

## 仮想マルチメディアオブジェクトを用いた図書のカスタム化システム

原田 正則 宝珍 輝尚 中田 充 都司 達夫  
福井大学 工学部 情報工学科

### あらまし

次世代情報システムの一つとして電子図書館が研究開発されており、電子図書館では、様々な図書から独自の図書を作成し、要求に応じて種々のメディア変換ができ、電子的な付箋を付けてメモを書き込むことが要求されている。本論文では、このような電子図書カスタム化システムを仮想マルチメディアオブジェクト (VMO) を用いて実現する。ここで、VMO とは、マルチメディアデータの一部、または、様々な演算やデータ付加の結果得られるオブジェクトである。VMO システムを用いると少ない開発コストで電子図書カスタム化システムを構築することができ、VMO システムを利用する性能オーバーヘッドは実用上問題とならないことを明らかにする。

## Digital Document Customizing System Using Virtual Multimedia Objects

Masanori HARADA Teruhisa HOCHIN Mitsuru NAKATA Tatsuo TSUJI  
Department of Information Science, Faculty of Engineering, Fukui University

### Abstract

The digital library has been studied and developed as one of the next generation information systems. In the digital library, creating original documents by taking out parts of various books, converting data formats, and filling out memo are required. A digital document customizing system (DDCS) has been constructed by using virtual multimedia objects (VMOs). Here, a VMO is a part of multimedia data, the object created by some operation, or the one with the specific data. We clarify that the construction cost of the DDCS can be reduced by using the VMO system, and the performance overhead in using the VMO system can be ignored in the real application.

## 1 はじめに

マルチメディア情報処理技術，マルチメディア通信技術，大容量記憶装置技術などの進展により，テキスト，イメージ，グラフィックス，ビデオなどのマルチメディアデータが計算機で手軽に扱えるようになってきている。また，インターネット上には多数の WWW(World-Wide Web)情報蓄積サーバが構築されてきており，それら世界中のサーバにアクセスして自分が必要とする情報を居ながらにして入手することが可能になってきている。そして，次世代の魅力的な情報システムの一つとして「電子図書館」が研究開発されている [1, 2, 3]。電子図書館とは，必要な情報を，どこにあるかを問わず，いつでも，どこにいても，簡単に入手することができ，楽に読むことができるとともに，関連のある情報へと導いてくれるシステムである [1]。電子図書館を利用する端末では，利用者の要求に応じて種々のメディアに変換でき，電子的な付箋を付けてどこにでもメモを書き込んだり，アンダーラインをしたりすることが要求されている [2]。

そこで，本論文では，仮想マルチメディアオブジェクト (VMO) を用いた電子図書カスタム化システムの実現について述べる。ここで VMO とは，マルチメディアデータの一部分であるだけでなく，様々な演算やデータ付加の結果得られるものであり得るマルチメディアオブジェクトである。VMO システムを用いると少ない開発コストで電子図書カスタム化システムを構築することができ，VMO システムを利用する性能オーバーヘッドは実用上問題とならないことを明らかにする。

以下，2章では，電子図書館について概説する。3章では，仮想マルチメディアオブジェクト (VMO)，ならびに，VMO を実現するシステムについて述べる。4章では，VMO を用いて実現した図書のカスタム化システムについて述べる。最後に，5章でまとめを行う。

## 2 電子図書館

電子図書館は，伝統的な図書館とは，次のように，多くの点で異なるといわれている [1]。

- 電子図書館は図書館の単なる電子化ではない。  
すなわち，電子図書館は，現在の図書館資料・業務を電子化した電子化図書館とは異なる。
- 電子図書館は孤立した大規模集中システムではない。  
世界中に分散している大小様々の情報源の集まりが，電子図書館である。
- 電子図書館は単なる情報検索システムではない。  
必要な情報を見つけ出す機能は，電子図書館にとって必須の機能ではあるが，それに加えて，情報源の拡大，読書をしやすくする機能などが必要である。
- 電子図書館は他のシステムやサービスと親和性が高い。  
電子図書館の要素技術は，他の様々なシステムやサービスと共通する部分が多いため，電子図書館は他のシステムやサービスとうまく融合しうる。

以上のような電子図書館の主要な点 [2] を以下に示す。

- (1) 電子図書館が取り扱う情報の単位は一冊単位の本や雑誌ではなく，それらの中の任意の章，節，項，あるいは頁など，任意の単位である。

- (2) 情報の検索は書誌的事項やキーワードによる検索でなく、それらよりはるかに精度高く必要箇所を取り出せる階層構造検索であり、全文検索はそれらの検索でも満足できない場合に行う。
- (3) 関連のある情報は任意の大きさの情報単位で相互関連づけられて理想的に情報を引き出し、ダイナミックな形でハイパーテキスト（メディア）システムを構成することができる。
- (4) 読書端末は利用者の要求に応じて種々のメディア変換ができ、以下に示す読書のための補助機能により電子読書を可能とする。
  - (i) 文字の大きさや配列、頁の形式などを自由にかえ、読みやすくする。
  - (ii) 自動朗読機能によって取り出した文章を朗読する。
  - (iii) 分からない単語を辞書引きできる。
  - (iv) 外国語は自動翻訳してくれる。
  - (v) 電子的な付箋を付けて、どこにでもメモを書き込んだり、アンダーラインをしたりできるし、電子的なしおりをはさんで続きは明日読むことができる。
  - (vi) 関連する参考資料を取り出して本文とともに並行して読んだり、ノートブックを開いて資料を参考にしながら論文を書いたりすることができる。

(1)と(3)より、様々な図書から自分に必要な部分を任意の単位で抜き出すことが望まれている。必要な部分を抜き出すことについては「トランスクリューション」という概念が提案されてきており、これを実現するシステムのプロトタイプが構築されてきている[3]。これに加え、(4)の(v)で述べたような機能も必要とされている。

ここで、筆者らの提案している仮想マルチメディアオブジェクト（VMO）を用いると、上記の、様々な図書から自分に必要な部分を抜き出し、メモを付け、アンダーラインをする機能を容易に実現できる可能性がある。そこで、この機能を持つ電子図書カスタム化システムをVMOを用いて構築してみた。以下、VMOについて説明し、電子図書カスタム化システムの構築について述べる。

### 3 仮想マルチメディアオブジェクト

#### 3.1 仮想マルチメディアオブジェクト

ここでは、利用者によるマルチメディアデータの自由なカスタム化を可能とする仮想マルチメディアオブジェクト（VMO）[4]について述べる。

仮想マルチメディアオブジェクト（VMO）は次のように定義される。

- マルチメディアデータの一部分を抜き出してできるオブジェクト
- VMO から0以上のデータを用いて変換されたオブジェクト。
- VMO から0以上のデータを用いて生成されたオブジェクト。
- 1以上のVMO から生成されたオブジェクト。

ここでの“変換”とは、画像に対するエッジ抽出といった画像処理やデータフォーマット変換を意味する。また、“生成”とは、利用者固有のデータを付加した新たなVMOを生成することを意味する。例えば、文字列“田中 太郎, 山田 次郎, 鈴木 三郎”からされた文字列“○田中, 鈴木, 山田”はVMOである。これは、例えば、もとの文字列から3つの文字列“田中,”、“山田,”、“鈴木”を取り出し、文字列“田中,”の先頭に“○”を付け、文字列“鈴木”の最後に“,”を付け、並び替えて一つの文字列とすることで生成できる。

VMOの考え方によれば、もとのマルチメディアデータに存在しないデータを付加できる。また、定義より、VMOに対してさらにデータを付加したり変換を行うことが可能である。この際にデータの複

写を行う必要がない。

ここでは、VMO ともとなるマルチメディアデータ間の対応関係を表現する、ユーザ定義の構造体（以降、指示エン트리と呼ぶ）を利用する。ここで、VMO のもととなるマルチメディアデータ（または、VMO）を対象データと呼ぶ。指示エント리는、ユーザ定義の手続きである導出手続きにより生成される。また、VMO は、ユーザ定義の関数である対応関数により対象データと指示エン트리から導出される。また、対象データと VMO のデータフォーマットは同一とは限らないので、データフォーマット変換を考慮する必要がある。

VMO は5つ組 ( $D_{od}$ ,  $D_{pe}$ ,  $P_{drv}$ ,  $P_{map}$ ,  $P_{conv}$ ) で表される。ここで、 $D_{od}$  は対象データの集合、 $D_{pe}$  は指示エント리의集合、 $P_{drv}$  は導出手続き、 $P_{map}$  は対応関数、 $P_{conv}$  はデータフォーマット変換関数である。指示エント리는  $D_{od}$  中のデータと VMO との対応関係を表現する。導出手続きは  $D_{od}$  をもとに指示エント리를生成する。対応関数は  $D_{od}$  と  $D_{pe}$  から VMO を導出する。 $P_{conv}$  は対応関数により導出された VMO のデータフォーマット変換を行う。

VMO に対して型を定義することができる。VMO の型は7つ組 (Name,  $f_{org}$ ,  $f_{mo}$ ,  $f_{pe}$ ,  $P_{drv}$ ,  $P_{map}$ ,  $P_{conv}$ ) である。Name は VMO の型を一意に識別する名前である。 $f_{org}$  は対象データのデータフォーマット、 $f_{mo}$  は VMO のデータフォーマット、 $f_{pe}$  は指示エント리의データフォーマット、 $P_{drv}$  は導出手続きの集合、 $P_{map}$  は対応関数、 $P_{conv}$  はデータフォーマット変換関数である。対象データ、VMO、指示エント리의データフォーマットが同一で、同じ対応関数をもつ VMO は同じ型に属することができる。

### 3.2 プロトタイプシステム

VMO の定義に基づいたプロトタイプシステムの実現について述べる。プロトタイプシステムは、Solaris2.5.1 上で C++ 言語を用いて実現した。設計には、オブジェクト指向方法論 OMT [9] を用いた。

プロトタイプシステムにおいては、VMO に関する定義情報の管理と VMO に対応する永続的なオブジェクトの作成を行わなければならない。そこで、プロトタイプシステムを仮想オブジェクトサブシステムと格納サブシステムの2つのサブシステムにより構成した。仮想オブジェクトサブシステムでは VMO の作成、ならびに、VMO に関する定義情報（すなわち、VMO 型）の管理を行う。また、格納サブシステムは仮想オブジェクトサブシステムで生成されたオブジェクトの永続的な管理を行う。以降では仮想オブジェクトサブシステムと格納サブシステムについて説明する。

#### 3.2.1 仮想オブジェクトサブシステム

仮想オブジェクトサブシステムの OMT におけるオブジェクト図を図1に示す。

「VMO Type」クラスのメンバ関数では、ユーザが定義した VMO の型に対応するクラスの作成と VMO のデータフォーマット変換方法の決定を行う。例えば、VMO 名が “Test” である VMO の型が定義されると、格納サブシステムを用いて「VMO」クラス “Test” を作成する。また、データフォーマット変換方法は、対象データのデータフォーマット名と VMO データフォーマット名をもとにシステムに登録されているデータフォーマット変換表を用いて決定する。たとえば、対象データのデータフォーマット名が “GIF” で、VMO データフォーマット名が “JPEG” の場合は、コマンド convert を使用するという具合である。

「Original Data」クラスは、対象データを管理するクラスである。「Original Data」クラスは対象データの種別をもとに対象データを管理する。「Real Data」クラス、「Virtual Data」クラスは、各々、VMO の対象データとなるマルチメディアデータと VMO を管理するクラスであり、「Original Data」クラスのサブクラスである。「Real Data」クラスは対象データとなるマルチメディアデータへのファイルパスを管理する。「Virtual Data」クラスは対象データとなる VMO の管理を行う。「Virtual Data」

クラスは、導出元となる対象データの VMO を導出先の VMO の親として登録することにより再帰的な VMO の導出を可能にしている。対象データが VMO でかつ複数存在するときは、それらのデータから一つの一時的なデータを作成し、導出手続き、ならびに、対応関数を適用している。

指示エントリの作成、ならびに、VMO の導出は「VMO」クラスのメンバ関数により行う。仮想オブジェクトサブシステムにおいて、ユーザ定義の手続きである導出手続きと対応関数の実行は、ユーザが指定した手続き名と手続きのオブジェクトファイルへのファイルパスに基づいて、Unix でサポートされているダイナミックリンク機能を利用して行う。「VMO」クラスはユーザが指定した対象データをもとに導出手続きを実行し、指示エントリを作成する。また、対象データと指示エントリをもとに対応関数を実行し、VMO を導出する。このとき、データフォーマット変換が必要な場合は、「VMO Type」クラスで決定したデータフォーマット変換関数により VMO のデータフォーマット変換を行う。

仮想オブジェクトサブシステムのプログラムサイズは C++ で約 1,200 行（コメントを除く）である。

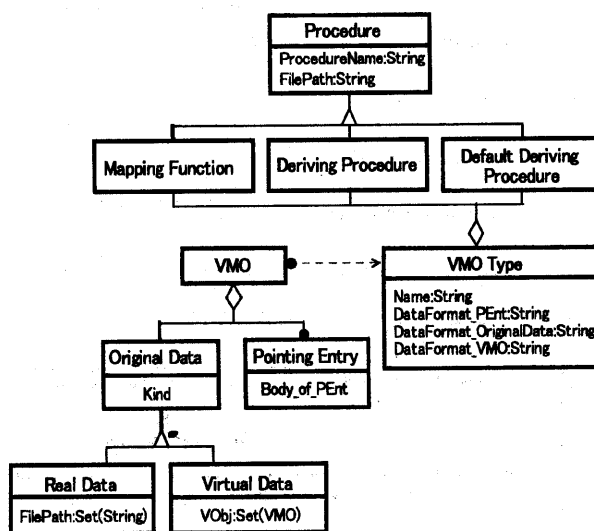


図1 仮想オブジェクトサブシステムに対するオブジェクト図

### 3.2.2 格納サブシステム

格納サブシステムは、仮想オブジェクトサブシステムで作成されたクラスやインスタンスの永続的な管理を行うシステムであり、二次記憶デバイスとしてファイルを用いている。格納サブシステムでは仮想オブジェクトサブシステムにおける1つのクラスを3つのファイルを用いて管理する。1つは、クラスのインスタンス（データ実体）を格納するファイルである。2つ目は、各データ実体のファイルの先頭からのオフセットを格納するファイルである。最後は、クラスの属性情報（属性名、ドメイン名と長さ）を管理するファイルである。また、対象データの更新を可能とするため、格納サブシステムでは履歴データ型を用いた版管理データモデルを利用する。履歴データ型とは、履歴型による入れ子型リレーショナルモデルの枠組みの中で、版の導出関係と時間情報の管理を可能とするデータ型である。履歴型に導出元となるインスタンスを導出先であるインスタンスの親の版として登録することにより、更新前のインスタンスを特定できるようにしている。また、このために、対象データが実データの場合、それが最初に使用されるときに、その対象データをコピーして格納サブシステム内で保持している。

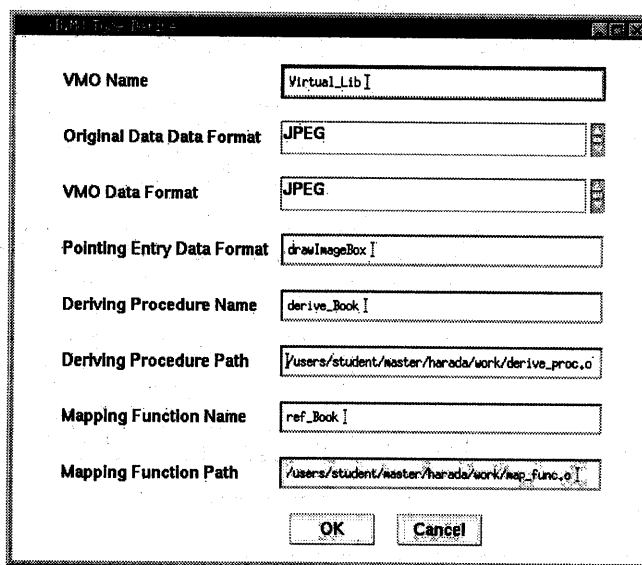
格納サブシステムのプログラムサイズは C++ で約 6,000 行（コメントを除く）である。

## 4 図書のカスタム化システム

ここでは、VMO を用いた図書のカスタム化システムの実現方法について述べる。本システムでは、自分専用の図書の生成を導出手続きにより行う。また、導出手続きで生成された図書の参照は対応関数により行う。利用者は、あらかじめ、必要とする図書を対象データとして登録しておく必要がある。

次に、電子図書館システムの実行例を示す。

利用者は図2のような VMO 型「Virtual\_Lib」を定義したとする。図2の定義を行うと、システムは「VMO Type」クラスのインスタンスに図2の定義情報を格納し、「VMO」クラスに対応する「Virtual\_Lib」クラスを格納サブシステムを用いて作成する。



VMO Name	Virtual_Lib
Original Data Data Format	JPEG
VMO Data Format	JPEG
Pointing Entry Data Format	drawImageBox
Deriving Procedure Name	derive_Book
Deriving Procedure Path	/users/student/waster/harada/work/derive_proc.o
Mapping Function Name	ref_Book
Mapping Function Path	/users/student/waster/harada/work/map_func.o

図2 VMO 型定義

ここでは、冊子体の遺跡発掘調査報告書 [8] をスキャナで読み込んで電子化したものをもとに新たな図書を作成する方法について述べる。

利用者が対象データとして登録した遺跡発掘調査報告書の中から図3に示す2ページより導出手続き“derive\_Book”を実行し、新たな図書を生成することを考える。

導出手続きにおいて、利用者はウィンドウ上に表示されている対象データより自分が必要とする部分をマウスで指定し (図4(a))、新たな図書に貼り付け自分専用の図書を生成する (図4(b))。この際、導出手続きで生成された指示エン트리 “drawImageBox” のインスタンスは格納サブシステムの「drawImageBox」クラスに格納される。同様にして、テキストの必要部分も切り出して同一ページに貼り込むことができる。導出手続きが終了すると、「Virtual\_Lib」クラスのインスタンスとして「drawImageBox」クラスのインスタンスの集合と対象データ (いずれも、オブジェクト識別子) が保持される。図書の参照は対応関数 “ref\_Book” により行われる (図4(c))。「Virtual\_Lib」クラスのインスタンスにメッセージを送ることにより対応関数 “ref\_Book” が呼び出され、対象データと指示エン트리 (「drawImageBox」クラスのインスタンス) を用いて VMO を生成する (図4(c))。このようにして、利用者は自分専用の図書を生成することが可能になる。

これをもとにして再び新たな図書を作成することができる。

2. 図書の抽出 (抽出) 抽出された図書の抽出結果を抽出結果ファイルに抽出結果を抽出する。  
 3. 図書の抽出 (抽出) 抽出された図書の抽出結果を抽出結果ファイルに抽出結果を抽出する。  
 4. 図書の抽出 (抽出) 抽出された図書の抽出結果を抽出結果ファイルに抽出結果を抽出する。  
 5. 図書の抽出 (抽出) 抽出された図書の抽出結果を抽出結果ファイルに抽出結果を抽出する。  
 6. 図書の抽出 (抽出) 抽出された図書の抽出結果を抽出結果ファイルに抽出結果を抽出する。  
 7. 図書の抽出 (抽出) 抽出された図書の抽出結果を抽出結果ファイルに抽出結果を抽出する。  
 8. 図書の抽出 (抽出) 抽出された図書の抽出結果を抽出結果ファイルに抽出結果を抽出する。  
 9. 図書の抽出 (抽出) 抽出された図書の抽出結果を抽出結果ファイルに抽出結果を抽出する。  
 10. 図書の抽出 (抽出) 抽出された図書の抽出結果を抽出結果ファイルに抽出結果を抽出する。

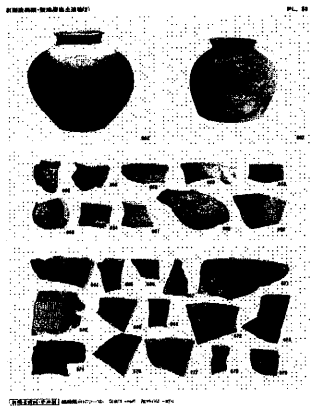
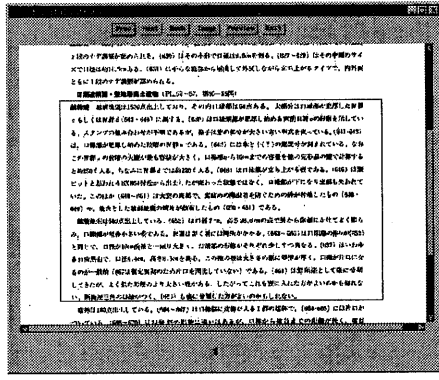
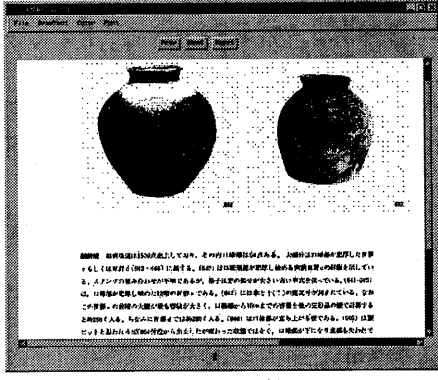


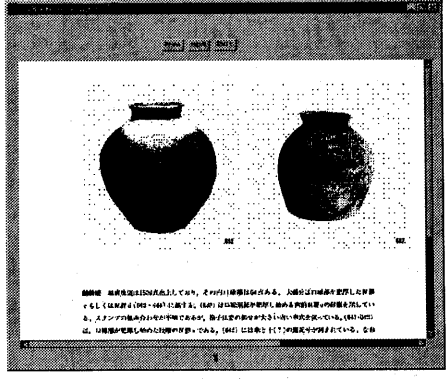
図3 対象データ



(a) 図書の抜き出し



(b) 図書の貼り付け



(c) 図書の参照

図4 図書カスタム化システム実行例

本システムを作成するためのプログラム量は、C++でコメントを除いて約900行(導出手続き約500行, 対応関数約400行)である。これは、プロトタイプシステムを用いないで作成する場合の約1/8

の規模である。

また、VMO 型の定義、指示エントリの作成、VMO の導出には、各々、0.27 秒、0.78 秒、0.56 秒を要する。これらは利用者とのインタラクションの時間（数秒～数十秒）と比較して極めて短く利用上問題とならない。

## 5 まとめ

本論文では、仮想マルチメディアオブジェクト (VMO) を用いた図書館のカスタム化システムの実現について述べた。VMO は、マルチメディアデータもしくは VMO の一部分を取り出してできるオブジェクト、VMO から変換によって得られるオブジェクト、VMO に新たなデータを付加して得られるオブジェクト、または、1つ以上の VMO から生成されたオブジェクトである。VMO を用いることにより、利用者が、もとのマルチメディアデータを損なうことなく、また、利用者ごとにもとのマルチメディアデータを複写することなく、独自の操作やデータの付加を行うことが可能となる。VMO を実現するプロトタイプシステムを用いて、電子図書館のカスタム化を行うシステムを構築した。この結果、VMO を使用しない場合と比較して、約 1/8 程度のプログラム記述量で電子図書館カスタム化システムが構築でき、プロトタイプシステムを利用することによる時間的なオーバーヘッドはほとんど無視できることを明らかにした。

本稿で述べた電子図書館カスタム化システムは画像ファイルのみに対応している。SGML 文章、HTML 文章や Latex 文章に対する対応は今後の課題である。

謝辞 一乗谷朝倉氏遺跡第 24 次発掘調査報告書の使用を許可して頂いた福井県立一乗谷朝倉氏遺跡資料館文化財調査員水村伸行氏に感謝致します。

本研究は、一部、文部省科学研究費重点領域研究 (1) (課題番号 09204113) による。

## 参考文献

- [1] 原田 勝：「電子図書とはなにか」, 人文学と情報処理, No. 9, pp. 3-8 (1995).
- [2] 長尾 誠：「電子図書館の理念」, 人文学と情報処理, No. 9, pp. 9-16 (1995).
- [3] 藤澤 浩道, 絹川 博之：「仮想個人図書館」と個人情報環境, デジタル図書館, No. 6 ([http://www.dl.ulis.ac.jp/Dljournal/No\\_6](http://www.dl.ulis.ac.jp/Dljournal/No_6)) (1996).
- [4] 原田 正則, 宝珍 輝尚, 都司 達夫：マルチメディアデータのカスタム化を目指した仮想マルチメディアオブジェクト, 電気関係学会北陸支部連合大会 (1997).
- [5] 原田 正則, 宝珍 輝尚, 中田 充, 都司 達夫：データ型に基づくマルチメディアデータ参照機構の実現とその有効性, 情報処理学会論文誌, Vol.38, No.8, pp.1603-1612 (1997).
- [6] Hochin, T. and Tsuji, T. : On the Application Interface of the LIQUID data type for flexible manipulation of multimedia data, Proc. of Int'l Symposium on Advanced Database Technologies, pp.200-206 (1994).
- [7] Zdonik, S. B. : Incremental Database System : Database from the Ground Up, Proc. of ACM SIGMOD 1993, pp.408-412 (1993).
- [8] 福井県立一乗谷朝倉氏遺跡資料館：一乗谷朝倉氏遺跡発掘調査報告 IV (1993).
- [9] ランボー, J., ブラハ, M., プレメラニ, W., エディ, F., ローレンセン, W. (著), 羽生田栄一(監訳)：オブジェクト指向方法論 OMT, トッパン(1992).