

ソフトウェア部品管理リポジトリ

6K-2

北野 拓哉

NEC C&C 研究所

1 はじめに

本稿で議論するリポジトリは、ソフトウェア部品の開発や運用のために作成した仕様書や説明書などのソフトウェアドキュメントを管理するリポジトリである。ソフトウェアドキュメントはSGML(Standard Generalized Markup Language)の構造化文書であったり、特定の情報を含んでいたり、ハイパーテキスト化されていたりするなど、ある構造を有している。本リポジトリはオブジェクト指向データベース(OODB)を用いてこの構造をスキーマとして管理する。よって、スキーマに基づくリンク関係を維持しながらドキュメントの生成や変更を行ったり、リンク関係や文書部品の親子(包含)関係などをたどりながら目的の情報を引き出ししたりすることなどが可能となる。

2 リポジトリの基本構造

本リポジトリは、ソフトウェアドキュメントを図1のメタモデルに従って管理する。このモデルは、SGMLのDTD(Document Type Definition)で定義される親子関係と文書部品間のリンク関係で構成される。ドキュメントの構造化にはタグを用いる方法を採用し、タグの構造的関係をクラススキーマとしてOODBが管理する。DTDで定義されるタグの親子関係は、クラス「タグ」が他の0以上のタグを包含する関係でモデル化される。文書部品間のリンク関係は、クラス「リンクタグ」の集合によってモデル化される(例が3章の図2にある)。実際にドキュメントに張り付けるタグは、クラス「タグ」、「リンクタグ」のインスタンスとして定義される。そしてこのタグによって識別される文書部品は、クラス「文書部品」、「文書リンク部品」のインスタンスとして管理される。また「文書リンク部品」の文書部品は、それを包含する「文書部品」を有するものとしてモデル化する。これは個々の「文書リンク部品」が、木構造をなすDTDに基づくSGML

構造化文書のどの「文書部品」中に出現しているのかを管理するためである。

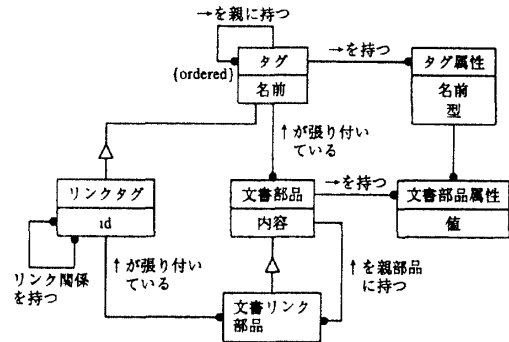


図1: ソフトウェアドキュメントのメタモデル

3 ソフトウェアドキュメントの管理

ソフトウェアドキュメントをリポジトリで管理するには、ソフトウェア部品の情報、すなわちソフトウェアドキュメントの文書部品を引き出す枠組として、タグの構造的関係を決定しなければならない。

MicrosoftのOCXに関するソフトウェアドキュメントの管理を例に挙げる。OCXの設計情報にはクラス、属性、操作、関連などの情報があり、運用情報にはメソッド/プロパティや他のドキュメントファイルなどがある。これらの情報を示すドキュメント中の文書部品やそのタグを、それぞれ図1のクラス「文書リンク部品」と「リンクタグ」のインスタンスとして管理するため、まず「リンクタグ」のクラススキーマを与える。図2のリンクタグの部分がそのスキーマである。

次にこのモデルの各クラスのインスタンス化を行って、ドキュメント中の文書部品をOODBで管理できるようにする。以下のソフトウェアドキュメント[†]をリポジトリへ登録する。

全国を下記のように分割し、`<at id="1"> 駅名 </at>` や`<cl id="1"> 路線 </cl>`の絞り込みをスムーズにさせます。`<cl id="2"> 地域 </cl>`を限定するとその、`<cl id="2"> 地域 </cl>`以外の`<at id="1"> 駅名 </at>`、`<cl id="3"> 会社 </cl>`、`<cl id="1"> 路線`

A repository to manage software components
Takuya Kitano
C&C Research Laboratories, NEC Corporation

[†](株)ヴァル研究所 駅すばあと全国版のヘルプより抜粋

</cl> は指定できません。
 ※ 「<pr id="1"> 使用路線 </pr>」を設定している場合は <cl id="1"> 路線 </cl> の指定が優先します。

ドキュメント中のタグをそれぞれ図 2 の各リンクタグのクラスのインスタンスとして登録する。

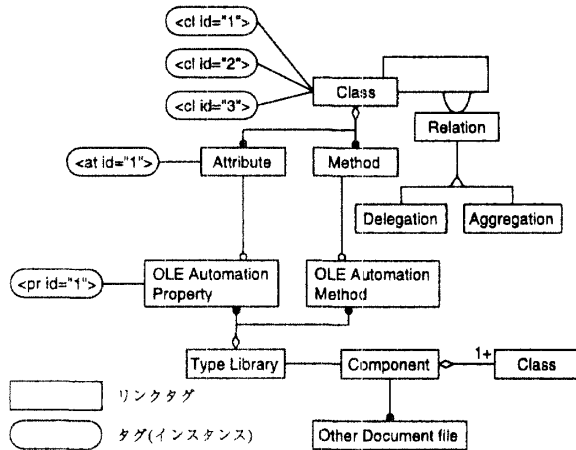


図 2: ソフトウェアドキュメント内のタグの構造化とその登録

タグ付けされた文書部品を図 1 のクラス「文書リンク部品」のインスタンスとして登録する (図 3)。このとき、図 3 のタグを結ぶ実線 (破線ではない) が示す通り、図 2 のクラススキーマに基づいたタグのリンク関係も登録する。図 3 の id の値がないタグは、その対象となる文書部品が存在しないためまだ未登録であるが、図 2 のクラススキーマの制約に基づき、将来登録されなければならないことを示す。

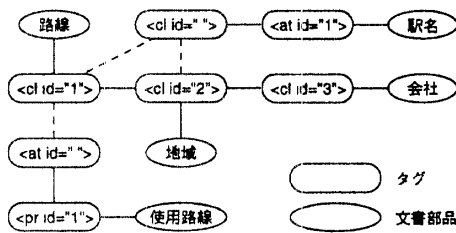


図 3: 文書部品とそのリンク関係の登録

このようにソフトウェアドキュメントを文書部品単位でリポジトリに登録することで、構造の制約に基づいた情報洩れのない一貫したドキュメントの作成が行える。

4 ソフトウェア部品の情報検索

3章で述べた方法でソフトウェアドキュメントを構造的に生成し管理すれば、全文検索エンジンなど

と組み合わせて以下のような構造に基づく情報検索が可能となる。

情報検索の一例

「路線」を検索ワードとして、これに関するソフトウェア部品の情報を引き出す。(注意: 以下では図 3 で未登録であった id の値がないタグやその文書部品、リンク関係 (破線) は登録されたものとする。)

1. 全文検索エンジンによりソフトウェアドキュメント中に検索ワード「路線」を発見する。
2. 「路線」にはタグ <cl id="1"> が張り付いているので、ある Class の情報であることが分かる。
3. Class 「路線」の Attribute は「使用路線」という OLE Automation Property を持つことが分かる。
4. Class 「路線」と関連のある Class 「地域」と Attribute が「駅名」である Class を得ることができ、ソフトウェア部品のクラス構成が分かる。

また DTD に従ったタグの親子関係も、図 1 のクラス「文書リンク部品」、「文書部品」、「タグ」の関連と「タグ」の再帰的関連でモデル化して管理しているので、「路線」を含む文書部品 (例えば文書中の節) やさらにその親部品 (例えば文書中の章)、兄弟部品 (例えば文書のタイトルや著者) も表示させることもできる。

5 まとめと今後の展望

ソフトウェア部品に関するドキュメントは、ある決まった情報を含み、かつそれらの情報が決まった構造を形成して存在する。よって、そのような構造の規則に従ってソフトウェアドキュメントは生成される必要がある。本リポジトリはこれを OODB を用いて管理することにより実現させた。またその決まった構造はソフトウェア部品の種類や開発グループの開発方法によって異なるが、タグの木構造やリンク構造はスキーマとして生成し管理するので、そのスキーマの定義次第で柔軟に対応できる。さらに、情報の検索も構造に基づくナビゲーションによって次々得ることができ、単純なリンクをたどる HTML のような情報の探し方は不要となる。

今後は WWW への対応を予定しており、OCX に関する情報の検索、ドキュメントのキーワード検索、ハイパーテキストナビゲーションなどの操作を Web ブラウザ上で行えるようにする予定である。