

XMLデータベースを用いた実験・シミュレーション統合Webシステム

上島 豊（株式会社キャトルアイ・サイエンス）,

西原 功修（大阪大学レーザーエネルギー学研究センター, 株式会社キャトルアイ・サイエンス）

Integrated Web system of experiment & simulation using XML database

Yutaka Ueshima(Quatre-i Science Co., Ltd.) and

Katsunobu Nishihara (Institute of Laser Engineering, Osaka University and Quatre-i Science Co. Ltd.)

Abstract

Recently a lot of data are being produced because it has become easy to digitize and create data after the IT revolution. In addition, procedure of experiment and simulation and analysis methods of their data are getting more complicated. These facts make researchers more difficult to reach conclusions and reproduce same results. In the other words, efficiency and accuracy of the R&D processing, we call it R&D Chain, are getting lower. Especially the forefront fields such as space science, bio, nano, medical, nuclear engineering, and accelerator physics consider this problem seriously. In these fields, the cycle of changing methods is shorter; the number of output parameters is larger. This means that adopting fixed control system and solid relational database are not suitable. We developed a basic middleware, named "R&D Chain Management System Software (RCM)", to give appropriate solution of the problem. We report on an integrated system of experiment and simulation, as an example of better workflow management with application of RCM.

1. はじめに

近年、高次元・高時空間分解計測器や高性能計算機、ネットワークの普及、高度可視化処理の増大により、大学・公的研究所のみならず民間においても多種多様な実験計測や数値シミュレーションおよびそれらの可視化解析が行われるようになってきた。

結果として出力されるデータが多種多様となり、データ分析のためのデータ組み合わせや解析手法のパターンも膨大になってきている。その結果、可視化解析画像や「その結論がなぜ導き出されたか？」の理由などの点で研究の再現性が保証し難くなってきた。研究の品質管理(QC:Quality Control)、品質保証(QA:Quality Assurance)の不確実性が高まってきているとも言え換えることができるだろう。

一方、公的研究機関の独立行政法人化やインターネットの普及により、外部機関との共同研究など、機関外部との連携が活発化してきている。これら外部機関との連携において、ITは、申請、契約などの事務手続き（電子申請）、研究者同士のコミュニケーション（電子メール）の簡便化をもたらしている。

しかしながら、研究データの管理や共有、公開の仕組みは、ITが普及する前より悪化している。PCの普及により、個人個人が独立で進める研究スタイルが進んだことが原因である。研究者一人一人の短期的視点では、PCは研究者の自由度を最大限に生かせ、研究に集中できる環境をもたらした。しかし、研究組織や中長期的観点、つまり、研究データの管理や共有、公開といった観点からは、旧来の中央集約的な仕組みの方が有利であった。

もちろん、現在のように大量のデータが発生する研究環境においては、旧来の方法でデータを管理していくことは、原理的に不可能で新たな方法の模索が必要である。

我々は、“R&Dの流れをマネジメント”し、これらの課題を解決するための基盤ミドルウェア（R&D Chain Management System Software）を開発した。本論文では、“R&Dの流れをマネジメント”するために必要な機能について明らかにし、それらを満たす基盤ミドルウェアの適用例に関して報告する。

2. R&Dマネジメントに必要な技術コンポーネント

様々な非定型・非定常データ・情報が錯綜する研究開発現場において、“R&Dの流れをマネジメントする”ためには、何が必要であろうか？

研究データの管理や共有、公開を行うためには、研究にQC、QA的観点が無ければならない。QC/QAにおける最低限の条件は、研究が再現可能であることであり、トレーサビリティ性を持つことである。以下、研究・開発系業務の再現性、トレーサビリティを確保するために、必要な技術コンポーネントを説明していく。

1) XMLデータベース

生産系や事務系の業務においては、業務手続のマニュアル化およびそれを支える各種情報のデータベース化を行うことでQC/QAを維持している。それを制度面から認定するのがISOであり、技術面から支援するのがERP(Enterprise Resource Planning)、SCM(Supply Chain Management)やグループウェアといったITシステムであ

る。その根幹に現在のデファクトスタンダードであるリレーショナルデータベース（RDB）が存在する。

研究・開発系の業務においては、業務系と比べ業務手続き＝プロトコルやデータフォーマットの変更が頻繁であり、プロトコルを見出すこと自体が仕事の場合さえある。一方、RDBは、データ構造をあらかじめ決定しておかなければならないので、研究の定型部分のみしか扱えない。つまり、RDBは、変更が多い研究・開発系業務の再現性、トレーサビリティを確保する目的での活用は、困難ということになる。実際、研究を包括的に管理するデータベースが運用されていない一番の理由は、RDBのこの特性にあるといっても過言ではない。

テーブル設計、正規化が必要なRDBではなく、データ構造を途中からも変更可能なXMLデータベース（XML-DB）を研究・開発系業務のコアに位置づけることで、上記課題をクリアできる可能性がある。

RDBデータベース構造

実験番号	初期値1	初期値2	結果File	結果jpg
1	3	0.2	0.3.02.dat	0.3.02.jpg
2	7	0.1	2.7.01.dat	2.7.01.jpg
3	2	0.2	5.2.02.dat	5.2.02.jpg
.....				
10	4	0.3	7.4.03.dat	7.4.03.jpg

初期値3を追加し、値を0.4にする

データベース再設計

XMLデータベース構造

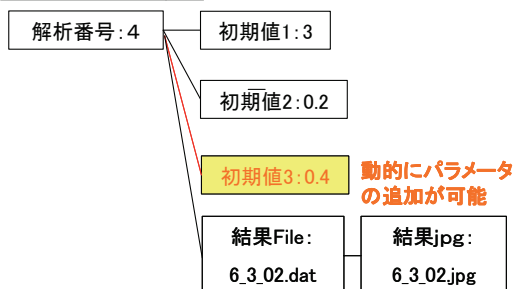


図.1 RDBとXML-DBの比較

2) Webポータル、Webサービス

XML-DBを採用するだけで研究・開発系業務の再現性、トレーサビリティを確保することが可能になるかというと、そうではない。一昔前のように研究者各人にPCなどが無く、端末室や実験室に直接移動し、専用のシステムからしか研究業務を行わない場合は、そのシステムとXML-DBを連結するだけで事足りたはずである。しかし、現状は、ほとんどの業務が自分の机からPCで処理できるので、XML-DBをどこかに設置したところで、各研究者のPCと連携できなければ、ほとんど無意味である。

通常の事務系システムでは、各PCに専用のソフトウェアをインストールして、そのソフトウェアが、データベースとやり取りすることで、上記問題をクリアしている。

しかしながら、その方法では、専用のソフトウェアであるがゆえに、固定的な作業しかできず、折角のXML-DBのフレキシビリティが損なわれてしまう。

PCのブラウザをベースとしたWebサービスによるポータルは、上記課題を解決し、XML-DBのフレキシビリティを生かしたままシステムを構築できる可能性のあるものである。また、Webサービスを基盤としたシステムは、他の専用システムと比べ、ミドルウェアが充実しており高度なシステム構築が比較的容易で、インターネットとの親和性が高く、Public Key Infrastructure（公開鍵方式の暗号化基盤）によるセキュリティ確保やファイアーウォールを透過させることも容易であるという利点を持っている。

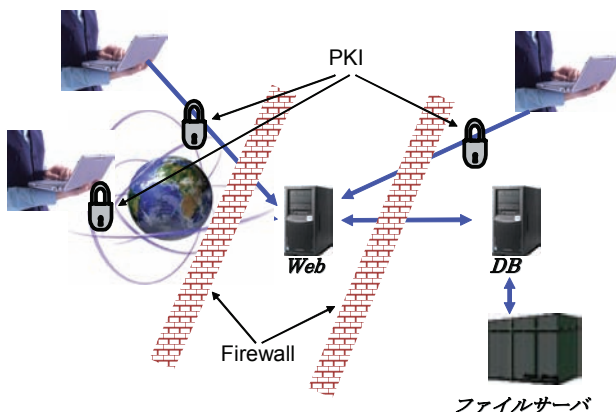


図.2 Webサービスによるポータル化イメージ

3) リモートサーバ制御、ワークフロー

XML-DB、Webサービスを採用することで、変化の激しい研究・開発系の情報を格納するデータベースと各研究者の机に分散されているPCは連携が可能になる。しかし、研究・開発系業務の再現性、トレーサビリティを確保するには、まだ不十分である。

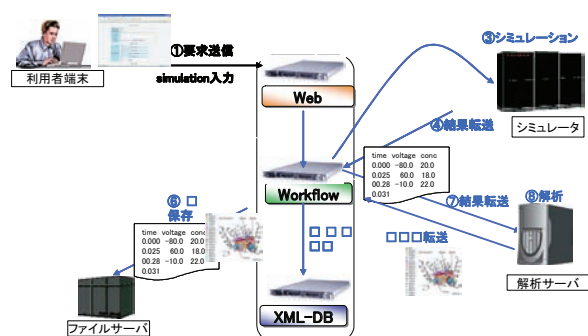
研究開発系業務は、研究者のクライアントPCだけでなく、実験装置やスーパーコンピュータ、専用の可視化装置など様々なコンピュータを使って作業が進められる。もちろん、ほとんどがネットワークに接続され、IPベースで情報、ファイルのやり取りができるようになっている。その意味では、研究者は、自分のデスクから離れないでほとんどの作業が実施可能である。（SSHを使ってリモートサーバのターミナルを開き、SCPでファイルをPCに持ってくるといった作業が行われるのだろう）

上記作業の通りであれば、Webサービスは使われないし、当然、その先につながっているXML-DBにも何の情報も入らなくなり、やはり、システムとしては、意味を成さない状態である。Webサービスを使って、リモートサーバのファイルにアクセスでき、リモートサーバで解析ができる機能、つまり、リモートサーバアクセス・制

御機能が必要なのである。

Web、リモートサーバ制御、DBの3つの機能は、Webサービスシステムでは、標準的であり、3階層から構築されているところから、一般的に3-Layer Web systemと呼ばれている。

また、リモートサーバを利用するだけでなく、それらの作業内容をXML-DBに記憶させたり、記憶させた情報をXML-DBから引き出したりするためには、当然、作業が増えるので、それらを省力化するワークフローエンジンも必要である。



3. システム開発指針と基盤ソフトへの要求条件

前節で議論した技術コンポーネントをつなぎ合わせることで、R&Dをマネジメントするシステムが構築される。しかし、必要機能を実現するだけでは、R&D組織で十分に活用、運用できるシステムにはならない。

事務系の業務システムであれば、技術コンポーネントをつなぎ合わせて必要な機能が実装できれば、十分であり、その実装手法、システム化方法は、開発者の裁量で決められることが多い。これは、事務系の業務システムは、一度、仕様を決めてしまうと変更することは少なく、長期間同じシステムで運用を継続することが多いからである。

しかしながら、R&D組織では、あらかじめ十分な機能、仕様を列挙することが困難なだけでなく、長期間全く同じシステムで運用を継続しなければならない環境は、R&D活動そのものを阻害することさえある。

それでは、R&D組織を対象にしたシステム開発の場合、どのような要求条件があるのだろうか？必要と思われる4つの条件を以下で列挙しておく。

1) アプリケーション、ワークフロー、フレームワークごとに独立に開発・更新が可能なこと。

(a) **アプリケーション**とは、解析、可視化、データフォーマット変換等の機能要素であり、サーバを跨がないスタンドアローン動作が可能なソフトウェアを指す。

(b) **ワークフロー**とは、サーバ内で複数のアプリケーション

を連携させたり、他サーバへのファイル・情報を移動させることで、サーバ間のアプリケーション連携を可能にするためのインタプリタ言語である。

(c) **フレームワーク**とは、システム(Web, Controller, DB)を支える基盤部で、ユーザインタフェース、通信、ワークフロー言語を解釈し実行するワークフローエンジン、データベース等のWebシステムに必要な機能を提供するミドルウェアである。

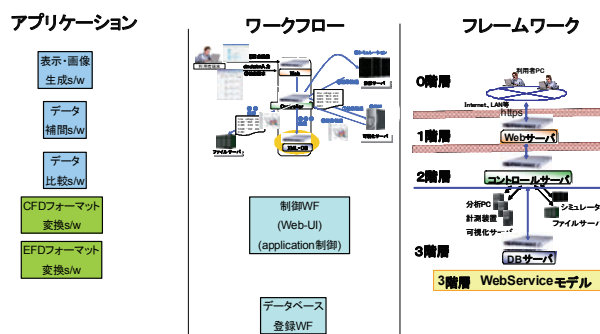


図.4 アプリケーション、ワークフロー、フレームワークの独立性

2) アプリケーションは、なるべく体系的な動作を取り込まずスタンドアローン動作ができるように作成すること。開発速度、メンテナンス性およびFieldviewやAVSといった市販ツールとの連携という観点において重要である。逆に言うと、スタンドアローン動作できるようにさえアプリケーションを作成すればシステム化が可能なようなワークフロー、フレームワーク基盤が必要という意味である。サーバ間を跨ぐ処理は、ワークフローによるファイル授受で行うべきであるが、その部分がボトルネックとなる場合、共有ファイルシステム等をハード側で準備し、対処すべきである。

3) ワークフローは、デバック、改良が簡単のようにインタプリタ言語で書かれるべきである。ただし、記述言語は、PerlやPythonのような自由度の高い汎用言語ではなく、ワークフローを記述するのに最低限度必要な自由度のみを持つものが好ましい。なぜなら、汎用言語のように自由度が高いとワークフローに統一性と可読性を持たせるのが困難となるからである。また、ワークフロー自体の共有や書換禁止等のアクセス制御や検索絞込みが容易なようにXML-DBへ格納可能なXMLで記述されるべきである。同じ処理を各ワークフローに直接記述すると煩雑かつメンテナンス性が悪いので、他ワークフローをsubroutine的呼び出し連携ができる機能を有することが好ましい。

4) 任意のアプリケーション、ワークフローの変更ごと

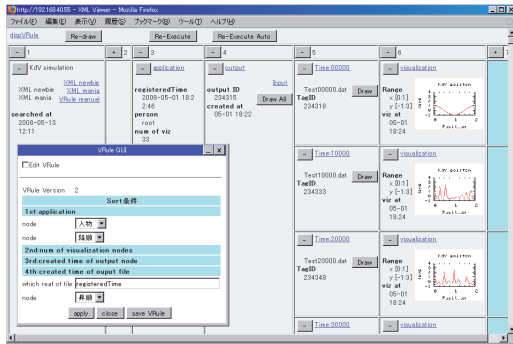


図.7 検索結果表示画面

- 5) 検索結果表示画面にWorkflow実行ボタンを付けることもできる。リモートサーバのファイル取得も同じ機構を使っており、URLリンクではなく、Workflowに基づく処理（認証、権限管理確認を伴う）であり、クラッキングに対する防護性が高くなっている。

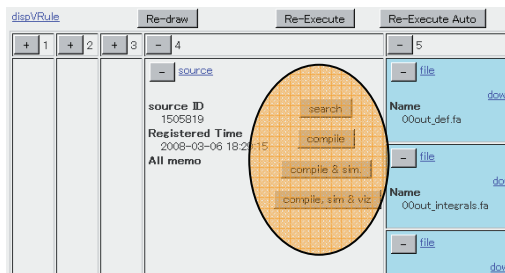


図.8 Workflow実行ボタンの設置

- 6) 検索結果表示画面に直接、コメントを入力したり、ファイルを添付、任意の研究者へメール発信をすることができ、研究ノート、グループ内の意見交換プラットフォームとしても利用できる。

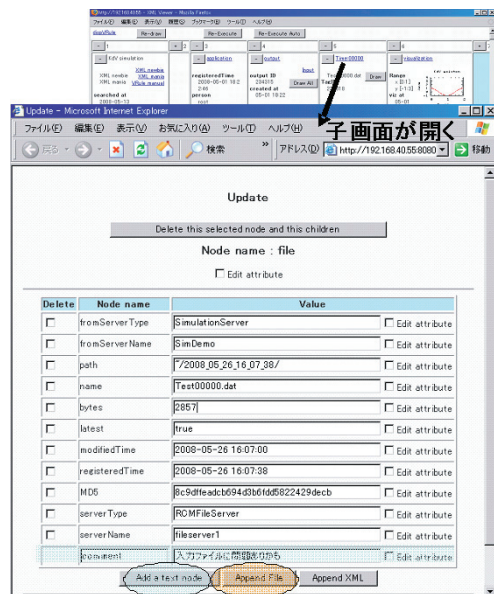
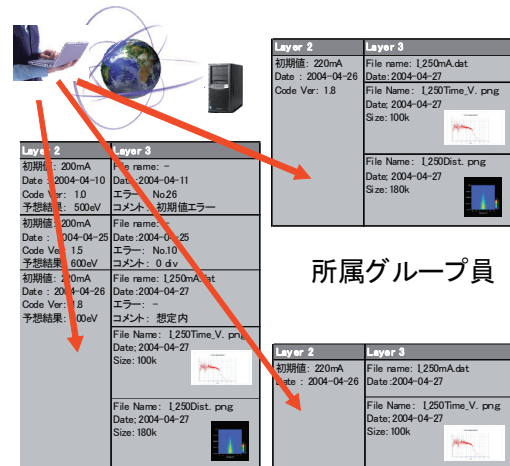


図.9 電子ノート感覚のデータベース書き込み

- 7) すべての情報（ファイルやその他の情報）は、XML-DBにより閲覧、書込のアクセス権限が設定されており、実体ファイルや情報をCOPYすることなく、機密情報、仮想グループ内共有情報、公開情報などを容易に管理できる。また、各ユーザが自由に仮想グループを定義することが可能であり、細かな権限管理がユーザレベルで容易に設定でき、運用において高い機動性を有している。



研究者本人

共同開発者

図.10 XML-DBによりアクセス制御

- 8) Java Web Start技術を使うことで、予めクライアントPCにサーバアプリケーション（SSHサーバ Tomcatサーバ等）の準備なしで、フレームワークからPC上のファイルをWorkflowで扱うことが可能である。

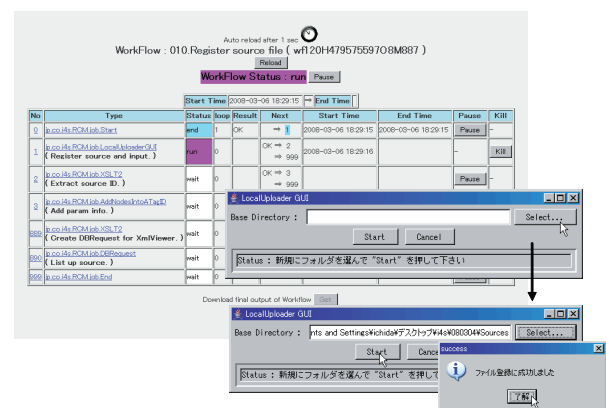


図.11 JavaWebStartによるPC内作業のWorkflow化

- 9) 履歴管理付きリポジトリ機能により、個人PC内のファイルをサーバにアップロードでき、更新履歴などの履歴管理ができる。
- 10) シミュレーションのソース管理、コンパイル、実行、可視化などを統合的に扱うことができるデータベース

