

# 拡張可能な会話型文芸的プログラミング支援システム

2000MT021 本田 宜生

2000MT067 寧 静

2000MT104 山本 裕也

指導教員 真野 芳久

## 1 はじめに

本研究では、文芸的プログラミングの支援のため、ユーザの与える様々な視点<sup>\*1</sup>によって、文芸的プログラミングで記述された文書から情報を抜き出して表示する拡張可能なビュー<sup>\*2</sup>(マルチビュー)を提供し、複数のプログラム言語と文書整形言語を扱う仕組(マルチランゲージ)を提供するシステム XMulTAS を考案と試作を行った。

### 1.1 文芸的プログラミングについて

文芸的プログラミングは、プログラムの作成と同時に説明文書を記述していくプログラミング方法である。そのため、記述する文書にはプログラムと説明文書が対として混在している。そして、説明文書のみを抜き出して読むことによって人間がプログラムを文章で理解できる。また、文芸的プログラミングはセクションという意味ある塊を単位としてプログラミングし、セクションからセクションを呼び出すことができる。

そして、プログラム作成時に説明文書を記述しながらプログラムを作成していくことで、プログラム作成者は思考の整理ができ、正しいプログラムを作成しやすくなる。また、保守時には、説明文書を読むことでプログラムの内容を容易に思い出したり、他人に伝えたりできる。

このプログラミング方法を支援するシステムとして WEB ( CWEB ) システム<sup>\*3</sup>、MulTAS<sup>\*4</sup>、Xas-LP<sup>\*5</sup>などが存在する。

### 1.2 問題点

文芸的プログラミング支援システムである WEB システムの問題点のうち、本研究で扱う問題点をあげる。

- WEB 文書はセクションの羅列となるため、プログラム構造の理解が困難
- セクション間にどのような関係があるのかの判断が困難
- セクション分けによってプログラムが分割されるため、プログラム言語の構文規則(変数参照可能

領域等)の判断が困難

これらの問題を本研究で扱うことにする。

### 1.3 目的・目標

ユーザが簡単な拡張を行うことにより、全てのプログラム言語や文書整形言語を扱え、ユーザの期待する情報を出力し、文芸的プログラミングを支援するシステムの作成を目的とする。

文芸的プログラミングを支援する1つの方法として、未解決である「プログラム構造の理解が困難」という問題を解決するために、MulTAS や Xas-LP を踏まえて、新たな支援の土台として XMulTAS を作成する。そして、ユーザの与える視点によって抜き出した情報でプログラムの構造を表すことによって、ユーザが容易にプログラムを理解できるようにする新たな支援について検討する。

## 2 XMulTAS

### 2.1 XMulTAS 実現に向けて

MulTAS や Xas-LP では、いくつかある WEB システムの問題点の内の次の問題を解決している。

- WEB 文書<sup>\*6</sup>編集時に説明文書を参照することができない ( MulTAS )
- 開発言語・整形言語が固定されている問題 ( Xas-LP )

MulTAS は4つの固定された視点 ( Edit, Browser, Tree, Index ) をもち、ユーザはそのビューを参照しながら WEB 文書を記述する。Edit には WEB 文書、そして WEB 文書の変更を MulTAS に伝えることによって Browser に整形された説明文書、Tree にプログラムの構造を示すためのセクションの呼び出し関係、Index に重要語句の登録状況をそれぞれ表示する。また、Xas-LP では XML を使用して WEB 文書を表し、プログラムや説明文書の抜き出しに XSLT を使用することで、プログラム言語・整形言語の固定問題の解決法と XSLT の数だけビューを表示できることを示している。MulTAS における Browser, Tree, Index は WEB 文書からそれぞれの視点によって抜き出したビューと考えられる。そこで、XML での WEB 文書から XSLT で抜き出すことによってこの3つのビューは表現できると考える。

また、文芸的プログラミングはセクションを単位としてプログラムする方法であることから、各セクションを別々に編集することにする。そして、セクション毎の編集のために、SMulTAS ( Section-MulTAS ) を用意す

<sup>\*1</sup> 情報から必要な情報を抜き出すときの注目点

<sup>\*2</sup> 視点を基に抜き出した情報を表示したもの

<sup>\*3</sup> 文芸的プログラミング提唱者 Donald E. Knuth によって開発されたシステム、WEB システムはプログラム言語に Pascal、整形言語に TeX を使用し、この2つを WEB 言語という独自の言語によって制御している。また、プログラム言語に C 言語を使用できる CWEB というシステムも存在する。これらは主にプログラム保守時に支援を行い、プログラム作成時の支援は乏しい。

<sup>\*4</sup> 我々の研究室がこれまでに作成したシステムの1つである。CWEB をもとにして作成されており、CWEB の問題点であるプログラム作成時の支援の不足を解消するために作成された。

<sup>\*5</sup> XML タグで WEB 言語を表し、XSLT の使用によりデータを変換することによって、開発言語と整形言語の固定問題を解決しようとした試作品である。

<sup>\*6</sup> WEB システムによって扱われる文書

る。このため、XMulTAS は SMulTAS を呼び出すシステムといえる。

## 2.2 WEB 文書の構造について

XMulTAS における WEB 文書は、開発言語・整形言語固定の問題を回避するために Xas-LP に引続き XML タグを WEB 言語として使用することにする。そして、XML タグを使用して記述した WEB 文書を XMLWEB 文書とする。SMulTAS によって管理される 1 セクション分の XMLWEB 文書は以下の構造になる。

```
<sec title="考察" id="1">
  <doc>
    1 から 1000 までの素数の計算を行う。
  </doc>
  <path title="標準ヘッダー" id="2"/>
  <path title="エラストテネスのふりい" id="3"/>
  <path title="メイン関数" id="4"/>
</sec>
```

各セクションの XMLWEB 文書にあたる<sec>タグ内には説明文書にあたる<doc>と対応するセクション先を指定する<path>、プログラム言語で記述されたプログラムコードにあたる<program>が 0 個以上自由に配置できる。そうすることで、ユーザにセクション分けを強要することなく、ユーザの自由なセクション分けを許している。

## 3 本研究で扱う問題点について

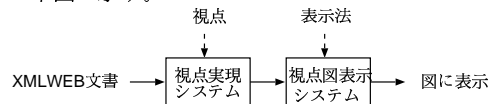
文芸的プログラミングでは、プログラムをセクション分けすることによってセクションという塊でプログラムを理解できる。しかし、これは利点であると同時に欠点でもある。セクション分けは内容を中心として行われるため、プログラム言語によって記述されるプログラムコードにおいて塊であったとしても、内容によってさらに細分化され、プログラム言語の構文規則からみるとプログラムコード理解の面で困難になっている。MulTAS では Tree でプログラムの構造を表現することによってこの問題を解決しているが、Tree はセクション間の呼び出しの表示のみに留まっており、プログラムコード面の支援はできていない。そのため、プログラムコード面における様々な視点を与えることによって拡張され、プログラムの構造を表示する支援について検討する。

■プログラム構造の図表示 MulTAS の Tree は、セクションの呼び出しを視点として木構造でプログラムの構造を表示したものである。そして、呼び出しは WEB 言語で記述されたものを調べれば見つけられる。XMulTAS の XMLWEB 文書においては<path>がそれにあたり、<path>を探ることによってセクションの呼び出しを視点としたプログラム構造を表示できる。また<path>は、XMLWEB 文書という文字列の中のある特定の文字列ということができ、セクションの呼び出しの情報は<path>によって抜き出したということが出来る。そのため、他の文字列で抜き出した情報からは違ったプ

ログラムの構造を表示できると考えられ、XMLWEB 文書という文字列を解析することによってプログラム構造を表示できないかと考える。また、文芸的プログラミングはセクションを単位としているため、プログラム構造はセクション単位で表示することにする。

## 4 拡張機能について

ユーザから視点を受け取り、図に表示するまでの流れを下図に示す。



表示までを 2 段階に分け、視点実現システムは受け取った視点を基に XMLWEB 文書から情報を抜き出し、視点図表示システムは抜き出した情報を表示する。そして、それぞれどのような情報を抜き出すかの視点とどのように表示するかの表示法の指定が必要となる。

### 4.1 正規表現

セクションを中心に構造を表示するため、まず各セクションの XMLWEB 文書を解析して、セクション間の関係を探し、表示構造を組み立てる方法を考える。プログラムコード面に重点を置いて、視点として変数の宣言参照関係を表示したい場合、XMLWEB 文書内の変数名文字列とその前後を調べることによって関係を見付けられると考えたが、予想できるユーザの要求の多くがプログラムコード面に対する要求であること、得られるプログラムコードの情報が少ないこと、またプログラムコード面における要求全てに対処できないことからプログラムコード面に重点を置くこととし、文脈自由文法でプログラムコードを解析して関係\*7を探ることにする。

### 4.2 視点実現システム

#### 4.2.1 検討

このシステムは以下の 3 つの理由により、文脈自由文法を用いて XMLWEB 文書のプログラム部分のみを解析する。

■ユーザの視点の実現の要求について XMLWEB 文書に対するユーザの要求を考えると、関数の呼び出し関係の表示、ある関数で宣言されている変数の集合の表示など、プログラムに関係した要求が多いと考える。よって、XMLWEB 文書に対して視点の実現を行おうとしたとき、プログラムの構造を利用して解析する必要があると考える。

#### ■XMLWEB 文書の性質 1

XMLWEB 文書は、XSL を使用することによってプログラムコードを生成することが可能である。よって XMLWEB 文書の解析の方法として字句解析、構文解析などのプログラムを解析する方法を適用することができる。

#### ■XMLWEB 文書の性質 2

\*7 ユーザの与える視点に基づいた情報を含むセクションとセクションのペア

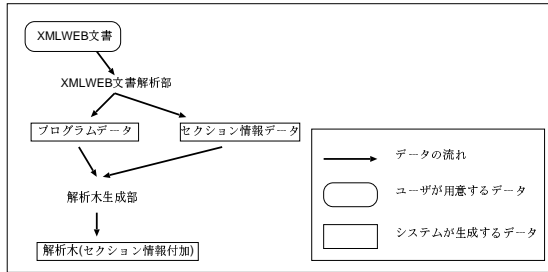
XMLWEB 文書のプログラムコード部分と説明文書部分は互いに関係していて、プログラムコード部分の間の関係と説明文書部分の間の関係は同じと考えられる。

これは、プログラムの関係にのみ着目して関係の抜き出しを行っても説明文書間の関係も同じような関係で抜き出されたと考えることができる。

#### 4.2.2 実現方法

##### ■処理の流れ

視点実現システムの処理の流れは以下ようになる。



視点実現システムはユーザが作成した XMLWEB 文書を視点実現システムが持っているプログラム言語の文法を表す BNF をもとに XMLWEB 文書のプログラムコード部分を解析する。

視点実現システムは、XMLWEB 文書を XMLWEB 文書解析部に通し、プログラムデータとセクション情報データを出力する。このプログラムデータは XMLWEB 文書のプログラム部分のみを抜き出したものであり、セクション情報データはプログラム部分抜き出しによって失われてしまうプログラムコードの一部がどのセクションに属しているかという情報を表すものである。

この2つのデータをもとに解析木生成部がプログラムコードの字句解析と構文解析を順に行い、解析木を生成する。またその解析木の終端記号にはセクション情報データをもとにその記号がどのセクションに属しているかという情報を付加する。

##### ■解析木について

プログラムコード部分の構造を解釈するため字句解析と構文解析を経て解析木を生成し、その可視化された情報をもとに視点を実現しようとする。

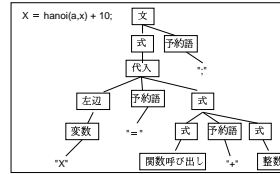
プログラムの構造を表す方法として解析木を使うか抽象構文木を使うかという選択がある。

解析木は BNF を用いて解析した結果そのものの木の構造になっている。抽象構文木はコンパイラが実行コードを生成するために必要な本質的な部分のみを抜き出すので、そのために必要のない非終端記号は削除される。それによって木の全体のサイズが小さくなり、木の探索時間も短くなる。

視点実現システムが解析木を使用した理由は、ユーザが視点の実現のために視点実現システムに理解しやすい定義を行わないとにならないが、この視点実現システムでは解析木の構造をもとに指定できると考える。

##### ■簡単な例

代入文のプログラム断片を解析した結果を示す。



変数“X”の代入されている場所を知りたいとユーザが望んだ場合、非終端記号\*8“代入”の子の要素である“左辺”の子の要素である“変数”が“X”である場所を探せばよい。何が代入されているかは“代入”の子の要素である“式”のなかを見ればわかる。

#### 4.3 視点図表示システムについて

視点実現システムから解析木を受け取り、視点図表示システムが図に表示するためのデータベースとして保存する。ここで情報を分類し、XML 文書を作成する。XML を使うことで、解析木の構造をタグの入れ子で容易に表現することができ、DOM ( Document Object Model ) を使って XML 文書から容易に情報を抜き出すことが可能である。

情報の種類は解析木で表したプログラムコード情報と、プログラムコードがどのセクションに含まれるかの情報の2種類がある。

解析木の非終端記号毎にタグを作成しそれを要素\*9とする。非終端記号の文字列は要素名として記述され、非終端記号のすぐ下に終端記号がくる場合に、終端記号の文字列は要素の内容として記述される。終端記号がどのセクションに含まれるかの情報は要素の属性として記述される。

図1は変数 X の宣言文を表す部分的な解析木を XML 文書に変換する例である。

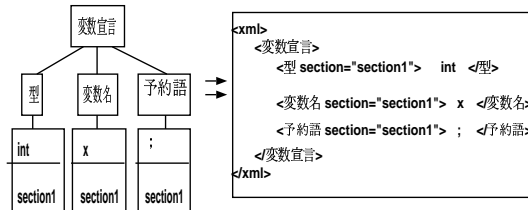


図1 XML 文書の変換の例

解析木の構造をそのまま XML 文書に表現することで、視点図表示システムがプログラムコードを人間がプログラムに対する認識に近い方法で分類でき、明示的な情報量が増える。

XML 文書は解析木の構造を表現しているため、ユーザの視点に基づく情報を XML 文書にしたとしても、同じように抜き出すことが可能である。XML 文書から視

\*8 文法を形式的に扱う際の、<関数><変数>などの文法上のものの総称。

\*9 <要素名 属性名=" 属性値" >内容 </要素名> 開始タグから終了タグまで要素という。

点に基づく情報を抜き出し、視点の情報をタグの属性として付けることで、視点毎に解析木を作り、XML 文書に変換するよりファイルのサイズが小さくなる。

またプログラムの構造は解析木だけでなく、セクション間の関係でも表せると考えられるため、セクション間の関係情報をセクションのペアで表し、視点の意味を持つ情報で作られたタグに囲むことで表す。

#### <視点の名前>

```
< section1 path="section2" />
< section1 path="section3" />
```

#### </視点の名前>

### 4.4 ユーザによる視点の定義方法について

#### 4.4.1 ユーザによる視点の定義方法

視点を定義してもらう方法としてこのシステムでは解析木の構造を利用している。ユーザは注目したいプログラム断片つまり解析木の部分木を示すためにその部分木のルート要素(非終端記号)を記述する。そのあとに、その部分木の特徴(子の要素の構造がどのようになっているかなど)を記述することによりユーザが求めている要素を指定できる。

例えば変数 X に値が代入されているプログラム断片を知りたいければ、次のように指定をする。

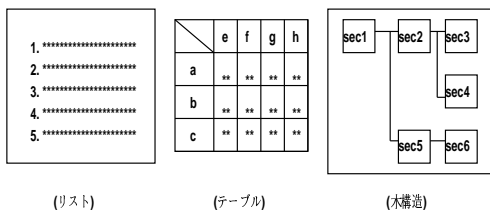
```
(代入){ 代入 < 左辺 < 変数 = "X" }
```

始めの“代入”は注目した部分木のルート要素で“{}”に囲まれた部分はその部分木の構造を示している。この例はルート要素が“代入”であり、その子の要素が左辺でその子の要素が変数でありその先の終端記号が“X”である部分木が抜き出される。

#### 4.4.2 ユーザによる表示の指定方法

ユーザが視点を与えることで、視点に基づく情報が得られる。そして、ユーザがその情報をどう表示するかを指定し、望み通りの表示を参照することによって、よりよく理解できると考える。

表示方法はリスト構造、テーブル、木構造の3種類に限定し、ユーザに表示方法と表示内容を指定してもらう。リスト構造やテーブルは基本的な表示方法であり、木構



造は関係やプログラムコードの構造を表現するのに適すると考えられる。

ユーザが視点の名前とリスト項目になる情報の決め方を指定することでリスト構造を表示する。

ユーザが視点の名前とテーブルの列、行項目になる情

報の決め方を指定することでテーブルを表示する。

木構造は、『XML 文書の非終端記号を木構造の要素として表現した木構造』と『セクション間の関係に基づき、セクションを木構造の要素として表現した木構造』が考えられる。

ユーザが表示させたい部分木のルート要素と要素名、属性、内容のどれを表示させるかを指定することで XML 文書の非終端記号を木構造の要素として表現した木構造を表示する。

ユーザが表示させたい視点と木構造のルート要素になるセクションを指定することでセクション間の関係に基づき、セクションを木構造の要素として表現した木構造を表示する。

木構造には木の要素をクリックした時にその要素に関する詳細な情報をユーザの指定によって、参照できるようにする。こうして、1つの視点の表示から、その視点に基づく複数の情報を参照することが可能になる。

表示する視点に基づく情報は XML 文書から抜き出すので、XML 文書に記述されている要素名、属性名、属性値、要素の内容の全てがユーザの指定対象とする。

## 5 今後の課題

### 5.1 視点実現システムにおけるユーザ負担の軽減

今回検討したシステムは解析木を使用して、その木の構造に基づいてユーザに指定してもらっているが、解析木の構造を理解できない人もいるかもしれない。より直感的に理解しやすい表現を用いるべきである。

### 5.2 視点実現システムの拡張可能性

XMulTAS は拡張により複数の言語に対応するが、視点実現システムでは、あらかじめ用意された文法を基に解析木を作成する。よって文法が用意されていない言語については文法をユーザが用意しなければならないが、それを用意するのはユーザに対し負担が大きい。文法作成の支援を行う必要がある。

### 5.3 XMLWEB 文書が更新された場合の処理

XMLWEB 文書を更新した場合、解析木も新たに作ることになる。そして、新たに解析木を作成することによって視点表示システムにおける XML 文書も書き直され、視点に関する情報が全て初期化されてしまう。そのため、何度も視点を入力する必要があるため、変更のあったセクション部分の XML 文書のみ更新や、視点を保存しておくなどの工夫が必要である。

## 参考文献

- [1] 板谷 尚哉: 文書としてのプログラム記述方式とマルチビューシステムの設計, 南山大学 卒業論文 (2000)
- [2] 余吾 幸彦: 文芸的プログラミングにおける利用者インタフェースの検討と試作, 南山大学 卒業論文 (2001)
- [3] 荒引 朱美: 拡張可能な CWEAB 拡張システムに関する考察, 南山大学 卒業論文 (2003)