

Title	分散知識サービス環境に関する研究
Author(s)	林, 正治
Citation	
Issue Date	2004-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/494
Rights	
Description	知識科学研究科, 修士

修士論文

分散知識サービス環境に関する研究

北陸先端科学技術大学院大学
知識科学研究科 知識社会システム学専攻

林 正治

2004 年 3 月

Copyright © 2004 by Masaharu Hayashi

修士論文

分散知識サービス環境に関する研究

指導教官 吉田 武稔 教授

北陸先端科学技術大学院大学
知識科学研究科知識社会システム学専攻

250050 林 正治

審査委員: 吉田 武稔 教授(主査)

宮田 一乗 教授

野口 尚孝 教授

藤波 努 助教授

2004 年 2 月

目次

第 1 章 序論-----1

1.1	研究の背景-----	1
1.2	研究の目的-----	1
1.3	本論文の構成-----	2

第 2 章 本研究に関わる概念と研究の位置付け---3

2.1	Mediator-----	3
2.2	データフュージョン・メディエーター-----	3
2.3	Bean Markup Language (BML)-----	5
2.4	DAML (DFM Application Markup Language)-----	6
2.5	XML (Extensible Markup Language)-----	8
2.6	データベースへのXMLデータの格納-----	9
2.7	XML文書への問い合わせ-----	11
2.8	電子カルテ-----	12
2.9	HL7 (Health Level Seven)-----	13
2.10	MML (Medical Markup Language)-----	13
2.11	研究の位置付け-----	14

第 3 章	電子カルテ	15
3.1	電子カルテとは	15
3.2	電子カルテに求められているもの	16
3.3	電子カルテ・システムプロトタイプの開発	17
3.4	開発方針	17
3.5	開発環境とシステム構成	18
3.6	DBServlet	19
3.7	知的活動支援システムとしての電子カルテ	19
第 4 章	分散知識サービス環境	21
4.1	従来のアプリケーションとデータベースの関係	21
4.2	システムへの要件	22
4.3	Bridge サーバ	23
4.4	XML 文書にマッピング	24
4.5	認証	27
4.6	動作実験	27
第 5 章	結論	30
5.1	本研究のまとめ	30
5.2	今後の課題	30
謝辞		31
参考文献		32
付録 A	「電子カルテ・システム ソースコード抜粋」	35

図目次

2-1	データフュージョン・メディエーター-----	2
2-2	Document Object Model-----	9
3-1	電子カルテ・システム構成図-----	18
3-2	電子カルテ・システム-----	20
4-1	データベースの修正に対応する(従来のやり方)-----	22
4-2	データベースの修正に対応する(Bridge サーバあり)-----	23
4-3	データベース構造を XML 文書にマッピング-----	26
4-4	電子カルテ・システム起動時-----	27
4-5	電子カルテ・システム 問い合わせ時-----	28

表目次

2-1	BML 言語の要素-----	6
4-1	データベースの項目の例-----	24
4-2	要素一覧-----	25
4-3	属性一覧-----	26
4-4	USER テーブル-----	27
4-5	picharge テーブル-----	29
4-6	picharge テーブルの属性-----	29

第1章

序論

1.1 研究の背景

本研究では、人間の意思決定などの知的活動を支援する情報システムを研究開発することを最終目標としている。しかし、なにが知的活動に必要とされているのか、知的活動を支援するにはどのような問題が考えられるのかなどの、知的活動を支援する情報システムの要件が明らかになっていない。

そこで、本研究では医師が行う知的活動の1つである、診療行為に注目した。医師の診療行為とは、高度な教育を受けた医師が患者と向き合い、必要となる検査データ、医療情報を統合し、医師の知識を用いることで、病気を治そうとする一連のプロセスである。その一連のプロセスを支援しているツールに、患者の情報を記録したカルテがある。医師はこのカルテに記述されている患者についての医療情報に、医師の知識を用いることで、診療をおこなう。この医師の知的活動を支援するツールであるカルテを情報システムとしたものが電子カルテであり、電子カルテは知的活動を支援する情報システムの1つであると言える。

1.2 研究の目的

本研究では、人間の知的活動として医師の診療行為に着目し、それを支援する情報システム、電子カルテを例題として取り上げる。その中で人間の知的活動を支援する情報システムについて考察し、人間の知的活動に必要な機能、問題点を明確にすることを目的とする。

1.3 本論文の構成

本論文は、序論である本章を含め全部で 5 つの章によって構成される。第 2 章では、本研究に関わる概念を述べた後に研究の位置づけを示す。第 3 章では、一般的な電子カルテについて説明したあと、電子カルテ・システムのプロトタイプの開発について述べ、その考察を行う。第 4 章では、第 3 章で、判明した問題の解決手法について検証を行う。最後に、第 5 章で本研究の研究成果をまとめるとともに、今後の研究課題について述べる。

第 2 章

本研究に関わる概念と研究の位置付け

本章では、まず本研究に関わる概念とその関連技術について説明した後、本研究の位置付けを説明する。

2.1 メディエーター (Mediator)

メディエーターの役割は、個々のユーザやアプリケーションに対して、それらの要請する情報を、ヘテロな情報源から収集し、それらの解釈可能なデータ形式へと融合させる仲介機構である⁽¹⁾。ヘテロな情報源にある情報は、個々のユーザやアプリケーションにとって、必ずしも理解可能なものではなく、欲しい情報形式になっているとは限らない。メディエーターは、このようなデータベースの View のような概念をもっている。

2.2 データフュージョン・メディエーター

データフュージョン・メディエーター(以降、DFM)は、メディエーターの概念と、分散オブジェクト単位のセキュリティ機構を備えたサーバ・システムである。DFM では、IBM 社の BML(Bean Markup Language)を拡張した DAML を実装し、オブジェクトごとにアクセス許可の定義を行うことにより、メディエーターの概念のユーザに応じて異なるデータ融合を図る機能を実現している。DFM は、WWW(World Wide Web)上に偏在する JavaBean あるいは Java クラスをコンポーネントとして組み立てられた Java アプリケーションをクライアントである Web ブラウザ上で実行する。

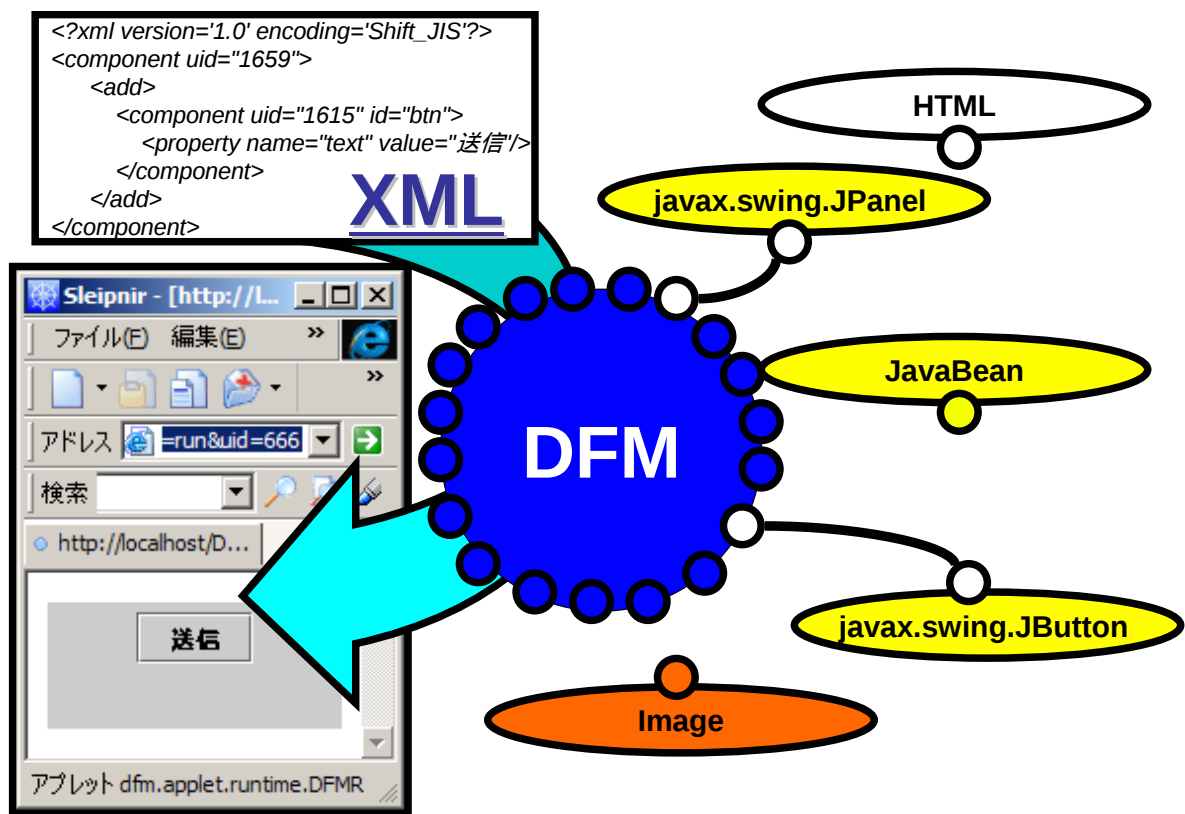


図 2-1. データフュージョン・メディエーター

分散オブジェクトとは、これらのコンポーネントを含む WWW 上に偏在する HTML(Hyper Text Markup Language)文書等の情報である。また、分散オブジェクトを統合することで作成されたアプリケーションも分散オブジェクトとして利用することが可能である。DFM はセキュリティ管理機構を実現する分散オブジェクト管理機構を備え、分散オブジェクトを統合するアプリケーション記述を解釈実行するランタイムをアプレットとしてクライアントに提供する機能をもつ。DFM は、メディエーター機構として、2 つの機能を有する。1 つ目は、分散オブジェクトあるいは分散オブジェクトを構成する要素が、その値の獲得のために認証を必要とする、あるいは、ユーザの WWW 上のアクセスするドメインによって利用を制限されているなどの際に、アプリケーション実行の不許可、あるいは、実行可能性を保証するために代替の値を提供する機能である。2 つ目の機能は、ユーザやユーザの環境からの要請で意図的に分散オブジェクトの値の獲得先を指定することが可能であることである。DFM では分散オブジェクトを location ファイルで管理しており、それらにはユニークな ID が付与されている。新たな分散オブジェクトは location ファイルに書き加えられていくことで、DFM で扱うことが可能となる。

DFM の特徴を、以下にまとめる。

- ・ 分散オブジェクトを統合して Java Applet として実行させる
- ・ 分散オブジェクト単位でセキュリティ機構を備えている
- ・ XML アプリケーションである

DFM サーバ上で動作するアプリケーション(以降、DFM アプリケーション)は、DAML という IBM の XML ベースの BML を拡張した言語で記述されている。DFM はこの DFM アプリケーションをクライアントに提供する。

2.3 Bean Markup Language(BML)

Bean Markup Language(以降、BML) ⁽¹⁵⁾⁽¹⁶⁾は、IBM 社の alphaWorks によって開発された、JavaBean を組み合わせてアプリケーションを記述するための XML アプリケーションである。BML には、BML ファイルを Java コードに変換する BML コンパイラと、BML ファイルを読み込み、動的にインスタンス化する BML プレイヤーが存在する。BML プレイヤーは、XML パーサを使って、BML 文書を読み込み、リフレクションを用いて実行時にメソッドの特定を行うため、ある程度の時間的なオーバーヘッドを伴う。BML には、JavaBean を相互接続するために、Java Bean を階層的にインスタンス化する機能、イベントを通じて JavaBean を接続する機能、フィールド/プロパティ値を割り当てる機能、JavaBean のメソッドを呼び出す機能を備えている。しかし、BML は新しい Java クラスを作成するためのツールではないため、JavaBean が最初から存在する必要がある。表 2-1 は BML 言語の要素の一覧である。BML ではこれらの要素で、JavaBean を組み合わせたアプリケーション開発を実現している。

表 2-1 BML 言語の要素

要素	説明
<bean>	新しい bean を作成し、検索する
<args>	コンストラクターの引数を指定する
<string>	新しい文字列を作成し、検索する
<property>	Bean のプロパティを設定し、検索する
<field>	Bean のフィールドを設定し、検索する
<event-binding>	1つの Bean から別の Bean にイベントをバインドする
<call-method>	Bean のメソッドを呼び出す
<cast>	Bean のタイプを変換する
<add>	1つの Bean に別の Bean を追加し Bean の階層を作成する
<script>	BML スクリプトまたは他のスクリプトを作成する

2.4 DAML (DFM Application Markup Language)

先で述べたように、DFM アプリケーションは、DAML で記述されている。DAML は分散オブジェクトを連結させて動作させるための言語である。DFM で管理されている分散オブジェクトには UID という固有の数字を付与されている。DAML ではその UID を指定して、DFM アプリケーションを記述していく。

次に、DAML で記述されたボタン追加の例を示す。

```
<add>
  <component uid="17" id="btn">
    <property name="text">OK</property>
  </component>
</add>
```

上の例にある UID="17"とは、Java での `java.awt.Button` を表している。この例を Java で書き直すと以下ようになる。

```
java.awt.Button btn = new java.awt.Button("OK");
add(btn);
```

コンポーネントへアクセスするには、JavaBean と同様にプロパティを操作する。ボタンを押した時などに発生するイベント処理には、event-binding を用いて、以下のように記述する。

```
<add>
  <component uid="17" id="btn">
    <event-binding name="action">
      <script>
        ~ ボタンが押されたときの処理 ~
      </script>
    </event-binding>
  </component>
</add>
```

次のように、直接コンポーネントのメソッドを、call-method を使って呼ぶことも可能である。

```
<call-method target="testcomp" name="sendMail"/>
```

2.5 XML(Extensible Markup Language)

Web の表記言語として一般的な HTML は、Web ページ表記のための言語であり、データの抽出には向かない。そこで、データに意味づけが可能である XML に注目が集まってきている。タグを用いて情報に意味づけをおこなう HTML、XML はマークアップ言語と呼ばれている。そのタグに囲まれることで、情報に意味づけがおこなわれる。また、マークアップ言語は、階層構造で表現されるためプログラムで解釈しやすい言語である。

XML はユーザが独自のタグを宣言することで自由に拡張することが可能になるメタ言語である。そのため、情報交換技術として注目されている。その、XML を用いて記述されたデータを XML 文書と呼ぶ。次に、XML 文書の例を示す。

```
<?xml version="1.0" encoding="Shift_JIS"?>
```

```
<書籍情報>
```

```
  <書名>XML について</書名>
```

```
  <著者>何答太郎</著者>
```

```
  <出版社>何デモ出版</出版社>
```

```
  <定価>2500</定価>
```

```
</書籍情報>
```

様々な技術標準を定めている W3C(World Wide Web Consortium)では、XML 文書をプログラムにとって扱いやすくするために、XML 文書にアクセスするための API である DOM(Document Object Model)が規定されている。DOM は XML 文書をオブジェクトとして扱うことを実現

しており、オブジェクト指向言語との相性がよいため、プログラムは容易に XML 文書にアクセスすることが可能となっている。

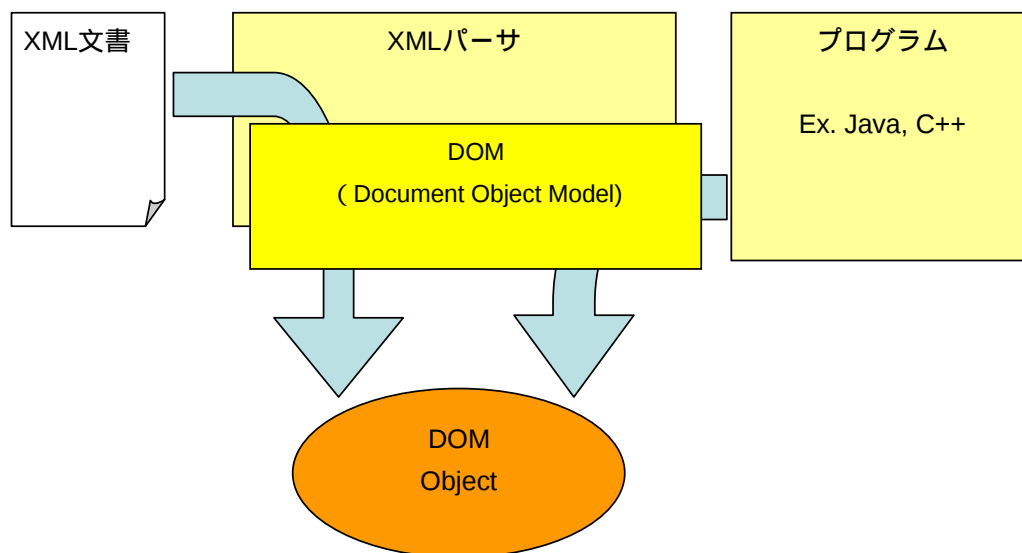


図 2-2 Document Object Model

2.6 データベースへの XML 文書の格納

前節でも述べたが、XML はユーザが独自のタグを宣言することで自由に拡張することが可能になるメタ言語である。この XML を用いてさまざまな情報交換がおこなわれている。また、そのような中で XML 文書をデータベースに格納したいという要求も出てきている。その XML 文書をデータベースに格納する際に一番問題となるのは XML の利点でもあるデータ構造が一意に定まっていないという点である。

このような XML 文書をデータベースに格納するのに用いられる手法は大まか分けるとファイル格納、テーブル格納、木構造格納の 3 種類に分類される。

・ ファイル格納

ファイル格納手法は、XML 文書をファイル形式そのまま格納する手法である。この手法では、X

ML 文書構造をそのまま保存することはできるが、ファイル形式のままデータベースに格納するため、目的の XML 文書を探すことが困難である。その為、インデックスを用意するなどして、高速化を図っている。

・ テーブル格納

テーブル格納手法は、XML 文書をデータベースにマッピングして格納する手法である。XML 文書を構造化データと見なし、データベースに格納することにより、高速な検索を行うことを目的としている。データベースのマッピングには XML の文書構造を定義するスキーマ言語である DTD(Document Type Definition)を利用するなどして、オブジェクトリレーショナルデータベースのテーブルに XML の要素をマッピングして格納する手法などが研究されている⁽⁵⁾。

・ 木構造格納

木構造格納手法は、XML 文書の木構造を保持したままデータベースに格納する手法である⁽⁴⁾。XML 文書の木構造をノード単位に分けてデータベースに格納し、そのノード間のリンク情報も格納する。木構造そのまま格納するため、どのような XML 文書でも格納することが出来る。XML 文書を木構造で格納すると、木構造を辿るように XML 文書を検索することが可能である。

本研究では、テーブル格納手法を用いて XML 文書にリレーショナルデータベースの構造をマッピングし、またその逆もおこなっている。

2.7 XML 文書への問い合わせ

XML の応用が進むにつれ、膨大な量の XML 文書から必要な要素を検索したいという要求が出てきている。また、リレーショナルデータベースのように SQL のような検索言語で問い合わせを行うことを目指す研究もおこなわれている⁽⁸⁾⁽⁹⁾⁽¹⁰⁾。そのような中、W3C(World Wide Web Consortium)ではその要求に応えるために XML Query⁽⁶⁾, XPath⁽⁷⁾仕様の策定を進めている。XML Query は W3C における、XML による問い合わせ言語に関する活動の名称である。そのような中、XML-QL⁽¹⁹⁾, XQuery⁽²⁰⁾などが相次いで発表されている。

・ XML-QL

XML-QL は半構造型データを扱う問い合わせ言語として開発された。XML-QL は XML の特徴であるユーザ定義可能なタグや入れ子型要素を扱うことができる宣言的問い合わせ言語である。XML-QL の特徴は、XML 文書からデータを抽出し、変数へのバインドと、全く新しい XML 文書を組み立てることが出来る点にある。また、2 つ以上の XML 文書をジョインすることも可能である。XML-QL は従来のデータベースクエリに近い考え型に XML の階層型の考えを加えたものといえる。

・ XQuery

XML-QL やその他の XML による問い合わせ言語は、W3C が "XML Query Requirements"⁽⁶⁾や "XML Query Use Cases"⁽²⁰⁾で提示した XML による問い合わせ言語に必要な項目を満たせていなかった。そのような中、W3C によって XQuery が XML の問い合わせ言語⁽¹⁹⁾として提示された。XQuery では、単にデータベースに格納された XML 文書に対して検索が行えるだけでなく、射影、選択、繰り返しといった、リレーショナルモデルのような演算も可能である。

・ XPath

XPath では、XML 文書の特定の部分を指し示す構文を規定する。XPath を利用すれば、XML 文書中の任意の位置を指し示すことができる。XPath では、XML 文書のノードを“/”で表す。ルートノードの下にある要素 book を表すには“/book”と記述する。同じように、要素 book の子要素 name を表すには、“/ book/name”と記述する。また、“//book[id="001"]”と記述すると、ある XML 文書中に存在するすべての要素 book の中で、その要素の id 属性が“001”であるものを選択する。

本研究では、XML 文書の問い合わせには XPath を用いた。

2.8 電子カルテ

電子カルテの定義については日本医療情報学会が電子カルテの定義に関する見解⁽¹¹⁾を発表しているが、まだ明確な定義は存在していない。そのため現状では、オーダエントリーシステムの延長上にあるものが多く、未だに様々な試みがおこなわれている。電子カルテのためのデータのフォーマットや通信手段の標準がまだないことも原因の1つである。そのような電子カルテの医療情報交換通信手段として有名なものにアメリカで制定された HL7 (Health Level Seven) がある。HL7 が扱う情報の範囲は広範囲に及ぶが、日本とアメリカの医療制度の違いから適用しにくくなっている。また日本では、厚生省電子カルテ研究開発事業の研究会に置いて MML (Medical Markup Language) の開発が行われている。MML の目的は医療データベース同士の情報交換をおこないやすいものにある。現在でも電子カルテ研究会の MML working group が中心となって開発を維持しているが、いまだに標準化に至っていない。医療の IT 化には、データベースとの情報交換が避けられないものとなっており、データフォーマット標準化の早期解決が期待されている。

2.9 HL7(Health Level Seven)

HL7 とは、1987 年にアメリカで設立された任意団体が規定した、医療情報交換のためのプロトコルである。HL7 は OSI 参照モデルの第7層、アプリケーション層で規定されており、HL7 という名もそこから由来している。HL7 には、交換されるデータの定義、公刊のタイミング、アプリケーション特有のエラー通信について定義している。HL7 が扱う情報の範囲は、患者管理、オーダ、照会、財務、検査報告、マスタファイル、情報管理、予約、患者紹介、患者ケア、ラボラトリオートメーション、アプリケーション管理、人事管理などである。しかし、日本とアメリカの医療制度の違いから、会計、看護オーダなど、そのまま日本で使いにくいものも多い。

2.10 MML (Medical Markup Language)

MML とは、異なる医療機関、特に電子カルテ間でのデータ交換を想定した規格である。1995 年5月に開かれた Seagaia Meeting(日本医療情報学会電子カルテ研究会年次総会)において、発案が行われ、SGML(Standard Generalized Markup Language)を用いた方法での研究を進められた。その後、1996 年度に結成された厚生省電子カルテ開発プロジェクトの研究グループに継承され発展し、現在では、MML は XML によって開発されており、電子カルテ研究会の中の MML working group が中心となって開発を継続している。MML が扱うデータの範囲は、患者、健康保険、診断履歴、生活習慣、基礎的診療、初診特有、経過記録、手術記録、臨床サマリー、検歴、報告書、紹介状など多岐にわたっている。しかし、開発が進むにつれ診療分野に特化した構造が必要となっており、XML の Namespace で MML をモジュール化して、モジュールを組み合わせて使う手法を採用した。MML はデータベースをマッピングすることで、施設毎の独立性を保ちながら、全国の医療機関とデータ交換を容易にすることを目的としている。

2.11 研究の位置付け

本研究が目指すところは、人間の意思決定などの知的活動を支援する情報システムを構築するところにある。しかし、知的活動を支援する情報システムに必要な機能、問題が不明瞭である。そこで、医学という高度な知識を持つ医師を支援する電子カルテを例に問題の抽出を図り、その問題の解決を目指す。よって、本研究の位置付けは、人間の知的活動を支援する情報システムに必要な機能の抽出、基盤の構築にある。

第 3 章

電子カルテ

前章で、本研究に関わる概念として、電子カルテを紹介したが、本章でも電子カルテについて詳しく説明した上、電子カルテ・システムプロトタイプの開発手法について述べる。

3.1 電子カルテとは

現在、電子カルテには明確な解釈が存在しないため、様々な電子カルテが乱立している状態にある。その最中、日本医療情報学会が 2003 年 2 月に「電子カルテの定義に関する日本医療情報学会の見解」⁽¹¹⁾として電子カルテの定義を発表している。その中で、電子カルテの機能の最低限の要件として、オーダ通信システム、診療情報の蓄積、不特定の大量施設との情報連携、画面表示性・操作性、電子保存の 3 条件(真正性の確保、見読性の確保、保存性の確保)を提示している。様々な企業が多種多様の電子カルテを開発しているが、現状では電子カルテはオーダリングシステムの延長上のものが多い。

このような背景には、国が医療改革を押し進めていることがある。政府・与党社会保障改革協議会が 2001 年 11 月に「医療制度改革大綱」⁽¹⁴⁾の中で、医療提供体制の改革の一環として、電子カルテ・レセプト電算化などの医療の IT 化の推進を掲げていることも一因に挙げられる。その「医療制度改革大綱」の中では、「電子カルテ等について目標と達成年次を年内に策定し、その実現に向けた支援措置を講じる」と提示している。また、同年 12 月に厚生労働省が設置した保健医療情報システム検討会がまとめた、「保健医療分野の情報化にむけてのグランドデザイン」⁽¹²⁾の最終提言の中でも、電子カルテについて「平成 16 年度までに全国の 2 次医療圏毎に少なくとも一施設は電子カルテ・システムの普及を図る」、「平成 18 年度までに全国 400 床以上の病院の 6 割以上に普及、前診療所の 6 割以上に普及を図る」との達成目標を設定しており、電子カルテの普及は急務となっている。

3.2 電子カルテに求められているもの

厚生労働省の「保健医療分野の情報化にむけてのグランドデザイン」では、医療の課題として、「情報提供」、「質の向上」、「効率化」、「安全対策」の4つを掲げており、その解決手法に、電子カルテ・システム、レセプト電算処理システム、「根拠に基づく医療(Evidence based Medicine)」支援、遠隔診療支援、オーダリングシステム、個人・資格認証システム、電子商取引を提案している。また、電子カルテ・システムの効果として、「比較可能なデータの蓄積と活用」、「見やすく読みやすく分かりやすい情報」、「医療従事者間での情報提供や診療連携」、「患者の診療データの一元管理・共有化・情報の解析等による新たな臨床上の根拠の創出」、「コスト削減」を要求している。医療情報を扱うので、プライバシーの保護などセキュリティにも注意しなければならないとしている。

この様に、電子カルテには次の条件が要求されている。

- ・ 医療情報の蓄積と活用
- ・ 紙よりも使いやすい情報源
- ・ 他診療施設との連携
- ・ EBM のような診療支援
- ・ コストの削減
- ・ セキュリティ(真正性、見読性、保存性)

要求されている他診療施設との連携には、情報交換には用語の統一が必要になる。現在、医療情報交換技術には HL7 や、MML などがある。HL7 とは、1987 年にアメリカで設立された任意団体が規定した、医療情報交換のためのプロトコルだが、アメリカで規定されたものであり、現在も日本支部で改訂中である。また MML とは、異なる医療機関、特に電子カルテ間でのデータ交換を想定した規格である。1995 年 5 月に開かれた Seagaia Meeting(日本医療情報学会電子カルテ研究会年次総会)において、発案が行われた。はじめは SGML 技術を取り入れていたが、XML 技術を用いたものに変更された。その MML はデータベースをマッピングすることで、施設毎の独立性を保ちながら、全国の医療機関とデータ交換を容易にすることを目的としている。しかし、

どちらも未だ標準化には至っておらず、今も開発が続けられている。

3.3 電子カルテ・システムプロトタイプの開発

本研究では人間の知的活動を支援するシステムに必要な機能、問題点を明確にするために、実際に電子カルテ・システムのプロトタイプ(以下、電子カルテ・システム)を構築する。電子カルテ・システムを開発する工程での、特徴、問題の抽出を行えると考えたためである。電子カルテ・システムには、石川県地域結集型共同研究事業 次世代型脳機能計測・診断支援技術の開発、医用ナレッジ・ハンドリング技術の開発の研究成果として開発された、DFM を用いる。

3.4 開発方針

電子カルテ・システムは、以下のことを目標に開発した。

- ・ DFM サーバ上で動作する電子カルテ・システム
- ・ 複数のユーザに対応し、ユーザグループごとに動作変更可能
- ・ 電子カルテ表示部は紙のカルテのように表示

電子カルテ・システムは DFM サーバ上で動作するアプリケーションとして構築する。電子カルテ・システムが動作する DFM サーバは、WWW 上に散らばった画像やテキストなどのリソースや Java クラス、JavaBean などの分散オブジェクトを統合して、Java アプリケーションとして、クライアントに提供するサーバシステムである。

電子カルテ・システムでは、医療情報データベースや電子カルテ・コンポーネントが分散オブジェクトにあたる。また、DFM にはセキュリティ機構が備わっており、分散オブジェクトに対してアクセス制限をかけることが可能である。特徴的なのが DFM アプリケーション内にもアクセス制限をかけることができる点である。これにより、同じ DFM アプリケーションでも、アクセス制限により挙動

を変える事が可能である。DFMサーバは動的に分散オブジェクトからDFMアプリケーションを生成するため、この機能によりセキュリティは高いものとなる。これらDFMの機能を利用して、電子カルテ・システムの開発を行った。

3.5 開発環境とシステム構成

電子カルテ・システムのコアプログラムの開発にはJava™ 2 SDK Standard Edition version.1.4.2を用いた。

DFMは、Tomcat上で動作するWebアプリケーションであるため、サーバマシンにはApache HTTP Server version.2.0.48、Tomcat version.4.1を導入した。また、電子カルテ情報を扱う医療情報データベースには、MySQL version. 4.0.16を用いた。

図3-1はこのシステム構成のイメージ図である。

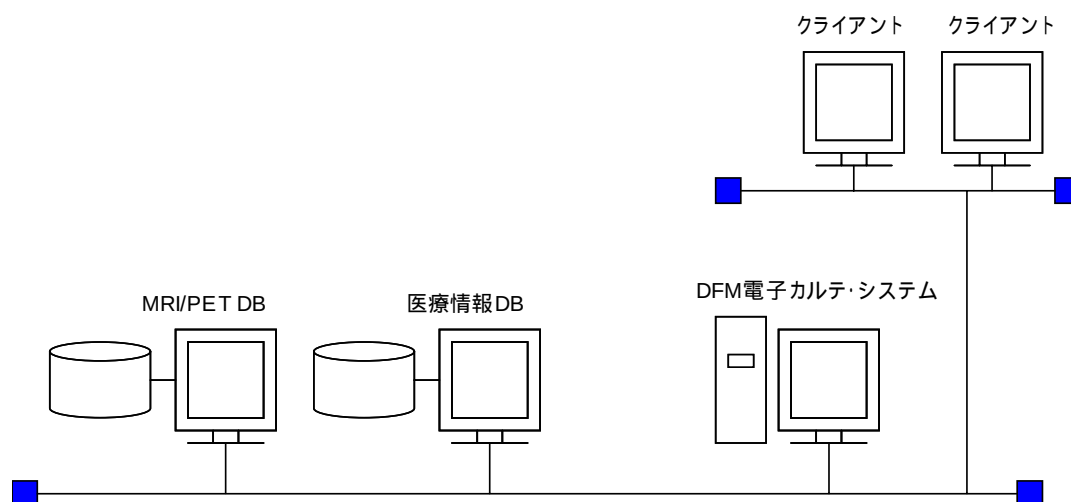


図 3-1 . 電子カルテ・システム構成図

DFMの特徴の1つとして、DFMアプリケーションをJava Appletとして動作させることが挙げられる。通常Java Appletは、データベースに接続して問い合わせを行うことは困難であるため、データベースとの接続にJava ServletのDBServletを用意した。また、電子カルテの表示には

JSP(Java Server Pages)を用いた。

本電子カルテ・システムは、データベース設計、システム設計、プログラミングの順で開発を行った。

3.6 DBServlet

電子カルテ・システムでは、データベースとの接続に DBServlet を用いている。電子カルテ・システムは HTTP ストリーム内のパラメータセットを使って DBServlet に SQL 文を送信する。DBServlet は HTTP ストリーム内のパラメータを読み取り、SQL 文を実行、その問い合わせ結果を、HTTP ストリームを使って送信する。

3.7 知的活動支援システムとしての電子カルテ

電子カルテ・システムの開発を通じて明らかになった、電子カルテの問題点を以下にまとめる。

- ・ **入力方式**
- ・ **カルテの枠組みが固定である**

電子カルテ・システム開発を通じて判明した問題点として、第 1 に入力方式が挙げられる。これは、キーボード、マウス、などのインターフェイス周りの問題と、アプリケーション側の問題があるが、今回の電子カルテ・システムでは考慮しなかった。

第 2 に、カルテの枠組みは固定であるということである。今回のプロトタイプを含め、一般的な電子カルテ・システムは、データベースを構築後にアプリケーションを開発しており、データベースに依存した形となっている。そのような状態の打開策として、MML などの標準化が進められているのである。

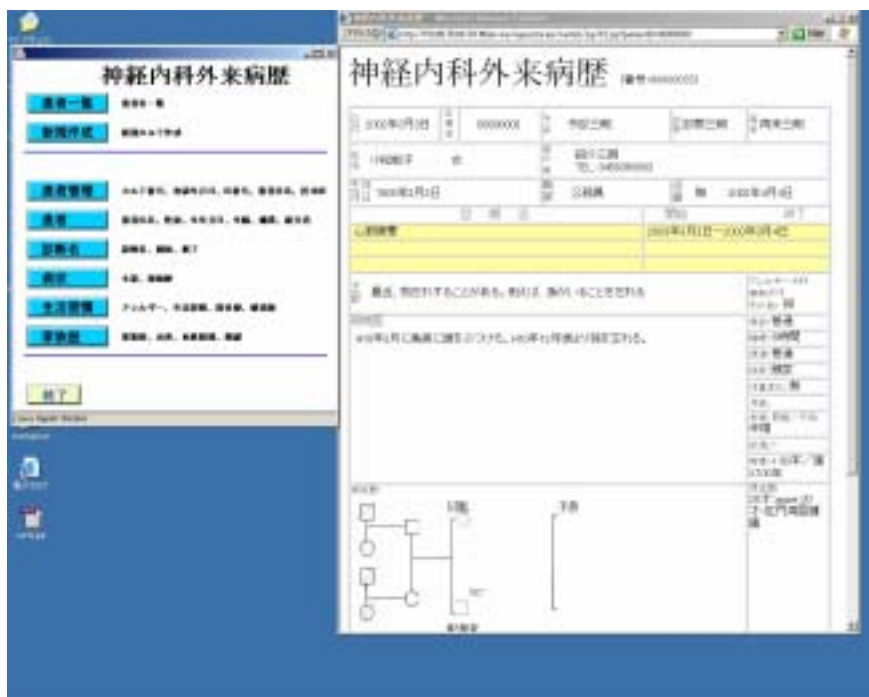


図 3-2 . 電子カルテ・システム

医師の知的活動は、電子カルテから必要な情報を得て、医師の持つ知識から、診療・治療を行い、診断結果を電子カルテに記録していく、診療行為のことであり、その知的活動を支援しているのは電子カルテである。医師の知識は電子カルテのフォーマットに格納されていくが、一般的な人間の知的活動を支援する情報システムとして考えた場合では、ダイナミックに変化する知識をそこに格納する必要があるためフォーマットは固定ではない。つまり、カルテの枠組みは固定ではないのである。

このダイナミックに変化する知識を格納するということを、情報システムとして考えると、問題となってくるのは、データベース構造とアプリケーションのダイナミックな関係を維持することである。

第 4 章

分散知識サービス環境

前章で、知的活動を支援するシステムの条件として、データベースにダイナミックに変化する知識を格納する必要性を示した。この要件を情報システムとして考えてみると、データベース構造とアプリケーションのダイナミックな関係維持を達成する必要性があると置き換えることができる。

本章では、データベース構造とアプリケーションのダイナミックな関係維持を達成するための情報システムの仕組みについて述べ、この仕組みを分散知識サービス環境と呼ぶことにする。

4.1 従来のアプリケーションとデータベースの関係

従来のデータベースを利用したアプリケーションでは、データベース設計を行ってデータ構造を決定した後、アプリケーションの開発を行っていた。その為、アプリケーションとデータベースは密な関係にあり、データベースに修正を加えるには、一度サービス全体を停止させてから修正を行う必要があった。

図 4-1 は従来のアプリケーションとデータベースの関係を表している。従来の関係は、データベースのアプリケーションは密な関係であり、データベースの項目の変更や、移動などの修正には、アプリケーション側との打ち合わせが必要であった。

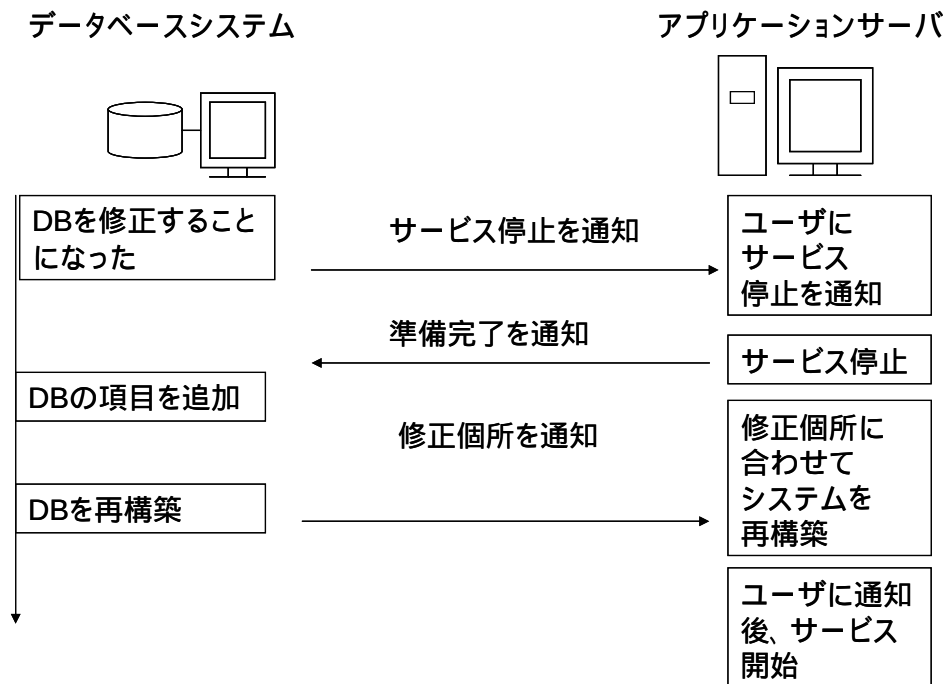


図 4-1. データベースの修正に対応する(従来のやり方)

4.2 システムへの要件

前節で述べた従来の手法では、知識のダイナミックな変化に対応するアプリケーションの構築には問題が存在する。知識の変化に対応するには、まず、データベースとアプリケーションの関係を疎にする必要性が出てくる。何故ならば、データベースとアプリケーションが密着した状態では、データベースの修正、新規作成、物理的な場所の移動などの変化に対応することに無理が生じるからである。データベースとアプリケーションが互いに独立した関係になれば、データベースの修正に対して、アプリケーションが混乱に陥ることは少なくなると考えられる。アプリケーションとデータベースの互いの関係が疎であるほど、アプリケーションに影響を与えにくくなる。だが、互いが疎の関係ではアプリケーションとデータベースの連携は成り立たなくなってしまう。つまり、何らかの仲介役が必要になる。その仲介役を通してアプリケーションとデータベースの情報を交換すればよい。

4.3 Bridge サーバ

本研究ではアプリケーションとデータベースの仲介役として、Bridge サーバを開発した。また、データベースの修正に対応するために、データベース構造を XML 文書にマッピングすることで問題の解決を図った。Bridge サーバは、データベース構造を XML 文書へマッピングする機能の他に、認証機能を実現している。

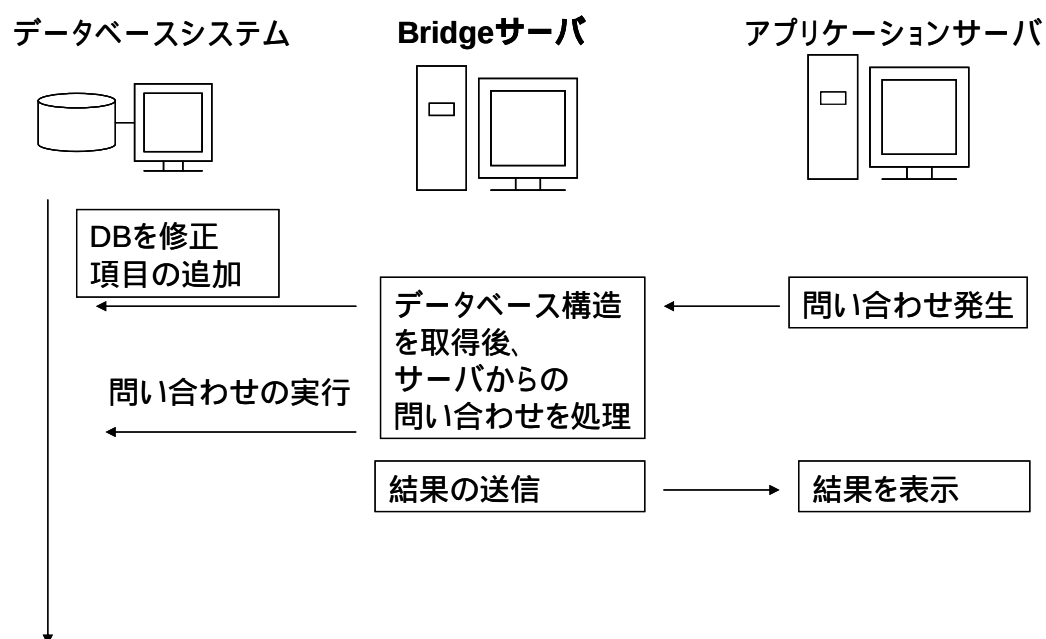


図 4-2. データベースの修正に対応する(Bridge サーバあり)

図 4 - 2 は、Bridge サーバを介したデータベースの修正について表した図である。この場合は、データベースに修正を行うのに、従来のやり方(図 4-1 参照)とは違い、アプリケーション側に通知を行わなくても良い。Bridge サーバが仲介役となって、データベースへの処理を行うことで、それを可能としている。また、マッピング後の XML 文書から、データを取得するには XPath を利用している。Bridge サーバでは、アプリケーションからの接続が起きたときに、アプリケーションの認証を行っている。その後、そのアプリケーションに許可されたデータベースの構造を取得し、XML 文書として、アプリケーションに通知する。アプリケーションからの問い合わせには、Bridge サーバがアプリケーションに許可したデータベースに問い合わせを行った後、XML 文書として、アプリ

ケーションに通知する。アプリケーションはその XML 文書から XPath を用いて、データを取得する。Bridge サーバが、データベースのクッション役として動作することで、アプリケーションとデータベースのダイナミックな関係を維持している。

4.4 XML 文書にマッピング

データベースの項目には、少なくとも項目名、データタイプ、データサイズが定義されている。データベースの修正では、これらの項目のサイズ変更や、データタイプの変更などが行われる。また、項目が定義されている表名が変更されることも考えられる。データベースの追加では、新たに表、項目がデータベースに追加されることが考えられる。

表 4-1 . データベースの項目の例

項目名	データタイプ	データサイズ
NAME	VARCHAR	128

アプリケーションがデータベースの修正にダイナミックに対応するには、何らかの方法でデータベースの構造、項目の定義をアプリケーションに伝える必要がある。表 4-1 のデータベースの項目の例の様に、それぞれの項目には、名前、データタイプ、データサイズなどの属性が存在する。

このようなデータベースの構造を、独自の形式を用いてアプリケーションに伝える手法も考えられる。しかし、本研究ではデータのマッピングや、データベースのマッピングによく使われる XML を用いることにする。XML を用いていれば今後 MML などが標準化に至ったときにも、インターフェイスを作成することなどで対応できると考えたからである。

```

<?xml version="1.0" encoding="UTF-8"?>

<dataroot xmlns:od="urn:schemas-microsoft-com:officedata">

<employee>

    <id>0</id>

    <name>masa</name>

</employee>

</dataroot>

```

上記の XML 文書は、Microsoft Access で、テーブル employee を XML 形式で出力したものである。リレーショナルデータベースを XML 文書としてマッピングする時には、XML の要素名としてデータベースの項目を使うことが多い。しかし、本研究ではデータベースの構造を XML 文書としてマッピングすることを目指しているため、以下の形式をとることにした。要素数を少なくすることで、XML 文書の軽量化を図った。

```

<?xml version="1.0" encoding="shift_jis"?>

<tables>

<table name="表名">

<column class="Java クラス" name="列名" size="列サイズ" type="列タイプ">データ</column>

</table>

</tables>

```

表 4-2. 要素一覧

要素名	意味
Tables	root 要素
Table	表、複数個存在してもよい
Column	列、複数個存在してもよい

表 4-3. 属性一覧

属性名	意味
Name	名前
Class	Java のクラス名
Size	列が格納できるサイズ
Type	SQL でのデータタイプ

表 4-2、表 4-3 の様に、データベースの構造を表すことで、Bridge サーバはデータベース構造を XML 形式へとマッピングし、システムに伝えることができる。つまり、システムはデータベースの構造を常に知ることが可能となるのである。これにより、アプリケーションはデータベースの構造を気にせずにデータベースを扱う事が可能となる。

pipatient

patientID	name	sex	birthday	occupation
0000001	吉田恋子	0	1930年1月1日	無職

```
<table name="pipatient">
  <column class="java.lang.String" name="PatientID" size="8" table=""
    type="VARCHAR">00000001</column>
  <column class="java.lang.String" name="Name" size="64" table="" type="VARCHAR">吉田恋子</column>
  <column class="java.lang.String" name="Sex" size="8" table="" type="VARCHAR">0</column>
  <column class="java.lang.String" name="Birthday" size="64" table="" type="VARCHAR">1930年1月1日
    </column>
  <column class="java.lang.String" name="Occupation" size="64" table="" type="VARCHAR">無職</column>
</table>
```

図 4-3 データベースの構造を XML 文書にマッピング

図 4-3 では、実際に Bridge サーバで、テーブル pipatient を XML 文書にマッピングした時の例である。この様に XML はデータ構造を表すことが出来るメタ言語の性質をもつ。また、XML はマークアップ言語であるため、コンピュータにとって操作しやすい、そのため編集も容易である。

4.5 認証

Bridge サブレットでは、ユーザ管理データベースを独自に持っている。ユーザ管理データベースは、以下のようなテーブルをもつ。

表 4-4. USER テーブル

ID	NAME	PASSWD	ADMITDB
----	------	--------	---------

アプリケーションからの問い合わせが発生すると、Bridge サーバはユーザ名、パスワードでアプリケーションの認証を行う。承認されたアプリケーションは、許可されたデータベースへのアクセス権限を得る。

4.6 動作実験

Bridge サーバを実際に、電子カルテ・システムに組み込んで動作実験を行った。システム構成の変更は、前電子カルテ・システムに Bridge サーバを加えただけである。電子カルテ・システムに Bridge サーバを組み込んだ時の、Bridge サーバの動きは以下の図の通りである。

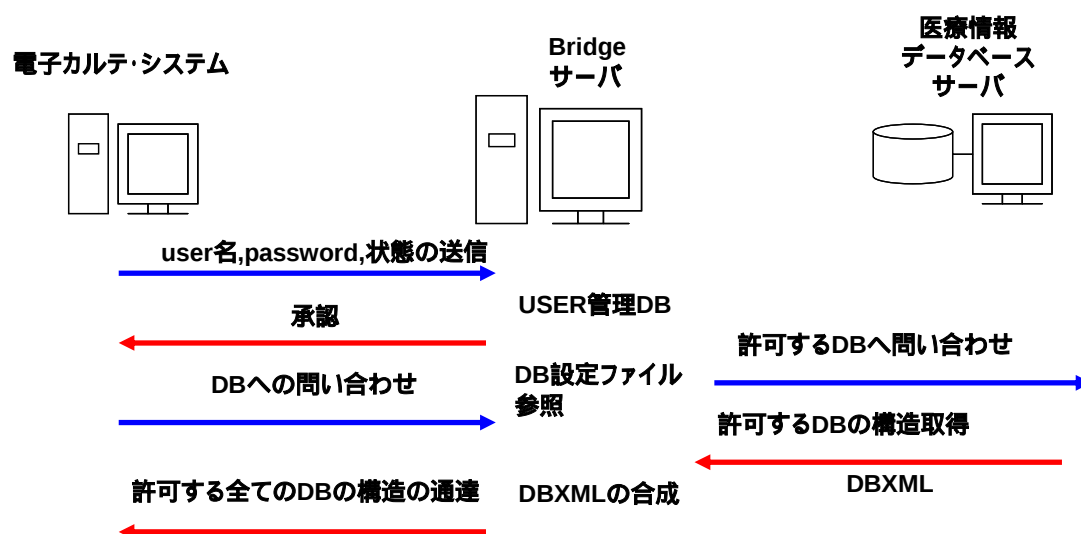


図 4-4 電子カルテ・システム起動時

図 4-4 は、電子カルテ・システムを起動したときのデータベース、電子カルテ・システム、Bridge サーバ間のデータ交換を表したものである。

次の図 4-5 は、同じく電子カルテ・システムから問い合わせが発生した場合を表したものである。

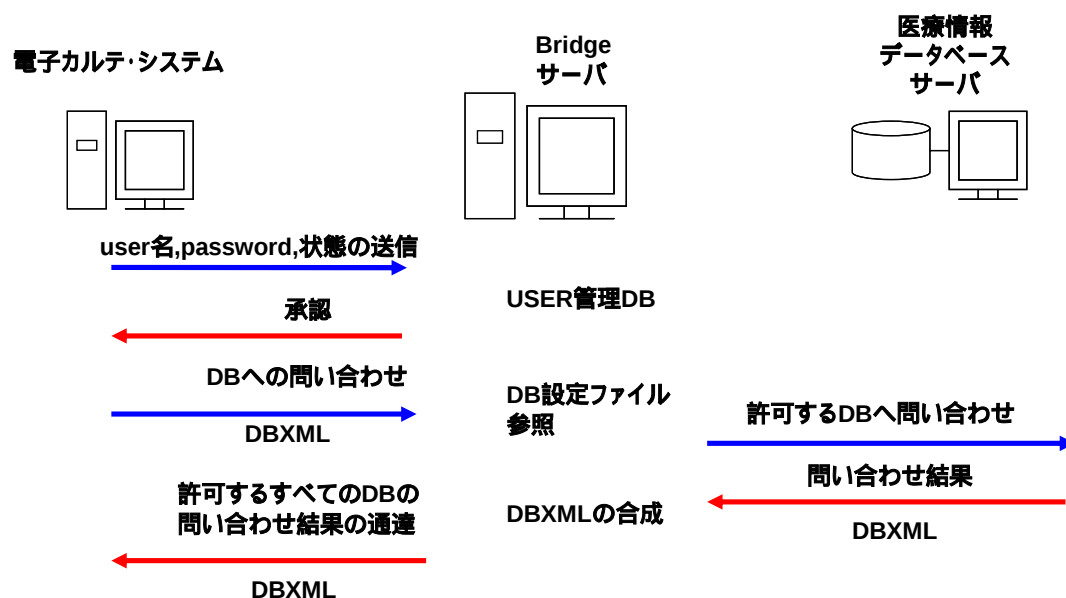


図 4-5 電子カルテ・システム 問い合わせ時

動作実験は、下記の 4 パターンを用いて、Bridge サーバの動作を確かめた。

- ・ テーブルに列を追加した場合
- ・ テーブルの列の属性を変更した場合
- ・ テーブルを別データベースに移動させた場合
- ・ テーブルの列を削除した場合

実験環境のデータベース carte には、picharge , picontrol ,picrinalhistory , pifmaily, piintroducer,pilife,pimaincomplaint,pimri,pipatient,pipet の 10 個のテーブルが定義されている。今回は、テーブル picharge に上記の 4 パターンの修正を加えることを試みた。

表 4-5. picharge テーブル

PatientID	MainDoctor	FirstDoctor	SubDoctor
-----------	------------	-------------	-----------

表 4-6. picharge テーブルの属性

項目名	データサイズ	データタイプ
PatientID	8	VARCHAR
MainDoctor	128	VARCHAR
FirstDoctor	128	VARCHAR
SubDoctor	128	VARCHAR

表 4-5,4-6 は、picharge テーブルを構成する項目とその属性を表している。

動作実験は、以下の結果となった。

第 1 のパターン、「テーブルに列を追加した場合」では、この picharge テーブルに項目の追加を試みた。電子カルテ・システム動作中に追加を試みたが何事もなく動作していた。第 2 のパターン、「テーブルの列の属性を変更した場合」では、MainDoctor のデータタイプ VARCHAR 型を int 型に変更した。このデータベースの修正にも電子カルテ・システムは対応出来ていた。第 3 のパターン、「テーブルを別のデータベースへ移動させた場合」では、データベース carte2 に picharge テーブルを移動させた場合においても、電子カルテ・システムに問題は生じなかった。最後のパターン、「テーブルの列を削除した場合」では、picharge テーブルから PatientID の削除を試みたところ、電子カルテ・システムに障害が起きた。このような電子カルテ・システムにとって重要な項目を消してしまった場合に問題が起こることが判明した。

以上の結果より、Bridge サーバは完全にデータベースとの関係を疎にするという目標は達成できなかったが、ある程度、アプリケーションとデータベースのダイナミックな関係は維持できたという結論に至った。

第 5 章

結論

5.1 本研究のまとめ

本研究では、知的活動を支援するシステムについて、必要な機能、問題、特徴を明確にするために、電子カルテのプロトタイプを作成することで、考察を行った。電子カルテ・システムとしては、プロトタイプではあるが、今後の開発に向けて、医師より前向きな意見を頂いた。また、電子カルテ・システムより、知的活動を支援する情報システムの要件の1つを抽出することができた。そして、そのデータベース構造をアプリケーションのダイナミックな関係維持を達成するための情報システムとして分散知識サービス環境を提案した。これにより、人間の知的活動を支援するシステムの要件のデータベース構造とアプリケーションのダイナミックな関係維持を達成に有効性を示せたといえる。

5.2 今後の課題

今回の研究は、知的活動を支援するシステムに必要な機能、特徴を抽出し、知的活動を支援するシステムの要件である、データベース構造とアプリケーションのダイナミックな関係維持を達成する為の情報システムの基盤を開発した。しかし、要件を1つ解決したのみで、知的活動を支援するシステムを示したとはいえない。ナレッジマネジメントツールの現状を調査して、知的活動を支援するツールについて考察していく必要がある。今後は、より知的活動について研究を行い、人間の知的活動を支援するより汎用的なツールの研究開発を目指したい。

謝辞

本研究をおこなうにあたって、主指導教官である吉田武稔教授には、適切なご指導や助言を頂いたのみならず、様々な研究活動のチャンスを与えて頂いたことに深く感謝致しております。

中間審査では、野口尚孝教授、林幸雄助教授、藤波努助教授に、ご指導や助言を頂いたことを深く感謝致しております。また、後期課程進学審査では、中森義輝教授、橋本敬助教授、佐藤賢二助教授にも、ご指導や助言を頂いたことを深く感謝致しております。

また、システム実装にあたっては、株式会社 PFU の東田雅宏氏、牧野倫幸氏には多大なる技術指導、助言を頂いたことに深く感謝致します。

また、本研究を暖かく見守って頂いた研究室のメンバー、また学業に限らず、様々な相談に乗って頂いた学友に感謝いたします。

最後に、遠くより暖かい支援を送って頂いた、家族、祖父母、親戚に感謝いたします。

参考文献

- (1) Y.Papakonstantinou, S.Abiteboul, H.Garcia-Molina : “Object Fusion in Mediator System”, Proceedings of the Twenty-second International Conference on VeryLargeDatabases,pp412,1996
- (2) T.Shimanaka, I.Mejuev, M.Higashida, N.Makino: “Visual Environment for Distrivuted Content Integration”, Software Symoposium 2000, pp16-19,2000(in Japanese)

嶋中徹, I.Mejuev, 東田雅宏, 牧野倫幸: 「VEDICI 分散コンポーネントを用いたアプリケーション構築基盤の開発」, Software Symposium 2000, pp16-19, 2000
- (3) 里村陽一: 「電子カルテが医療を変える 改訂版」、日経 BP 社、2003,ISBN4-8222-8173-6
- (4) 金政泰彦、久保田和己、石川博: 「XML 問い合わせ処理システム(xQues)のデータ格納管理」、情報処理学会 データベースシステム、 pp243-248,119-41,1999
- (5) 西岡秀一、鬼塚誠: 「XML のオブジェクトリレーショナルマッピングに関する一手法」、情報処理学会 データベースシステム,pp17-24,122-3,2000
- (6) World Wide Web Consortium: “XML Query (XQuery) Requirements W3C Working Draft 12 November 2003”,
<http://www.w3.org/TR/xquery-requirements/>,2003

- (7) World Wide Web Consortium: “XML Path Language (XPath) Version 1.0 W3C Recommendation 16 November 1999”,
<http://www.w3.org/TR/xpath> ,1999
- (8) 久保田和己、金政泰彦、石川博:「XML 問い合わせ処理システム(xQues)の問い合わせ処理系」、情報処理学会 データベースシステム、pp25-30, 119-5,1999
- (9) 石川博、久保田和己、金政泰彦:「XML 問い合わせ処理システム(xQues)の問い合わせ言語」、情報処理学会 データベースシステム、pp19-23,119-4, 1999
- (10) 絹谷弘子、吉川正俊、植村俊亮:「スキーマのない多様な XML 文書のリポジトリに対する問い合わせ処理について」、情報処理学会 データベースシステム、pp129-136,121-19,2000
- (11) 日本医療情報学会:「電子カルテの定義に関する日本医療情報学会の見解」,2003
- (12) 厚生労働省:「保健医療分野の情報化にむけてのグランドデザイン」,2001
- (13) Jayavel Shanmugasundaram, Kristin Tufte, Chun Zhang, Gang He, David J. DeWitt, and Jeffrey F. Naughton. *Relational databases for querying XML documents: Limitations and opportunities*. In Proceedings of 25th International Conference on Very Large Data Bases (VLDB'99), pages 79-90. Morgan Kaufmann, 1999.
- (14) 政府・与党社会保障改革協議会:「医療制度改革大綱」, 2001
- (15) Sanjiva Weerawarana and Matthew J. Duftler : Bean Markup Language (Version 2.3) User's Guide , September 22, 1999

- (16) Sanjiva Weerawarana and Matthew J. Duftler : Bean Markup Language (Version 2.3) Tutorial, September 22, 1999
- (17) World Wide Web Consortium: "XML-QL: A Query Language for XML Submission to the World Wide Web Consortium 19-August-1998", <http://www.w3.org/TR/NOTE-xml-ql/> ,1998
- (18) World Wide Web Consortium: "XQuery 1.0: An XML Query Language W3C Working Draft 12 November 2003" , <http://www.w3.org/TR/xquery/>, 2003
- (19) World Wide Web Consortium: "XML Query Use Cases W3C Working Draft 12 November 2003" , <http://www.w3.org/TR/xquery-use-cases/>, 2003

付録 A 「電子カルテ・システム ソースコード抜粋」

ファイル名: KcarteApp.xml

用途 : メイン画面

```
<?xml version="1.0" encoding="Shift_JIS"?>
<entity uid="1040" id="controlPanel">
<stub id="main">
  <attribute name="carte" component="4000"/>
  <attribute name="fail" class="boolean"/>
</stub>
  <component uid="1659" id="btnPatientAdmin"/>
  <component uid="3013" id="browser">
    <property name="menuBar" value="false"/>
    <property name="locationBar" value="false"/>
    <property name="statusBar" value="false"/>
    <property name="toolBar" value="false"/>
    <property name="width" value="760"/>
    <property name="height" value="700"/>
    <property name="resizable" value="false"/>
  </component>
  <property target="browser" name="visible" value="true"/>
  <attribute target="main" name="carte">
    <component uid="4000" id="carteObj"/>
  </attribute>
  <attribute target="main" name="fail">
    <cast class="boolean" value="true"/>
  </attribute>
  <property target="carteObj" name="patientID">
    <string value="null"/>
  </property>
  <property name="locationJSP" target="carteObj">
    <location uid="280"/>
  </property>
  <property name="urlOfDBServlet" target="carteObj">
    <location uid="1300"/>
  </property>
  <property name="urlImageFamilyChart" target="carteObj">
    <location uid="1301"/>
  </property>
  <property name="uriBridge" target="carteObj">
    <location uid="2000"/>
  </property>
</add>
```

```

<entity uid="1007">
  <add>
    <component uid="1646" id="lblTitleSmall">
      <property name="text" value="神經內科外來病歷  "/>
      <property name="font">
        <entity uid="1004"/>
      </property>
    </component>
    <string value="East"/>
  </add>
</entity>
</add>
<add>
  <entity uid="1038">
    <add>
      <entity uid="1020" id="btnPatientList">
        <event-binding name="action">
          <script>
            <proc uid='146'>
              <attribute name="browser">
                <bean source="browser"/>
              </attribute>
              <attribute name="carte">
                <bean source="carteObj"/>
              </attribute>
              <property name="running" value="true"/>
              <attribute target="main" name="fail">
                <attribute name="fail"/>
              </attribute>
            </proc>
          </script>
        </event-binding>
      </entity>
    </add>
    <add>
      <component uid="1646">
        <property name="text" value="患者名一覽"/>
      </component>
    </add>
  </entity>
</add>
<add>
  <entity uid="1038">
    <add>
      <entity uid="1021" id="new">

```

```

<event-binding name="action">
  <script>
    <proc uid='148'>
      <attribute name="carte">
        <bean source="carteObj"/>
      </attribute>
      <attribute name="browser">
        <bean source="browser"/>
      </attribute>
      <property name="running" value="true"/>
      <attribute target="main" name="fail">
        <attribute name="fail"/>
      </attribute>
    </proc>
  </script>
</event-binding>
</entity>
</add>
<add>
  <component uid="1646">
    <property name="text" value="新規カルテ作成"/>
  </component>
</add>
</entity>
</add>
<add>
  <entity uid="1012"/>
</add>
<add>
  <entity uid="1038">
    <add>
      <alternative id="btnPatientAdmin">
        <entity uid="1022">
          <property name="enabled" value="false"/>
          <event-binding name="action">
            <script>
              <proc uid='147'>
                <attribute name="carte">
                  <bean source="carteObj"/>
                </attribute>
                <attribute name="browser">
                  <bean source="browser"/>
                </attribute>
                <property name="running" value="true"/>
              </proc>
            </script>
          </event-binding>
        </entity>
      </alternative>
    </add>
  </entity>
</add>

```

```

        </script>
    </event-binding>
</entity>
<bean source="btnPatientAdmin"/>
</alternative>
</add>
<add>
    <component uid="1646">
        <property name="text" value="カルテ番号、初診年月日、ID 番号、患者氏名、担当医"/>
    </component>
</add>
</entity>
</add>
<add>
    <entity uid="1013">
        <add>
            <entity uid="1023" id="btnPatient">
                <property name="enabled" value="false"/>
                <event-binding name="action">
                    <script>
                        <proc uid='140'>
                            <attribute name="carte">
                                <bean source="carteObj"/>
                            </attribute>
                            <attribute name="browser">
                                <bean source="browser"/>
                            </attribute>
                            <property name="running" value="true"/>
                        </proc>
                    </script>
                </event-binding>
            </entity>
        </add>
        <add>
            <component uid="1646">
                <property name="text" value="患者氏名、性別、生年月日、年齢、職業、紹介者"/>
            </component>
        </add>
    </entity>
</add>
<add>
    <entity uid="1013">
        <add>
            <entity uid="1024" id="btnDiagnosis">
                <property name="enabled" value="false"/>
            </entity>
        </add>
    </entity>
</add>

```

```

<event-binding name="action">
  <script>
    <proc uid='141'>
      <attribute name="carte">
        <bean source="carteObj"/>
      </attribute>
      <attribute name="browser">
        <bean source="browser"/>
      </attribute>
      <property name="running" value="true"/>
    </proc>
  </script>
</event-binding>
</entity>
</add>
<add>
  <component uid="1646">
    <property name="text" value="診断名、開始、終了"/>
  </component>
</add>
</entity>
</add>
<add>
  <entity uid="1013">
    <add>
      <entity uid="1025" id="btnDisease">
        <property name="enabled" value="false"/>
        <event-binding name="action">
          <script>
            <proc uid='142'>
              <attribute name="carte">
                <bean source="carteObj"/>
              </attribute>
              <attribute name="browser">
                <bean source="browser"/>
              </attribute>
              <property name="running" value="true"/>
            </proc>
          </script>
        </event-binding>
      </entity>
    </add>
    <add>
      <component uid="1646">
        <property name="text" value="主訴、現病歴"/>
      </component>
    </add>
  </entity>
</add>

```

```

        </component>
    </add>
</entity>
</add>
<add>
    <entity uid="1013">
        <add>
            <entity uid="1026" id="btnHabit">
                <property name="enabled" value="false"/>
                <event-binding name="action">
                    <script>
                        <proc uid='143'>
                            <attribute name="carte">
                                <bean source="carteObj"/>
                            </attribute>
                            <attribute name="browser">
                                <bean source="browser"/>
                            </attribute>
                            <property name="running" value="true"/>
                        </proc>
                    </script>
                </event-binding>
            </entity>
        </add>
        <add>
            <component uid="1646">
                <property name="text" value="アレルギー、生活習慣、既往歴、感染歴"/>
            </component>
        </add>
    </entity>
</add>
<add>
    <entity uid="1013">
        <add>
            <entity uid="1027" id="btnFamily">
                <property name="enabled" value="false"/>
                <event-binding name="action">
                    <script>
                        <proc uid='144'>
                            <attribute name="carte">
                                <bean source="carteObj"/>
                            </attribute>
                            <attribute name="browser">
                                <bean source="browser"/>
                            </attribute>

```

```

        <property name="running" value="true"/>
    </proc>
</script>
</event-binding>
</entity>
</add>
<add>
    <component uid="1646">
        <property name="text" value="家族歴、出身、血族結婚、親戚"/>
    </component>
</add>
</entity>
</add>
<add>
    <entity uid="1013">
        <add>
            <entity uid="1048" id="btnMRI">
                <property name="enabled" value="false"/>
                <event-binding name="action">
                    <script>
                        <proc uid='149' id="imageList">
                            <attribute name="carte">
                                <bean source="carteObj"/>
                            </attribute>
                            <attribute name="type">
                                <string value="MRI"/>
                            </attribute>
                            <property name="running" value="true"/>
                        </proc>
                    <proc uid="151">
                        <attribute name="carte">
                            <bean source="carteObj"/>
                        </attribute>
                        <attribute name="type">
                            <string value="MRI"/>
                        </attribute>
                        <attribute name="imageName">
                            <attribute target="imageList" name="imageName"/>
                        </attribute>
                        <property name="running">
                            <attribute target="imageList" name="selected"/>
                        </property>
                    </proc>
                </script>
            </event-binding>

```

```

    </entity>
  </add>
  <add>
    <component uid="1646">
      <property name="text" value="MRI 画像"/>
    </component>
  </add>
</entity>
</add>
<add>
  <entity uid="1013">
    <add>
      <entity uid="1049" id="btnPET">
        <property name="enabled" value="false"/>
        <event-binding name="action">
          <script>
            <proc uid="149" id="imageList">
              <attribute name="carte">
                <bean source="carteObj"/>
              </attribute>
              <attribute name="type">
                <string value="PET"/>
              </attribute>
              <property name="running" value="true"/>
            </proc>
            <proc uid="151">
              <attribute name="carte">
                <bean source="carteObj"/>
              </attribute>
              <attribute name="type">
                <string value="PET"/>
              </attribute>
              <attribute name="imageName">
                <attribute target="imageList" name="imageName"/>
              </attribute>
              <property name="running">
                <attribute target="imageList" name="selected"/>
              </property>
            </proc>
          </script>
        </event-binding>
      </entity>
    </add>
  <add>
    <component uid="1646">

```

```

        <property name="text" value="PET 画像"/>
    </component>
</add>
</entity>
</add>
<add>
    <entity uid="1012"/>
</add>
<add>
    <entity uid="1013">
        <add>
            <entity uid="1031" id="btnExit">
                <event-binding name="action">
                    <script>
                        <call-method target="uid:2601" name="getCurrentApplet" id="capplet"/>
                        <call-method target="capplet" name="logout"/>
                    </script>
                </event-binding>
            </entity>
        </add>
    </entity>
</add>
</entity>
</add>
<event-binding name="propertyChange" filter="fail" target="main">
    <script>
        <property target="btnPatientAdmin" name="enabled" value="true"/>
        <property target="btnPatient" name="enabled" value="true"/>
        <property target="btnDiagnosis" name="enabled" value="true"/>
        <property target="btnDisease" name="enabled" value="true"/>
        <property target="btnHabit" name="enabled" value="true"/>
        <property target="btnFamily" name="enabled" value="true"/>
        <property target="btnMRI" name="enabled" value="true"/>
        <property target="btnPET" name="enabled" value="true"/>
        <property target="browser" name="address">
            <property name="locationJSP" target="carteObj"/>
        </property>
        <property target="browser" name="visible" value="true"/>
        <attribute target="main" name="fail">
            <cast class="boolean" value="true"/>
        </attribute>
    </script>
</event-binding>
<component uid="1659">
<alternative>
    <entity uid="1039" id="idmap">
        <component uid="549" id="pid"/>
    </entity>
</alternative>

```

```

<call-method target="pid" name="append">
  <call-method target="idmap" name="get">
    <cast component="534">
      <call-method target="uid:2601" name="getRunningUser"/>
    </cast>
  </call-method>
</call-method>
<property target="carteObj" name="patientID">
  <call-method target="pid" name="toString"/>
</property>
<call-method name="loadAll" target="carteObj"/>
<attribute target="main" name="fail">
  <cast class="boolean" value="false"/>
</attribute>
</entity>
</alternative>
</component>
</entity>

```

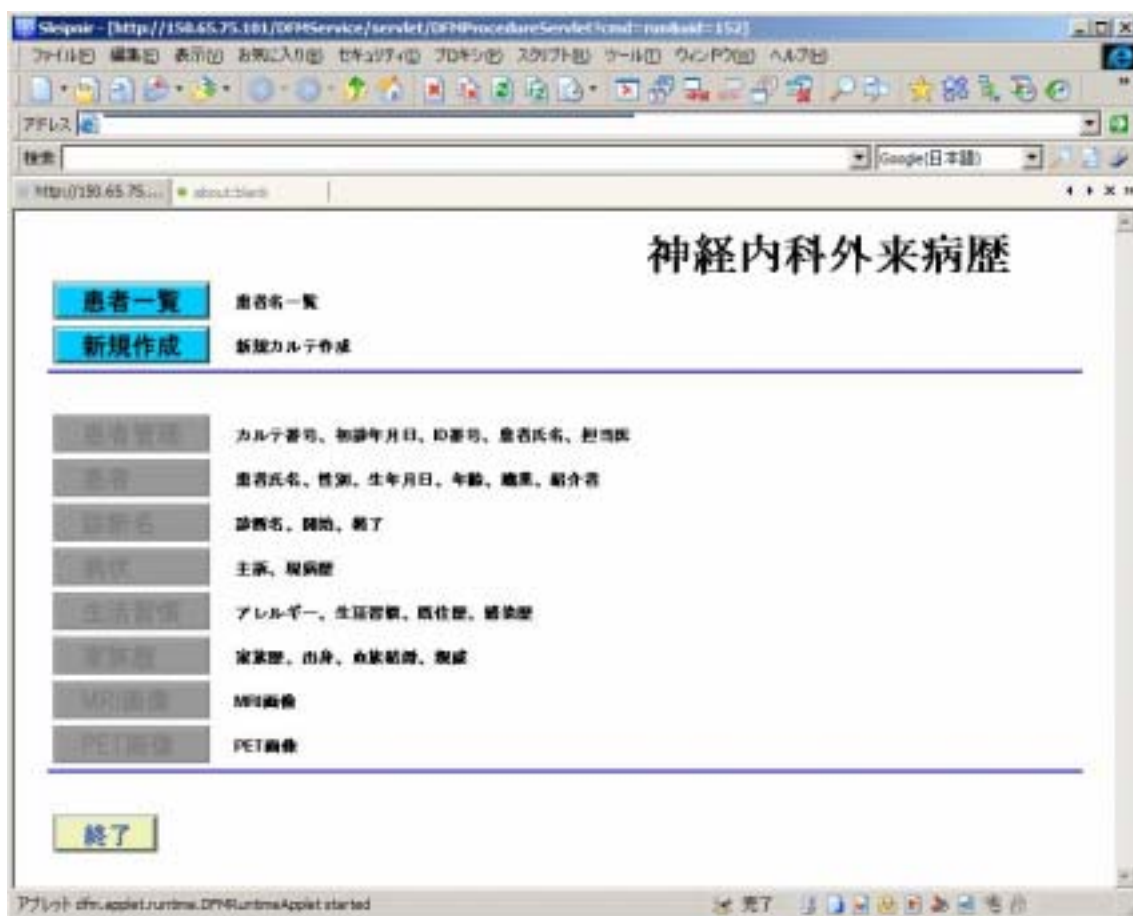


図.メイン画面

ファイル名	: EButtonp1.xml
用途	: 患者一覧ボタン
entity uid	:1020
location uid	: 1020

```
<?xml version="1.0" encoding="Shift_JIS"?>
<entity uid="1019">
  <property name="icon">
    <component uid="1609">
      <args>
        <component uid="618">
          <args><location uid="1100"/></args>
        </component>
      </args>
    </component>
  </property>
  <property name="rolloverIcon">
```

```
    <component uid="1609">
      <args>
        <component uid="618">
          <args><location uid="1101"/></args>
        </component>
      </args>
    </component>
```

```
javax.swing.ImageIcon(
new  java.net.URL("http://localhost/DFMService/repositories/carte/img/p1_f2.gif")
);
```

```
  </property>
  <property name="pressedIcon">
    <component uid="1609">
      <args>
        <component uid="618">
          <args><location uid="1102"/></args>
        </component>
      </args>
    </component>
  </property>
</entity>
```

ファイル名	: EButton01.xml
用途	: ボタンテンプレート
entity uid	:1019
location uid	: 1019

```
<?xml version="1.0" encoding="Shift_JIS"?>
<component uid="1615">                                <!-- javax.swing.JButton -->
  <property name="background">
    <entity uid="1008"/>
  </property>
  <property name="font">
    <entity uid="1010"/>
  </property>
```

```
  <property name="border">
    <call-method target="uid:1581" name="createRaisedBevelBorder"/>
  </property>
```

```
java.swing.JButton.setBorder(javax.swing.BorderFactory.createRaisedBevelBorder());
```

```
  <property name="margin">
    <component uid="60">
      <args><cast class="int" value="12"/><cast class="int" value="14"/><cast class="int" value="12"/><cast class="int" value="14"/></args>
    </component>
  </property>
```

```
java.swing.JButton.setMargin(java.awt.Insets(12,14,12,14));
```

```
</component>
```

ファイル名 : EcolorBtRs.xml
用途 : R180G255B200 色指定



entity uid :1008
location uid : 1008

```
<?xml version="1.0" encoding="Shift_JIS"?>
<component uid="24"> <!-- java.awt.Color -->
<args>
    <cast class="int" value="180"/>
    <cast class="int" value="255"/>
    <cast class="int" value="200"/>
</args>
</component>
```

```
Color c = new Color(180,255,200);
```

ファイル名 : EserifBold14.xml
用途 : Font 指定 Serif Font.PLAIN 14
entity uid :1010
location uid : 1010

```
<?xml version="1.0" encoding="Shift_JIS"?>
<component uid="42"> <!-- java.awt.Font -->
<args>
    <string value="Serif"/>
    <cast class="int" value="0"/>
    <cast class="int" value="14"/>
</args>
</component>
```

```
Font f14 = new Font("Serif",Font.PLAIN,14);
```
