

ソフトウェア設計文書テンプレート方式の検討

3Q-6

飯尾和彦 忠海 均

NTTソフトウェア研究所

1. はじめに

高品質なソフトウェアを作成するための数々の取り組みがなされている。特に、開発の上流工程で品質を向上させる、すなわち機能設計書、詳細設計書などのソフトウェアドキュメントの品質を向上させることが、高品質なソフトウェアを作成する上で重要であることが知られている。

国際規格ISO8879で定められているSGML(Standard Generalized Markup Language)を使うことによってドキュメントの品質の向上が期待できるが、現状では操作性や運用性などに問題がある。

本稿では、既存の設計文書を分析することによって、ソフトウェアドキュメント作成や管理の際の操作性や運用性を向上させる方法を検討した。

2. ドキュメントの品質

高品質なソフトウェアドキュメントが満たすべき条件¹⁾に対し、SGMLを使うことの効果を評価すると以下ようになる。

A. 効果が認められる点

アクセシビリティ	リンク構造を定義することで、瞬時に関連箇所を引き出す構造を実現できる。
管理性	機能別にファイルを分けて作成し、管理情報は属性としてSGMLファイルに埋め込むという構造をとることができる。
コーディングリストへの追跡性	リンク構造を定義することで、瞬時に関連箇所を引き出すしくみを実現できる。
ドキュメント簡潔記述性	意味的なタグを多く定義しておくことで、表で整理するように簡潔に表現できる。

A. 効果が認められる点（続き）

統一性	組織内でドキュメントドキュメント作成ガイドに沿って書かれていることがコンパイル時にチェックできる
-----	--

B. 効果のない点

ダイアグラム簡潔記述性	ソフトウェアで使われているダイアグラムについて簡潔に記述できる能力はない。
-------------	---------------------------------------

3. SGML適用時の問題点

品質の面からは、設計ドキュメントをSGML化する利点は大きいといえる。しかし、SGMLでドキュメントを記述していく場合には以下のような問題が存在する。

- ・文脈によって使用できるタグが変わる。
- ・タグの名前を覚えておかななくてはならない。
- ・ドキュメントの量が増えたときの管理が複雑になる。
- ・新たにSGMLエディタの操作を覚えたくない。
- ・現状では、まだ使いやすいSGMLエディタは存在しない。

4. 方式の説明

既存の要求仕様書や作成ガイドラインを分析したところ、文書構造の共通性を多く見つけた。これにより、共通項目をテンプレート化する方式が、作成・管理の文書の操作性が向上する上で有効と考えた。

A. 各ワープロごとにテンプレートを作成する。

ドキュメント種別毎にSGMLで記述した共通のDTDに従い、ワープロ毎にテンプレートを作成する。これによってエンドユーザが自分の使いなれた好みのワープロを使い、SGML文書を作成できる。また、コンパイルによるチェック機能により、文書構造の共通化ができる。

A Study of Template Application Method for Software Design Document

Kazuhiko Iio, Hitoshi Tadaumi

NTT Software Laboratories

3-9-11 Midori-cho Musasino-shi Tokyo 180 Japan

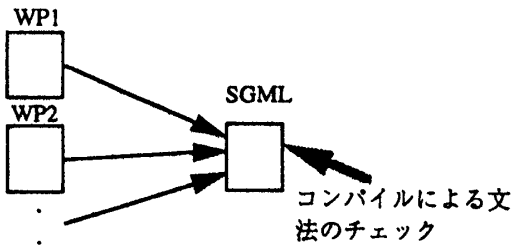


図1. テンプレート方式によるSGML文書の作成

タイトル	料金業務システム用設備
バージョン	2.3
社名	日本電信電話株式会社
所属	ソフトウェア研究所
機能概要	<機能概要>
目的	<目的>
位置づけ	<位置づけ>
...	...

図2. テンプレートの例

B. 共通項目の再利用

「リファレンス」、「用語」、「運用条件」、「品質条件」の箇所などパターン化できるものが多く見つかった。この部分は、general entityの機能を用いて入力を省力化できる。例えば運用条件の箇所の「センタ設備のソースプログラムのファイル転送、.....等については、センタ設備と連動して行えること」というエンティティを

```
<!ENTITY softcondition1 "センタ設備のソース
.....センタ設備と連動して行えること">
```

のようにエンティティ定義しておき、SGML文書作成の際に

```
&softcondition1;
```

と入力することで、パーサが自動的にエンティティの中身を展開することができる。

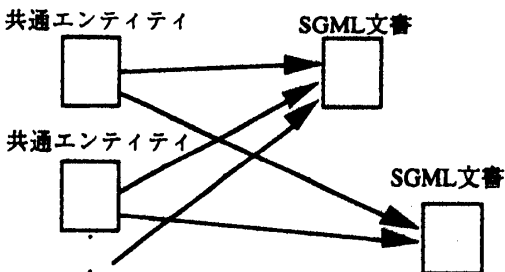


図3. 共通エンティティ方式

このようなリンク構造をとることによって、共通エンティティに変更があるときでも、再コンパイルによって即座に対応できる。

C. 管理項目とドキュメントの一体化

ファイルが大量になっても整理して管理できるように管理項目をSGMLのattribute機能を使って保存する。作成年月日やバージョンはもちろんのこと、整理番号やドキュメントの状態なども記憶させる。これにより、より多くの検索・分類が可能となる。

D. 文書の管理方式

さまざまな種類のワープロで作成しても、整合のとれた文書の構成管理をしておく必要がある。整合をとるためには、以下のような方式が一例として考えられる。

SGML文書構成管理者は、原ワープロ文書からSGML文書が一意に生成できることを確認し、原文書とSGML文書をペアにしてデータベースに保管する。原文書とSGML文書の一貫性を保つために、修正が生じた場合には、ドキュメント作成者は原文書の方を訂正し、パーサのチェックによりSGMLに反映する。将来的に使いやすいSGMLエディタが出てきた場合にはSGML文書のみ管理に体制をシフトする。

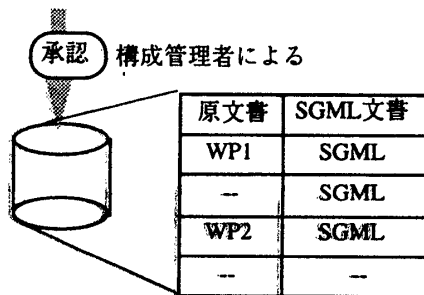


図4. 設計書文書管理モデル

5. 今後の課題

今後は、上記の方式を実際に適用して評価する。

また、ソフトウェアドキュメントにはさまざまなダイアグラムの書式が存在する。構造を持つ表現方法としてそれらを再利用するために、ダイアグラムの書式をSGML文書に、組み込む方式などを考察していきたい。

参考文献

- (1) Michael S.Deutsch, Ronald R.Will著, 成田光彰訳, 「ソフトウェア品質工学」.日経BP社, 1990
- (2) 藤井他: ソフトウェア設計情報共有化方式の一考察, 信学技報, SSE93-127