鹿児島大学リポジトリ
Kagoshima University
Repository

コンパイラ・コンパイラを用いたメタデータ作成支援システムの開発

著者	徳守 淳也, 小野 智司, 木場 隆司, 北山 信一, 中山 茂
雑誌名	情報知識学会誌
巻	17
号	1
ページ	41-50
発行年	2007
別言語のタイトル	Development of Metadata Generation Support System Using Compiler Compiler
URL	http://hdl.handle.net/10232/1135

コンパイラ・コンパイラを用いたメタデータ作成支援システムの開発

著者	徳守 淳也, 小野 智司, 木場 隆司, 北山 信一, 中山 茂
雑誌名	情報知識学会誌=Journal of Japan Society of Information and Knowledge
巻	17
号	1
ページ	41-50
別言語のタイトル	Development of Metadata Generation Support System Using Compiler Compiler
URL	http://hdl.handle.net/10232/00003377

コンパイラ・コンパイラを用いた メタデータ作成支援システムの開発

Development of Metadata Generation Support System Using Compiler Compiler

徳守 淳也[1]，**小野 智司**[1*]，**木場 隆司**[2]，**北山 信一**[2]，**中山 茂**[1]
Junya TOKUMORI, Satoshi ONO, Takashi KOBA, Shinichi KITAYAMA, and Shigeru NAKAYAMA

[*1]鹿児島大学 工学部 情報工学科
Department of Information and Computer Science, Faculty of Engineering, Kagoshima University
E-mail: {sc102038, ono, shignaka}@ics.kagoshima-u.ac.jp

[2] 鹿児島大学 附属図書館 情報管理課
Division of information management and processing, Kagoshima University Library
E-mail: {koba,iria}@lib.kagoshima-u.ac.jp

学術情報の円滑な流通を図るため，国立情報学研究所（NII）の主導のもと，大学などの研究機関において学術リポジトリの構築が進められている．しかし，学術情報のメタデータ作成は労力がかかるため，より簡便にメタデータを作成できるツールやサービスの開発が望まれている．本研究では，学術機関リポジトリの構築支援を目的として，理工系の研究者の間で研究者に広く用いられている BIBTEX 形式のメタデータを，NII メタデータ記述要素に準拠するメタデータに変換するシステムを開発する．提案するシステムを利用することで，メタデータ記述の労力を 5 分の 1 程度に抑えることができる．また，コンパイラ・コンパイラを利用することで，開発の労力を抑えることができた．

Many universities and other research organizations have proceeded construction of academic resource repositories under the leadership of National Institute of Informatics (NII). Because it requires much labor to create metadata contents of academic resources, development of tools or services enabling to easily create metadata contents is expected. This paper proposes a system which converts BIBTEXmetadata of academic resources – widely used by researchers in science and engineering – into NII metadata format in order to support constructions of academic resource repositories. The proposed system enables to reduce almost four-fifth of workloads of creating metadata contents. In addition, utilizing compiler compiler to develop the proposed system reduced the develop cost.

キーワード：メタデータ作成支援，コンパイラ・コンパイラ
Metadata generation support，Compiler compiler

1　はじめに

国立情報学研究所（National Institute of Informatics: NII）では現在，国内外の有用な学術情報資源との連携を目標とした学術コンテンツ・ポータルの構築を進めている[1–3]．国内の大学・研究機関がインターネット上で発信している学術情報資源の二次情報をデータベース化することにより学術情報の円滑な流通を図り，各大学の研究成果を広く世界に発信することを支援することを目的としている．各大学でも機関リポジトリに関する様々な取組みが行われ[4–6]，NII が提唱するダブリンコア（Dublin Core: DC）[7] を拡張したメタデータ記述要素[8] に従ってメタデータ（JuNii 形式）を作成し，メタデータ交換技術（Open Archives Initiative Protocol for Metadata Harvesting: OAI-PMH）[9–11] を通じて NII にメタデータを提供している．しかし，メタデータの作成は労力がかかり，タグに関する知識や記述が不要なツールを用いても，人手で作業を行うと 1 件あたり 5 分程度の時間を要する．このため，より簡便にメタデータを作成できるツールやサービスの開発が望まれている．

本研究では，学術機関リポジトリの構築に向けたメタデータの作成支援を目的として，BIBTEX 形式のメタデータ[12,13] を，NII メタデータ記述要素に準拠する JuNii 形式メタデータに変換するシステムを開発する．

BIBTEX は，理工系の研究者の間で広く利用されている文書作成システム TEX に付随する文献管理システムである．近年，国立情報学研究所 論文ナビゲータ CiNii[14] や，CiteSeer.IST[15]， OXFORD JOURNALS[16] ， American Association for the Advancement of Science[17]，Association for Computing Machinery(ACM) PORTAL[18]，

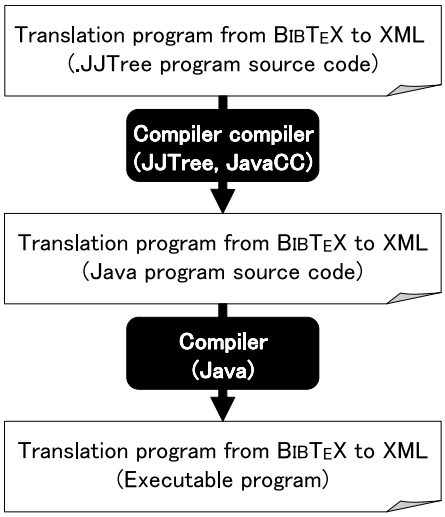

図 1: Develop flow.

IEEE Computer Society[19] などの様々な学術情報ポータルにおいて，BIBTEX メタデータの提供を行うポータルが増加しており，BIBTEX の有用性を高めている．提案するシステムは，BIBTEX 形式のメタデータを利用することで，最低限の手作業により NII メタデータ記述要素に準拠するメタデータを作成することができる．

また，本システムは，コンパイラ・コンパイラ[20–22] を利用することで開発の省力化を図っている．コンパイラ・コンパイラは，パーサジェネレータとも呼ばれ，実用的なコンパイラの作成からシステム定義ファイルの解析などの一般的なテキスト処理まで，幅広い用途で有用なツールである．著者らはこれまでに，分散並列処理プログラミング言語 Espace を提案し，Java を基本とするコンパイラ・コンパイラ JavaCC[22] を用いて Espace のコンパイラを開発した[23]．

メタデータ変換システムの開発にコンパイラ・コンパイラを用いる利点は，以下の 3 点である．

- **プログラムを宣言的に記述することができる．** このため，直観的なコー

```
<Start> ::= <CompilationUnit> EOF
<CompilationUnit> ::= ( <Document> )+
<Document> ::= "@" <DocumentType> "{" REFERENCE_NAME (
        "," ( <Field> ) )+ "}"
<Field> ::= <FieldType> "=" ( ( "\"" | "{" ) <FieldData> (
        \verb! "\"" | "}" ) | IDENTIFAR | LETERAL )
<DocumentType> ::= ARTICLE | BOOK | CONFERENCE | ...
<FieldType> ::= AUTHOR | EDITOR | YOMI | TITLE | BOOKTITLE ...
```

図 2: BIBTEXgrammer in BNF expression.

ディングが可能であり，既にバッカスナウア記法（Backus-Naur Form: BNF）[24] で表されている文法を対象とする場合，開発の労力を大幅に削減できる．

- **解析処理部を記述する必要がない．** Perl や C 言語など，通常のプログラミング言語を用いてメタデータ変換システムを開発する場合，字句解析や構文解析を行う処理部を開発し，解析対象文書の文法を実装する必要がある．コンパイラ・コンパイラを用いることで，字句解析や構文解析を行う処理部を記述する必要がなく，解析に関する詳細な知識が無くとも変換システムを作成することができる．

- **様々な形式の言語やファイルの文法記述ファイルが公開されている．** JavaCC の場合，HTML や DTD などから C/ C++言語や Java 言語まで，様々な文法ファイルが公開されており [25]，メタデータを開発する際や，開発したメタデータ変換システムの移植を行う際に，それらの文法定義ファイルを利用できる．

コンパイラ・コンパイラを用いたメタデータ変換システムの開発は，EndNote などの他の文献情報メタデータを利用する変換システムや，学術情報に限らず一般のメタデータ変換システムを開発する際にも有効なアプローチである．

2 コンパイラ・コンパイラを用いたメタデータ作成支援

2.1 開発手順

提案するメタデータ作成支援システムの中核となる変換プログラムの開発手順を図 1 に示す．本研究では，開発する変換プログラムを様々な計算機環境で実行できるよう，Java のソースプログラムを出力するコンパイラ・コンパイラ JavaCC[21, 22] を利用する．また，JavaCC を用いる場合，解析木に対して変換などの処理を行うが，本研究では，意味的に冗長性を含む解析木よりも単純な構文木に対して変換処理を行えるよう，JavaCC のプリプロセッサ JJTree を用いる．

変換元データの文法をもとに，コンパイラ・コンパイラの入力となる JJTree プログラムを作成する．BNF に従って表した BIBTEX の文法の概要を図 2 に，JJTree ソースプログラムの一部を図 3 に示す．図 2 および図 3 に示すように，JJTree ソースコードは，BNF 表記に対応する形で宣言的に記述することができ，図 4 に示すような解析処理部の Java ソースコードは JavaCC が自動生成するため，容易に変換システムを作成できる．また，解析木では

```
//BibTeXParser.jjt
options {
        STATIC=false;
        LOOKAHEAD=1;
        UNICODE_INPUT=true;
      ⋮
}
PARSER_BEGIN(BibTeXParser)
public class BibTeXParser {
        int nParseErrors=0;
        String errorMessage="";
      ⋮
}
PARSER_END(BibTeXParser)
SKIP :
{
  " " | "\r" | "\t" | "\n"
}
TOKEN :
{
        < LBRACE : "{" >
|       < RBRACE : "}" >
|       < DOUBLEQUO : "\"" >
|       < ATMARK : "@" >
}
TOKEN :
{
        < ARTICLE : "article" >
|       < BOOK : "book" >
|       < CONFERENCE : "conference" >
      ⋮
}
SimpleNode Start() #Start :
{ }
{
        CompilationUnit() <EOF> { return jjtThis; }
}
void CompilationUnit() :
{
        String error="";
}
{
        (
        {token_source.SwitchTo(DEFAULT);}
         ( <ATMARK> | <E_AT> )
        try{
        (
                Document()
        |
                String()
        )
        }catch(ParseException e){
                ++nParseErrors;
                System.err.println("Doc error");
      ⋮
}
void Document() #Document:
{
        Token t;
        int i;
      ⋮
```

図 3: Source code example in JJTree expression

```
/* Generated By:JJTree & JavaCC:
   Do not edit this line. BibTeXParser.java
*/
public class BibTeXParser/*@bgen(jjtree)*/
  implements BibTeXParserTreeConstants,
  BibTeXParserConstants
{/*@bgen(jjtree)*/
  protected JJTBibTeXParserState jjtree
    = new JJTBibTeXParserState();
  int nParseErrors=0;
  String errorMessage="";
     ⋮

  final public SimpleNode Start()
    throws ParseException
  {
    /*@bgen(jjtree) Start */
    ASTStart jjtn000 = new ASTStart(JJTSTART);
    boolean jjtc000 = true;
    jjtree.openNodeScope(jjtn000);
    try {
      CompilationUnit();
      jj_consume_token(0);
      jjtree.closeNodeScope(jjtn000, true);
      jjtc000 = false;
      {if (true) return jjtn000;}
    } catch (Throwable jjte000) {
      if (jjtc000) {
        jjtree.clearNodeScope(jjtn000);
        jjtc000 = false;
      } else {
      ⋮

    }
    throw new Error(
      "Missing return statement in function"
    );
  }
  final public void CompilationUnit()
    throws ParseException
  {
    String error = "";
    label_1:
    while (true) {
      token_source.SwitchTo(DEFAULT);
      switch ((jj_ntk == -1) ? jj_ntk() : jj_ntk) {
      case ATMARK:
        jj_consume_token(ATMARK);
        break;
      case E_AT:
      ⋮
```

図 4: Source code example generated by JavaCC

なく構文木に対して処理を行うため，コンパイラの知識が無くとも容易に変換処理部の実装を行える．加えて，様々な形式のプログラムコードやデータフォーマットを解析するパーサも公開されており，実装した変換処理の移植が容易である．

本システムを構築するために記述したJJTree形式のプログラムは約700行，生成した構文木を処理するプログラムが約200

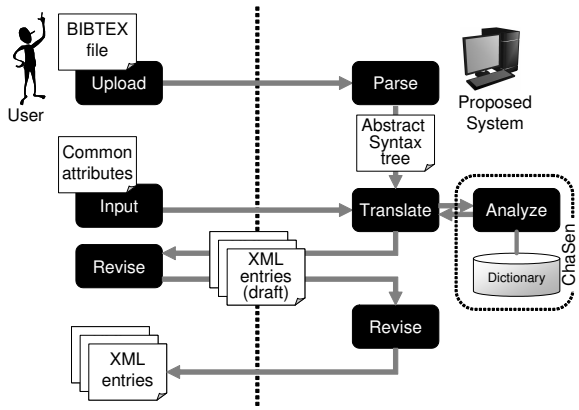

図 5: Process flow of the proposed system.

図 6: Screenshot of revising mode.

行であった．また，JavaCC によって生成されたプログラムは約 6,000 行となった．

2.2 提案するシステムの処理手順

本システムを用いて BIBTEX 形式のメタデータを JuNii 形式に変換する手順を図 5 に示す．本システムは Web アプリケーションとして実装し，ユーザは Web ブラウザ上から操作を行う．

表 1: Attribute examples of general BIBTEXmetadata.

Attribute name	Contents
author	All author names.
yomi	Pronounce of author names
title	The item's title.
journal	The journal name carrying the item.
volumne	The volume of a journal or multivolume book carrying the item.
number	The number of a journal, magazine, etc. carrying the item in a series .
pages	The first and last pages of the item.
url	The WWW Universal Resource Locator that points to the item being referenced.
year	The year when the journal was published.

表 2: Element examples of JuNii metadata.

Element	Qualifier	Scheme
title	Transcription, Alternative	
Creator	Transcription, Alternative	
Subject		NII
Description		
Publisher	Transcription, Alternative	
Contributor		
Date		W3CDTF
Type		NII
Format		IMT
Identifier		URL, ISBN, ISSN, DOI
Source		URL, ISBN, ISSN
Language		ISO639-2
Relation		URL
Coverage	Temporal	NII
Rights		
Comment		
FANO		
UserID		

まず，ユーザは BIBTEX ファイルを変換システムにアップロードする．また，BIB-

```
@article{Nakayama2006a,
  author = "中山 茂 and 今別府 孝洋 and 小野 智司 and 飯村 伊智郎",
  yomi = "Shigeru Nakayama and Takahiro Imabeppu and Satoshi Ono
          and Ichiro Iimura",
  title = "量子風進化的アルゴリズムにおける対交換戦略の検討",
  journal = "電子情報通信学会論文誌",
  volume = "J89-D",
  number = "9",
  pages = "2134-2139",
  year = "2006",
  url = "http://search.ieice.org/bin/summary.php?id=j89-d_9_2134
         &category=D&lang=J&year=2006&abst=&auth=1"
}
```

図 7: Example of bibtex data.

TEX ファイルには含まれない要素のうち，全文献エントリに共通する要素（共通要素）を，Web ブラウザのフォームを通じて入力し，変換システムに送信する．変換システムは，前節の手順で開発した変換プログラムを用いて，BIBTEX ファイルを解析した構文木を作成する．変換システムは次に，構文木およびユーザが入力した共通要素をもとに，各文献ごとに JuNii 形式メタデータ案を作成する．

対象とする文献情報が日本語で記述されている場合，文献のタイトルや著者名のカナ表記（Transcription 限定子）が必要となる．このため本システムでは，日本語で記述された要素に対して形態素解析を行い，カナ表記の要素データを生成する．日本語で記述された要素については，英語表記（Alternative 限定子）も必要となる．本システムでは，著者名のローマ字表記を自動的に生成するが，文献タイトルの英語表記はユーザに入力を依頼する．

作成した JuNii 形式メタデータ案を初期値とするフォームをユーザに提示し，不足する要素の入力および変換誤りの修正をユーザに依頼する．ユーザの修正をもとに JuNii 形式メタデータ案を修正し，XML ファイルとして出力する．

2.3 要素の一覧

BIBTEX における標準的な要素の一覧を表 1 に，JuNii 形式における要素の一覧を表 2 に示す．各記述要素については文献[8] などを参照されたい．JuNii 形式に含まれて BIBTEX に含まれない要素のうち，UserID および参加組織レコード ID を表す FANO は，共通要素としてユーザに入力を依頼する．type 要素は BIBTEX における文献の種類（article，proceedings，book など）から，date 要素は BIBTEX における year および month 要素から，language 要素は BIBTEX における yomi 要素の有無をもとに生成する．Subject，Identifier およびその他の任意の要素は各論文ごとに，ユーザに入力を依頼する．

3 評価実験

本システムを用いて文献情報エントリを 226 件含む BIBTEX ファイルを変換する実験を行った．各文献情報エントリにおいて，適切に変換された要素数，手動で修正・入力が必要であった要素数を調べた．また，システムが変換に要する時間，および，ユーザが修正を行った時間を測定した．

本実験では，日本語で記述された論文 124 件および英語で記述された論文 102 件

```
<OAI-PMH xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
     http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-11-24T07:37:22Z</responseDate>
  <request verb="GetRecord" metadataPrefix="junii"
    identifier="oai:ics.kagoshima-u.ac.jp:00002957:S">
    http://metasv.nii.ac.jp:80/cgi-bin/oai/oai2.0
  </request>
  <GetRecord>
    <record>
      <header>
        <identifier>
          oai:ics.kagoshima-u.ac.jp:00002957
        </identifier>
        <datestamp>2006-11-24T16:35:35Z</datestamp>
      </header>
      <metadata>
        <meta xsi:schemaLocation="http://metasv.nii.ac.jp/oai
          http://metasv.nii.ac.jp/oai/junii.xsd">
          <code>00002957</code>
          <userid/>
          <fano>FA003647</fano>
          <adate>20061124</adate>
          <udate>20061124</udate>
          <institution>鹿児島大学</institution>
          <title>
            量子風進化的アルゴリズムにおける対交換戦略の検討
          </title>
          <title.transcription>
            リョウシ フウ シンカ テキ アルゴリズム ニオケル
            タイ コウカン センリャク ノ ケントウ
          </title.transcription>
          <creator>
            中山，茂
          </creator>
          <creator.transcription>
            シゲル，ナカヤマ
          </creator.transcription>
          <creator.alternative>
            Shigeru, Nakayama
          </creator.alternative>

                ⋮
```

図 8: Example of derived XML data.

のメタデータを生成した．日本語の文献における平均著者数は 2.6 人，英語で記述された論文における平均著者数は 2.3 人であった．

実験結果を図 9 に示す．1 件の文献情報エントリあたり，JuNii 形式では平均 18.8 要素のデータを記述する必要があったが，本システムを利用することで平均 14.1 要素を正しく生成することができた．正しく生成することができた 14.1 要素のうち，Title，Creator，Date などの 7.1 要素は BIBTEX 形式のメタデータから変換を行い，a-

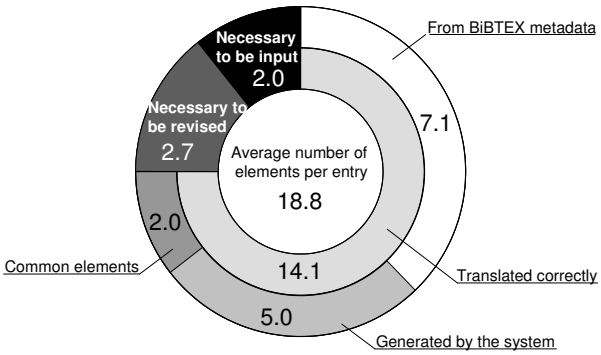

図 9: Experimental results.

date，institution，format などの 5.0 要素はシステムが現在の日時などを参照し，自動的に付加した．FANO および UserID の 2 要素は共通要素であり，BIBTEX ファイル送信時にユーザが入力した情報をもとにした．

また，18.8 要素のうち，Title – Transcription，Creator – Transcription，Creator – Alternative など，平均 2.7 要素は，BIBTEX 形式のメタデータから変換を行ったもののユーザによる修正が必要であった．これは，形態素解析に用いた ChaSen が漢字かな変換や形態素の認定を誤ることがあったこと，Creator の要素において人名の姓と名の間に空白がない場合に姓名の識別を行えなかったことなどが原因である．なお，本実験では日本語で記述された要素の Transcription 限定子において，分かち書きの正確さは考慮しなかった．形態素解析における文節の認定結果や，知識ベースを用いた日本語の分かち書き方式[26, 27] の利用により，適切な分かち書きを行えると考える．

Subject と Identifier に関しては手動による入力が必要であった．本実験で用いた BIBTEX データには含まれていなかったが，BIBTEX データにおいて url 要素を適切に記述することで，JuNii 形式の Identifier 要素を自動的に生成することができると考える．Subject 要素の自動生成は困難であるが，タイトル内のキーワードの自動抽出および抽出したキーワードからの研究分野の推定について，今後検討を行う予定である．

なお，文献情報エントリ 1 件あたりの自動変換に要する時間は約 0.2 秒，修正に要する時間は約 40 秒であった．よって，本システムを用いることで，JuNii 形式の要素のうち，約 8 割を自動的に生成することができ，従来 4，5 分程度必要であった入力時間を 1 分程度に短縮できることがわかる．

4 おわりに

コンパイラ・コンパイラ JavaCC を用いて，BIBTEX 形式のメタデータを NII メタデータ形式へと変換するシステムを作成した．従来は 1 件の学術情報につき 5 分程度の時間を費して手作業でメタデータを作成していたのに対し，本システムを利用することで，手作業で入力する要素は 5 分の 1 程度に減少し，約 1 分でデータを入力できることがわかった．よって，本システムは学術機関リポジトリの構築，拡充に貢献できると考える．

コンパイラ・コンパイラを用いた変換プログラムの作成は，BNF 表記に従って文法を宣言的に記述することで解析プログラムを生成できるため，開発効率が高く，様々な形式の既存データをメタデータ作成へと活かすための有用なアプローチである．

今後，本システムの運用を開始するとともに，他形式からの JuNii 形式準拠メタデータの生成システムの開発や，1 次データ，すなわち論文の LATEX ソースファイルからのメタデータ生成システムの開発を行う予定である．

参考文献

[1] 国立情報学研究所：「メタデータ・データベース共同構築事業」, http://www.nii.ac.jp/metadata/, (2006年11月27日参照).

[2] 国立情報学研究所：「GeNii 学術コンテンツ・ポータル」, http://ge.nii.ac.jp/genii/jsp/index.jsp, (2006年11月27日参照).

[3] 国立情報学研究所：「大学 web サイト資源検索 JuNii 大学情報メタデータ・ポータル試験提供版」, http://ju.nii.ac.jp/, (2006年11月27日参照).

[4] 文部科学省研究振興局情報課：「学術情報発信に向けた大学図書館機能の改善について（報告書)」, 2003.

[5] Association of Research Libraries: "Collection & access for the 21st-century scholar: changing roles of research libraries", ARL bimonthly report 225, http://www.arl.org/newsltr/225/main.html, (2006年11月27日参照).

[6] Jackson, Mary E.: "The advent of portals", Library Journal, Vol.127, No.15, pp.36–39, 2002.

[7] Dublin core metadata initiative: "DCMI Metadata Terms", http://dublincore.org/documents/dcmi-terms/, (2006年11月27日参照).

[8] 国立情報学研究所:「NII メタデータ・データベース入力マニュアル 2.0 版」, http://www.nii.ac.jp/metadata/manual/, (2006年11月27日参照).

[9] "The Open Archives Initiative Protocol for Metadata Harvesting – v.2.0", http://www.openarchives.org/OAI/openarchivesprotocol.html, (2006年11月27日参照).

[10] Hochstenbach, Patrick; Jerez, Henry; Van de Sompel, Herbert: "The OAI-PMH static repository and static repository gateway", In 2003 Joint Conference on Digital Libraries, pp.210–217, 2003.

[11] 国立情報学研究所：「OAI-PMH の NII メタデータ・データベースへの適用について」, http://www.nii.ac.jp/metadata/oai-pmh/implementation_as_repository.html, (2006年11月27日参照).

[12] Patashnik, Oren:"BiBTeXing",1988.

[13] Patashnik, Oren:"Designing BiBTeX styles", 1988.

[14] 国立情報学研究所: 「論文ナビゲータ CiNii」, http://ci.nii.ac.jp, (2007年1月22日参照).

[15] College of Information Sciences and Technology: "CiteSeer.IST", http://citeseer.ist.psu.edu/, (2007年1月22日参照).

[16] OXFORD UNIVERSITY PRESS: "OXFORD JOURNALS", http://www.oxfordjournals.org/, (2007年1月22日参照).

[17] American Association for the Advancement of Science: “Science”, http://www.sciencemag.org/, （2007 年 1 月 22 日参照）.

[18] The Association for Computing Machinery: “ACM Portal”, http://portal.acm.org/portal.cfm, （2007 年 1 月 22 日参照）.

[19] The Institute of Electrical and Electronics Engineers, Inc.: “IEEE Computer Society”, http://www.computer.org/, （2007 年 1 月 22 日参照）.

[20] Johnson, Steven C: “Yacc: Yet another compiler compiler”, In UNIX Programmer’s Manual, Vol.2, pp.353–387, New York, NY, USA, Holt, Rinehart, and Winston, 1979.

[21] Futamura, Yoshihiko: “Partial evaluation of computationpartial evaluation of computation process: An approach to a compiler-compiler process: An approach to a compiler-compiler”, Higher-Order and Symbolic Computation, Vol.12, No.4, pp.381–391, 1999.

[22] Sun Microsystems: “Java compiler compiler (javacc) - the java parser generator”, https://javacc.dev.java.net/, （2006 年 11 月 27 日参照）.

[23] 岩川建彦；小野智司；中山茂：「分散並列処理プログラミング言語 Espace の開発」，システム制御情報学会論文誌，Vol.19，No.7，pp.296–298，2006.

[24] Backus, John W. : “The syntax and semantics of the proposed international algebraic language of the zurich acm-gamm conference”, In Proceedings of the International Comference on Information Processing, pp.125–132, 1959.

[25] CollabNet, Inc.: “Repository of JavaCC grammers”, https://javacc.dev.java.net/servlets/ProjectDocumentList?folderID=110, （2007 年 1 月 22 日参照）.

[26] 小野智司；鈴木恵美子；宮下和雄；西原清一：「事例・ルール間変換による知識編成方式と日本語点字翻訳の分かち書きへの応用」，情報処理学会論文誌，Vol.41，No.11，pp.3037–3045，2000.

[27] 小野智司；高木喜次；浜田佳延；水野一徳；西原清一：「事例知識を用いた日本語点字翻訳とエラー修正支援」，ヒューマンインタフェース学会論文誌，Vol.5，No.4，pp.491–498，2003.