

XMLを用いた情報システムの実現に関する研究

2000MT043 北川 直毅 2000MT070 野村 和

指導教員 野呂 昌満

1 はじめに

近年、情報システムのデータの表現形式として、XML が利用される。XML は、文書の交換形式の標準として広く普及している。[1][3]。情報システムの入出力データは、関係データベースで保持され、XML を用いることで、入出力データを木構造で表すことができる。検索、登録など各機能を実現するプログラムは、入出力データの対応関係から実現する。

入出力データの対応関係からプログラムを導く方法論として JSP(Jackson Structured Programming) がある [2]。入出力の対応関係が決まれば自動的にプログラムを導くことができることから、本研究の情報システム開発は JSP を利用することが有効だと考える。情報システムを開発する場合に以下の問題点がある。

1. 情報システムのプログラムは類似している部分が多いにもかかわらず、プログラムの再利用がされていない。
2. 情報システムのデータのデータ構造は類似しているが、その類似性を活用していない。

本研究の目的は情報システムを開発するときの問題を解決し、開発の手間を軽減することである。

本研究で上述した問題点は以下のようにして解決する。

1. JSP を適用して実現した住所録システムのプログラムから再利用可能なプログラムを抽出する。住所録システムの機能である登録、表示、検索、削除の入出力の木の写像を考え、デザインパターンを用いて機能を実現するプログラムを作成する。本来、JSP は手続き指向言語を対象にしたプログラミング方法論である。本研究では、プログラムの作成においてオブジェクト指向言語を対象にデザインパターンを用いる。
2. 各情報システムに共通のデータ構造を抽出し、メタ情報として定義する。メタ情報を定義することで入出力の木の構造と要素間の写像を自動的に導くことができる。入出力の写像からプログラムの木を自動生成できるので、汎用的な情報システムを実現することができると考えられる。

以上の手順から情報システムを実現する。情報システムの開発者はメタ情報を作成するだけで情報システムを実現することができる。

北川は主に JSP とメタ情報について、野村は情報システムのプログラム作成を主に担当した。

2 JSP の概要

JSP は、プログラムへの入出力データに注目し、データの構造を考察し、そのデータの構造からプログラムの構造を導くという着想にもとづいている [2]。

基本 JSP は次の 4 段階の手順から構成されている。

1. データ構造の定義 (データ・ステップ)
入出力データをそれぞれジャクソンの木構造図によって定義する。
2. プログラム構造の導出 (プログラム・ステップ)
入力の木から出力の木への写像の存在を確かめ、入力の木と出力の木からプログラムの構造図、すなわち、プログラムの木を導出する。
3. 手続きの列挙 (オペレーション・ステップ)
必要な手続きを列挙し、プログラムの木に割り付ける。
4. プログラムの完成 (テキスト・ステップ)
プログラムの実行に必要な条件を付加し、プログラム・テキストを完成する。

3 情報システムの概要

XML は W3C(World Wide Web Consortium) が標準化をおこなっている文書規約である [1][3]。W3C 勧告のもと広く普及している XML パーサとして DOM パーサがある [4]。DOM は応用言語に依存することなく XML 文書を解析するためのモデルであり、DOM パーサは XML 文書を DOM に従った木構造で表現する。情報システムは、DOM パーサを利用し、JSP のプログラム方法論に従って実現する。

3.1 本研究の情報システムの機能

本研究での情報システムは、住所録や図書などのデータの一覧表示、検索、削除、登録ができる機能をもつ。それぞれの機能の入出力を以下のように考えた。

1. 一覧表示・・・入力は情報システムのデータを表現した XML 文書 (以下、入力 XML 文書とする) とメタ情報とする。出力は情報システムのデータとし、出力の表示形式は情報システム開発者によって変えることができる。
2. 検索・・・入力は、入力 XML 文書とメタ情報、情報システム利用者が入力した検索キーとする。出力は、検索キーに一致した情報システムのデータとする。
3. 削除・・・入力は、入力 XML 文書とメタ情報、情報システム利用者が入力した検索キーとする。出力は、検索キーに一致した情報システムのデータを削除した XML 文書とする。
4. 登録・・・入力は、入力 XML 文書とメタ情報、情報システム利用者が入力した登録データとする。

出力は登録データを追加した情報システムのデータを表現した XML 文書である。

本研究の情報システムの機能を以上のように考えたことで、次の特徴がある。

- 機能を限定したことで情報システムの機能を実現するためのプログラムは類似している。
- プログラムの異なる部分は、情報システムのデータを表現した XML 文書のデータ構造に依存する部分である。
- 情報システムのデータの構造は、類似している部分が多い。

これらの点から、共通のデータ構造を抽出しデータ構造を限定することで、汎用的なプログラムが実現できると考える。

4 JSP を用いた住所録システムの開発

JSP の方法論を用いて住所録システムを開発する。住所録システムの入出力の木構造図を定義し、要素間の対応を考える。対応関係からプログラムの木構造図を導出し、プログラムを作成する。プログラムの作成は、オブジェクト指向言語で作成し、デザインパターンを用いて再利用可能なプログラムを実現する。

住所録システムの開発

1. データ構造の定義 (データ・ステップ)
住所録システムの機能を実現するための入出力データを木構造図で定義する。
2. プログラム構造の導出 (プログラム・ステップ)
入力の木から出力の木への要素間の対応を調べ、プログラムの構造図を導出する。住所録システムの一覧、検索、登録の木構造図と要素間の対応を示したものを図 1、図 2、図 3 に示す。
3. 手続きの列挙 (オペレーション・ステップ)
必要な手続きを列挙し、プログラムの木に割り当てる。手続きの列挙を図 4 に示す。また、手続きをプログラムの木に割り当てる。
4. プログラムの作成 (テキスト・ステップ)
プログラムの作成はオブジェクト指向言語で作成し、デザインパターンを用いることで、再利用可能なプログラムを実現する。
導出したプログラムの木から、プログラムの木の要素を走査しながら、処理すれば、機能が実現できる。Visitor パターンを用い、木を走査するクラスと、各要素で処理をするクラスに分ける。木を走査するクラスは再利用できる。Visitor パターンを用いたクラス図を図 5 に示す。
また、情報システムの機能は、入力データの要素を処理することで実現している。機能は複数の処理で実現し、各機能は、同じ処理をする場合がある。Command パターンを用い、処理を command とすることで、複数の command を組み合わせる新しい機能として再利用することができる。Command パターンを用いたクラス図を図 6 に示す。

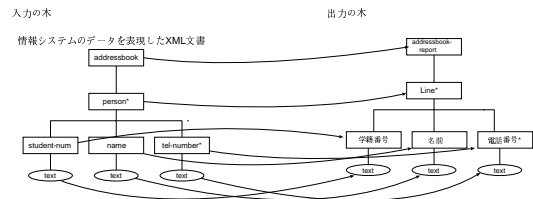


図 1 一覧表示の入出力の木

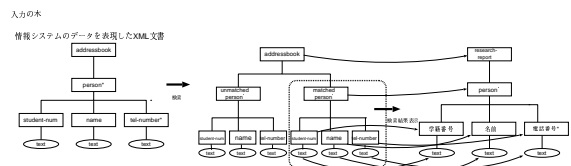


図 2 検索の入出力の木

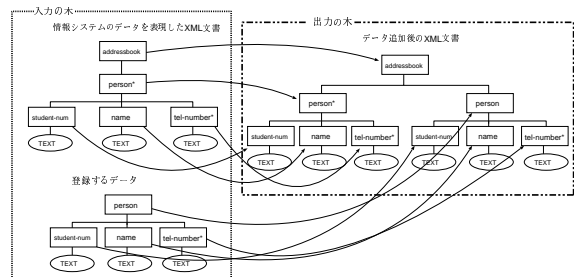


図 3 登録の入出力の木

図 5 は、表示をするときの Visitor パターンを用いたクラス図である。ノードをたどるクラスを Walker クラスとして、Visitor クラスに各ノードの処理をさせる。実際の処理は、Visitor クラスのサブクラスである PrintText クラスと PrintHTML クラスが行う。PrintText クラスは、コマンドライン上で表示をするためのクラスである。PrintHTML クラスは、Web ブラウザ上で表示をするためのクラスである。Visitor のサブクラスでノードタイプが ELEMENT の場合と TEXT の場合ノードに対して処理を行う。

5 メタ情報の定義と JSP の適用

JSP を用いた情報システムの開発に、メタ情報を適用する。メタ情報は、情報システムの入力データのデータ構造を定義する。本研究の情報システムは入力の木と出力の木の構造が一致するので、メタ情報は出力の木の構造も決定する。また、各情報システムごとに入出力の木の写像を調べていたが、入出力の木とメタ情報の対応関係から、入出力の要素間の対応が自動的に決定する。また、入力の要素名と出力の表示名の対応を InfoMap として定義する (図 9)

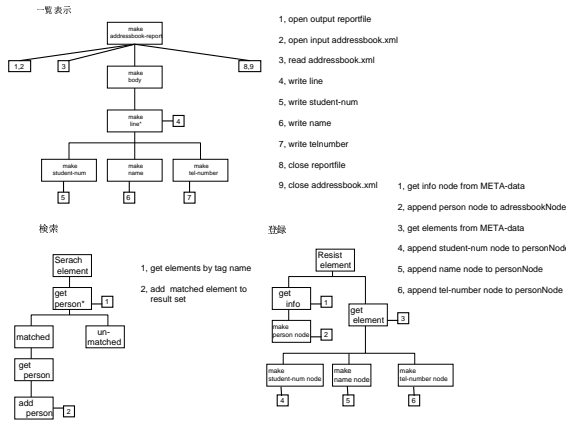


図4 プログラムの木

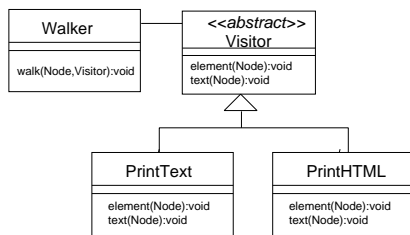


図5 Visitorパターンを用いたクラス図

メタ情報の定義

本研究での情報システムの共通のデータ構造を以下のように特徴づける．作成したメタ情報の DTD を図 7 に示す．

1. 情報システムのデータのデータ構造は，レコード型要素の繰り返しで構成されている．情報システムの機能を実現する際，レコード型の要素を単位として処理することが多い．レコード型要素を info 要素として定義する．
2. レコード型を構成する要素はメタ情報の element 要素で表現する．element 要素の子要素である elementName 要素，occurs 要素，attr 要素で要素名，出現頻度，属性を表す．

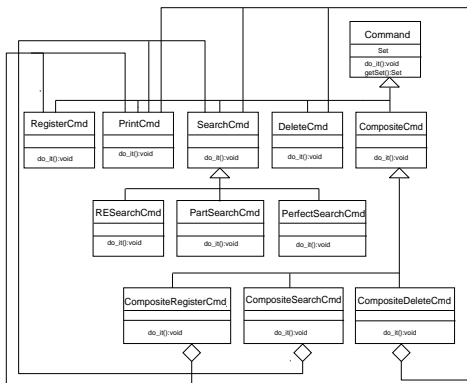


図6 Commandパターンを用いたクラス図

```

<ELEMENT META-DTD (root,info)>
<ELEMENT root (rootName)>
<ELEMENT rootName (#PCDATA)>
<ELEMENT info (elementName,child)>
<ELEMENT elementName (#PCDATA)>
<ELEMENT child (element?,string?,empty?)>
<ELEMENT element (elementName,occurs,attr?,child)>
<ELEMENT occurs (#PCDATA)>
<ELEMENT attr (attrName,enum)>
<ELEMENT attrName (#PCDATA)>
<ELEMENT enum (value+)>
<ELEMENT value (#PCDATA)>
<ELEMENT string EMPTY>
<ELEMENT empty EMPTY>

```

図7 メタ情報の DTD 文書

図7のメタ情報の DTD は，rootName で情報システムの名前，info でデータ全体を構成するレコード型要素の名前を定義する．element でレコードを構成する要素の名前，出現頻度，属性，要素の親子関係を定義する．

メタ情報の DTD に従って作成したメタ情報の XML 文書を図9に示す．

```

<META-DATA>
<root rootName=addressbook/>
<info>
<elementName=person/>
<child>
<elementName=student-num/>
<occurs=once/>
<child>
<string>
<element>
<elementName=name/>
<occurs=once/>
<child>
<string>
<child>
<element>
<elementName=tel-number/>
<occurs=overZero/>
<attr>
<attrName=sort/>
<enum>
<value=home/>
<value=mobile/>
</enum>
</child>
</string>
</child>
</element>
</info>
</META-DATA>

```

図8 住所録システムのメタ情報の XML 文書例

図9はレコード型の person 要素が子要素，student-num 要素，name 要素，tel-number 要素をもつ．出現頻度はそれぞれ 1 回，1 回，0 回以上であることをメタ情報の XML 文書の occurs 要素が表す．tel-number 要素は属性名が sort で，属性値が home，mobile のいずれかをとることをメタ情報の XML 文書の attr 要素で表している．

メタ情報を用いた JSP の適用

メタ情報を定義し JSP に適用することで，以下の利点がある．

- 入出力の木の構造が自動的に決定する．
- 入出力の要素間の対応関係が自動的に決定する．

住所録システムを例にすると，図1の入力の木の人要素と出力の木の Line 要素は，メタ情報の info 要素と対応している．また，図1の入出力の木の student-num,name,tel-number 要素はメタ情報の element 要素と対応している．このことから，入力の木と出力の木の要素間の写像はメタ情報を用いて決定することができる．写像が自動に生成されることから，プログラムも自動生成できる．メタ情報と入出力の木の対応，入出力の木の要素間の対応を図9に示す．

6 考察

本研究で，JSP を適用した情報システムの利点と，Command の組み合わせによる機能について考察する．ま

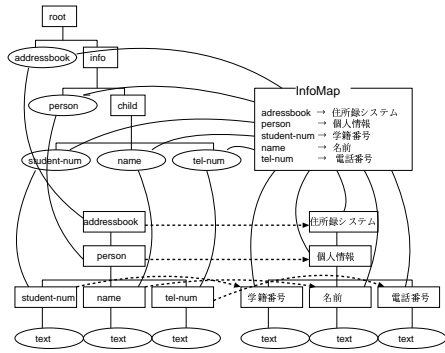


図9 メタ情報を用いた一覧表示の入出力の木

た、本研究の情報システムは入力の木構造と出力の木構造が一致している場合について考えたが、入出力のデータ構造が異なる場合の対応について考える。

6.1 本研究で JSP を適用した情報システムの利点
 本研究では、JSP をオブジェクト指向言語に応用する。JSP を用いたときの利点を以下に示す。

1. プログラムの再利用ができる
 本研究では JSP を用いて、情報システムの入出力のデータからプログラムを導く。その際に、オブジェクト指向のデザインパターンを用いることで、作成したプログラムが再利用できる。
2. 入出力の木の写像が自動的に決まる
 情報システムの共通なデータ構造を抽出し、メタ情報として定義したことで、入出力の木の写像が自動的に決まる。入出力の木の写像が自動的に決まることから、プログラムの木も自動的に決まり、プログラムが自動生成できる。

以上のことから、メタ情報を入力として、プログラムの異なる部分を自動的に作成できるので、情報システム開発者は、メタ情報を作成するだけで、情報システムを実現することができる。

6.2 Command の組み合わせについての考察
 Command パターンは複数の機能を組み合わせることで、新しい機能を追加することができる。本研究の情報システムは、複数の Command を組み合わせることで実現している。例えば、情報システムの検索は、利用者は検索したい項目と検索キーを入力することで、SearchCmd を用いて、一致したレコード型の要素を集合として返す。返された集合に対して再び SearchCmd を利用することで AND 検索をすることができる。また、削除は、削除するレコード型の要素を探すときに SearchCmd を用いる。

以上のことから、Command を組み合わせることで新たな機能を追加できると考えられる。例えば、情報システムのデータを修正するとき、検索と登録の Command を組み合わせると実現できると考えられる。

6.3 構造不一致の場合の考察
 情報システムの機能の拡張として、一覧表示したときの出力の表示形式を自由に変更できる機能を考える。しかし、表示形式をかえると、入出力間のデータ構造が一致

しない場合が考えられる。構造不一致の問題を解決を考える。以下の手順のよって、入出力間の構造不一致の問題を解決できると考える。

1. 本研究の構造不一致になる場合をパターン化する。
 構造不一致になる場合は以下の 2 通りがあると考えられる。(図 10)
 - 表示項目の順序による不一致
 - 入力データの結合による不一致
2. それぞれの構造不一致なデータ構造に対して処理するプログラムを作成する。
3. 出力のメタ情報を定義する。
 入力メタ情報と出力メタ情報を比較することで、木のどの要素の部分で、どのパターンの構造不一致なのかを調べる。

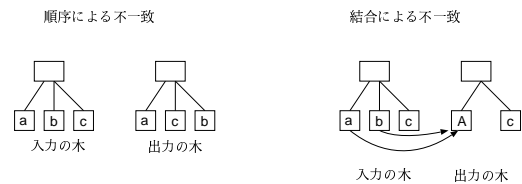


図 10 構造不一致の例

7 おわりに

本研究では、情報システムのプログラムの類似性とデータ構造の類似性から、メタ情報を作成した。メタ情報を作成するだけで情報システムを実現することができることから、情報システム開発者はプログラム作成の手間を大きく軽減することができる。今後の研究課題を以下に示す。

- ・XML データをデータベースに格納して情報システムの実現。

謝辞

本研究を進めるにあたり、二年間御指導いただいた野呂昌満教授、有益なアドバイスをいただいた張漢明助教授、蜂巢吉成講師、親身になって相談ののってくださった大学院生の熊崎敦司さん、森貴彦さん、藤原泰昌さん、後藤修平さんに深く感謝致します。

参考文献

- [1] Elliotte Rusty Harold : XML パイプル, 日経 BP 社, 2001
- [2] Jackson, M.A : Principles of Program Design, pp. 299, Academic Press, London(1975). (邦訳 鳥居宏次: 構造的プログラム設計の原理, 日本コンピュータ協会 (1980))
- [3] Tim Bray and et al : W3C: Extensible Markup Language (XML) 1.0 (Second Edition) <http://www.w3.org/TR/REC-xml>
- [4] W3C, Document Object Model (DOM) <http://www.w3c.org/DOM/>