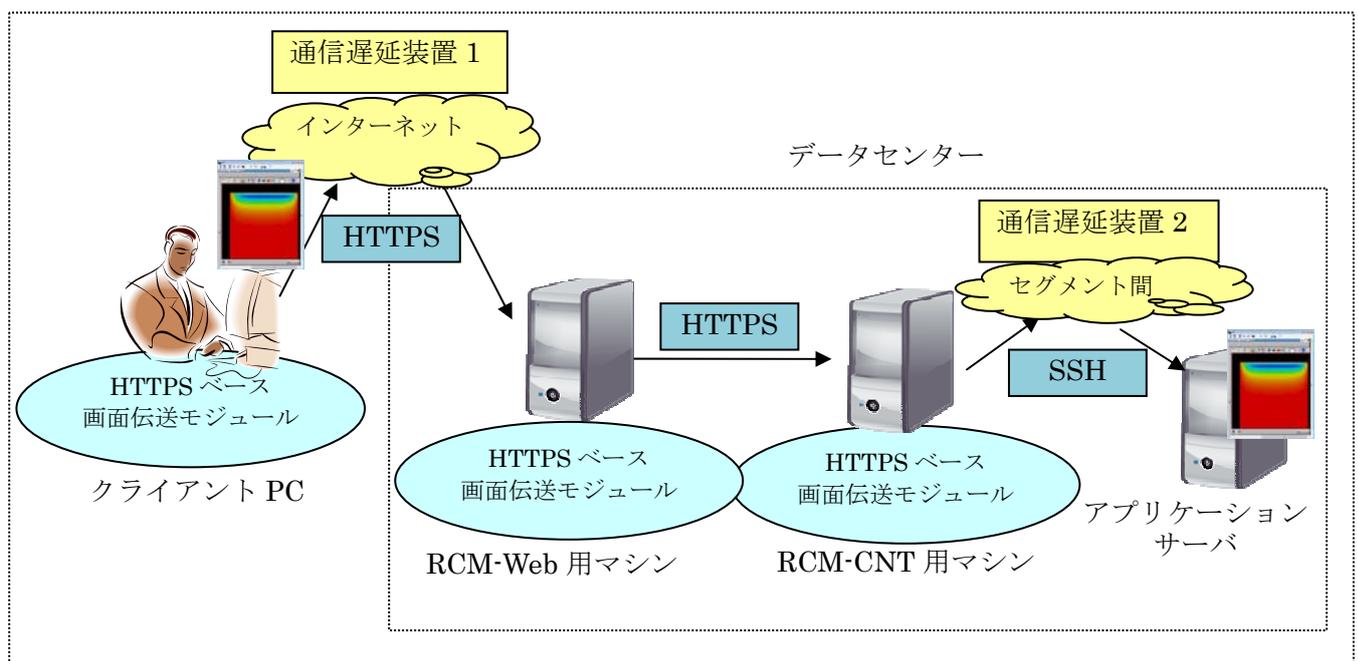
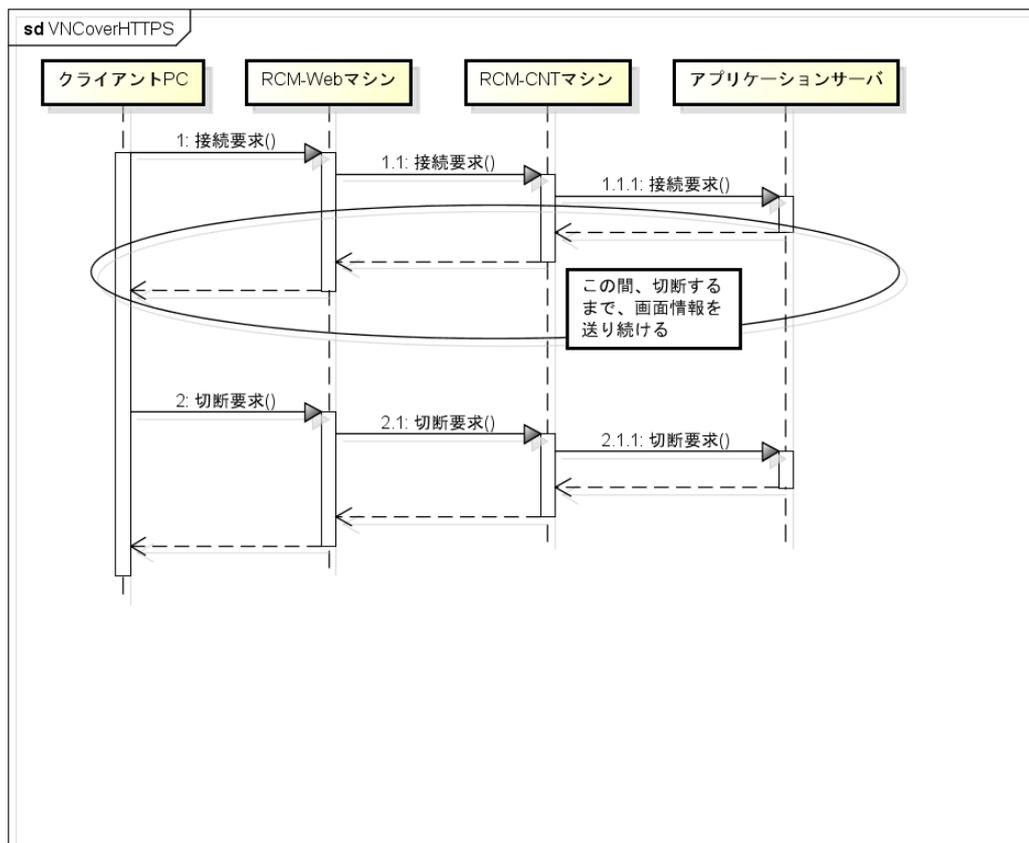


図 1 HTTPS ベースの画面伝送路

具体的な通信のシーケンスを次ページの図 2に示す。



powered by astah

図 2 VNCoverHTTPS のシーケンス

一般的に、RCM-Web マシンへは、インターネット回線の外部から接続できるが、アプリケーションサーバは、インターネット回線の外部からは直接接続できないように Firewallなどでブロックされている。

本機構は、クライアント PC に、Firewall 内部のアプリケーションサーバの画面をインターネット回線を通じて転送することを目的としている。

本シーケンス図の通信の流れに示す通り、クライアント PC は RCM-Web マシンのみと通信を行いつつ、Firewall 内部のアプリケーションサーバの画面データを受信することができる。

- (2) 画面伝送の性能チューニングを行い、実使用に耐えうる伝送速度を確保した。

平成 21 年度と同課題の性能評価において、画面伝送を実用的な応答速度で操作するためには、10Mbps あれば「ほぼ問題なく使用できる」ことが評価できている。また、5Mbps あれば、「やや難があるが、操作可能」であることが評価できている。

そのため、図 1 に示す RCM 上の通信路において、5Mbps 以上の通信速度を確保するため、性能チューニングを行い、表 1 の通信速度を得た。

(クライアント PC : HTTPS ⇔ HTTPS ⇔ SSH : アプリケーションサーバ)

表 1 HTTPS ベースの通信速度

設定 RTT(ms)	PLR 0% (Mbps)	PLR 2% (Mbps)	PLR 4% (Mbps)
1 未満	29.87	10.21	4.62
10	12.90	3.24	2.16
20	8.87	2.27	1.38
50	3.78	1.07	0.77
100	2.17	0.69	0.43
200	1.18	0.33	0.30

日本国内のインターネット回線では、一般的に、RTT ≤ 20ms, PLR 0%の環境を想定することができるため、表 1から、8.87 ~ 29.87Mbps の通信速度を、HTTPS ベースの画面伝送で確保できることが評価できた。

RTT(Round Trip Time)：信号を発信してから応答が返って来るまでの時間。
PLR(Packet Loss Rate)：パケットロス率。

すなわち、HTTPS ベースの画面伝送において、「やや難があるが、操作可能」または「ほぼ問題なく使用できる」伝送速度を確保できた。

4-1-3 達成状況及び今後の課題

- (1) 平成 22 年度は、RCM に組み込んだ HTTPS ベースの画面伝送において、リモートアプリケーションサーバの画面伝送を実施できる機構を組み込んだ。また、性能評価を行い、実用に耐えうる画面伝送速度を確保した。
- (2) 平成 23 年度は、複数の利用者が本機構を利用しているときの画面伝送性能を安定的かつ実用的な速度で伝送できるように、性能の向上と評価を実施する。

4-2 RCM システムの負荷分散、冗長機構の研究開発

4-2-1 研究開発内容

- (1) RCM-Web, Control, DB サーバの分散化を行い、ロードバランス機能により負荷を分散させる機構を開発し、基礎性能評価を実施した。
- (2) RCM-Web, Control, DB サーバの一部に障害が発生した場合は、他の RCM-Web サーバで処理が継続することを可能にする機構の開発・評価を行った。
- (3) ハードウェアで本機構を実現できることを、実機を用いて検証し、その基礎性能を評価した。

4-2-2 実施状況

- (1) 複数の RCM をロードバランサの配下に接続し、負荷分散機構を実現した。

実施内容 1

RCM システムに、稼働中の系列を一意に示す機構を開発し、ロードバランサを通じて、負荷を分散できることを評価した。その結果、正常に負荷分散がなされていることを確認し、ロードバランス機能による負荷分散機構の開発

を達成した。

表 2 評価に使用したロードバランサ

Hardware	Vender	Spec
IPCOM EX 1200 LB	Fujitsu Co.,Ltd.	FW:E10L51 NF0201 B03

また、RCM-DB の負荷分散は、データの整合性を確保するために、RCM-ParallelDB に支配下サーバとして、複数の RCM-DB サーバ(下図の DB1, DB2, DB3)を配置する。

(RCM-Web, RCM-CNT)は、WorkFlow を実行する際に、RCM-ParallelDB にトランザクション番号と呼ばれる一連の処理番号を渡すことがあり、系列 1 の(RCM-Web, RCM-CNT)と 系列 2 の(RCM-Web, RCM-CNT)が重複したトランザクション番号を発行しないように、系列ごとにユニークなトランザクション番号を発行する機構を実装した。

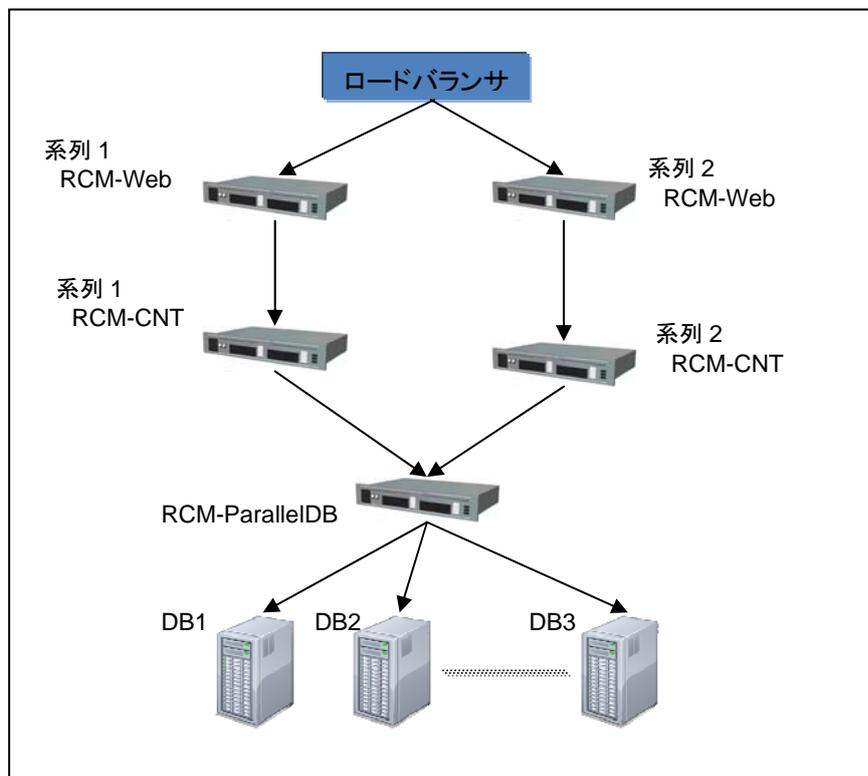


図 3 ロードバランサによる負荷分散

- (2) 複数の RCM をロードバランサの配下に接続し、障害が発生した RCM を切り離し、障害の発生していない RCM-Web サーバで処理を継続できることを評価した。その結果、障害の発生していない RCM-Web サーバで処理が継続できることを確認し、RCM-Web サーバ、RCM-Control サーバ、RCM-DB サーバの一部に障害が発生した場合に、他の RCM-Web サーバで処理を継続する機構を達成した。

実施内容 1

RCM システムに障害が発生した場合は、ロードバランサが障害を検知できるように、死活監視機構を RCM に開発した。これにより、障害が発生した RCM システムはロードバランサから、命令を転送されなくなり、他の、稼働している RCM-Web サーバで処理を継続することを評価した。

RCM システムは、系列ごとに、ロードバランサからの死活監視要求に対して、応答を返す。このとき、RCM-Web または RCM-CNT に障害があれば、死活監視要求に応答しない、または障害通知を返信することで、ロードバランサから切り離される。

- (例) 系列 1 の RCM-CNT に障害が発生した場合 RCM-Web は、ロードバランサからの死活監視要求に対して、障害通知を返信します。

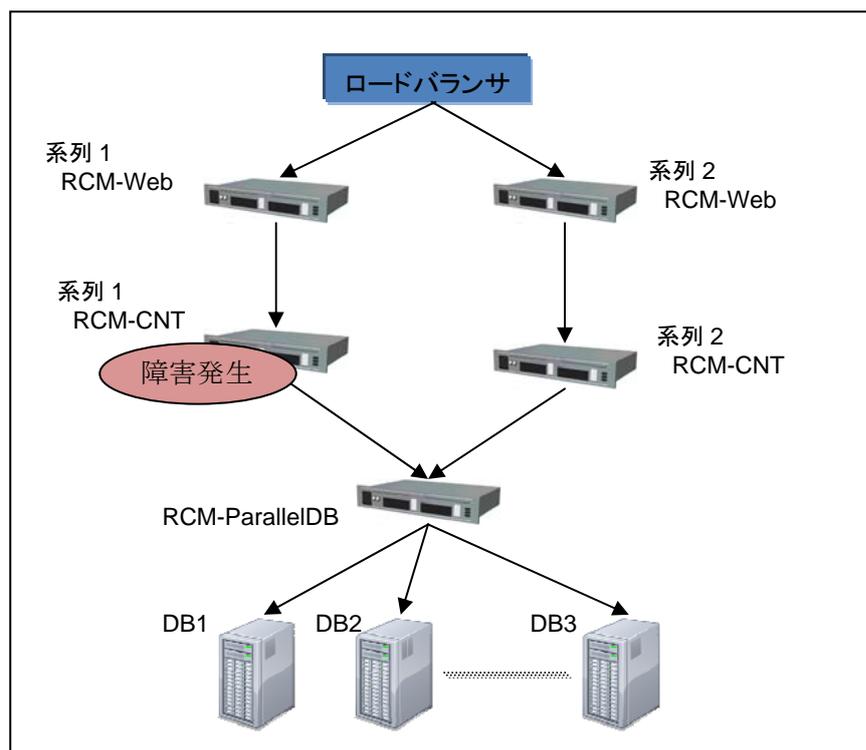


図 4 死活監視による障害通知

ロードバランサは、障害通知を受けた系列の RCM には、それ以降、要求を転送せず、系列 2 の RCM にのみ、要求を通知します。

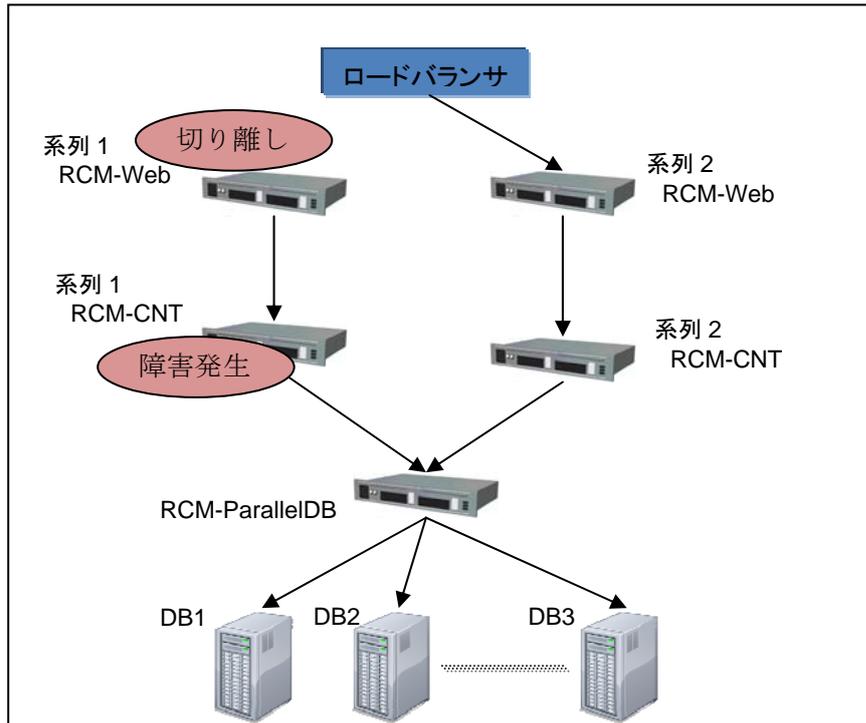


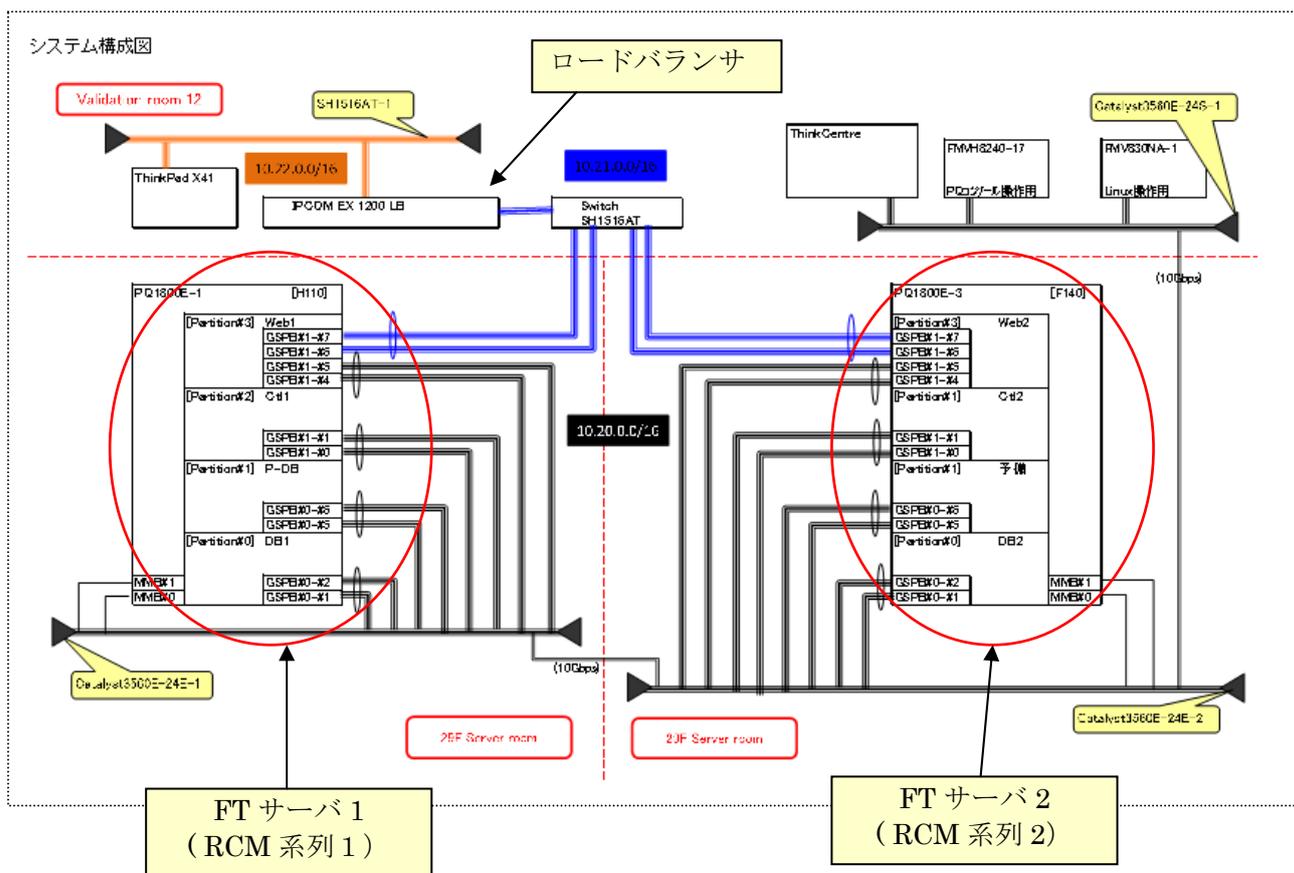
図 5 障害通知後のロードバランサの要求振り振り分け

(3) Fault Tolerant サーバ(FT サーバ)を用いて、ハードウェアによる冗長構成の動作確認を行い、評価を行った結果、ハードウェア障害時にも RCM が処理を継続できることを確認し、ハードウェアによる冗長構成の実現を達成した。

表 3 評価に使用した FT サーバ

Hardware	Vender	Spec
PrimeQuest1800E	Fujitsu Co.,Ltd.	(Xeon E7540 * 2,Memory 16GB) * 4SB

(1), (2), (3)について、下図の機器構成で評価を行った。



4-2-3 達成状況及び今後の課題

(1) 平成22年度は、RCM-Web, Control, DBサーバの分散を行い、ロードバランスにより、負荷を分散させる機構を開発し、基礎性能評価を実施した。また、RCMシステムの一部に障害が発生した場合に備え、冗長機構を開発し、他のRCM-Webサーバで処理を継続することを可能にする機構の開発・評価を行った。

(2) 平成23年度は、データ保存に関する負荷分散や冗長機構の開発を行う。具体的には、複数のファイルサーバに格納される実ファイルに関する負荷分散、冗長機構の開発と評価を行う。

4-3 データベースの高機密化(排他的記録)機構の研究開発

4-3-1 研究開発内容

- (1) 支配下サーバを動的に追加できる試作機能の基礎評価を実施した。
- (2) 任意のデータをよびメタデータを別ハードウェアに格納するためのDBリクエストを開発し、それ以外のデータとハードウェア的に分離した管理が可能となる機構の開発・評価を行った。

4-3-2 実施状況

- (1) 支配下サーバを動的に追加できる試作機能の基礎評価を実施した。

実施内容

支配下サーバは、図 6に示すように、RCM-並列 DB 制御サーバマシンにおいて、並列データベース群として管理されている。
本機構では、並列データベース群の追加が必要になった時点で、動的に追加する機構の評価を行い、実運用に対応できることを確認した。

例えば、図 6において、RCM-XDB04 は、当初は存在しないものとする。ProjectC用のデータ登録が増大する時点で、RCM-XDB04 を動的に追加し、データ登録の容量を増大させる運用が可能であることが評価できた。

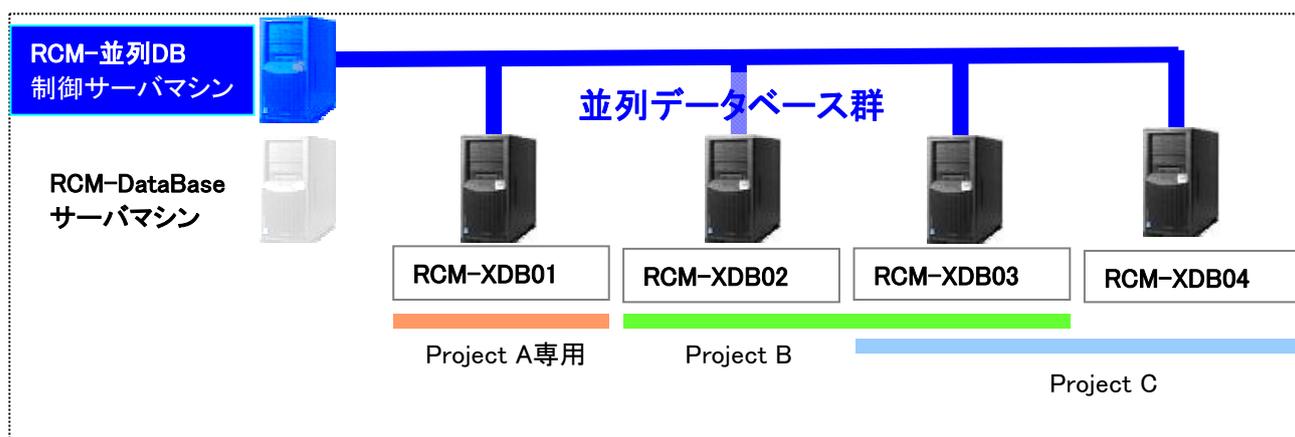


図 6 支配下サーバの動的追加機構

- (2) 任意のデータおよびメタデータを別ハードウェアに格納するための DB リクエストを開発し、データの種別に応じて、ハードウェア的に分離した管理が可能となる機構の開発・評価を行った。その結果、データ種別によりハードウェア的に分離した管理ができることを確認した。

実施内容

図 6を例に、評価内容について説明する。

- (1) RCM-並列 DB 制御サーバマシンに、データ種別=ProjectA のデータ登録を送信する。RCM-並列制御サーバマシンは、支配下サーバから ProjectA のデータ種別を格納する 「RCM-XDB01」を選択し、データ登録を行う。
- (2) RCM-並列 DB 制御サーバマシンに、データ種別=ProjectB のデータ登録を送信する。RCM-並列制御サーバマシンは、支配下サーバから ProjectB のデータ種別を格納する 「RCM-XDB02」または「RCM-XDB03」を選択し、負荷分散などを考慮してデータ登録を行う。

4-3-3 達成状況及び今後の課題

- (1) 平成 22 年度は、支配下サーバを動的に追加できる試作機構の基礎性能を評価した。また、任意のデータおよびメタデータを別ハードウェアに格納するための DB リクエストを開発し、評価を行った。
- (2) 平成 23 年度は、初期に設定した機密レベルを、一括して変更することでできる機構の開発を行う。

4-4 Workflow や高品位な UI を GUI で設定、変更を可能にする機構の研究開発

4-4-1 研究開発内容

- (1) XML-Workflow(入力系 UI、出力系 UI を含む)の GUI 画面での設定を可能とする機構の開発、評価を行った。
- (2) XML-Workflow 設定 GUI 画面では、次の機能を満たすことを可能にする機構の開発・評価を行った。
 - ・XML のタグ名入力は不要にする
 - ・ドロップダウン選択機構を有する
 - ・フォーマットを誘導、チェック機構を有する
 - ・job 間の相関性誘導機構を有する
- (3) 入力系 UI、出力系 UI 設定を、レイアウト画面で設定することを可能とする機構の開発・評価を行った。

4-4-2 実施状況

- (1) データ解析ウィザード GUI の開発を行い、XML-Workflow(入力系 UI、出力系 UI を含む)の GUI 画面での設定を可能とした。

実施内容

XML-Workflow の入力系 UI では、次ページに示す 図 7 の通り、各コンポーネント(処理単位)ごとに、XML-Workflow の実行に必要なパラメータを UI 上から入力できるようにした。

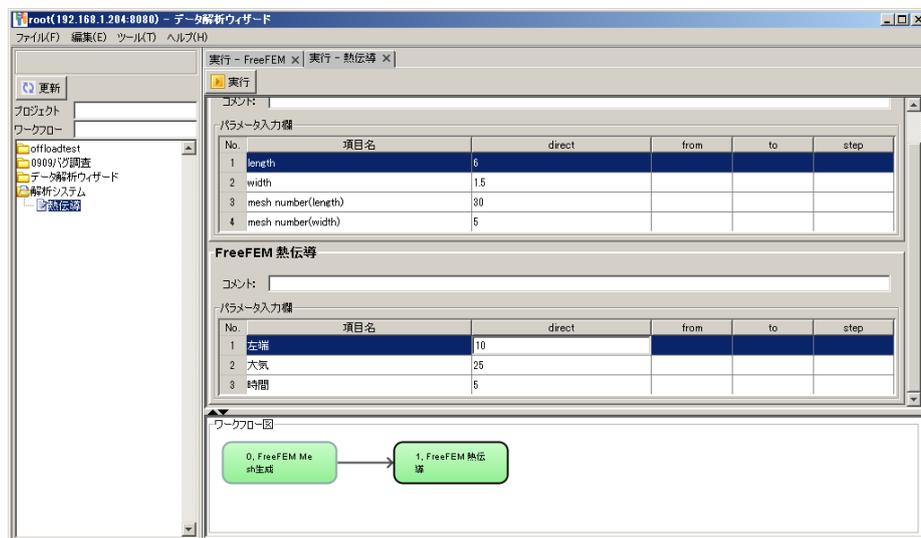


図 7 データ解析ウィザード GUI の入力系 UI での設定画面

また、出力系 UI では、登録したデータの表示にあたり、表示する項目を選択し、表示状態をカスタマイズできる機構の開発を行った。

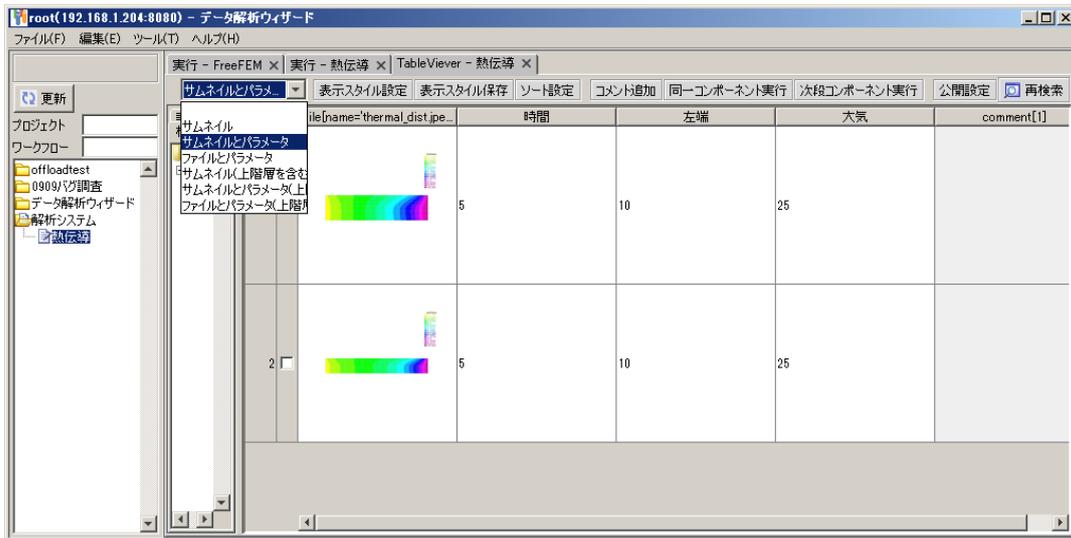


図 8 データ解析ウィザード GUI の出力系 UI での設定画面

(2) XML-Workflow 設定 GUI 画面での入力方法について開発・評価を行った。その結果、以下の実施内容を達成した。

実施内容

- XML のタグ名入力は不要にする
- ドロップダウン選択機構を有する
- フォーマットを誘導、チェック機構を有する
- job 間の相関性誘導機構を有する

XML-Workflow を GUI で作成する手順を説明する。

(step1) コンポーネントを登録する

データ解析ウィザード GUI で新規コンポーネント作成を実行します。

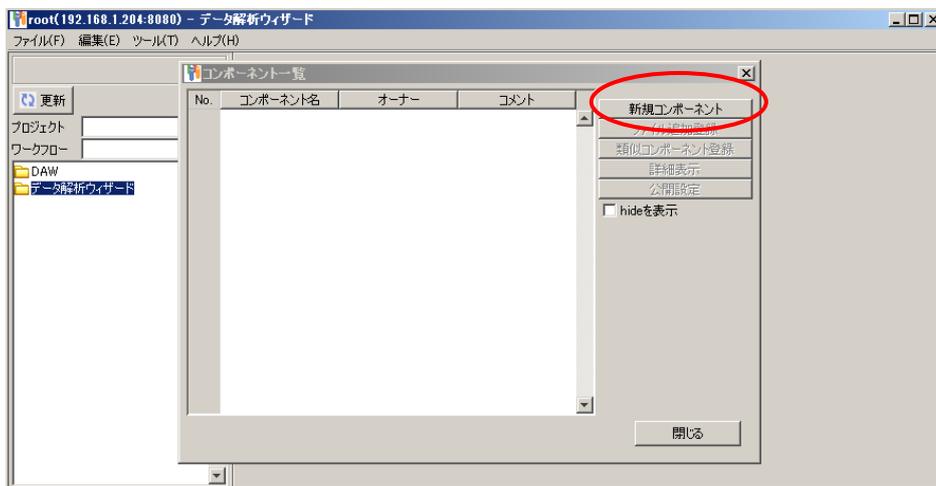


図 9 新規コンポーネント作成

コンポーネントを構成するファイルを、登録します。
登録するファイルは、CAE を実行する際に、手作業で実行するプログラムを指定します。

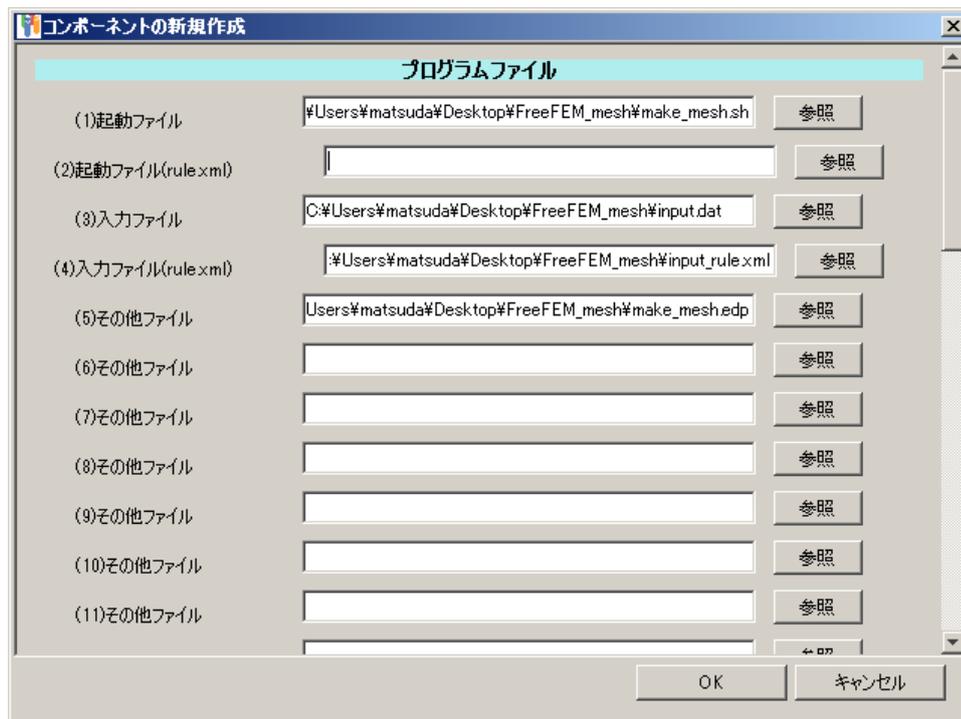


図 10 コンポーネントへのファイル登録

(step2) Workflow にコンポーネントを追加する

新規にワークフロー登録画面を開き、Workflow を作成します。

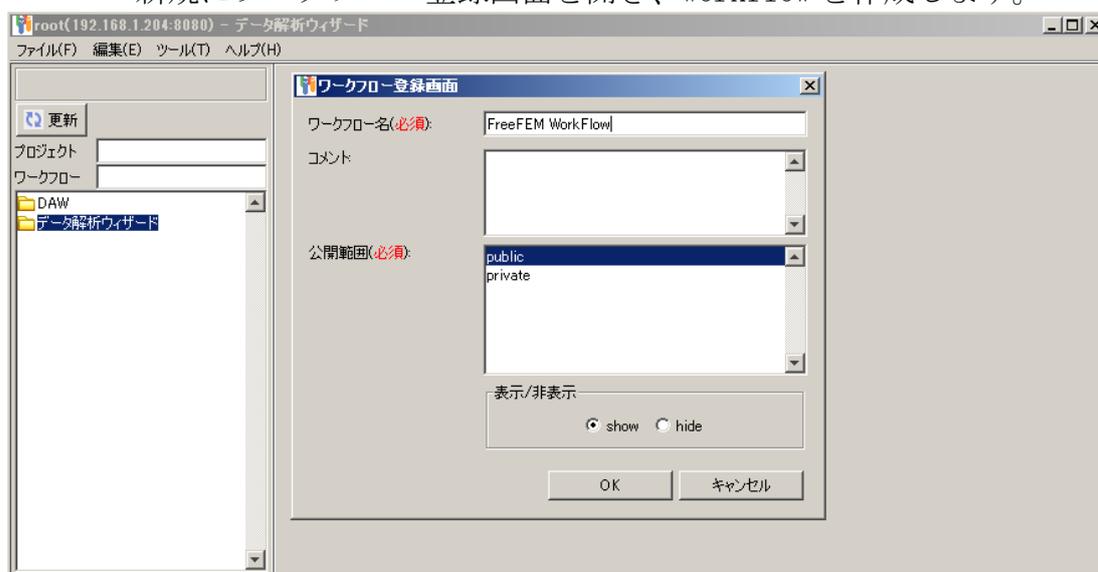


図 11 新規ワークフローの作成

作成した Workflow の編集画面を開き、作成済みのコンポーネントをマウスでドラック&ドロップして追加します。

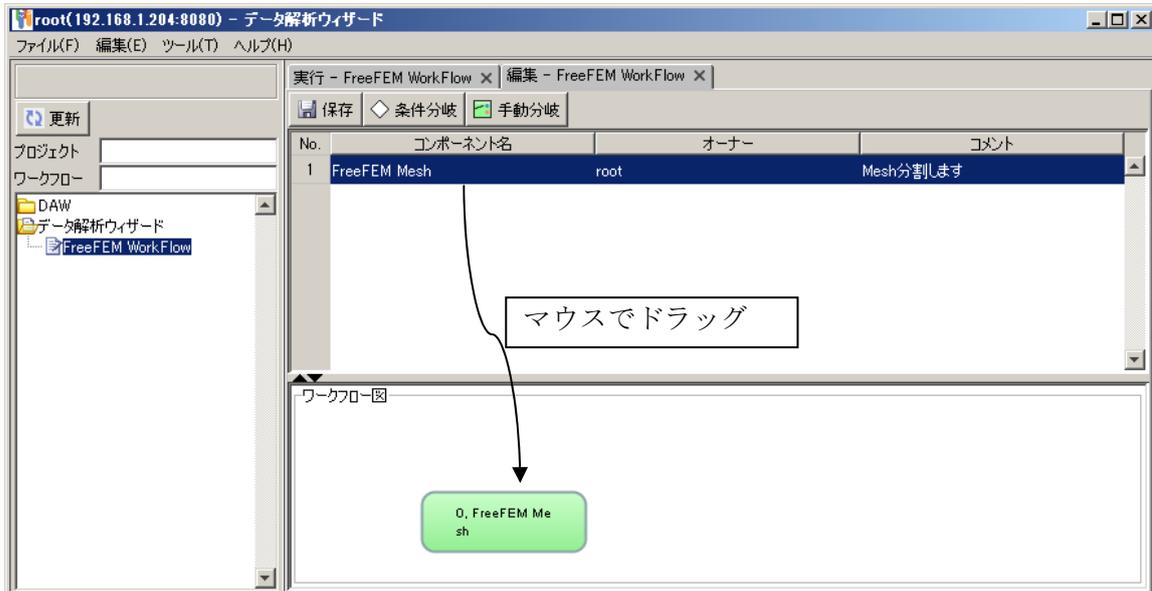


図 12 Workflow をマウス操作で作成する

(step3) Workflow の実行画面を開く

ここまでで、XML のタグ名入力を行わずに、Workflow を作成できました。また、コンポーネントの入力ファイルを解析して、入力欄を自動的に生成しています。

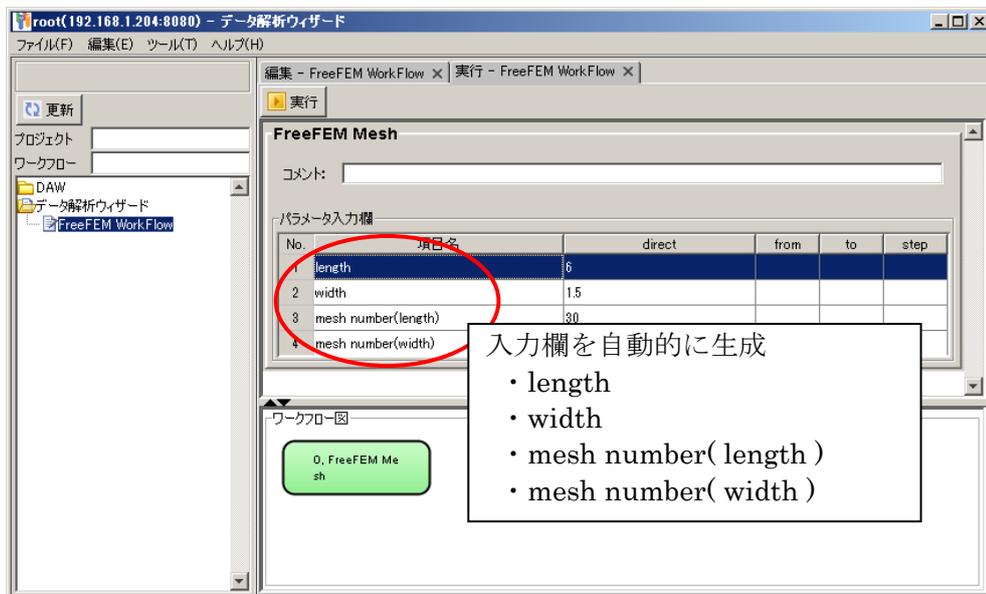


図 13 Workflow の実行画面

また、「フォーマットを誘導、チェック機構を有する」の例として、図 14に示すように、入力のフォーマットチェックを行い、入力誤りについてはエラー通知を行う機構の開発・評価を行った。その結果、入力誤り検知時にエラー通知がなされることを確認し、フォーマットの誘導、チェック機構の開発を達成した。

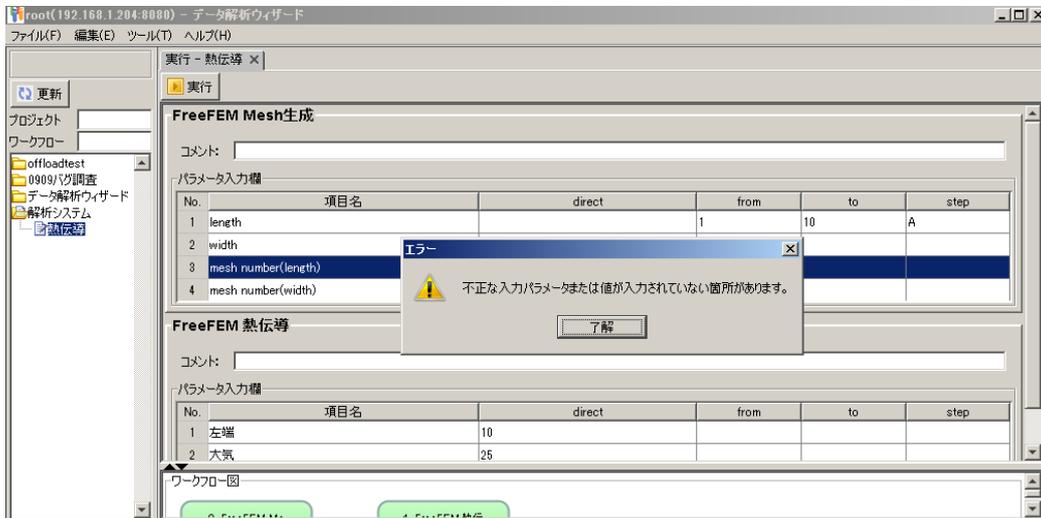


図 14 データ解析ウィザード GUI の「フォーマットを誘導、チェック機構」

また、「job 間の相関性誘導機構」では、図 15に示すように、job の前後関係を GUI 上から設定し、連続で複数の Job を指定した順序で実行する機構を開発し、評価を行った。その結果、相関性を定義した複数の job が指定した順序で実行されることを確認し、GUI による job 間の相関性入力の開発を達成した。

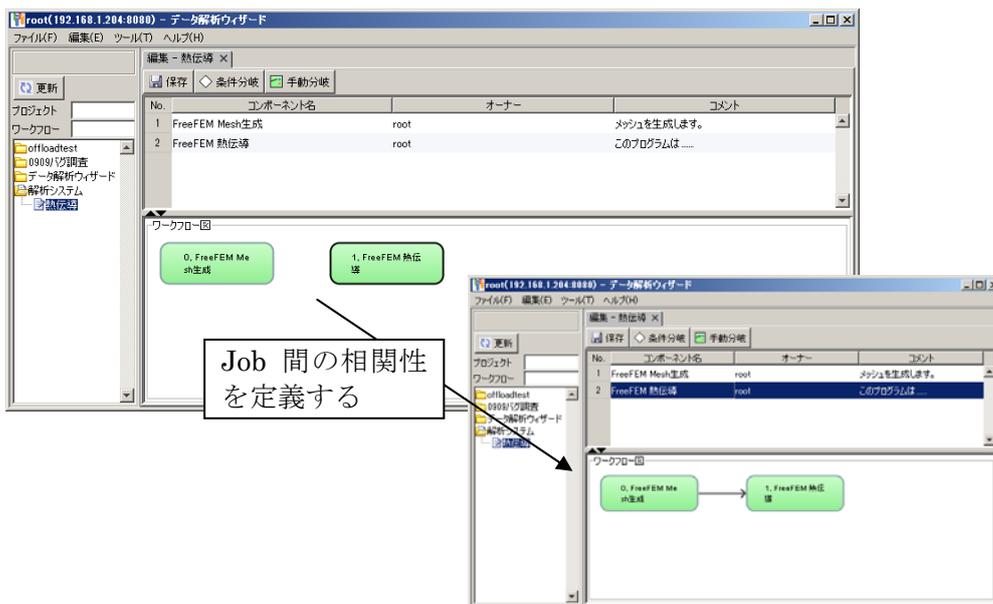


図 15 データ解析ウィザード GUI の「job 間の相関性誘導機構」

- (3) 入力系 UI、出力系 UI 設定を、レイアウト画面で設定することを可能とする機構の開発・評価を行った。その結果、入力系 UI では Job の連続実行をマウス操

作で設定できることを、出力系 UI では表示データの内容（画像ファイル、テキストファイル等）をマウス操作で切り替えられることを確認し、HomePage 作成ツールのようなレイアウト画面で入力系 UI、出力系 UI を作成する機構の開発を達成した。

実施内容

入力系 UI の設定では、レイアウト画面において、Job の連続実行の ON/OFF をマウス操作で設定して、実行する機構などの開発・評価を行った。その結果、レイアウト画面でマウス操作により Job の連続実行の ON/OFF を切り替えられることを確認した。

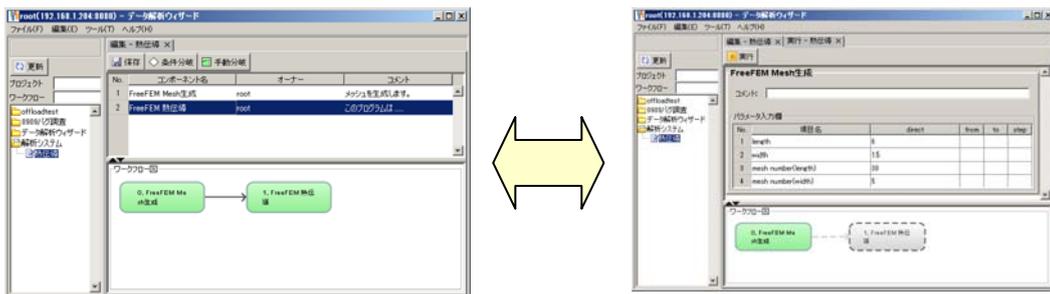


図 16 データ解析ウィザード GUI の 入力系 UI 設定機構

同様に 出力系 UI では、出力データの表示データ種別を、マウスで簡単に切り替えることができる機構などの開発・評価を行い、出力データの比較を、従来よりも直感的に実施することが可能となった。

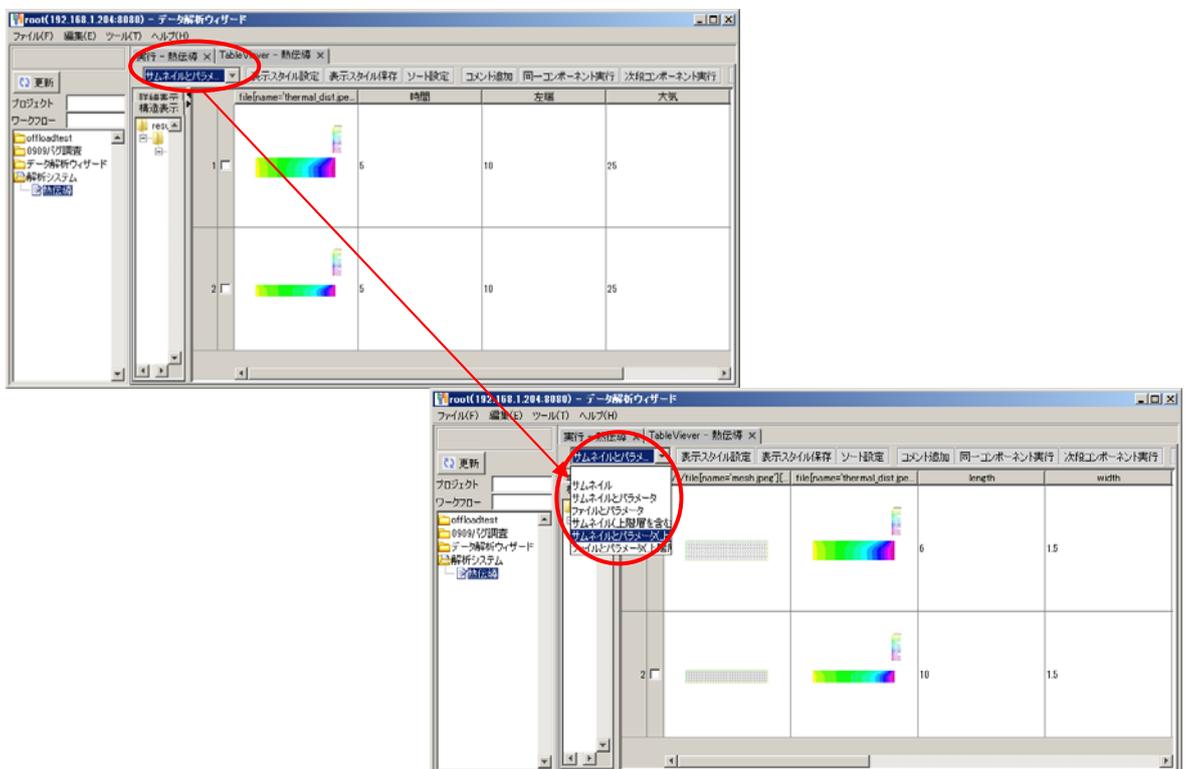


図 17 データ解析ウィザード GUI の出力系 UI 設計機構

4-4-3 達成状況及び今後の課題

平成 22 年度は、データ解析ウィザード GUI を開発し、入力系 UI, 出力系 UI を GUI 上から操作、設定できる機構の開発・評価を行った。

平成 23 年度は、データ解析ウィザード GUI の自動試験機構の開発およびテキストパターンマッチングツールである picker の組み込みを行う。

4-5 RCM システム間 (WebServer-WebServer) の連携機構の研究開発

4-5-1 研究開発内容

- (1) Workflow 内で異なる RCM システムで実行される job について、他方の RCM システムに処理を依頼し、その戻値を受け取ること、および、job 内で異なる RCM システム支配下のサーバを利用する場合、他方の RCM システムと連携して1つの job 機能を果たすことを可能とする機能の開発・評価を行った。

4-5-2 実施状況

- (1) 他方の RCM システムに処理を依頼し、その戻値を受け取る機構の開発・評価を行った。また、他方の RCM システムと連携して1つの job 機能を果たす機構の開発・評価を行った。

実施内容

一方の RCM システムから、他方の RCM システムに処理を依頼するための Job を新規に定義し、他方の RCM システムと通信を行い、処理結果を受領する機構の開発・評価を行った。その結果、連携した RCM システムが1つの job 機能として動作することを確認し、他方の RCM システムと連携して1つの job 機能を果たす機構の開発を達成した。

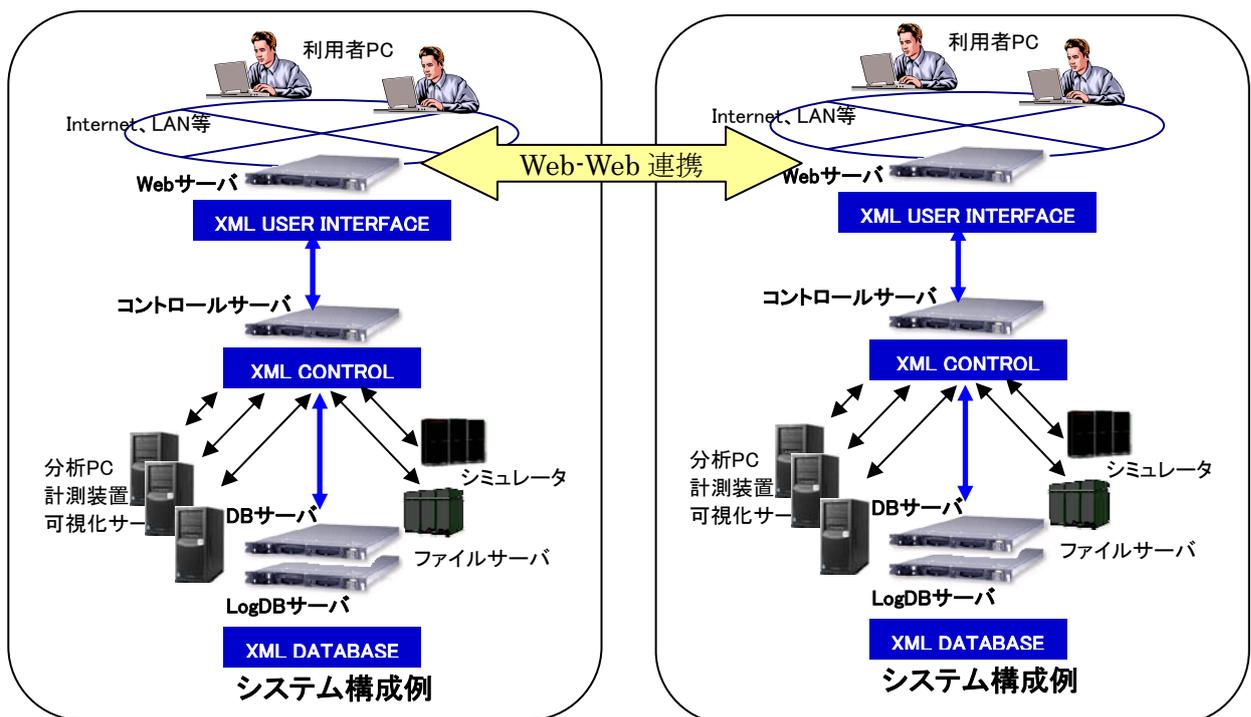


図 18 他方の RCM システムに処理を依頼し、その戻値を受け取る機構

他方の RCM システムへの管理画面を作成し、ここに、利用可能な RCM システムを登録しておく。

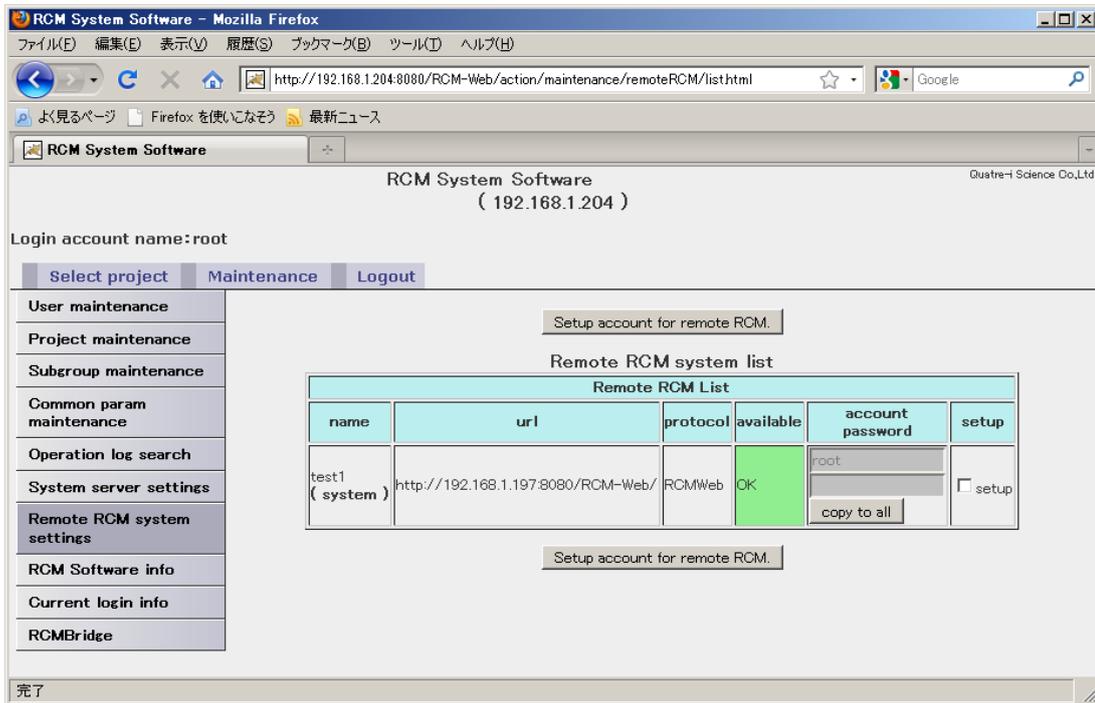


図 19 他方の RCM システムの管理画面

次に、他方の RCM システムに処理を依頼するには、新規に開発した ExecRemoteWorkFlow Job を実行する（次ページの 図 20、表 4 を参照）。本 Job は、他方の RCM での処理結果を受領するので、元々の RCM 内で、他方の RCM での処理結果を参照して、1つの job 機構を果たすことができる。

```

<job no="〇〇" jobClass="ExecRemoteWorkFlow">
  <remoteRCM>①</remoteRCM>
  <templateID>②</templateID>
  <parameter name="④">③</parameter>
  ...
  <label name="⑥" dataType="⑦">⑤</label>
  ...
  <wait>⑧</wait>
</job>

```

図 20 他方の RCM システムに処理を依頼する Job 定義

表 4 他方の RCM システムに処理を依頼する Job の設定内容

No	タグ名・属性名	値	説明
①	remoteRCM (必須)	remoteRCM 名	
②	templateID (必須)	remoteRCM 上の template 番号	
③	parameter (省略可) (複数記述可)	WorkFlow に渡す preset 定義値(任意)	WorkFlow の preset 値を初期化する 設定を記述します。

④	属性 name (必須)	WorkFlow 内の preset 名	
⑤	label (省略可) (複数記述可)	WorkFlow のラベルに 事前にセットするデータを 指定する文字列	WorkFlow のラベルに事前にセットする ファイルを指定します。本タグの値は、 dataType 属性に指定する値によって、 job 識 別名を記述する場合と、ラベル名を記述すべ き場合があります。
⑥	属性 name (必須)	WorkFlow 内のラベル名	
⑦	属性 dataType (必須)	stdout, jobXml, label のいずれか	受け渡すファイルを取得する方法を指定しま す。
⑧	wait (省略可)	true false のいずれか	false : WorkFlow の終了を待たない 省略時 または true : WorkFlow の終了を待つ

4-5-3 達成状況及び今後の課題

平成 22 年度は、RCM システムが、異なる RCM システムに処理を依頼する機構の開発を行った。

平成 23 年度は、データ解析ウィザード GUI 上で RCM システム間の連携を実現するための機構の開発と評価を行う。

4-6 既存（非 RCM）社内 R&D システムとの連携機構の研究開発

4-6-1 研究開発内容

- (1) 任意の通信プロトコルを受信し、応答するためのブリッジモジュール機構の開発・評価を行った。
- (2) ブリッジモジュールにおいて SOAP プロトコルを使う場合、ESB を使った通信をサポートできる機構の開発・評価を行った。
- (3) ブリッジモジュールの稼働・停止状態を RCM システムの、メンテナンスモードから監視、制御できる機構の開発・評価を行った。

4-6-2 実施状況

- (1) 任意の通信プロトコルを受信し、応答するためのブリッジモジュール機構の開発・評価を行った。

実施内容

既存システムの通信プロトコルは、規格化されていないものが多く、独自プロトコルで実装されていることが一般的である。

このため、RCM では、任意の TCP/IP 通信を待ち受けし、通信を受け付けると RCM の WorkFlow を起動して、既存システムと RCM 間での通信方法を確立した。

- (2) 通信プロトコルの中でも、規格化されている SOAP プロトコルに対応し、ESB を使った通信をサポートできる機構の開発と評価を行った。

実施内容

RCM システム内に、SOAP サービスを行うブリッジモジュールを登録し、RCMブリッジ管理画面から起動する(図 21)。

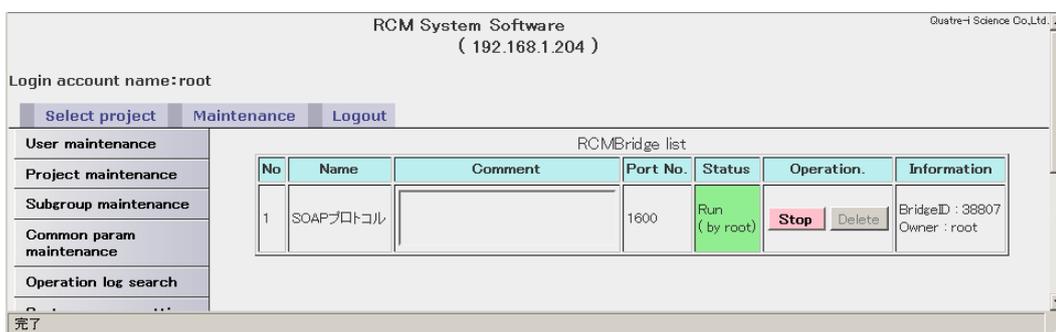


図 21 RCMブリッジ 管理画面

SOAP サービスが起動されていることは、wsdl 情報を表示して確認することができる。

wsdl(Web Services Description Language) : Web サービスを記述する為の、XML をベースとした言語仕様。

```
-<wsdl:definitions targetNamespace="http://RCM">
  <wsdl:documentation>RCMBridge</wsdl:documentation>
  -<wsdl:types>
    -<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://util.java/xsd">
      -<xs:complexType name="Map">
        -<xs:sequence>
          <xs:element minOccurs="0" name="empty" type="xs:boolean"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
    -<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://RCM">
      <xs:import namespace="http://util.java/xsd"/>
      <xs:import namespace="http://io.java/xsd"/>
      -<xs:element name="sendCommand">
        -<xs:complexType>
          -<xs:sequence>
            <xs:element minOccurs="0" name="aArgMap" nillable="true" type="ax22:Map"/>
            <xs:element minOccurs="0" name="aFileMap" nillable="true" type="ax22:Map"/>
            <xs:element minOccurs="0" name="aOut" nillable="true" type="ax24:OutputStream"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      -<xs:element name="getArgsResponse">
        -<xs:complexType>
          -<xs:sequence>
```

図 22 RCMブリッジ SOAP モジュールの wsdl

- (3) ブリッジモジュールの稼働・停止状態を RCM システムの、メンテナンスモードから監視、制御できる機構の開発・評価を行った。

実施内容

ブリッジモジュールを RCM 上の管理画面から監視、制御する機構を開発し、RCM 上から、ブリッジモジュールの起動、稼働状態の監視、停止を GUI 画面で操作、表示できることを確認した。

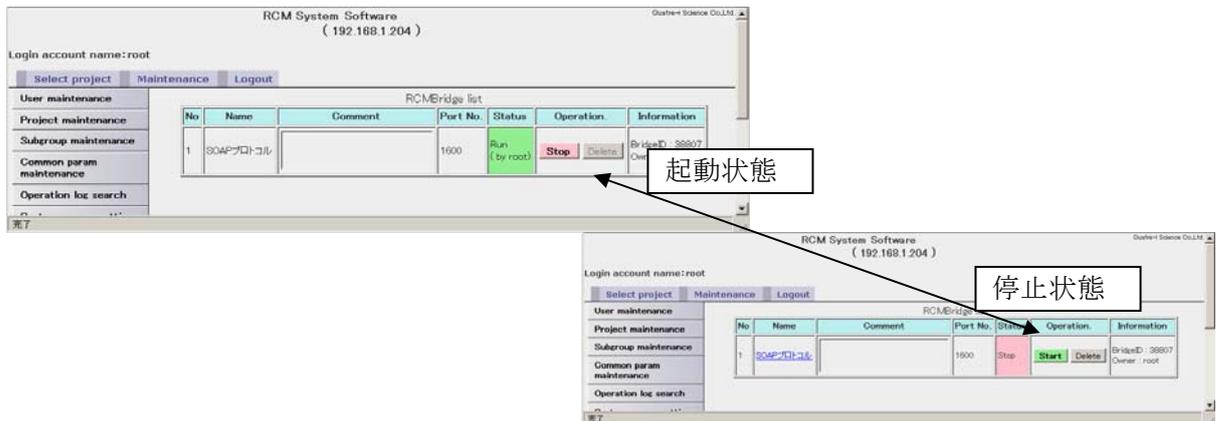


図 23 RCM ブリッジモジュールの管理画面

4-6-3 達成状況及び今後の課題

平成 22 年度は、任意の通信プロトコルで既存システムとの連携を開発した。
平成 23 年度は、システム連携の実証を行う。

具体的には、LDAP 等の実際に利用されている既存システムとの連携を実地で検証する。

4-7 実証システムの構築と運用の研究開発

4-7-1 研究開発内容

- (1) 2 つ以上の RCM システムを別ネットワークで構築、運用し、サブテーマ(1)～(6)の機能が正しく動作するかを 1 ヶ月以上検証した。
- (2) サブテーマ(1)～(6)の機能の性能を評価し、性能、機能面で利用上問題がある部分に関してボトルネックポイントを同定し、改善プランを策定した。
- (3) 上記(1)の検証時に運用モデルを明確化し、運用マニュアル、運用における留意点をドキュメントにまとめた。また、そのマニュアルに従って、運用を実際に行って、問題点がないかを確認した。

4-7-2 実施状況

- (1) 2つ以上のRCMシステムを別ネットワークで構築、運用し、サブテーマ(1)～(6)の機能が正しく動作するかを1ヶ月以上検証した。

実施内容 1

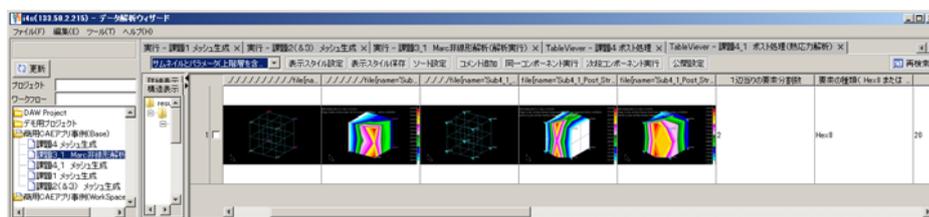
6箇所の実証サイトを立ち上げ、RCMシステムを6箇月にわたり連続運用を行った。

実施内容 2

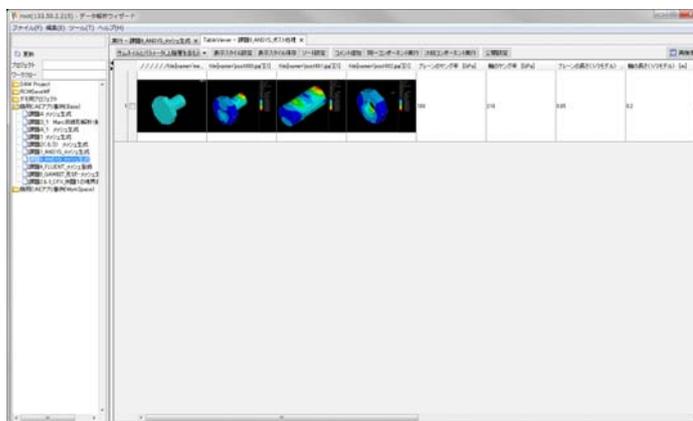
各実証サイトにおいて、各種のアプリケーションとRCMとの連携評価を行った。

アプリケーションとの連携評価の例(商用アプリ)

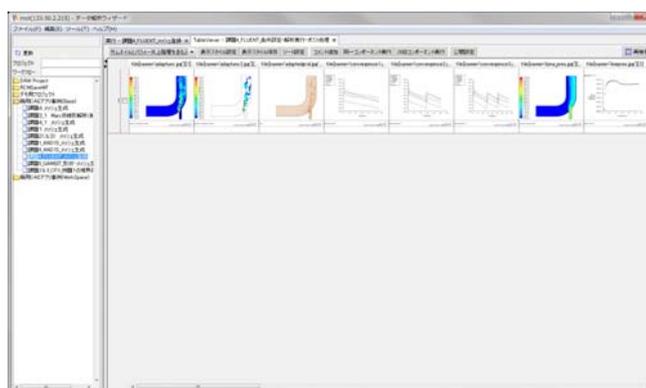
(例1) MSC.NASTRAN



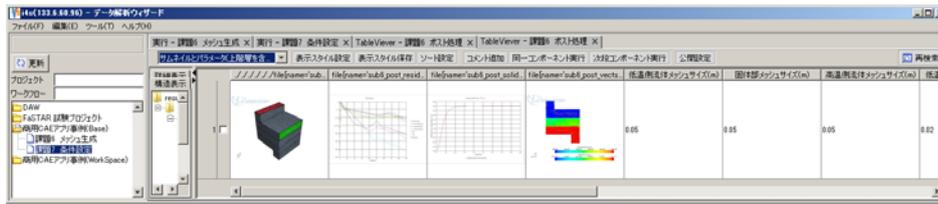
(例2) ANSYS



(例3) FLUENT



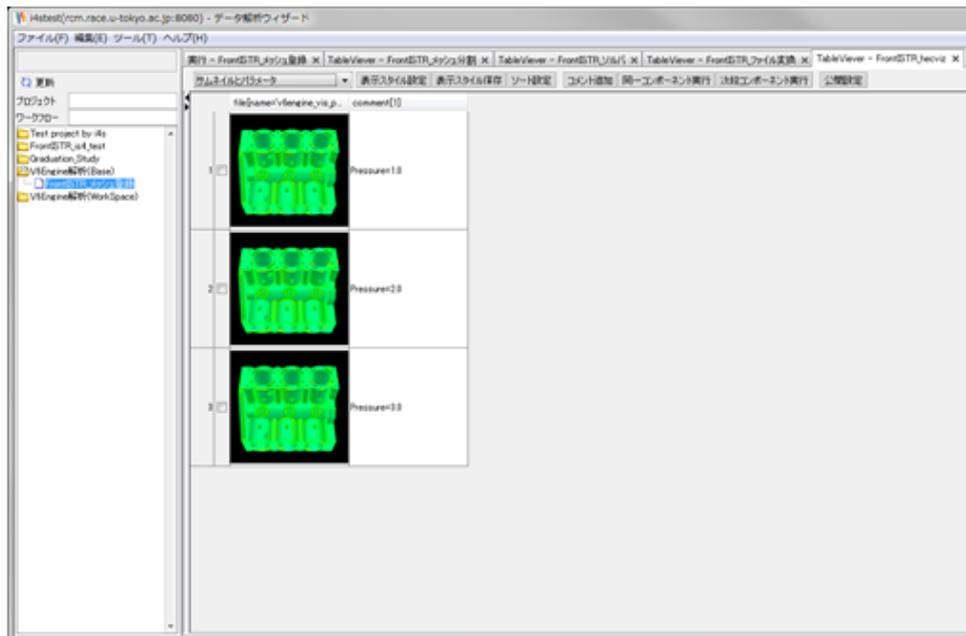
(例4) Start-CCM



アプリケーションとの連携評価の例(オープンソース)

(例1) FrontISTR

「5.FrontISTR_hecviz」の結果検索



実施内容 3

各実証サイト間を連携して、CAE アプリの統合クラウド化の運用評価を行った。

(実証サイト連携の例)

1. 実証サイト A でソルバ計算を行った結果を実証サイト B に転送し、
実証サイト B で可視化を行う。
RCM 利用者は、RCM を通じて、可視化結果の画像を取得することができる。
 2. 実証サイト C で解析計算を行った結果を、実証サイト B に転送し、
実証サイト B で可視化を行う。
RCM 利用者は、RCM を通じて、可視化結果の画像を取得することができる。
- (2) サブテーマ (1) ~ (6) の機能の性能を評価し、性能、機能面で利用上問題がある部分に関してボトルネックポイントを同定し、改善プランを策定した。

実施内容

1. 長期間、連続運用すると、RCM から計算サーバへの SSH 接続を自動的に切断してしまうことがあり、接続状態を維持する改善を行った。
 2. 長期間の運用において、RCM システム自体のバックアップ方式を体系化するため、RCM システムに定期バックアップ機能を追加した。
- (3) 上記 (1) の検証時に運用モデルを明確化し、運用マニュアル、運用における留意点をドキュメントにまとめた。また、そのマニュアルに従って、運用を実際に行って、問題点がないかを確認した。

実施内容

以下のドキュメントを整備した。また、本ドキュメントを用いて、弊社のスタッフ以外でも、実証サイトに RCM をインストールし、運用を行えることを検証した。

- ・「インフラ環境ヒアリングシート」
- ・「RCM 設置ヒアリングシート」
- ・「RCM サーバ設定・動作確認シート」

各ドキュメントのサンプル

「インフラヒアリングシート」の例

インフラヒアリングシートでは、RCM サーバを設置する実証サイトのネットワーク構成などの情報で、必要な情報を事前にヒアリングする。

2.1 ネットワーク通信の制限事項に関する確認事項

ネットワーク構成図に記述したネットワークセグメントについて、セグメント間の通信制限についてご記入ください。

表 1 ネットワーク通信制限情報記述例（ネットワークセグメント間）

ネットワークアドレス		L4 プロトコル	ポート番号	L7 プロトコル	通信可否	特記事項
FROM	TO					
192.168.1.0/24	172.146.3.0/24	TCP	80	HTTP	可	
		TCP	443	HTTPS	可	
		TCP	8080	HTTP	否	
		TCP	6000-6999	X11	可	
		TCP	22	SSH	可	
		TCP	21	FTP	可	
		TCP	5900-5999	RDP(VNC)	可	
		TCP	5500-5599	RDP(VNC)	可	
192.168.1.0/24	202.171.156.200/28	全て許可				
192.168.1.0/24	172.10.5.0/24	TCP	587	SMTP	可	
		上記以外全て拒否				

表 2 ネットワーク通信制限情報記述例（ネットワークセグメント～特定ホスト間）

ネットワークアドレス	ホストアドレス	L4 プロトコル	ポート番号	L7 プロトコル	通信可否	特記事項
FROM	TO					
192.168.1.0/24	172.146.3.15	TCP	80	HTTP	可	
		TCP	443	HTTPS	可	
		TCP	8080	HTTP	否	
		TCP	6000-6999	X11	可	
		TCP	22	SSH	可	
		TCP	20, 21	FTP	可	
		TCP	5900-5999	RDP(VNC)	可	
		TCP	5500-5599	RDP(VNC)	否	
上記以外は全て拒否						

図 24 インフラヒアリングシートの例

「RCM 設置ヒアリングシート」の例

RCM 設置ヒアリングシートは、RCM サーバの物理的なマシン情報などを実証サイトの RCM 管理者に記入してもらうシートである。

RCM の設置に、何か障害がある場合、本シート of 情報を参照することで、問題の切り分けが容易になる。

■3 RCM サーバ構成情報

..

RCM サーバ設置のためのサーバH/W情報と設定内容について、下表にご記入をお願いします。

..

■3.1 RCM-Front サーバ

..



表 1 RCM-Front サーバ基本情報

ホスト名	
MAC アドレス(ライセンス発行用)	
IP アドレス	
ネットマスク	
デフォルトGW	
DNS	
NTP	
HTTP Proxy (もしあれば)	
コンピュータ管理者の部門	
コンピュータ管理者の名称	
コンピュータ管理者のE-mail	
製品名	
サイズ	
電源容量(定格)	
重量	

OS と Version	CentOS x64 5.9	
CPU/Memory/HDD		
パーティション情報	パーティション	サイズ (GB)
通信可能ポート IN - カンマ区切りで列挙		
通信可能ポート OUT - カンマ区切りで列挙		
ログインシェル		
[確認コマンド echo \$SHELL]		
ホームディレクトリのパス		

図 25 RCM 設置ヒアリングシートの例

「RCM サーバ設定・動作確認シート」の例

RCM サーバをインストールし、実際に動作しているかをチェックする。

実証サイトの RCM 管理者は、本シートの動作テストを一通り実施することで、RCM を正しくインストールできているかを確認することができる。

表 23 IP アドレス変更後の接続試験状況

論理サーバタイプ	論理サーバ名	正常時 接続結果	結果 (合格/不合格)	確認日(確認者)	補足(表示されたメッセージなど)
RCMDBServer	DB1	OK			
RCMFileServer	file1	OK			
RCMBackupServer	backup1	OK			
RCMLogDBServer		N/A			
RCMLogFileServer		N/A			
RCM-LogBackupServer		N/A			
DataServer	dataIn	OK			
SimulationServer	sim1	OK			
AnalyzeServer	ana1	OK			
XXXServer	test1	NG			
MailServer	smtp	OK			
	pop3				
RCMSaveServer	save1	OK			
	save2	OK			

11.3 RCM サーバのネットワーク設定

サーバの IP アドレス、ネットマスク、DNS サーバなどのネットワーク設定を、「インフラヒアリングシート」と「RCM 設置ヒアリングシート」に、実際の使用環境で使用するものに変更します。

表 24 出荷前サーバ設定

RCM サーバマ シン	ホス ト名	IP (Netmask)	Default GW	DNS	SELinux	FireWall		ntp	特記事項
						設 定	許可サービス 又は ポート		
RCM-Front	abcd	xxx.xxx.xxx.xxx (xxx.xxx.xxx.xxx)	xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx	disable	有 効	SSH HTTP	xxx.xxx.xxx.xxx	
RCM-DB	efgh	xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx	xxx.xxx.xxx.xxx	disable	有 効	SSH	xxx.xxx.xxx.xxx	

図 26 RCM サーバ設定・動作確認シートの例

4-7-3 達成状況及び今後の課題

平成 22 年度は、実証サイト群の構築を行い、長期運用の評価を行った。その結果、構築した実証サイト群の連携が正常に機能し、長期運用の場合でも、性能をおとすことなく動作することを確認し、2 つ以上の RCM システムを別ネットワークで構築・運用する検証を達成した。平成 23 年度は、下記の開発、評価を行う。

- (1) 昨年度の実証システムの導入で明らかになったシステム導入マニュアル、システム運用マニュアルの問題点、改善が必要な点を加味したマニュアル改訂を行う。
- (2) 作成したシステム導入マニュアルおよびシステム運用マニュアルに従い、今年度も新規に実証サイトを構築することでマニュアルの評価を行う。
- (3) 実証サイトの導入、運用を支援するツールを開発し、より効率的な構築、運用を実現する。

4-8 総括

本研究開発は、R&D 系業務をシステム化した現行の RCM システムを拡張し、PaaS 化するものである。研究開発期間は平成 21 年 11 月から平成 23 年 10 月までの 2 年間で、今回は平成 22 年度の 12 ヶ月間を対象とする成果報告である。

平成22年度は、次のサブテーマの研究開発を実施した。

ア HTTPS ベースでリモートアプリケーションサーバの画面を高速に伝送する機構の研究開発

(ア) RCM に、HTTPS ベースによる、リモートアプリケーションサーバの画面伝送機構を開発した。これにより、HTTPS 通信路上で、クライアントからリモートアプリケーションサーバのインタラクティブアプリケーションの操作を行うことが可能となった。

(イ) HTTPS ベースの画面伝送のインタラクティブ性能のチューニングを行い、実用上問題のない速度を確保した。具体的には、SSH 直接接続時の速度に比べ、1/5 以内の FPS を確保した。

イ RCM システムの負荷分散、冗長機構の研究開発

(ア) RCM-Web、Control、DB サーバの分散化を行い、ロードバランス機能により負荷を分散させる機構を開発し、基礎性能評価を実施した。

(イ) RCM-Web、Control、DB サーバの一部に障害が発生した場合は、他の RCM-Web サーバで処理が継続することを可能にする機構の開発・評価を行った。

(ウ) ハードウェアで本機構を実現できることを、実機を用いて検証し、その基礎性能を評価した。その結果、本機構が実現できており、正常に動作することを確認し、ハードウェアでの負荷分散、冗長機構の開発を達成した。

ウ データベースの高機密化（排他的記録）機構の研究開発

(ア) 支配下サーバを動的に追加できる試作機構の基礎性能を評価した。

(イ) 任意のデータおよびメタデータを別ハードウェアに格納するための DB リクエストを開発し、それ以外のデータとハードウェア的に分離した管理が可能となる機構の開発・評価を行った。その結果、任意のデータ、メタデータとそれ以外のデータとをハードウェア的に分離した管理が可能であることを確認した。

エ Workflow や高品位な UI を GUI で設定、変更を可能にする機構の研究開発

(ア) XML-Workflow（入力系 UI、出力系 UI を含む）の GUI 画面での設定を可能にする機構の開発・評価を行った。その結果、XML-Workflow の GUI 画面での設定が可能であることを確認した。

(イ) XML-Workflow 設定 GUI 画面では、次の機能を満たすことを可能にする機構の開発・評価を行った。

- ・XML のタグ名入力は不要にする
- ・ドロップダウン選択機構を有する
- ・フォーマットを誘導、チェック機構を有する
- ・job 間の相関性誘導機構を有する

(ウ) 入力系 UI、出力系 UI 設定を、レイアウト画面で設定することを可能とする機構の開発・評価を行った。その結果、入力系 UI、出力系 UI 設定をレイア

ウト画面で設定できることを確認した。

オ RCM システム間 (WebServer-WebServer) の連携機構の研究開発

- (ア) Workflow内で異なるRCMシステムで実行されるjobについて、他方のRCMシステムに処理を依頼し、その戻値を受け取ること、およびjob内で異なるRCMシステム支配下のサーバを利用する場合、他方のRCMシステムと連携して1つのjob機能を果たすことを可能とする機構の開発・評価を行った。その結果、他方のRCMシステムと連携して1つのjobが動作することを確認した。

カ 既存 (非 RCM) 社内 R&D システムとの連携機構の研究開発

- (イ) 任意の通信プロトコルを受信し、応答するためのブリッジモジュール機構を開発・評価を行った。その結果、任意の通信プロトコルを受信し、応答することを確認した。
- (ロ) ブリッジモジュールにおいて SOAP プロトコルを使う場合、ESB を使った通信をサポートできる機構の開発・評価を行った。その結果、ブリッジモジュールで SOAP プロトコルを使用する場合に ESB を使った通信をサポートできることを確認した。
- (ハ) ブリッジモジュールの稼働・停止状態を RCM システムのメンテナンスモードから監視、制御できる機構の開発・評価を行った。その結果、ブリッジモジュールの稼働・停止状態を RCM システムのメンテナンスモードから監視、制御できることを確認した。

キ 実証システムの構築と運用の研究開発

- (ア) 4つの実証サイトで、RCMを構築し、4ヶ月以上、運用し、RCMの動作評価を行った。その結果、実証サイトが連携して正常に動作することを確認し、2つ以上のRCMシステムを別ネットワークで構築・運用する検証を達成した。
- (イ) 長期間、多人数で運用することによる改善ポイントを検討し、改善プランを策定した。
- (ロ) RCMを導入するためのヒアリングシートや運用マニュアルを整備した。

研究開発の最終年度となる平成23年度には、すべての機能、性能を実証システムおよび社内システムにおいて確認・改善を行い、運用を始めるために必要な総合的な性能、品質に目途がつくところまで昇華させる予定である。

5 参考資料

5-1 研究発表・講演等一覧

ア その他資料

- ・発表者名(所属) : 上島豊(株式会社キャトルアイ・サイエンス)
- ・発表題名 : CAE クラウド基盤技術
- ・発表会名 : NICT 民間基盤技術研究促進制度
ベンチャー支援制度 成果発表会
- ・発表年月日 : 平成 22 年 12 月 9 日

5-2 産業財産権

該当なし