

構文主導意味論に基づく文書構造モデル

木村高久

日本規格協会情報技術標準化研究センター

要旨

インターネット上で世界規模での文書共有が成功して以来、この技術を歴史的文献資料の公開に利用する活動が盛んになった。歴史的文献資料の公開には、解釈を表示する要請、伝統楽譜のように複雑な構造を持つ文書の対応に関する要請など、従来の文書記述言語を適用することが難しい問題が知られている。本報告では、歴史的文献資料向け高級文書記述言語の概念形成を目標として、数理言語学に基づく文書記述モデルを提案する。Fülöp の構文主導意味論に基づき、文書構造を記述する木構造と、この木構造に対する変換操作とを組み合わせ、文書記述機能モデルを構成する。例題を使って、この文書記述モデルの有効性を示す。

キーワード

構文主導意味論、木言語、木変換、 Σ 代数、文書記述言語、役割プログラミング

Document Structure Model based on the Syntax-Directed Semantics

TAKAHISA Kimura

Information Technology Research and Standardization Center(INSTAC)

Abstract

The success of the text-sharing on the net causes active researches for sharing historic documents on the net. Thorough these research activities, issues about sharing historic documents; researchers have to mark up to represent their interpretation of documents; to represent highly complicated document style like traditional musical notations, and so on. In this report, we propose a new document description model based on a mathematical language theory for establishing a notion of high-level document description language for sharing historic documents on the net. This document description model is established based on the Fülöp's syntax-directed semantics and consists of a pair of tree-structures and tree-transducers, which model document structures and document structure transformation respectively. We evaluate this document description model through applying it to realistic sample problems.

Key words

Syntax-directed semantics, Tree language, Tree transformation, Σ algebra, document description language, Role programming

1. はじめに

インターネットは世界規模での電子文書共有システムと考えることができる。インターネットが当初ホームページ公開で成功を収めて以来、歴史的文書資料を公開するためにこれを利用する可能性が検討されるようになった[1-5]。ホームページ公開と比べて歴史的文書資料の公開には、第一に、修辞法など文書の意味解釈を記述する必要があること[6,7]、第二に、伝統楽譜（声明譜[8]など）や図のデザインの一部として埋め込まれた文書（カリグラフィ[9]など）のような、複雑なレイアウト構造を持つ文書を電子文書化し、かつこれを使って語句検索等の処理を行うという要請があること、などの問題が知られている。文書記述言語は意味論を持たない言語として知られており、歴史的文書資料の意味解釈を記

述する問題に対応するには意味論を構築する必要がある。また、語句検索を許しかつ複雑なレイアウト構造を表現するためには、文書構造変換指定（文書構造プログラミング）の仕組みが重要な役割を持つと予想される。そこで本報告では、これらの基準を満たす文書処理方式の構成を目標として、木言語理論に基づく電子文書処理方式を導入する。

本報告では電子文書処理手順の説明に役割プログラミング法（以下 RP 法[10]）の記法を使う。RP 法はプログラム変換法であり、入力テキストの構造を記述する木構造を作製し、書き換え規則を適用してこの木構造を書き換えて、テキストをプログラムに変換する。文献[11]で言われている抽象アルゴリズムが電子文書処理モデルを規定できること、及びにこの抽象アルゴリズムが構文主導意味論（Syntax-directed semantics [13]、木変換系（[12,13]）を使う意味計算法）に基づいて具体化できることに着目して、RP 法を文書処理法として再構成した。再構成版 RP 法を、本報告では RRP 法（改訂 RP 法）と呼ぶ。

RRP 法ではデータ記述言語を使用して、木構造で文書構造を記述する。この木構造（以下、文書木とも言う）は木言語の構文と、意味記述とを持つ。一つの意味記述は整形 XML 文書（Well-formed XML document[14]）類似のタグ付きテキストになる。木から木への変換、及びに木から文字列への変換は、それぞれ、書き換え規則の適用によって実行する。具体的な変換操作は、変換規則集合を選択し、変換規則を適用順に並べて構成する。

RRP 法による電子文書処理手順を、文書構造変換プログラムの例を使って示す。華厳一乘法界図は複雑な文書構造を持つ文書例として知られており、師[15,16]により SVG（Scalable Vector Graphics [17]）を使った表現実験が試みられている。この SVG 文書を、文字並び文書、文字レイアウト並び文書、SVG テンプレート、の三つの文書から構成する文書構造変換プログラムの例を示す。この方法は図のデザインの一部としてプレーンテキストを埋め込むために利用できる。

これらの実例を通して、文書記述処理に木言語モデルを適用する意義を検討して、いくつかの課題を抽出する。十分な数の変換規則が整備されれば、RRP 法を高級文書記述言語のコンパイラとして利用できるようになると考えられる。文献資料の分析を通して共用性の高い変換規則を整備することが次の課題となる。

2. 文書木データの構文と処理手順

この節では改訂 RP 法（RRP 法）における文書木データの記述法と処理法を示す。

2-1. RRP 法の処理データ

RRP 法の処理データ、R 木構造、木構造変換規則、記号列表現生成規則（あるいは単に表現生成規則）、はそれぞれ以下の構文に従う。記号の衝突回避等、詳細は別途報告する。

(1) R 木構造

- ・ 節点定義 <節点識別名>=<関係名>(<関連リスト>)

ここで<関連リスト>は、名前の対”<引数識別名>:<節点識別名>”の並びから成る。

記述例： $I = P(R_1 : I_1, R_2 : I_2)$

特に<引数識別名>が順序数 1, 2, 3, ..., の場合、<引数識別名>を省略できる。

記述例： $I = P(I_1, I_2)$

後者は不定長リストを書くための便宜であり、前者の構造に還元される。

- ・ **R 木構造** 節点定義の有限個の集まり。

(2) 木構造変換規則

- ・ **変数記号** ①ないし④

節点定義を構成する各名前に前置して、名前が照合変数であることを表す。

④は<引数識別名>を省略した<関連リスト>にのみ現れる。

- ・ **照合パターン定義** 以下の条件を満たす節点定義

固定長形式 構成する名前が変数記号④で始まるものを含めない

可変長景色 構成する名前が変数記号④で始まるものを含む

- ・ **木構造変換規則** {<前提部>}→{<帰結部>}

ここで<前提部>、<帰結部>はそれぞれ、照合パターン定義の有限個の集まり。

(3) 表現生成規則

- ・ **表現生成規則** <関係名>=<記号識別名リスト><末尾記号>

ここで<記号識別名リスト>は対<記号並び><引数識別名>を一つ以上持つ並び。

<記号並び>は記号列または空。同様に<末尾記号>は記号列または空。

<引数識別名>は<関係名>に割当てられた<引数識別名>リストの要素。

記述例：text=<text> [char] </text> (text(char:文字)に適用される。)

2-2. RRP 法の処理の流れ

RRP 法の処理の流れを図 1 a に示す。初期木と終了木は前節に示した木構造記述言語で書かれる R 木構造データである。木から木変換は木構造変換規則の並びによって記述される。木から記号列変換は、表現生成規則の集まりによって記述される。記号列から木への変換は問題に応じて作成される記号列処理プログラムである(前節では規定していない)。

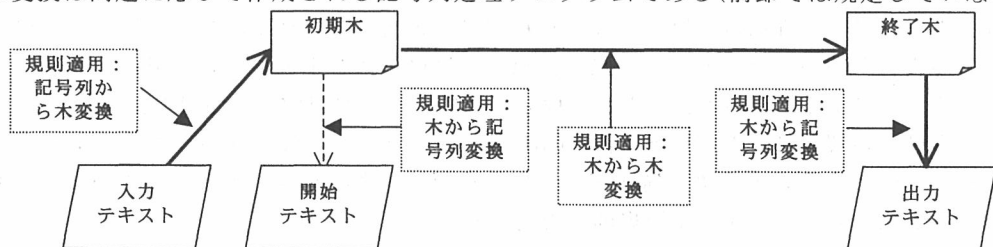


図 1 a. RRP 法の処理の流れ

プログラムスキーマ分野で知られる抽象アルゴリズムの規定[11]を図 1 b に示す。初期木と最終木は木言語の要素に対応付けることが可能であり、木変換系は木の書き換え規則の集まりで規定されることから、図 1 b が図 1 a のモデルを与えることが理解できる。

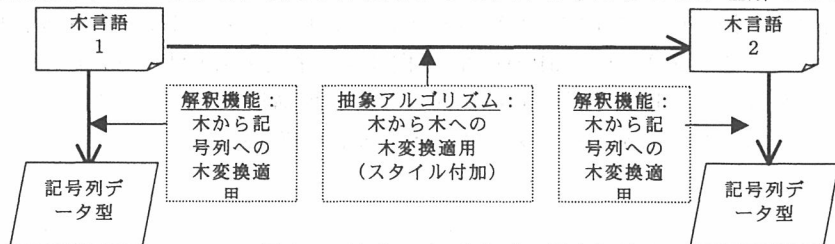


図 1 b. 抽象アルゴリズム規定[11]

以下では RRP 法が、文書記述処理系としてどのように機能するかを見る。

3. 文書木の構文と解釈

RRP 法による文書構造記述例を以下に示す。図 2 a は、2 節で示した文書木データ記述言語を使い、文献[18]の図 6.2 (131 頁)を参照して作成した文書構造記述例である。図 2 b は対応する SGML マークアップ例、図 2 c は同じ文書構造を名前の省略なしに記述した例である。等号右边を等号左辺の名前が示す位置に代入して、図 2 c から図 2 a を得る。

a) RRP 記法 (簡易版) <pre>POEM= poem(title: title(string: "The Sick Rose"), stanza: stanza(line(string: "O Rose thou are sick."), line(string: "The invisible worm,"), line(string: "That flies in the night"), line(string: "In the howling storm;")))</pre>	b) SGML マークアップ (参照[18]) <pre><poem><title>The SICK ROSE</title> <stanza> <line> O Rose thou are sick.</line> <line>The invisible worm,</line> <line> That flies in the night</line> <line> In the howling storm;</line> </stanza> </poem></pre>
c) RRP 法 (標準版) <pre>POEM= poem(title: TITLE_SICK_ROSE, stanza: STANZA_SICK_ROSE) TITLE_SICK_ROSE=title(string:THE_SICK_ROSE) THE_SICK_ROSE=TheSickRose() STANZA_SICK_ROSE=stanza(LINE#1, LINE#2, LINE#3, LINE#4) LINE#1=line(string:ORoseThouAreSick) ORoseThouAreSick=O_Rose_thou_are_sick_p() LINE#2=line(string:TheInvisibleWorm) TheInvisibleWorm=The_invisible_worm_c() LINE#3=line(string:ThatFliesInTheNight) ThatFliesInTheNight=That_flies_in_the_night() LINE#4=line(string:InTheHowlingStorm) InTheHowlingStorm=In_the_howling_storm_SC()</pre>	<pre>TheSickRose()= "TheSickRose" O_Rose_thou_are_sick_p() "O Rose thou are sick." The_invisible_worm_c() "The invisible worm," That_flies_in_the_night() "That flies in the night" In_the_howling_storm_SC() "In the howling storm;"</pre>

図 2. RRP 法による文書構造表現

a)RRP 法簡易版、b) SGML マークアップ、c)RRP 法標準版

RRP 法の文書構造記述では、検証用データを添付しない (整構造 XML 文書相当)。検証を外付けにした理由は、歴史的な文書資料では文書構造が所与ではなく解釈者の解釈に依存する場合があること、木言語理論の観点からは検証系を正規木言語に限定する必然性がないこと、による。正規木言語として検証する場合には DTD 相当の規則を適用する。

R 木構造の解釈は、木言語理論[11-13]に従って導入する。R 木構造の集合をある木言語でモデル化し、その木言語を構成する Σ 代数を使って意味を計算する。具体的な計算手順は、木を記号列に変換する変換規則で規定する。

解釈機能の設定例を以下に示す。引数のない関数は定数を返す。引数が一つの関数は入力文字列そのものを、複数個の引数を持つ関数は入力文字列を引数の並び順に連結した文字列を返す。この解釈は木の葉要素を抽出する操作に相当し、これに基づいて図 2 a の関数項を評価すると、最終的に、元の詩を表す文字列が POEM に代入される。

別の設定例を以下に示す。関数名を代表して f と書き、引数 x の評価結果を $\text{val}(x)$ と書く。このとき、 $f(x)=\langle f \rangle + \text{val}(x) + \langle /f \rangle$ で機能が定義される関数を利用する。 $\langle f \rangle$ と $\langle /f \rangle$ とをタグと見なすと、R 木構造の意味記述として、整構造 XML テキストを生成する。RRP 法では文書木の意味記述としてタグ付きテキストが導かれる。

4. 表示用文書木の生成

本節では、スタイルシートに相当する機能の具体化法を示す。

a) 華嚴一乘法界図表示例[16]	b) 華嚴一乘法界図を表示する SVG コード [16] <pre> <svg xml:space='preserve' width='4.8in'... <g> <text x='230' y='230'>法</text> <text x='200' y='230'>性</text> <text x='170' y='230'>圓</text> <text x='140' y='230'>融</text> <text x='110' y='230'>無</text> <text x='80' y='230'>二</text> <text x='50' y='230'>相</text> (中略) <text x='260' y='410'>舊</text> <text x='230' y='410'>來</text> <text x='230' y='380'>不</text> <text x='230' y='350'>動</text> <text x='230' y='320'>名</text> <text x='230' y='290'>爲</text> <text x='230' y='260'>佛</text> </g> </svg> </pre>
c) 文字の並び 法 性 圓 融 無 二 相 (中略) 舊 來 不 動 名 爲 佛 d) 文字の配置の並び <pre> position(x:'230', y:'230') position(x:'200', y:'230') position(x:'170', y:'230') position(x:'140', y:'230') position(x:'110', y:'230') position(x:'80', y:'230') position(x:'50', y:'230') (中略) position(x:'260', y:'410') position(x:'230', y:'410') position(x:'230', y:'380') position(x:'230', y:'350') position(x:'230', y:'320') position(x:'230', y:'290') position(x:'230', y:'260') </pre>	図 3. 法界図を表示する SVG コードの文字列、配置、地の文からの組み立て a) 法界図の一表示 b) SVG コード[16] c) 文字の並び d) 文字配置の並び 本文書例をお教え頂いた駒沢短期大学の石井教授ならびに上記 SVG コードの参照を許可頂いた花園大学の師専任講師に感謝いたします。

図 4 は、図 3 の組み立てを実施する文書構造変換規則の例である。

a) 初期文書 <pre> TEXT=charSequence(法, 性, 圓, 融, 無, 二, 相, 《中略》, 舊, 來, 不, 動, 名, 爲, 佛) LAYOUT=posSequence(position(x:230,y:230), 《中略》, position(x:230, y:260)) LAYOUT_TEXT=attrCharSequence() </pre> b) 変換規則前提部 <pre> { TEXT=charSequence(¥char,@chars) LAYOUT=posSequence(¥layout,@layouts) ¥layout=position(x:¥x,y:¥y) LAYOUT_TEXT=attrCharSequence(@attribChars) } </pre> c) 変換規則帰結部 <pre> { TEXT=charSequence(@chars) LAYOUT=posSequence(@layouts) ¥attribChar=text(char:¥char,x:¥x,y:¥y) LAYOUT_TEXT=attrCharSequence(@attribChars,¥attribChar) } </pre> d) SVG テンプレート <pre> SVG_CODE=HOKKAIZU(layoutText:LAYOUT_TEXT) </pre>

図 4. RRP 法による文書構造変換プログラム

a)初期文書（要約）、b)変換規則前提部、c)変換規則帰結部、d)SVG テンプレート

華嚴一乘法界図と呼ばれる文書を例題に使用する。師[15,16]は同文書の表現実験を試み、図3aの表示を図3bのSVGコードを使って実現した。図3aでは、文意を理解するために取るべき経路が線分で表示されている。この線分を地とする。図4の変換規則は、レイアウト情報のない漢字の並び（図3c）に、各漢字の配置情報を付加して、最終的なSVGコードを生成する。動作を以下で説明する。

図4で、節点識別子TEXTを左辺に持つ節点定義は文字の並びを記述するR木構造（文字並びR木構造）である。節点識別子LAYOUTを左辺に持つ節点定義は文字配置情報の並びを記述するR木構造（配置並びR木構造）である。節点識別子LAYOUT_TEXTを左辺に持つ節点定義は配置情報付きの文字の並びを記述するR木構造（文字配置並びR木構造）である。初期木として、以上の三本の節点定義から構成されるR木構造を使用する。節点識別子SVG_CODEを左辺に持つ節点定義は、配置情報付き文字の並び構造を代入する先となるR木構造である。このR木構造に適用される表現生成規則は以下の要領で作成する。法界図を表現するSVGコードにおいて、<text>タグに囲まれた文字の並びの全体を引数識別名に置き換えて得られる文字列を右辺に持たせる。

図4のR木変換規則は以下のように動作する。初期木に変換規則を適用する。前提部のcharSequence(¥char,@chars)が初期木のcharSequenceと照合されると、文字並びの最初の一文字が¥charに取り出され、残りが@charsに取り出される。同様に前提部のposSequence(¥layout,@layouts)の照合で配置並びR木構造の配置座標が¥layoutに取り出され、残りが@positionsに取り出される。取り出された文字情報と配置情報は、変換規則帰結部でtext節点定義に設定される。Text節点定義には表現生成規則text→<text x=[x], y=[y]>[char]</text>を適用して、タグ情報を作成する。変換規則帰結部では、文字並びR木構造及び配置並びR木構造から最初の一要素を削除し、文字配置並びR木構造にText節点定義を追加する。変換規則は初期木に適用可能な限り繰り返して適用されるので、文字並びR木構造及び配置並びR木構造の要素が尽きるまで繰り返し適用される。書き換え規則の適用の結果、文字配置並びR木構造にはText節点定義の並びが作成される。これを節点識別子SVG_CODEを左辺に持つ節点定義に代入し、SVGコード生成用表現生成規則を適用して表現生成を行い、SVGコードが得られる。この規則は図4には省略している。

初期木TEXT=charSequence(法, 性, 圓, 融, 無, 二, 相, 《中略》, 舊, 來, 不, 動, 名, 爲, 佛)はテキストファイルから簡単なフィルタで生成できるので、SVG表現が必要になった時点で変換規則を適用してSVGコードを生成する方法が使えるようになる。

5. 数理言語モデルの効用

2節では木構造変換を基礎とするテキスト変換法を示し、3節と4節の例を使って、このテキスト変換法を文書記述処理の機能モデルと考えることができる理由を説明した。数理言語モデルを利用する意義は、モデルを適用することで対象の詳細な性質を知り、問題に適合した解決法を選ぶ基準を得ることにある。以下に例題を示す。

5-1. ローカルなボキャブラリの定義法

XML 言語は文脈自由文法に従うことが知られている。また、正規木言語の葉言語（2 節の木の解釈参照）は文脈自由言語である。従って、正規木言語を模擬する R 木構造の集合は、XML と同程度の文書構造記述力を持つことが期待される。

記号列言語の場合同様、木言語には、正規木言語のクラスと文脈自由木言語（マクロ木言語）のクラスが定義される[13]。記号列言語の場合と同様に、正規木言語のクラスは木のブロック構造が表現できない。これは、正規木言語のクラスに基づく文書構造記述ではブロックにローカルなボキャブラリが定義できないことを意味する。

ブロックにローカルなボキャブラリを導入する一つの案は、ローカルなボキャブラリをグローバルにも一意に区別する手段を与えることである。XML の名前空間[19]はこの案に従った解決策と考えられる。数理言語の観点からは、ワインハルデン文法[20]の方法でボキャブラリに付加する接頭語を組織的に生成する解決策に相当すると考えられる。

もう一つの案は、マクロ木文法を文書記述に利用することである。文脈自由文法の性質からブロックにローカルなボキャブラリが自然に定義されると期待される。文書記述処理への適用例はまだないと思われる。

5-2. 意味記述

R 木構造と木変換機能とを木言語と木変換系に同一視することで、RRP 法の文書木は Σ 代数を使った解釈が定義される。文書木の解釈としてテキストが生成され、この生成されたテキストが機械または人によって解釈される。

文書木の解釈と生成されるテキストの解釈の境界は、文書木に情報を追加して（語句に説明を追加する、等の方法で）移動することができる。この情報追加操作を木言語モデルに基づいて説明するために、木言語の基盤である Σ 代数を抽象データ型と見なして、情報追加操作を抽象データ型の具体化操作に対応付ける方法を検討している。

5-3. 非決定的な文書構造

解釈が一意でない文は、語学・文学分野では、表現技法の一部として（正しい文書構造として）現れる。解釈が一意でない文書には一意でない文書木が必要になる。

例えば「かすみ立ち木の芽もはるの雪降れば花なき里も花ぞ散りける」という和歌（紀貫之）で、「木の芽もはるの季節」という表現は、「木の芽も張る、春の季節」という表現と等価であると言われている[21]。「はる」が二重の意味で使われている結果、この文は木構造をとらない（ $\langle \text{文} \rangle \text{木の芽も} \langle \text{文} \rangle \text{はる} \langle \text{文} \rangle \text{の季節} \langle \text{文} \rangle$ となる）。一つの文書が複数の構造を持つことを表すには、一つの文書に複数の文書木を対応付ける方法が考えられる。このためには非決定性正規木言語[12]ないし確率正規木言語が利用できる。

$$\left\{ \begin{array}{ll} \langle \text{文} \rangle \text{木の芽もはる} \langle \text{文} \rangle \langle \text{句} \rangle \text{の季節} \langle \text{句} \rangle & 0.5 \\ \langle \text{句} \rangle \text{木の芽も} \langle \text{句} \rangle \langle \text{文} \rangle \text{はるの季節} \langle \text{文} \rangle & 0.5 \end{array} \right\}$$

5-4. 高級文書記述言語

数理言語学モデルは XML と異なる特徴を持つ文書記述処理系が可能であり、あるものは有効であることを示している。これらに基づいて実際的な文書処理系を実現するためには、文書処理系の仕様を策定し、再利用性の高い木構造変換規則と表現生成規則とを用意する作業が必要になる。正規上位クラスの木言語で書かれた文書構造記述を XML に変換する RRP 法は高級文書記述言語の処理系と考えられる。有効な木言語のクラスを決定

するために、実在の文献資料の構造調査を構想している。

6. まとめ

本報告では文書記述言語の規格案策定のための新しい観点として、文書木処理中心の木言語モデルを示した。このモデルでは、XML などの文書記述言語における定式化と異なり、文書木には構文と解釈が定義される。文書木の意味記述の一つとしてタグ付きテキストが位置づけられる。例題を使って、人文電子系部門において従来の文書記述言語で扱いが困難な文書の取扱法を示した。

数理言語モデルから文書処理系の具体的な仕様を策定するためには、応用分野における基本的な要件を収集する必要がある。例えば、文書構造を記述するために必要となる木言語のクラス、文書構造変換の使い方、ツールの仕様、などの収集が必要である。そのためには、書誌学分野と情報科学分野の協調作業が必要になる。

参考文献

- [1] OASIS の学術応用一覧の頁 <http://xml.coverpages.org/acadapps.html>
- [2] Text Encoding Initiative のサイト <http://www.tei-c.org/>
- [3] 情報処理語学文学研究会の電子テキスト配布サイト
http://yoshi01.kokugo.edu.yamaguchi-u.ac.jp/kokugo/jal_ftp.html
- [4] 花園大学の電子達磨 <http://www.ijnet.or.jp/iriz/>
- [5] 大正新脩大藏経テキストデータベース <http://www.l.u-tokyo.ac.jp/~sat/japan/index.html>
- [6] 豊島正之、TEI から見た SGML のはなし、情報処理語学文学研究会会報 12 号、1992 年 12 月
- [7] 豊島正之、TEI-P3 について、情報処理語学文学研究会会報 15 号、1994 年 7 月
- [8] 木戸敏郎(編集), 声明〈1〉ー日本音楽叢書(3)ー, 音楽之友社, ISBN: 4276134331 (1990/07/01)
- [9] ジョルジュ・ジャン著、矢島文夫監修、文字の歴史、知の再発見叢書(創元社)01、1990 年
- [10] 小林要、木村高久、役割プログラミングの手引き、平成 8 年 4 月 15 日、Research Memorandum ISIS-RM-96-2J
- [11] Engelfriet, J.: SOME OPEN QUESTIONS AND RECENT RESULTS ON TREE TRANSDUCERS AND TREE LANGUAGES, in "Formal Language Theory, Perspective and Open Problems" R. V. Book editor, Academic Press (1980), pp.241-266
- [12] Géceg, F. M. Steinby: Tree Automata, Akadémiai Kiadó Budapest '84
- [13] Fülöp, Z., H. Vogler : Syntax-Directed Semantics: Formal Models Based on Tree Transducers, Sep. '98 (Springer)
- [14] XML 仕様書 <http://www.w3.org/TR/REC-xml>
- [15] Moro, Shigeki. "Complex Spatial Digitization Tasks for the SAT Project". (単著、英文、『電子佛典』第 3 輯、2001 年 12 月、東國大學校 EBTI)
- [16] 華厳一乗法界図の表現実験 <http://www.l.u-tokyo.ac.jp/~sat/japan/tech/experimental/svg.html>
- [17] SVG1.0 仕様書 <http://www.w3.org/TR/SVG/>
- [18] 根岸、石塚, SGML の活用, オーム社, 総合マルチメディア選書 (平成 6 年)
- [19] XML の名前空間 <http://www.w3.org/TR/REC-xml-names/>
- [20] A. Van Wijngaarden et al, Revised Report on the Algorithmic Language ALGOL 68, Acta Informatica 5:1-236 (1975)
- [21] 金田一春彦, 日本人の言語表現, 講談社現代新書