

リンク自動付与機能を持つ技術文書サーバの開発手法

3AC-3

荒木 円博, 手嶋 茂晴

豊田中央研究所

E-mail: foison@mes.tytlabs.co.jp, teshima@iclab.tytlabs.co.jp

1 はじめに

WWWブラウザの普及に伴い、仕様書、取扱説明書などの技術文書をHTML形式のハイパテキストとして提供するサーバの需要が高まっている。こうした技術文書向けのWWWサーバにとって重要な機能は一貫した方針に基づくリンクの自動付与である。

一貫した方針に基づくハイパテキストシステムの設計・開発支援に関しては、RMM, OOHDMMなど^[1]がある。これらの手法は、Entity-Relationshipやオブジェクト指向に基づく設計方法論、プロトタイプ生成ツールなどを提供している。

しかし、これらの手法では、いったんハイパテキストシステムを構築した後に、ノードの粒度の変更、例えばより細かく分割するのは難しい。このことは特にノードの数が多いうちに問題となる。そのため、システムを本格的に運用しはじめた後では、利用者の要望に応じたノードの粒度変更は困難である。

そこで我々は、ノードの粒度変更にも容易に対応できる開発手法を目指す。

2 技術文書向けWWWサーバとは

我々が想定する技術文書向けWWWサーバは、WWWブラウザからの参照要求を受けた時点で、参照要求に対応する文書やその構成要素にリンクを自動付与し、WWWブラウザにHTML形式で送るシステムである。

一貫した方針に基づいてリンクを自動付与するために、質問対リンク^[2]を利用する。そのためには、ノードがオブジェクト指向のインスタンスである必要があるため、文書をリンクづけに必要な構成要素まで分割し、各要素をインスタンスとする(図1参照)。

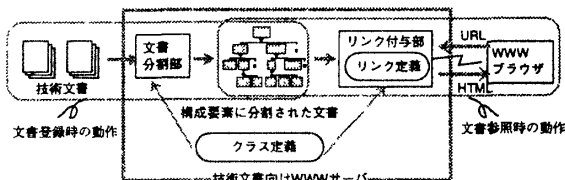


図1: 技術文書向けWWWサーバの基本構成

A method to develop technical document WWW server which can hyper-link automatically
 Mitsuhiro Araki, Shigeharu Teshima
 Toyota Central R&D Labs., Inc.

3 技術文書向けWWWサーバの開発手法

ノードの粒度の変更を容易に行うには、クラス定義の変更に応じて、もとの文書をインスタンスに分割し直し、さらに変更によって影響を受けるリンク定義を変更すればよい。そこで、下記のように、これらの変更を考慮したサーバ開発手法を検討した。

(1) オブジェクトモデル図(図2参照)の作成

対象とする技術文書について、必要なノードやリンクが現れるまでオブジェクト指向分析を行い(OMT法)^[3]、オブジェクトモデル図を作成する。リンクは関係として、ノードはクラスとして、リンクの制約条件で使う値は属性として扱う。

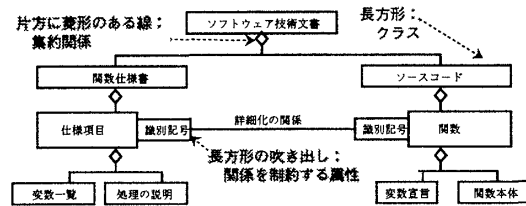


図2: オブジェクトモデル図の例

(2) オブジェクトモデル図からの定義などの作成

クラス定義: クラス, 属性, 集約関係(has-a), および汎化関係(is-a)をもとに作成する。

リンク定義: 関係と属性をもとに作成する。

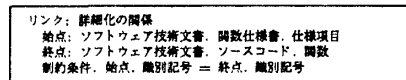


図3: リンク定義の例

文書分割部: クラス, 属性, および集約関係をもとに作成する。対象の文書がLALR文法に基づくテキストファイルならば、構文解析規則を作成すれば、yaccなどのコンパイラ・コンパイラによって文書分割部を作成できる。

HTMLのテンプレート: WWWブラウザに送るべき構成要素に対応するクラス名や属性名を列挙し、必要に応じ<FRAME>などレイアウト用のタグなどを追加する。なお、クラス名や属性名は、リンク付与部によって、実際のインスタンスの値に置き換えられる。

4 技術文書向け WWW サーバの実現

これまで、車載用制御ソフトウェアや産業用ロボット制御ソフトウェアの仕様書、ソースコード、マニュアルなどを対象にサーバを開発してきた。

まず、これらに対してオブジェクト指向分析を行った(図4参照)。

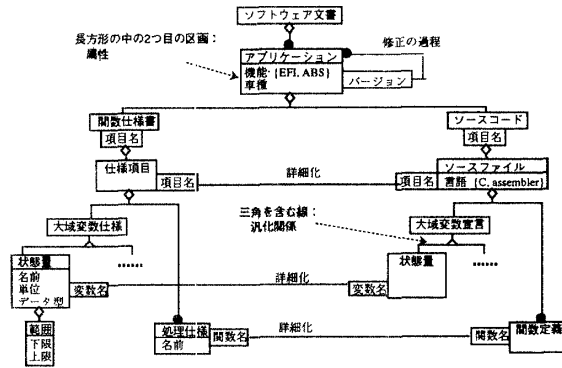


図4: 車載用制御ソフトウェアの分析結果概要

ハイパテキストやソフトウェア工学の分野では、下記のリンクの有用性が知られている。そこで、これらのリンクがオブジェクトモデル図に現れるまで分析を行い、文書をどこまで分割すればよいか明らかにした。

- (1) 集約関係を表すリンク
- (2) 汎化関係を表すリンク
- (3) 表現の詳細度が違う、同じもの間のリンク
- (4) 改訂前後の関係(修正の過程)を表すリンク
- (5) 構成要素の定義箇所と使用箇所との間のリンク
- (6) 作成者に関する情報と、作成物との間のリンク
- (7) 構成要素と、その使用例(実行例)との間のリンク
- (8) データフローに基づくリンク

そして、その結果に基づいて、下記のようにサーバの実現を行った(図5参照)。

分割後の技術文書の構造: 今回は OS ファイルシステム上での実装だったため、SGML の入れ子構造を利用して集約関係や汎化関係を表現した。同様に属性は SGML の属性で表現した(例: 図6)。

クラス定義: SGML による文書構造表現にあわせ、SGML の文書型定義(DTD)として実現した。

文書分割部: 文書ファイルの形式、中身の形式や種類ごとに作成した。例えば、plain text で書かれた A 社の関数仕様書用、Microsoft Excel で書かれた B 社の関数仕様書用、などである。

リンク付与部: 今回は、リンク定義を解釈する実装ではなく、直接コーディングした。今後、改良する。

また、クライアントとの通信、認証、暗号化を既存の WWW サーバに受け持たせた。リンク付与部自体は、WWW サーバから呼び出される CGI プログラムとして実装した。



図5: 技術文書向け WWW サーバの実装

```

<ソフトウェア文書>
<アプリケーション 機能=EFI 車種=Corona バージョン=1.2>
<関数仕様書>
<仕様項目 項目名=初期化>
<大域変数仕様>
<状態量 変数名=ptr_tmp 名前=水温 単位=Celsius アーク型=TEMP_ID>
<範囲 下限=-20 上限=80>
</状態量>
</大域変数仕様>
<処理仕様 関数名=init_fan 名前=初期化基本関数>
...
</処理仕様>
</仕様項目>
</関数仕様書>
</アプリケーション>
</ソフトウェア文書>
    
```

図6: SGML による構造表現の例

5 評価

ABS(Anti-lock Brake System) 制御用ソフトウェアの関数仕様書およびソースコードを対象に、実際の開発チームで評価を行い、有効性を確認した。

例えば、1つの関数を修正してから関連する全文書を修正するのにかかる時間について、導入前には1~2週間かかっていたのが、1~2日に短縮された。短縮の理由は、開発チームで必要としているリンクが過不足なく抽出できたためである。その結果、容易に関連文書が参照でき、修正期間が短縮された。

6 まとめ

一貫した方針に基づくリンクの自動付与や、ノードの粒度変更への対応を容易に実現することを目指して、オブジェクトモデル図に基づいて技術文書向け WWW サーバを開発する手法を検討した。また、その手法に基づいてサーバを実装し、手法の有効性を確認した。

今後は、より容易にサーバを実現できるように、オブジェクトモデル図から自動ないし半自動的にサーバを作成する技術を開発していく。

参考文献

- [1] "Designing Hypermedia Applications", CACM, Vol. 38, No. 8, pp. 26-112, 1995.
- [2] K. Tanaka et. al.: "Query Pairs As Hypertext Links", IEEE Data Eng. Conf., 1991.
- [3] J. Rumbaugh, et. al.: "Object Oriented Modeling and Design", Prentice Hall, 1991.