

3R-03 HTTP プロトコルの機能的限界と拡張の方向性

村上雄介

日本アイ・ビー・エム システムズ・エンジニアリング(株)

1. はじめに

1990年代に入り、グループウェアを始めとするクライアント/サーバーシステムの普及が著しい。これらのシステムでは、専用のクライアントや専用のプロトコルが用意されている。専門クライアントでは、優れたユーザーインターフェースの実現が可能であり、一般的にクライアント/サーバーシステムは機能的な完成度が非常に高い。こうした豊富な機能が好まれ、これらの製品は広く市場に回っている。ところが最近になると、専門クライアントのソフトウェア配布にかかるコストの観点などから汎用クライアントの重要性が認識されてきており、インターネット技術の普及につれて市場のニーズはイントラネットアプリケーションへと変化していった。こうした背景から、クライアント/サーバー製品の完結した技術は、イントラネットサーバーとしてのプロトコル変換技術へと発展を遂げつつある。

プロトコル変換技術の普及に伴い、近年多くの企業が既存システムのイントラネット化の検討を行ってきた。ところが、こうしたイントラネット化は検討の結果、実行を断念する場合が多い。本稿では、クライアント/サーバーシステムとHTTP (HyperText Transfer Protocol) で実現される世界との比較を行うことで、その失敗の原因の解明を試み、その概要を報告する。

2. HTTP の特徴

HTTP は、簡易なメソッドと認証方法が定義された通信プロトコルであり、静的文書の閲覧を行うアプリケーションを可能にする。この文書は、HTML (HyperText Markup Language) というメタ言語で記述されている。HTTP という軽量プロトコルは、汎用クライアントである WWW (World Wide Web) ブラウザが幅広く普及されていることも手伝い、インターネット基盤の上で大変発達した。シンプルなプロトコルは高機能なアプリケーションの実現には機能不足であったが、まもなくこうしたアプリケーションの実現に必要な補完技術が次々と出現し、HTTP 上での実現を可能とした。

3. クライアント/サーバーシステムのイントラネット化と Web 技術

次々と高機能の実現を可能としてきた Web 技術であるが、それでは何故クライアント/サーバーシステムのイントラネット化が失敗に終わるかを究明していく。

クライアント/サーバーシステムの特徴は、専用プロトコルと専用クライアントの使用による高機能なアプリケーションにある。特に、専門クライアント上で機能する優れたユーザーインターフェースは大きな特徴である。一方、Web 技術は軽量プロトコルと汎用クライアントによる技術であるが、アプリケーションの機能は様々な周辺技術として補強されており、必要とされる多くの機能は実現することが可能だ。汎用クライアントが導く限界もあるが、Java アプレットなど汎用クライアント上で動作可能なアプリケーション技術が代替を可能にしている。

こうした Web 技術の発達にもかかわらず、イントラネット化の試みが失敗に終わる背景は、主に次の 2 点に帰結することが多い。

- クライアント/サーバーシステム側での、未熟なプロトコル変換技術

プロトコル変換技術は日々成熟しており、純粋な HTML に対する変換だけでなく、Web 技術を支える拡張技術への自動変換などの領域へと進んできた。しかし、未だに高機能な元来の機能の全てを変換するには至っていない。

- 従来実現していた機能を Web 技術で開発する現実性

上記の自動変換技術の不足に対する機能の必要性は、Web 技術を利用した開発で実現する。こうした開発による実現の現実性と開発コストとの兼ね合いから、こうした開発が非現実的であると判断するケースがある。

なお、コストの話あげたが、従来のクライアント/サーバーシステムと比較し、Web におけるアプリケーションは実行環境をサーバー側に移すため、サーバーとしてのハードウェア台数が増えるのも、考慮点の一つとなることが多い。

4. 考察

こうした失敗の背景には、HTTP の機能的限界があったのであろうか。専門プロトコルや専門クライアントのアプリケーションをイントラネット化するには、現状の技術に不足があるであろうか。

そもそも HTTP の機能的限界とは何であろう。HTTP とは前述の通り、簡易メソッドと認証方法からなるアプリケーションプロトコルである。従って確かに専門プロトコルと専門クライアントで実現されていた豊富な機能を満たしていない点が多い。しかし、こうした弱点を Web 技術では周辺の補完技術で機能の補強をしており、グループウェアなどに代表されるアプリケーションで実現していた機能は、これらの補完技術がそのほとんどをカバーしている。特に Java アプレットの技術は、クライアントサイドの多くの障壁を取り除き、高機能を実現可能にする。

こうした経緯から、簡易な機能の HTTP を拡張していく考えは必ずしも正しいとはいえない。むしろ、機能面での弱点は補完技術の拡張で解決していく問題であろう。もし HTTP としての拡張が必要であれば、機能補完ではなく通信プロトコルの観点から進むべきであろう。

5. おわりに

高機能アプリケーションを Web 技術に変換するためには、成熟したプロトコル変換技術が鍵を握っており、特にビジネス観点からは重要である。変換技術の難しさは、HTTP を支える補完技術が多種多様にわたっていることや、アプリケーションとして高機能が求められていることにも起因する。しかし、これは変換技術だけに問題があるわけではない。クライアント側での機能が豊富すぎたシステムが、困難にしている要因なのだ。確かに、豊富な機能に慣れ親しんだ利用者にはクライアント側の機能低下は問題であるが、逆を言えばこうしたシステムと技術の利用は、イントラネットシステムを構築するにあたって、サーバー側で必要とされる多様な補完技術を、既に円熟した一体感のある技術で実現することを可能にする。

市場における高機能なアプリケーションの要求により、クライアントはますます肥大化する傾向にある。その一方で、モバイルなどに対応した軽量なクライアントにも脚光があたっている。高度な要求を満たすアプリケーションと、軽量なクライアント環境とを統合する適切な基盤とは何か。あるいはそれらは別の道を歩みはじめるのか。Web 技術の進歩は、こうした問題を議論すべき局面まで発展してきた。