

拡張可能な職業訓練履歴管理システムの研究

大野邦夫，須藤僚，荻原省吾，佐飛成幸，高橋雅也，竹林辰弥

職業能力開発総合大学校

ISO/TC232により非公式教育の標準化が進展している。その基本的な枠組は、品質に対するISO9000シリーズと類似であり、品質方針に対して教育訓練方針、品質手順に対して、組織・スタッフ・カリキュラムの構成、品質データに対して、教育訓練評価データが対応すると予想される。本報告では、教育訓練評価データを教育訓練対象者の履歴書と関係づけ、系統的に評価・参照することを可能とするシステムを検討した。システムの構築にCLOS (Common Lisp Object System) を用い、CLOSのオブジェクトとして管理されるデータをS式に変換し、さらにXMLに変換することにより、拡張性に優れた履歴書管理と教育訓練評価手法を目指す。

A Study on an Extensible Customizable Vocational Education and Training History Management System

Kunio Ohno, Ryo Suto, Shogo Ogiwara, Shigeyuki Sabi, Masaya Takahashi, Tatsuya Takebayashi

Polytechnic University

Standardization of non-formal education has been conducted by ISO/TC232. The basic methodology is the same as the quality system in ISO9000 series. Educational and training policy will correspond to quality policy, organization, staff, and calculi management will correspond to quality rules and processes, and educational and training data will correspond to quality data. The goal of this paper is to develop a system to retrieve and evaluate the vocational education and training data related to his or her resume. Common Lisp Object System (CLOS) has been chosen for the modeling language. We are planning an extensible vocational education and training document management system to convert CLOS instance to XML data through symbolic equation of lisp.

1. はじめに

情報技術が生活環境を変えつつある。特にインターネットの普及で情報環境はグローバル化し、そのインパクトで社会制度も変化を要求される。日本社会における就業形態、すなわち終身雇用から転職社会への変化もその一つであろう。転職が当たり前の社会になると、個々人にとっても採用する企業などの組織にとっても履歴書が重要になる。従って今後の雇用関係においては、履歴書とその関連情報が重要になることが予想される。そのためには履歴書の標準化が重要な課題になる。厚生労働省は履歴書の標準化とその精査を目的としてジョブカード制度を発足させた。ジョブカードは、従来の個人が書いていた履歴書をキャリアコンサルタントが評価し、さらに職業訓練におけるデータも包含させることを可能としている。

現状のジョブカードは電子化を前提にしたフォーマットではない。しかし今後はこのような個人の職業

やスキルを記述する履歴書情報がデータベース化され、それに基づいて求人・求職が行われるようになるであろう。米国に本拠を置くHR-XMLコンソーシアムは、すでに求人情報、求職情報、仲介者情報のフォーマットをWebの標準言語であるXMLを用いて標準化し、雇用ビジネスを推進している⁽¹⁾。EUでは、履歴書情報を含む個人的なスキルをEuropass CVとしてWebに登録し、EU内部で流通させ雇用を推進する制度を実施している⁽²⁾。

ところで、これらのXMLによるフォーマットは、固定的で関連データとの連携や拡張に対する考慮は払われていないが、今後はデータへの拡張性が要求されると考えられている。ISOは、非公式教育の標準化に取り組み、技術委員会のTC232では個人の職業スキルを国際的に流通させるシステムに取り組み、雇用関係のグローバル化への対処を進めている⁽³⁾。その枠組みは品質標準のISO9000シリーズを踏襲しており、品質方針が教育・訓練方針に、品質手順が、組織・ス

トップ・カリキュラムなどの構成に、品質文書が履歴書や成績、教育・訓練評価などに相当するようになると予想される。

今後、非公式教育標準化の国際的な取り組みが進展すると、履歴書情報や教育・訓練歴、さらに個人のスキル等に関する系統的かつ効率的な管理が要求されていくと考えられる。そこで、履歴書と関連情報の効果的な管理のための拡張可能な履歴書管理システムの検討を試みることにした。

2. 履歴書モデル

2.1 UMLによるジョブカードのモデル化

まず履歴書情報の明確化を行う必要があると考えた。履歴書は、JIS Z8303「帳票の設計基準」により内容の大枠が規定され、ジョブカードの様式1における総括表もその枠組に準拠している。ジョブカードの場合は、様式1の情報をさらに詳細化するために様式2以降で詳細な記述と共に、キャリアコンサルタントの記述欄や職業訓練履歴のデータなどが追加されている。

厚生労働省が管理するWebサイトでジョブカードの具体的内容が紹介されている⁽⁴⁾。履歴書情報の基本的内容は図1の様式1により記述されている⁽⁵⁾。

図1は「様式1 ジョブカード (総括表)」のサンプルデータを示しています。この表は、氏名、住所、電話番号、就業先・職務経歴、学歴・訓練歴などの項目を含みます。また、表にはいくつかの注釈が添えられています。

年月	就業先・職務経歴 等	特記事項
平成8年4月～ 9年3月	厚労物産株式会社 営業販売 (正社員)	自己都合退職
平成10年6月～ 12年3月	東洋プラス株式会社 製品のバリ取り作業 (アルバイト)	雇用期間満了
平成13年6月～ 16年10月	有限会社いろは電機工業 機械組立、製品検査等 (契約社員)	契約期間満了
平成18年1月～ 現在	凹凸コンピュータ販売株式会社 パソコン販売 (アルバイト)	

学歴・訓練歴

年月	教育・訓練機関名、学科(コース)名 等
平成4年3月	神奈川県立青葉高等学校普通科卒業
平成4年4月	東都大学経済学部経済学科入学
平成8年3月	東都大学経済学部経済学科卒業

図1 様式1のサンプルデータ

図1のジョブカードの様式1を分析した結果、図2に示すUML (Unified Modeling Language) のクラス図

が得られた。UMLは、オブジェクト指向技術に関する標準化団体であるOMG (Object Management Group) が標準化したオブジェクト分析設計とそのモデル化に関する規定で、種々の業界のシステム構築で幅広く用いられている。

オブジェクト指向プログラミングはクラス継承機能を持つことが大きな特徴である。クラス図における個々のブロックは、クラスを示している。内部が区分されているブロックは、上部がクラス名、下部がそのクラスに含まれる項目 (インスタンス変数) である。三角のマークは頂点で示すクラスを下位のクラスが継承する関係を示している。図2では、総括表1のクラスは、対象者基本情報、コンタクト情報、職務経歴、学習歴・訓練歴のクラスを継承していることを示しており、同様に総括表2のクラスは、アピール情報、資格・免許、社会体験活動歴を継承していることを示している。アスタリスク (*) が付けられているクラスは、複数のリストとなり得る場合を示す。

2.2 拡張機能の必要性

履歴書情報は個人の職業能力を提示する基本的な情報として位置づけられるが、履歴書の信頼度を向上させ有効性を増加させるためには、関連情報に対する拡張性が重要である。

情報の拡張性を実現する手段としては、関係データベースやハイパーリンクなどの技術もあるが、前者は、フラットな表形式のデータの管理にしか向いておらず、後者は拡張に伴うリンクの維持管理が問題となる。オブジェクト指向におけるクラス定義を用いると、定義されたクラスの性質をそのまま継承するサブクラスを定義し、さらに独自の機能を追加することが出来る。そのためにシステムにおける機能の拡張のためには好ましい特徴を持つ。特に履歴書のような、多様で逐次追加されるデータを持つような場合には最適であると考えられる。

3. モデルの実装

3.1 XMLによるモデルの検討

図1に示す労働一郎さんについて⁽⁶⁾、サンプルデータの一部をXML化すると、図3のように示される。

拡張機能を提供するためにXMLでは、DOM (Document Object Model) をインタフェースとするアプリケーションプログラムとの連携を必要とし、アプリケーションのために通常はJavaやJavaScriptによるプログラムが使われる。例えば、図3のXMLデータ (ファイル名Rodo_Ichiro.xml) をJavaScriptを用いてDOMインタフェースで処理する場合は、図4のようなプログラムを記述する必要がある⁽⁷⁾。

図4のプログラムは、最上位の次の階層の要素をアラート表示で逐次取り出す機能を実現している。以上

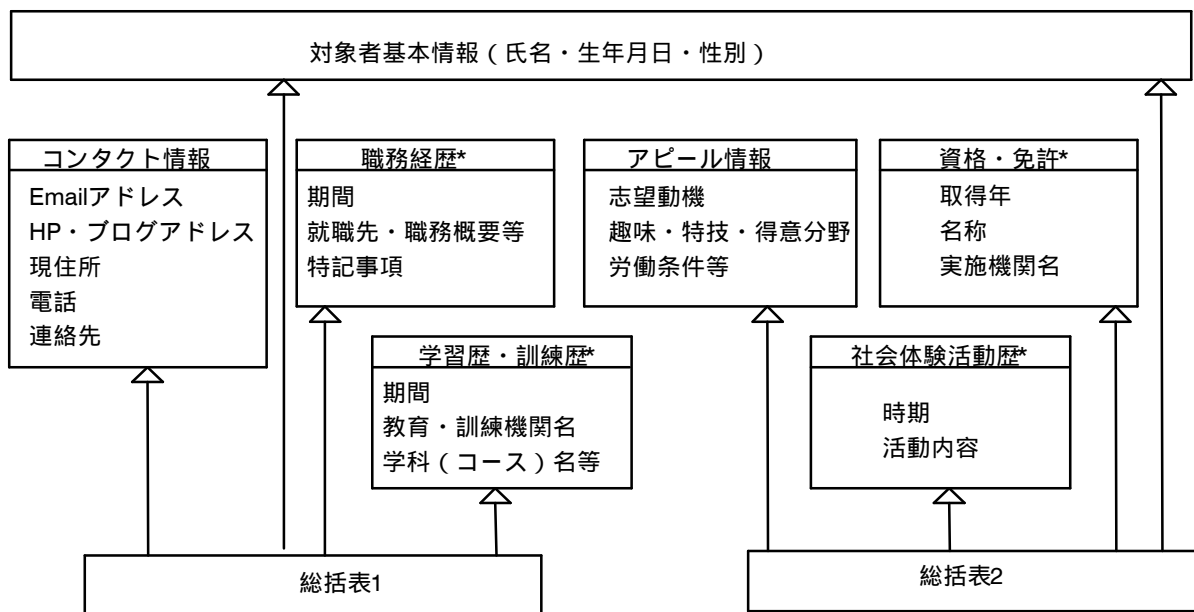


図2 ジョブカード様式1の情報構造

```

<?xml version="1.0" ?>
<integrated_1>
  <name>"労働一郎"</name>
  <email_address></email_address>
  <hp_blog_url></hp_blog_url>
  <birth_date>"1973-12-07"</birth_date>
  <gender>male</gender>
  <living_address>"神奈川県港南市東区朝日4-2"</living_address>
  <phone>"045-678-9012"</phone>
  <contact_address>"神奈川県港南市東区朝日4-2"</contact_address>
  . . . . .
</integrated_1>

```

図3 労働一郎さんのXMLフォーマット

は単純な例で、Web環境に埋め込まれたツールで処理可能だが、本格的なDOMのためのプログラム環境は、種々のツールやライブラリの支援を必要とし、その環境の専門家でない限り効果的なシステム構築は難しい⁽⁸⁾。さらにXMLとDOMによる処理は、オブジェクト指向による継承機能が使えないので、付加的な情報の記述が煩雑になり、多様な可能性を追求するプロトタイプシステムとしては必ずしも適合しない。従って、モデルベースで柔軟な拡張性を実現するシステムにとって以上のようなXMLの欠点は無視できない。

3.2 Lispによるモデルの実装

3.2.1 データ記述と処理

オブジェクト指向を効果的に活かせるプロトタイプの枠組みとしてLisp言語による処理を検討した。

Lisp言語は、XMLと同様な木構造データを括弧で階層化するS式で記述することが可能である。Lisp以外のプログラム言語でも木構造を記述することは可能であるが、XML以上に木構造のデータをコンパクトに記述できる言語はLisp以外には見当たらない。XML自体が開始タグと終了タグにより木構造を明示的に容易に記述できるからこそ、普及したと言える。

基本的には、XMLの開始タグがLispの開き括弧”(”に対応し、終了タグが閉じ括弧)””に対応する。

なお、XMLとLispのS式の対比に関しては、先行研究が存在する。紙名と玉井は、S式をXMLに変換する手法を検討している⁽⁹⁾。沖ソフトウェア(株)は、動的なインタフェース記述のためのDUELという言語をLispのS式とXML双方で記述可能とし、共通のモデルとしている⁽¹⁰⁾。平成12年度未踏ソフトウェア創造事業で「S式を用いたXMLサーバプログラミングツール」

```
<html>
<head>
<title>Jobカードの項目</title>
<script language="JavaScript">
<!--
var objDoc=new ActiveXObject("Msxml2.DOMDocument");
objDoc.async=false;
objDoc.load("Rodo_Ichiro.xml");
var objRoot=objDoc.documentElement;
var clnCld=objRoot.childNodes;
for(i=0;i<clnCld.length;i++){
    var objNod=clnCld.item(i);
    window.alert(objNod.xml);
}
//-->
</script>
</head>
<body>
<h1>DOMでJobカードの項目を抽出する</h1>
</body>
</html>
```

図4 DOMによるプログラム例

というテーマが採択され、成果がオープンソースとして公開されている⁽¹¹⁾。

図3のXML記述と同じ情報をLispで定義すると図5のようになる。冗長な開始タグ、終了タグのペアを必要

とする図3のXMLデータに比べると簡潔な記述になる。

```
(setf rodo_ichiro
  '((name "労働一郎")
    (email_address nil)
    (hp_blog_url nil)
    (birth_date "1973-12-07")
    (gender 'male)
    (living_address "神奈川県港南市東区朝日4-2")
    (phone "045-678-9012")
    (contact_address "神奈川県港南市東区朝日4-2")
    . . . . .
  ))
```

図5 Lisp言語 (S式) による労働一郎さんのフォーマット

XMLとS式による記述量を比較してみた。XMLの場合は、44行1199文字(半角換算)であった。それに対し、S式の場合は31行884文字となり、それぞれ、70.5%、73.7%に減少している。なお、文字数に関しては、スペースと改行は1文字として計算したが、インデントは無視した。

Lisp言語では、S式自体がプログラミング言語になるので、XMLの場合のようにDOMによるインタフェースを介してXMLファイルを読み込むアプリケーションプログラム環境とその処理は不要である。図4におけるJavaScript言語と同様な機能を実現するLisp関数は、下記のように極めて簡単である。

```
(defun first-level-print (x)
  (if (null x) nil
      (progn
        (print (car x)) (first-level-print (cdr x))))))
```

図5のデータの次の階層を逐次出力するには、定義されたfirst-level-printの関数に引数となる変数rodo_ichiroを与えて実行させれば良い。そのためには(first-level-print rodo_ichiro)をリスナ(Lisp言語の実行環境)上にタイプして実行させれば結果が得られる。

3.2.2 オブジェクト指向による継承の効果

オブジェクト指向拡張がなされたLisp言語であるCLOS(Common Lisp Object System)は、クラス継承機能が使用できるのでサブクラスを定義することにより付加的な情報の系統的な記述が可能となる。その結果、XMLの場合はもとより、S式の場合に比べても拡張性に富んだ効率的なシステム構築が可能になると考えられる。以上の考えに基づきシステムの構築を試みた。

拡張機能を実現するには、図5のようなフラットな構造のデータを定義するのではなく部品化された個別の情報を階層的に定義する手法が用いられる。図5の場合を例にすると、個人の基本情報として、氏名、誕生日、性別を定義する。基本情報は、個人を特定するために様々なデータから必要とされるからである。

```
(defclass person ()
  ((name :initform "労働一郎")
   (birth-date :initform "1973-12-07")
   (gender :initform 'male)))
```

defclassはクラス定義のためのマクロ命令で⁽¹²⁾、personという名前のクラスを定義する。personの次の括弧は、スーパークラスを記述するためのスロットで、この場合は空であるのでpersonが最上位のクラスとなることを意味する。上記クラス定義におけるname、birth-date、genderはインスタンス変数で、クラスの具体的なオブジェクト(インスタンス)が持つ変数の名称である。:initformは、生成されるインスタンスのデフォルト値を示す。

次にコンタクト情報、職歴情報、教育歴情報のクラスを定義する。

```
(defclass contact-info (person)
  ((email :initform nil)
   (hp-blog-url :initform nil)
   (living-address
    :initform "神奈川県港南市東区朝日4-2"))
```

```
(phone :initform "045-678-9012")
(contact-address
 :initform "神奈川県港南市東区朝日4-2"))
```

この場合には、personをスーパークラスとして継承するので、contact-infoクラスのインスタンスを生成すると、personで定義されたインスタンス変数とデフォルト値も含めたインスタンスが生成される。

3.2.3 クラス階層の詳細化と実装

前項で述べた継承の概念は単純なモデル化のための説明であり、実際のシステム化のためには状況に応じた最適なクラス階層化が要求される。そのようなクラス階層の構築のためには、UMLによるオブジェクト分析設計を通じたクラス図を用いるのが一般的である。本研究の場合も図2のクラス図に基づきプログラミングを行った。

CLOSによるクラス定義には、ミクシン(mix-in)と呼ばれるLispマシン以来の特徴ある多重継承機能を用いることが可能である。ミクシンの概念はクラスをフレーバと呼んだZetalisp文化の産物である⁽¹³⁾。かつてMITのLispプログラマ達は、バニラアイスクリームに、種々の香料(フレーバ)を追加して自分好みのアイスクリームを作り上げてそれを味わいながらプログラミングしていた。Xerox PARCのSmalltalk-80が提示したオブジェクト指向プログラミングパラダイムをLisp言語に導入するに当たり、彼らはアイスクリームに対するフレーバの混入、すなわち多重継承におけるミクシンを基礎概念としたことからLisp言語における独特なオブジェクト指向手法が確立された。

その結果、Lisp言語におけるオブジェクト拡張においては、スーパークラスを持たない素のクラスをバニラ(vanilla)と呼び、一般にはクラスで表現される概念をフレーバと呼び、種々の概念を多重継承により拡張していく手法をミクシンと呼んだのであった。これらの用語はZetalispとその関連言語の利用者で用いられたが、Common Lispの制定によりフレーバという用語が使用されなくなってからはあまり使用されなくなった。

抽象的な概念をクラスとして定義し多重継承で混合する過程を通じて徐々に具体化していく手法は、ここで検討対象とする履歴書のような場合には適合すると考えられる。

履歴書のような個人を扱う情報の場合、先ず個人から独立した基本概念が存在し、先ずそのクラスをバニラクラスとして定義する。基本概念を、具体的な個人のクラスとして定義するには、バニラクラスのサブクラスとしてデフォルト値を持つサブクラスを定義する。デフォルト値を持つクラスは、バニラクラスの具体的なパーツとなりそのようなサブクラスをミクシン手法により多重継承するサブクラスを定義することにより、全体的な履歴書情報が形成される。

オブジェクト指向プログラミングでは、概念をクラスとして定義し、その具体的なオブジェクトをインスタンスとして定義するが、具体的なインスタンスを継承して新たなインスタンスを生成することは出来ない。だがCLOSではデフォルト値を持つクラスをあたかも具体的なインスタンス変数値を持つインスタンスのように扱うことができる。しかもCLOSのクラスは多重継承が可能なので、ミクシンにおける概念部品として扱うことが可能となる。この機能とその活用法は、CLOSのみならず、Zetalisp⁽¹³⁾や日本のNTTが開発したTAO⁽¹⁴⁾のようなオブジェクト指向化されたLisp言語の特徴でありメリットであったと言える。

3.3 S式からXMLへの変換

Lisp言語はWebやインターネットを活用する環境においては必ずしも普及していないので、Webに表示するためのデータをS式のままで管理するのは必ずしも妥当とは言えない。そのために、S式によるデータをXMLに変換するLispの関数を用意し、必要とあらば随時XMLデータに変換することを検討した。なお、S式をXMLに変換する手法に関しては既に具体的な検討がなされている⁽⁹⁾。

図5の労働一郎さんのS式データをXML化する場合を例に説明する。(name "労働一郎")というS式データを、<name>"労働一郎"</name>というXMLフォーマットに変換するには、

```
(defun xml-line (tag value)
  (format t
    "~<~A>~A</~A>~%" tag value tag))
```

という関数を定義し、(xml-line '(name "労働一郎"))をリスナ上で実行すれば良い。これを図5で定義されたrodo_ichiroという変数に適用するには、タグと値とのペアが縦列的に記述されたS式に適用すれば良いので、下記の関数を定義する。

```
(defun xmlize (x)
  (if (null x) nil
      (prog1 (princ (xml-line (caar x) (cadar x))
        (xmlize (cdr x))))))
```

この関数を用いて、(xmlize rodo_ichiro)をリスナ上で実行すると、図3に示したのと同様なXMLデータが得られる。この手法を用い、Lispの処理系からそのデータ構造をXMLフォーマットで出力させることは、容易である。これは単一階層のフラットな表現に過ぎないが、階層関係を有するS式についても上記の関係を再帰的に適用することにより可能となる。

3.4 総括表2のモデル化

前章の方法を用い、総括表2についてもXML、S式、CLOSによる実装を試みた。前章と同様に行数と文字

数の比較を行ったところ、XMLは26行660字、S式は18行(69.2%) 489字(74.1%)であった。

3.5 様式2以降のモデル化

以上述べてきた方法を用い、ジョブカードの様式2以降についてもXML、S式、CLOSによる実装を試みた。CLOSを用いると、拡張するためのクラス定義は差分だけを記述すれば良いので少なくともすむ。

3.6 拡張性に好ましいシステム環境

前項までの検討から、履歴書システムの拡張部分については、XML、S式、CLOSの順に記述量が減少することが判明した。しかしながら、CLOSは最初の基本的な情報の定義においては記述量が多くなる。具体的に行数を調べてみると表の通りであった。

この表では、様式4、5のデータも含めて記載している。この表からXMLとS式に関しては対応関係が存在することが分かる。S式の行数は各項目に対してXMLの場合の約70%程度である。

表1 XML、S式、CLOSによるプログラム行数の比較

	XML	S式	CLOS
汎用基本情報	-	-	28
総括表1	44	31	58
総括表2	26	18	24
様式2	32	25	18
様式3	18	12	8
様式4	12	8	7
様式5	8	6	5
合計	140	100	148

それに対し、CLOSの場合は、初期の定義においては多くの行数を要し、後の追加定義においては少ない行数で済んでいる。ジョブカードの様式1から様式5までの記述では、全体の行数はCLOSが最大で、XML、S式の順で行数が減少しているがシステムがさらに拡張されるとCLOSの行数が減少し、最小になると考えられる。すなわち、拡張部分である様式2~5においては、CLOSの行数が最も小さいからである。

4. 検討結果の総括

以上の検討により、下記のことが明らかになった。

(1) XMLフォーマットのデータよりもLisp言語のS式の方が、プログラム行数について約30%程度の削減が可能で、冗長性の少ないコンパクトな記述が可能である。

(2) 履歴書データの拡張は、XMLの場合はDOMを用いて別の言語でプログラミングを行う必要があり煩雑になるが、Lispの場合は、自身のプログラミング機能を用いて関数定義やメソッド定義で処理できるので拡張が容易で融通性がある。

(3) UMLのクラス図で記述されるモデルの実装には、オブジェクト指向化されたLisp言語であるCLOSを用いると、クラス階層を自然にモデル化することが可能となり、モジュール化の観点でXMLおよびS式による実装よりも優れている。

(4) CLOSのミクシン手法は、サブクラスをオブジェクトの部品として組み合わせ、目的に応じて組み合わせを柔軟に変更して再利用することが可能なので、柔軟で多様なプロトタイプ構築に適している。

(5) LispによるS式データをXMLに変換するための基本的な関数を定義したので、本システムをベースとする履歴書データのWebへの表示やアプリケーション、さらにWebサービスへの適用なども容易である。

上記の総括から、種々の関連情報と連携可能な履歴書情報の管理のプロトタイプシステムの開発のためには、オブジェクト指向拡張されたプログラミング言語のLispを用いる方法の有効性が明らかになった。

なお、Lispを用いて履歴書を記述する場合でも、単にS式で記述する場合とCLOSのクラス階層を用いるオブジェクト指向の場合とを分けて考える必要がある。前項のまとめにおける(1)(5)項は前者に対応し、(3)(4)項は後者に対応する。(2)項は共通である。以上を考慮して、XML記述も含めて比較すると表2のようにまとめられる。以下個別項目について解説を加える。

表2 履歴書データ記述に関するXML、S式、CLOSの比較

	XML	S式	CLOS
記述情報量	×		(注)
クラス図との対応	×	×	
拡張性	×		
モジュール性	×		
情報隠蔽	×	×	
Webとの相性			×
習熟の容易さ	×		

(注) CLOSの場合は、クラス定義をして記述せねばならないので、共通部分の記述が多くなるが、拡張部分に関しては差分記述で済むのでS式の場合に比べ却って少なく済む。

5. 考察

5.1 履歴書とライフログとの連携

本研究は、アドレス帳や電話・メール等の通信履歴、スケジュール管理機能などのPIM情報を活用するオントロジの構築と、データセンターでPIMオントロジを活用して個人を支援するネットワークコンシェルジュに関する研究の応用として始めた⁽¹⁵⁾。最近、携帯電話のGPS等を使い個人の居場所を基準としたライフログシステム活用の研究が盛んである⁽¹⁶⁾。これらのライフログ情報を一括管理し実世界でのビジネスや社会生活に役立てる検討も試みられている⁽¹⁷⁾。

上記のようなライフログの取組に比べると、履歴書の職歴、学歴、ボランティア歴などは時間範囲の領域が異なるがライフログの一種である。現在の履歴書は、将来的には前記ライフログと連携する可能性がある。従ってこれらのデータと連携可能なデータモデルで履歴書のデータを管理可能な構造とすることが望まれる。そのような観点から、本研究では拡張可能で、様々な関連データを取り込める具体的なシステムをCLOSを用いて実装しその有効性を明らかにした。

5.2 多重継承の適用領域

拡張可能な履歴書管理システムの環境としては、CLOS環境が多くの方が好ましいことを明らかにしたが、その機能の多くはLisp言語におけるオブジェクト指向プログラミング技術に負っている。

ライフログを活用する最近の研究の多くは、XMLによるデータフォーマットを用いている。それらのデータを有機的に統合するためには、Webオントロジ言語のOWLが使用される場合が多い。OWLが使用される理由は、XMLによるクラス継承、プロパティ継承が可能な点にある。しかしOWLにおいては、深い知識としてのオントロジ機能が必ずしも適確に使用されているとは思えない。その理由は、別のプログラミング言語による推論エンジンの機能的な制約にあると考えられる。その点、本研究で検討対象としたCLOSは、データと処理が統合されており、オントロジ機能を実装するプログラミング環境としては優れていると言える。

5.3 CLOSからXMLへの変換

今後の情報システムがWebブラウザを標準的なインタフェースとするクラウド的な枠組となることを想定すると、CLOSによるモデルをXML化してWeb用のデータとすることが重要な課題である。S式からXMLへの変換は比較的容易であるが、CLOSによるカプセル化されたインスタンスを、Web用のXMLデータに変換するための一般的な手法は難しい。特定のクラスのインスタンスに閉じた変換法から検討する必要がある。具体的なサービスのためのプロセスとモデルをUMLベースで考慮すると種々の可能性が考えられる。

ISO/TC232における教育・訓練文書としての活用としてのジョブカードの各種様式がさしあたりの検討対象となるが、それだけでなく、履歴書データとライフ

ログとを統合することにより付加価値を生じる具体的なサービスを検討し、その分析に基づくドメインに特化

した手法の研究が必要と考えられる。その開発プロセスを図6に示す。

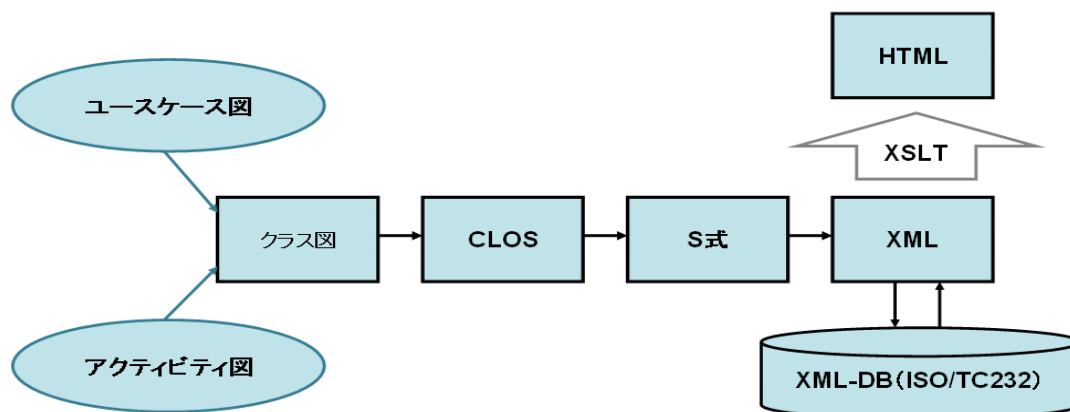


図6 UMLに基づくCLOSからXMLへの変換とXML-DBによる実装

6. おわりに

本テーマを当初卒業研究で検討し、基本的なプログラムを開発してくれたインドネシアからの留学生デウィヘラワティさんに感謝します。さらに本研究は、株式会社インターネットイニシアティブと協同で進めているネットワークコンシェルジュ関連の研究の応用であり、支援していただいているIJ研究所の新麗主任研究員に謝意を表します。

文献

- (1) HR-XML Consortium; "HR-XML Consortium Library, 2007 April 15", <http://www.hr-xml.org/>
- (2) National Europass Centres ; "The Europass Curriculum Vitae (CV)", <http://europass.cedefop.europa.eu/europass/home/vernav/Europass+Documents/Europass+CV.csp>
- (3) 額田順二; "非公式教育訓練分野の国際標準化に向けたISOの取り組み", 人材育成と教育サービスの国際化シンポジウム - Symposium on Internationalization of Human Resources Development and Educational Services, pp.3-12, (2009.9)
- (4) 厚生労働省; "ジョブカード制度のご案内", http://www.mhlw.go.jp/bunya/nouryoku/job_card01/index.html
- (5) 大野邦夫, デウィヘラワティ, 須藤僚; "情報社会における職業能力開発-ジョブカードの分析・モデル化と国際標準化動向の検討", 信学技報 AI2008-34(2008.11)
- (6) 厚生労働省; "ジョブカードサンプル労働一郎", http://www.mhlw.go.jp/bunya/nouryoku/job_card01/dl/roudou_ichirou.pdf

- (7) 山田祥寛; "10日でおぼえるXML入門教室(第2版)", 翔泳社, (2004)
- (8) 大野邦夫; "XMLとS式による情報記述と把握理解の比較", 画像電子学会VMA研究会報告(2009.7)
- (9) 紙名哲生, 玉井哲雄; "Lisp を基にした新しいXMLプログラミングツール実現手法", 情報処理学会情報学シンポジウム講演論文集(2002)
- (10) 沖エンジニアリング(株); "動的ユーザインタフェース環境記述言語DUEL(TM)とそのエンジンについて", <http://www.okisoft.co.jp/esc/DUEL/DUELabstract.pdf>
- (11) 湯浅太一; "S式を用いたXMLサーバプログラミングツール", 平成12年度未踏ソフトウェア創造事業採択案件評価書, <http://www.ipa.go.jp/NBP/12nendo/12mito/mdata/12-7h.htm>
- (12) Guy L. Steel Jr., (井田昌之監訳); "Common Lisp 第2版", 共立出版, (1991)
- (13) Hank Bromley; "LISP LORE: A Guide to Programming the Lisp Machine", Kluwer Academic Publishers, pp.17-41, (1985)
- (14) 山田康宏, 大野邦夫, 日比野靖, 竹内郁雄; "AIワークステーションELISの検討". 情報処理学会マルチメディア通信と分散処理研究会資料, 31-2, pp.1-7 (1986).
- (15) 大野邦夫, 須藤僚, 新麗; "ネットワークコンシェルジュの検討", 信学技報OIS2008-19 (2008.7)
- (16) 小塚宣秀; "ユビキタスネットワーク技術の研究開発 ~ ケータイ de ライフログ ~", 情報通信 BULLETIN, No.059, (2008.6)
- (17) 矢野和夫; "ライフログ経験: センサが人生を変える", 情報処理, Vol.50, No.7, July 2009, pp624-632