

ウェブ活用のシミュレーション構築・湛水リスク評価を例にとって

富永 直利*

Building Web Based Simulation:
Evaluation of Agricultural Reservoirs, As an Example

Naotoshi Tominaga*

アドバンスソフトでは内閣府官民研究開発投資拡大プログラム PRISM 気候変動対策プロ（豪雨対策）コンソーシアム共同研究のテーマ「農村地域における気象災害の発生リスク予測技術」において、ウェブを用いたシミュレーションの研究開発を行った。本稿ではその内容を紹介するとともに、技術背景についても解説する。

Keywords: ウェブ活用のシミュレーション、湛水リスク、分布型水循環モデル、情報モデル、データ中心設計、データベース、NetCDF、フロントエンド、バックエンド、大規模データ解析

1. 背景

アドバンスソフトでは内閣府官民研究開発投資拡大プログラム PRISM 気候変動対策プロ（豪雨対策）コンソーシアム共同研究のテーマ「農村地域における気象災害の発生リスク予測技術」の中でウェブ活用のシミュレーション構築を実施した。ソルバーは国立研究開発法人農業・食品産業技術総合研究機構（以下、農研機構という）において整備されたグリッド型の水文モデルによる河川流量および降水量を計算するプログラムを用いた。研究者などが単独で実施する数値シミュレーションのプリ・ポスト処理のようにスタンドアローン PC などでも実施するものでは、結果を複数の人が同時に確認することや、オンラインで時々刻々と配信されるデータに対して自動更新をするサーバーアプリケーションには不向きであるため、ウェブサーバーを用いたシミュレーション環境を構築することが解決策となった。

2. 技術背景、情報モデル

ここでは、一般的に用いられるスタイルの数値シミュレーションと、ウェブを活用した場合の数値シミュレーションとのシステムの差を情報モデルの差として論述する。

2.1. 一般的なスタンドアローン PC を用いた数値シミュレーション

スタンドアローン PC で利用されるシミュレーションソフトウェアは図 1 のように、プリ・ポスト処理とソルバー部分にプロセスが別れる。ソルバー部分は計算部分であるが、プリ処理においては、計算条件を作成し、ポスト処理では結果のプロットや可視化を行う。プリ処理はモデル作成と計算条件の設定であり、例えば流体解析においてはメッシュ生成や、ソルバー部分の入力形式に従った計算条件を記述した入力ファイル作成を意味し、ポスト処理は例えば 2 次元グラフやコンター表示などが例としてあげられ、計算結果が妥当であったか確認をするためにも重要である。

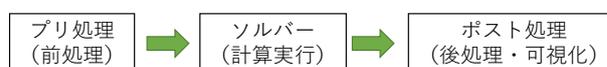


図 1 スタンドアローン PC による数値シミュレーションの一連の処理

*アドバンスソフト株式会社 熱流動エンジニアリングセンター

Thermal Hydraulics Engineering Center, AdvanceSoft Corporation

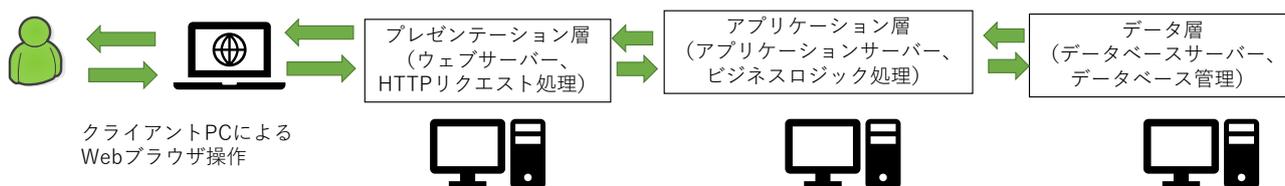


図2 ウェブ3層アプリケーション

2.2. 数値シミュレーションの概要

湛水リスク評価を行うソルバー部分は農研機構で整備されたモデル[1]を用いる。これは、図3に引用として示した河川・貯水池、水田灌漑地区の相互作用や河川から灌漑地区への実取水量、水田供給量、河道への還元量等が定量的に評価できるもので、流域に対しての雨量から変化する河川などの水量の変化を評価することで、湛水リスクを評価することに利用できる。このプログラムの雨量データは日々気象庁より提供される解析雨量・15時間先の予測データを利用できるように拡張している。システムにおいては、解析雨量により現在の物理量を更新すると同時に、同時に配信される15時間先の予測データを用いることで河川流量の変化を予測することで湛水リスクの評価をすることができる。

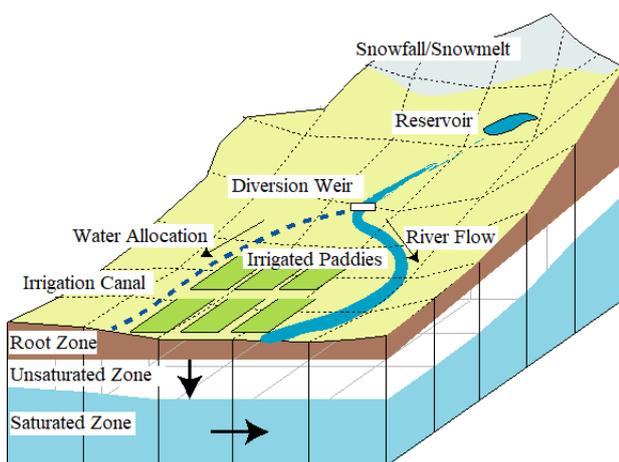


図3 用水配分・管理モデルを構成する要素とモデルの概念図（[1]より引用）

2.3. ウェブ3層アプリケーション

2.1節のスタンドアローンPCでのロジックを処理する数値計算アプリケーションに対して、ウェブアプリケーションについて対比的に説明するために、独立行政法人 情報処理推進機構がまと

めた資料[2]において、ウェブ3層アプリケーションは図2のようにユーザインタフェース、ビジネスロジック、データベースの相互依存を抑え、変更容易性、拡張性を高めることができるものとして、実際に運用されていることが述べられており、これを例として挙げる。ウェブ3層アプリケーションは、ISO/IEC 7498[3]に規定された通信機能を階層構造に分割したモデルであるOSI参照モデルのように、プレゼンテーション層、アプリケーション層、データ層にウェブシステムの構成要素を分割しサーバーの働きを分割したもので、[2]では以下の3つのように説明されている。

- プレゼンテーション層：ブラウザからのHTTPリクエストを処理し、必要に応じてアプリケーション層に処理を委譲する。また、アプリケーション層から処理結果を受け取り、ブラウザに返す。
- アプリケーション層：プレゼンテーション層からの要求に応じてビジネスロジックを処理する。必要に応じてデータ層にアクセスし、データの検索、保存などを行う。
- データ層：データを管理し、アプリケーション層に対してデータの検索、抽出、保存などのインターフェースを提供する。

3. 情報モデルの検討と適用

3.1. データ中心設計

これまで述べてきたスタンドアローンPCで利用されるシミュレーションソフトウェアをそのままウェブ3層アプリケーションとして適用する場合、サーバーが3台必要となるが、サーバーの負荷や、複雑度によっては、1台でも3層すべての役割を担うことは可能である。まず、弊社では

実際のプログラムに即したものとするために、農研機構より貸与いただいたコードを把握したうえで、設計を実施し、オフラインで画面の動作を確認するためのプロトタイプを作成してその状態を把握することとした。まず、前提としてデータから検討した。これは情報処理におけるデータベース分野ではデータ中心設計[4]として考えられる経験上の知見から、まずデータから検討した。従来のソルバー部分は Fortran 言語で記述されており、データは Fortran に依存したバイナリによる入出力が行われていたが、このままではデータベースを構築することができなかつたため、オープンな地理情報標準を策定する国際的なコンソーシアムである、Open Geospatial Consortium における国際標準であり、気象データで広く使用されている NetCDF 形式[5]を用いたデータベースを構成できるよう設計と実装を行った。NetCDF 形式は図 4 に示したヘッダー部分と、データ部分の二部構成で、ヘッダーに規定されたフォーマットが記載されており、プログラムの書き方に依存したデータ形式にはならない。

このことによって、データの取り扱いは気象データが変更されたとしても、データの入出力に関する部分の変更がなくデータに柔軟なプログラムを構築することができる。すなわち、決まったプロトコルによるデータ処理は、独自の抽出方

法などに依存することなく、SQL のクエリによる抽出をするような形式で、汎用的なデータベースの取り扱うことができる。

NetCDF 形式もバイナリ形式であるが、これは OS に依存しない。以上から、データ層とアプリケーション層がインターフェースを持つことができる。

3.2. 情報モデル設計

一方、プリ処理・ポスト処理についてはフロントエンドの整備が必要であった。フロントエンドとは、クライアント PC によるウェブブラウザでは決められた日付のデータを閲覧したい場合、グラフィックによるユーザーによる入力が必要となる。このようなウェブブラウザ側の要素をフロントエンドと呼び、サーバー側をバックエンドと呼び分けて区別する。シミュレーションにおける可視化自体はフロントエンドで表示できるように構築する必要があるが、可視化に必要なデータはバックエンドで処理する必要がある。すなわち、プレゼンテーション層とアプリケーション層はまとめて一つのバックエンドであるサーバーに集約し、フロントエンドが必要とする可視化用データを用意する。フロントエンドの構築においては広く JavaScript が多く利用されるが、バックエンドでは通常利用されない。本検討においては、開発効率性を考慮して、フロントエンドで使われている JavaScript をバックエンドでも利用できるようにした Node.js[6]という技術を用いたシステムとして設計した。以上の設計を考慮した数値シミュレーションのためのウェブ 3 層アプリケーションの情報モデルを図 5 に示す。

フロントエンド可視化については、地図上に計算結果の表示や拡大縮小・スライダー操作など、動的に処理する部分もあり、これは HTML・CSS・JavaScript を利用する。HTML は、ブラウザが読み込める文章形式で、これがそのままユーザーの視覚に反映され、またユーザーからの操作を受け付ける窓口となる。その裏では、表示の調整を CSS が行っており、単なるフォントの際の調整などだけではなく、回転や縮小、画像の位置調整や回転

NetCDF フォーマット例
<pre>netcdf_sample { dimensions: variable_size = 500; time = UNLIMITED; // (1000 currently) variables: int time(time); float variable_a(time, variable_size); float variable_b(time, variable_size); float variable_c(time, variable_size); (以下、データ部) }</pre>

図 4 NetCDF 形式の例

など、さまざまな調整をバックグラウンドで処理して、ブラウザ上に反映される。また、動的な処理については動的に表示している HTML や CSS を書き換えることができる JavaScript によって実現する。概要を図 6 に示す。

本アプリケーションにおいては、フロントエンド可視化の際に地図上に湛水リスク評価を表示するため地図データと、地図を拡大縮小・操作表示するためのフロントエンド用ライブラリが必要となった。これには leaflet[7]を利用した。これは BSD ライセンスであり、ライブラリとして利用するため、整備したソフトウェアへの影響がなく利用できる。また、地理データについては、国土地理院データを用いるものとし、国土地理院より公開されている地理院タイル[8]を利用することとした。地理院タイルは、アクセスするための仕様がオープンであり、商用非商用問わず、利用

するには地理院タイルの表示をサイト内に入れるのみでよいことからこれを採用して設計した。

3.3. プロトタイプの実装

以上の設計を実際に適用し、図 7 に示す表示でプロトタイプを作成した。日付、スライダーのユーザーコントロールを配置し、それを操作することで、動的に地図に示したチェックボックスがついたデータを更新するものとなっている。動的に動かすために JavaScript によるフロントエンドの実装と、データをサーバー側にリクエストした結果を取得できるようになっている。プロトタイプのため、このプロトタイプでは、湛水リスクの評価に必要な河川流量のみを表示しており、カラーカウンターによってその量が定量的に可視化されている。



図 5 数値シミュレーションのためのウェブ 3 層アプリケーション

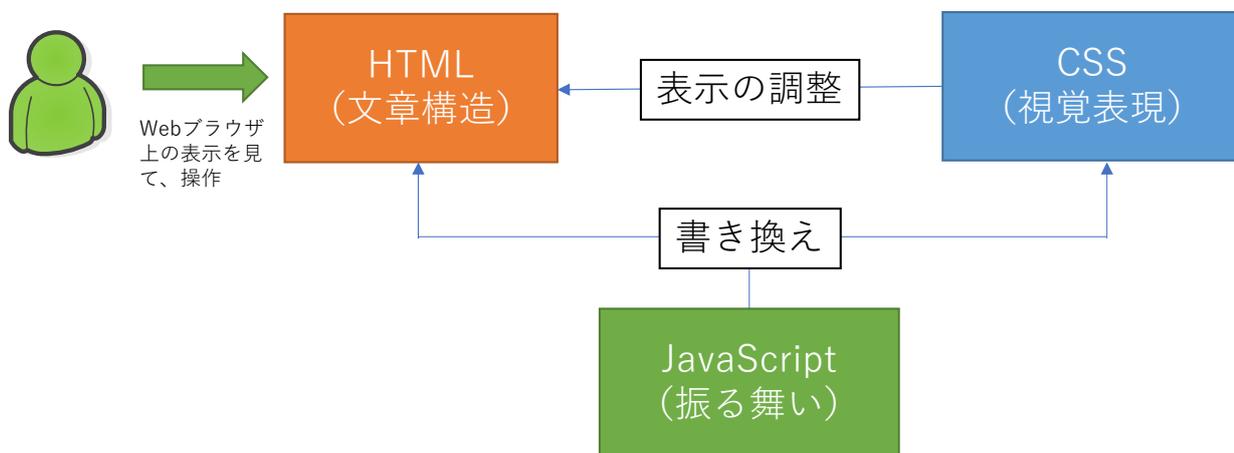


図 6 フロントエンドの概要

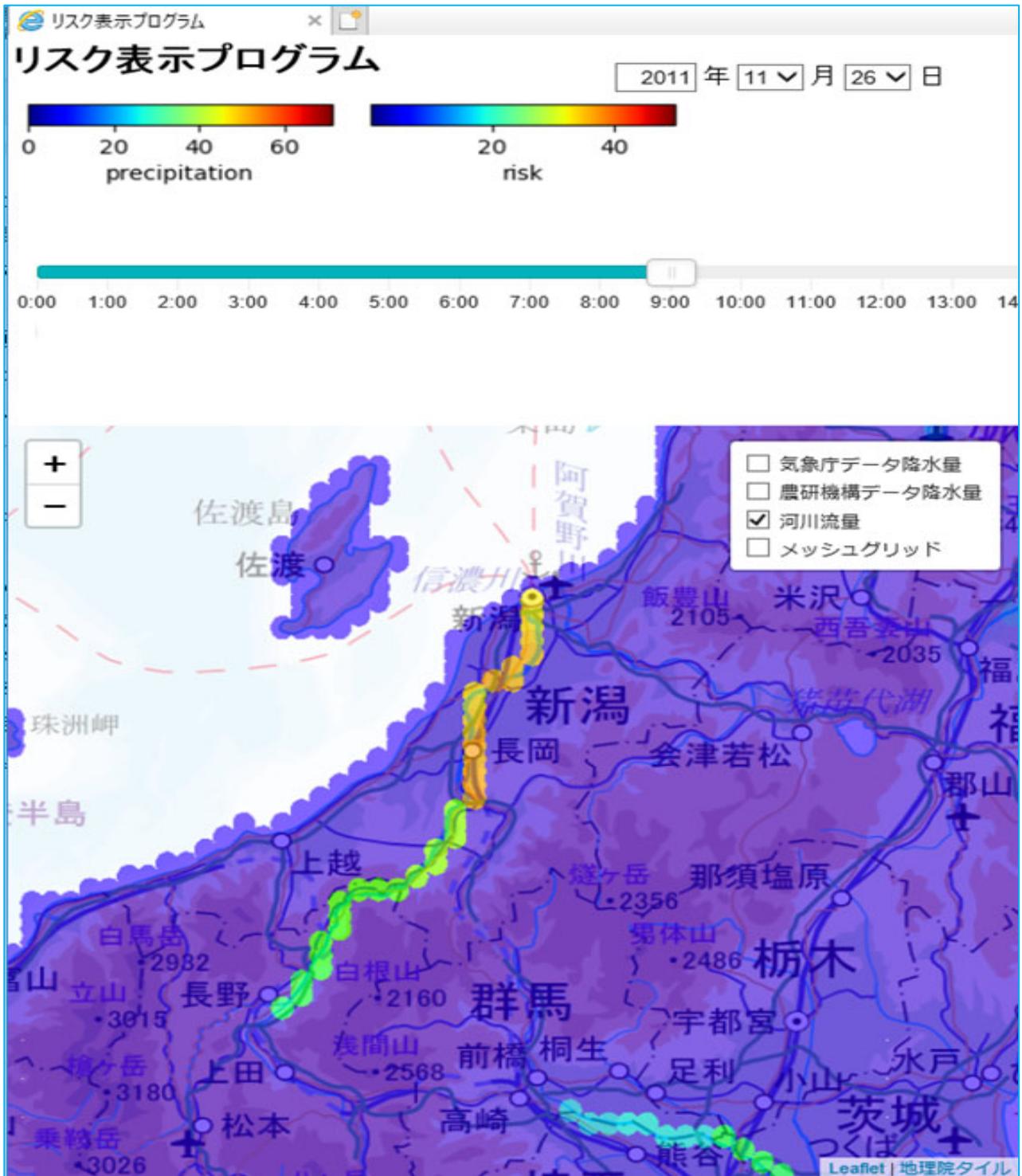


図 7 プロトタイプの外観

4. まとめと今後の展開

本稿では具体的な研究開発を事例として挙げ、ウェブ活用のシミュレーション構築についての解説を行った。ウェブアプリケーションは、インターネットを利用したビジネスには欠かせない技術であり、その技術者も多くかつ、活用される

業種も幅広い。また、ニッチな技術ではないため、技術の情報もインターネット上に多く共有されている。

今後の展開としては、フロントエンド・バックエンドの技術を利用したアプリケーション開発があげられる。本稿で開設したフロントエンドは

どの OS でも対応するよう整備され、ブラウザ上で動作する。これを利用したマルチプラットフォームによるブラウザベースアプリケーション開発の近年進められており、例えば、Microsoft 社が提供している統合開発環境である Visual Studio[9]は Windows でしか動作しないものであるが、近年オープンソースプロジェクトとして Visual Studio Code[10]という、ブラウザベースのアプリケーションも並行して開発が進められている。これは Windows 以外の Linux、Mac でも動作し、エディター機能やコンパイル機能が利用できる軽量なものである。これには、Electron[11]というフレームワーク技術が使われており、フロントエンドで作成した HTML、CSS、そして JavaScript で記述されたものがスタンドアロン PC で動くアプリケーション（例えば、run.exe などのような形式のバイナリ）としてパッケージ化して動作させることができるもので、この技術により、ウェブ活用のシミュレーション構築による成果はプラットフォーム、活用シーン（ウェブ・スタンドアロン PC など）を問わない継続的な技術統合を図ることが期待できる。

謝辞

本研究開発を進めるに当たり、国立研究開発法人農業・食品産業技術総合研究機構 農村工学研究部門 水文水資源ユニットの多大なる協力を得た。ここに記して感謝の意を表す。

参考文献

- [1] 吉田武郎, et al. "広域水田灌漑地区の用水配分・管理モデルの実装による流域水循環のモデル化.", 農業農村工学会論文集 80.1, (2012): 9-19.
- [2] OSS モデルカリキュラム V2 (2011 年 5 月公開) : IPA 独立行政法人 情報処理推進機構, https://www.ipa.go.jp/jinzai/renkei/ipedia/oss_mcv2
- [3] ISO - ISO/IEC 7498-1:1994 - Information technology — Open Systems Interconnection

— Basic Reference Model: The Basic Model, <https://www.iso.org/standard/20269.html>

- [4] 堀内一, データ中心システム分析と設計, オーム社, (1989).
- [5] Unidata | NetCDF, <https://www.unidata.ucar.edu/software/netcdf/>
- [6] Node.js, <https://nodejs.org/>
- [7] Leaflet - a JavaScript library for interactive maps, <https://leafletjs.com/>
- [8] 地理院地図 | 地理院タイル一覧, <https://maps.gsi.go.jp/development/ichiran.html>
- [9] Visual Studio IDE、コード エディター、Azure DevOps、App Center - Visual Studio, <https://visualstudio.microsoft.com/ja/>
- [10] Visual Studio Code - Code Editing. Redefined, <https://code.visualstudio.com/>
- [11] Electron | Build cross-platform desktop apps with JavaScript, HTML, and CSS., <https://www.electronjs.org/>

※ 技術情報誌アドバンスシミュレーションは、アドバンスソフト株式会社 ホームページのシミュレーション図書館から、PDF ファイルがダウンロードできます。(ダウンロードしていただくには、アドバンス/シミュレーションフォーラム会員登録が必要です。)