

サーバレット間通信による分散型臨床症例データベース管理システム

岡本 雅幸¹・納富 一宏²・山口 俊光³・石井 博章²・斎藤 恵一⁴・藤本 哲男⁵

- 1 博士課程前期 情報システム工学専攻
- 2 情報工学科
- 3 情報工学科 3年
- 4 東亜大学経営学部経営学科
- 5 芝浦工業大学工学部機械工学科

Distributed Database Management System for Clinical Cases with Servlet-Servlet Connection

Noriyuki OKAMOTO¹ Kazuhiro NOTOMI² Toshimitsu YAMAGUCHI³
Hiroaki ISHII² Keiichi SAITO⁴ Tetsuo FUJIMOTO⁵

Abstract

Shareing clinical cases are useful for medical field. We suggest a distributed database management system for clinical cases with Java Servlet. In this system, case informations were put two or more servers, and case index information using case searching were put on all servers. Communication between each servers were used Servlet-Servlet Connection. With this method, the system could handle various style of formatted data and system could be more extensive and more reliable.

Key Words: Distributed Object, Java Servlet, Clinical Case, Database, WWW

1 はしがき

臨床症例情報をデータベース化し利用することは、過去の症例を参照し新たな症例の病態を検討・類推する上で非常に重要である。症例情報を扱う際、複数の施設間で情報を共有し統一的に検索・閲覧が可能なシステムを構築することは非常に有用である。しかし、膨大な量の症例情報を単一のサーバへ蓄積し一括して扱う集中管理システムでは、登録・検索時の応答性が悪いとともにサーバ停止時の信頼性に欠ける。

そこで本稿では、臨床症例情報を分散管理するシステムについての提案を行う。臨床症例情報を各施設毎に分散蓄積し、検索に必要なインデックス情報のみを共有することで、高速な登録・検索

が可能となる。また、情報を複数のサーバで保持することも可能となり、システムの信頼性を向上させることができる。

本稿で提案するシステムでは、情報の登録及び閲覧を行うクライアントソフトウェアとして広く一般的に用いられている WWW(World Wide Web) ブラウザを用いる。サーバプログラムは Java サーブレットとして記述し、サーバレット間通信を用い相互に情報交換を行う。情報を分散蓄積することにより、1) 情報の複製 2) 情報の同期といった問題が発生するが、扱う情報を臨床症例情報に限定することでこれらの問題点を回避できることを示す。

2 システム構成

2.1 WWW を用いた 3 層システム

現在インターネットで最も広く用いられている WWW を利用することは、地理的に異なる場所に点在する施設間で情報を共有する手段として非常に有力である。また、システムのインタフェースとして広く出回っている WWW ブラウザを利用するので、システム導入時におけるユーザへの負担を最小限に抑えることが可能となる。

ユーザは WWW ブラウザを用いて WWW サーバへ接続する。WWW サーバ上では WWW サーバプログラムが処理プログラムを起動する。処理プログラムはデータベースへ接続し必要な情報の取得・更新を行う (Fig.1)。

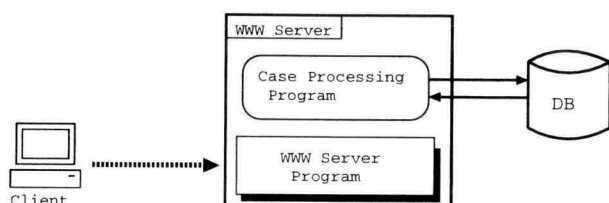


Fig. 1: WWW を用いた 3 層システム

3 層システムとは、

- クライアントアプリケーション
- アプリケーションサーバ
- データリソース

の 3 つの部分から構成されるシステムである。Fig.1 におけるブラウザがクライアントアプリケーションであり、WWW サーバがアプリケーションサーバである。

2.2 集中システムにおける問題点

全ての情報を一カ所で保持し、登録・検索・更新などの処理を行う集中システムでは、全てのクライアントからのアクセスが集中してしまう。このため以下の問題点が発生する。

負荷の集中 多くのクライアントから同時に接続される可能性が高い。負荷が集中し 1 つの接続に対する処理能力が落ちると、ユーザから見たシステムの応答性が悪化する。

高速な回線の必要性 サーバまでの回線速度がシステムの応答性に影響する。より快適にシ

ステムを使用するには、高速な回線が必要になってしまう。

障害に対する脆弱性 サーバには、ハードウェア故障などのトラブルに加えて定期的なメンテナンスが必要となる。集中システムではシステムの利用者全員に影響が及んでしまう。

本稿で提案するシステムでは、これらの問題を回避するために情報の分散蓄積を用いる。

2.3 分散システムにおける問題点に関する考察

情報を複数のサーバへ分散し蓄積することにより 2.1 の問題点を回避することができるが、情報の同期と複製に関する問題が新たに発生してしまう。これらの問題は、扱う情報を臨床症例情報に限定し検索に必要なインデックス情報と症例情報を分離して考えることにより無視できる。症例情報の検索手法については本論では論じない。検索に必要なインデックス情報としては、症例登録時に入力したキーワードや症例情報から抽出した単語の頻度情報等を仮定する。

以下に情報を分散蓄積する際に発生する問題点について述べる。

複製の問題 膨大な量の症例情報を全てのサーバで共有するのは現実的ではない。以下で述べるインデックス情報による検索手法を適用することにより、共有すべき情報の量を少なくすることができる。

同期の問題 臨床症例情報は過去に発生した症例情報であるので、本システムでは情報の即時性は必要ないと仮定する。あるサーバへ登録された症例情報は他のサーバへ段階的に伝えられる。

2.4 システム構成

本システムの構成図を Fig.2 に示す。各ユーザは WWW ブラウザを通して臨床症例情報の検索・閲覧及び登録を行う。

各施設での臨床症例情報は、その施設のサーバへと格納される。同時に症例情報がバックアップサーバへコピーされ、インデックス情報が全てのサーバへ配布される。

検索は全てのサーバに保持されているインデックス情報を元に行われるため、最も近隣のサーバに接続することで高速な検索が可能となる。検索

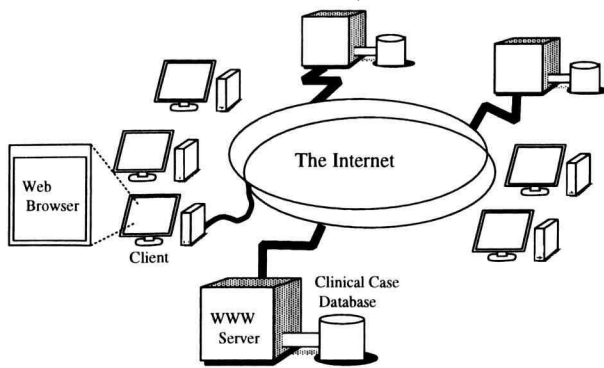


Fig. 2: システム構成

された情報の閲覧時には該当施設へアクセスするが、該当施設がメンテナンス等の理由により利用不可能であったとしても、バックアップサーバとして指定された施設からの情報取得が可能である。

3 Java サブレット間通信

3.1 Java サブレットの利用

後述するオブジェクト転送によるサーブレット間通信 (3.3) を利用するため、サーバ上で動作するシステムの構築には Java 言語を用いる。Java 言語を用いることによりアーキテクチャ非依存となり相互運用性が向上する。

システムは Java サブレットとして作成される。CGI(Common Gateway Interface) を用いたプログラムに比べ、以下の点において有利である。

高速な動作: 要求毎に処理プログラムを読み込む動作が無いいため、高速に動作する。

高い開発効率: クライアントとの通信部分やデータのエンコード/デコード処理はサーブレットエンジンに内包されているので記述するコード量が少ない。

3.2 HTTP トンネリング

各サーバ間の通信にはサーブレット間通信を用いる。各サーブレットはサーブレットエンジンを通して単一の通信ストリームによって結ばれる (Fig.3)。通信には HTTP が用いられるので、Proxy サーバを経由することも可能となり、Firewall と共存可能である。

3.3 Java オブジェクトの直列化と転送

HTTP は状態を持たないプロトコルであり、Servlet 間通信においては状態を持った複数のコマンドからなる通信プロトコルは適さない。そこ

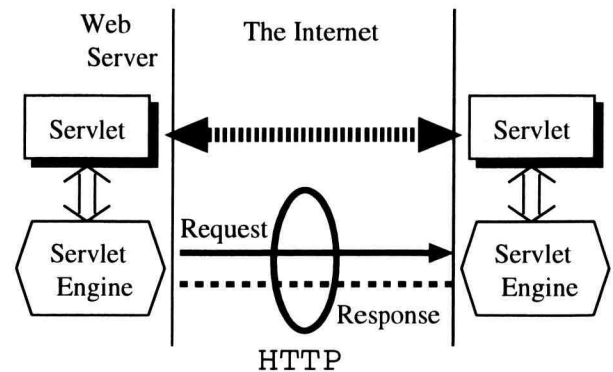


Fig. 3: HTTP Tunneling

で、要求を単一のオブジェクトとしてカプセル化し送信する手法を用いる (Fig.4)。操作に必要な情報を Command Object としてカプセル化し送信し、処理結果は Result Object としてカプセル化され返信される。オブジェクトのバイト列への直列化及び復元機能は JavaVM が備える。

CGI を用い通信プロトコルを定める従来の方式に比べ、次の利点がある。

- クライアントからの要求、及びサーバからの処理結果がオブジェクトとしてカプセル化され送受信されるので、拡張性を備える。
- 処理がをコマンドオブジェクト内に内包されるので、サーバプログラムの変更なしにサービスの拡張が可能となる。

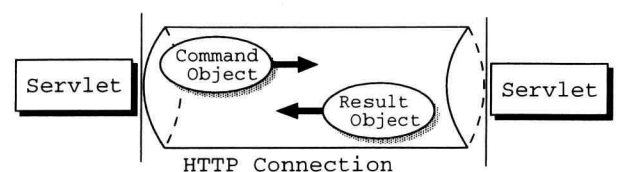


Fig. 4: Object Translation

4 臨床症例情報の閲覧

4.1 症例情報の登録

臨床症例情報はまず、ユーザにより各施設のサーバへ登録される。症例情報が登録されたサーバ (Main Server) は Backup Server へ症例情報のコピーを転送すると共に、他のサーバへインデックス情報のみを登録する (Fig.5)。登録すべき情報はコマンドオブジェクトとしてカプセル化され、送信される。

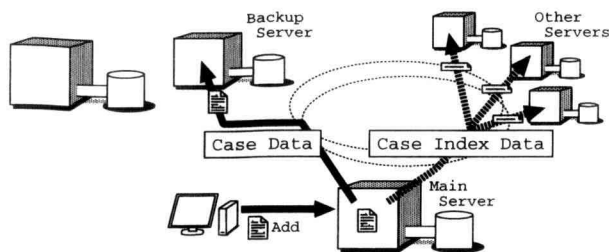


Fig. 5: 症例情報の登録

4.2 症例情報の検索と取得

臨床症例情報はインデックスからの検索となる。インデックス情報は全てのサーバで保持しているため、近隣サーバからの高速な検索が可能である。検索結果は一覧として表示される (Fig.6)。

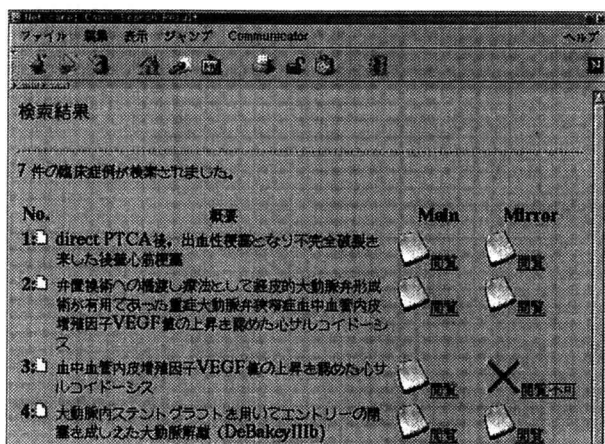


Fig. 6: 症例情報の検索結果画面

症例情報を閲覧する際、複数のサーバを選択することが可能である。症例が報告された施設のサーバが障害等により利用不可能な場合も、バックアップサーバからの情報の取得が可能である。

5 まとめ

本稿で提案を行ったシステムにより、臨床症例情報を複数の施設間で共有する際、高速な検索システムを利用でき、またシステム全体の柔軟性と信頼性を向上することが可能となった。今後の目標としてはサーバ間通信の標準化への対応があげられる。本稿では言及していない次の点についての標準化がなされることを期待している。

- 臨床症例情報の情報形式
- サーバ間通信の仕様

● 検索に用いるインデックス情報の形式

これらの標準化が進めば、臨床症例情報システムの有用性はさらに高まると考えられる。

参考文献

- [1] 納富, 斎藤, 他: WWW による臨床症例データベース検索システムの構築-自然言語処理による検索機能の実現- BMFSA 第 11 回年次大会講演論文集 (1998.11).
- [2] 高木 浩光, Sorblet: ORB over Servlet その使い方と実装 Java カンファレンス Servlet BOF セミナー資料 (1998.11).