

XML 文書の構造を利用した品質評価手法の提案と実現

金 京煥

内容概要

インターネットが発展するにつれ，Web を利用した電子文書の重要性が高まり，特に，W3C で提案された XML に対する研究が幅広く行われている．

電子文書は情報の共有や，文書の再利用性，検索性などを向上させることを目的に構造化言語を用いて記述する．しかし，全ての電子文書がこれらの性質を満たしているとは限らないため，電子文書の構造を調査し，実際にこれらの目的に即するかを評価する必要がある．

XML で記述された文書は様々な用途があるため，各々の文書が要求されている品質を持っているかどうか評価しなければならない．よって，XML で記述された文書の品質を評価する際には，用途に応じた評価基準を設定した上で，それに基づいて評価を行えばよい．これまで，電子文書の評価する際には，文書デザインや内容の評価に重点がおかれ，文書の持つ構造が評価対象となることは少なかった．このため，XML 文書の再利用性や文書作成の柔軟性などに着目した評価を行なうことは困難であった．

本研究では，XML で記述された文書を対象として，DTD 定義に基づいて評価するための手法を提案する．本手法は，多様な文書データに対応できるように DOM モデルより評価用のデータ構造記述を導出し，導出された構造に対して読みやすさ，文書作成の柔軟性，再利用性，メンテナンス性を評価するための基準を定義し，評価を行なった．

主な用語

XML (eXtensible Markup Language)

DTD (Document Type Definition)

DOM ツリー (DOM tree)

品質評価 (Product Evaluation)

目次

1	まえがき	4
2	構造化文書と XML	6
2.1	文書構造化	6
2.2	XML	7
2.2.1	要素	8
2.2.2	文書型	8
2.2.3	DTD	8
2.2.4	Valid XML 文書と Well-formed XML 文書	9
2.2.5	DOM	9
2.2.6	スタイル指定	9
2.3	XML , SGML , HTML の比較	10
3	構造化文書の評価	12
3.1	HTML 文書の評価	12
3.2	XML 文書への適用	14
4	XML 文書の品質評価手法	15
4.1	前提条件	15
4.2	XML 文書の品質評価手法の手順	15
4.3	評価用 DOM ツリー	16
4.4	DOM 変換ルール	17
4.4.1	モジュール	17
4.4.2	構造リンク	17
4.4.3	DOM 変換ルールの解析	17
4.5	品質特性	19
4.5.1	読みやすさ	19
4.5.2	文書作成の柔軟性	19
4.5.3	再利用性	19
4.5.4	メンテナンス性	20
4.6	評価基準	21
4.6.1	XML 文書の評価構造	21
4.6.2	評価基準の定義	21

5	本手法の実装	25
5.1	ツールの概略	25
5.2	ツールの構成	25
5.3	ツールの実行例	26
5.3.1	ユーザインタフェース	26
5.3.2	構文解析部	28
5.3.3	DOM ツリー変換部	29
5.3.4	構造計測部	31
6	評価実験	32
6.1	評価対象	32
6.2	実験方法	32
6.3	実験結果	33
6.4	考察	37
7	まとめ	38
	謝辞	39
	参考文献	40

1 まえがき

近年，インターネットの普及，情報技術の発展に伴い，WWW (World Wide Web，以降 Web 略称) を利用した電子文書の重要性が高まってきた．電子文書は情報の共有や，文書の再利用性，検索性などを向上させることを目的に構造化言語を用いて記述する．ところが全ての電子文書がこれらの性質を満たしているとは限らないため，電子文書の構造を調査し，実際にこれらの目的に即するかを評価する必要がある．

現在電子文書を利用している具体的な例としては，企業グループ内あるいは企業内インターネットがあげられる[13]．企業内インターネットでソフトウェア開発時 e-mail などを用いてほとんどの情報交換を済ませるようにしている．また従来紙ベースで作成してきたさまざまな文書を，HTML(Hypertext markup Language)という言語を用いて Web ページとして作成，閲覧できるようにしている．

しかし，HTML は拡張性が低いという弱点を持っている．そこで，この問題点を解決するために，近年 W3C (World Wide Web Consortium) で提案された XML (eXtensible Markup Language)[3,6,11,15,16,17,18]が登場し，電子文書やデータ構造の記述の際に XML を用いることが多くなってきた．

XML は HTML よりユーザの多様な要求を十分に満たすことができ，SGML より使いやすいという長所により Web 文書だけでなく，電子図書館，EDI (Electronic Data Interchange)，CALS (Commerce At the Light Speed)などを含んだ様々な用途[4]で XML が利用されているため，各々の文書が要求されている品質を持っているかどうか評価しなければならない．よって，XML で記述された文書の品質を評価する際には，用途に応じた評価基準を設定した上で，それに基づいて評価を行なう必要がある．これまで，電子文書は，読者のアンケートをもとにして，主にデザインや内容についての評価は行われているが，構造に対する評価はほとんど行われていないため，再利用性や文書作成の柔軟性などを評価することに着目した評価を行なうことは困難であった[2]．

そこで，本研究では，XML 文書を DTD (Document Type Definition)定義に基づいて評価するための手法について提案する．本手法は，多様な文書データおよび評価基準に対応するため，DOM モデルより評価用のデータ構造記述を導出し，導出された構造に対する評価を行なう．

まず，文書構造の品質を評価するために評価基準を定義し，実際に XML 文書の品質を評価できる計測ツール試作して，読みやすさ，文書作成の柔軟性，メンテナンス性，再利用生に対する評価を行なった．

以降，2 章では構造化文書と XML，3 章では HTML 文書に対する評価構造および評価基準について説明する．4 章では XML 文書の品質評価手法及び 4 つの品質特性，5 章では試

作したツールについて述べる．6章では実際にツールを使用して3種類のXML文書に対して評価を行なう．最後に7章では本論文のまとめと今後の課題について述べる．

2 構造化文書と XML

2.1 文書構造化

システムの巨大化や複雑化に伴い文書作成プロジェクトも大規模化してきた。このため、スケジュール管理、コスト管理、および品質管理が容易にできるような、複数の執筆担当者による分担作業ができる仕組みが必要になってきた[12]。このような文書作成作業に対する要求にこたえ、標準化された文書を効率的に作成するために文書構造化が導入された。

一般的な統合オフィスツールで文書を作成する場合は、情報を巻物のように流し込んでいく(図 1)。ほとんどのオフィス文書はこの形式で作成されている。ところがこのような流し込み方式で作ると文書内の情報を再利用したくても、特定のキーワードに関連する情報や図表だけを抽出することができない。これを行なうためには、あらかじめ文書に階層を持たせた構造化文書(図 1)を作成し、構造を解釈できるデータベースで管理する必要がある。それを可能にするのが SGML や XML で作成された構造化文書である。簡単に構造化文書(SGML/XML)と非構造化文書の違いを説明する。

- 非構造化文書は明確な論理構造を持たない。章タイトルと節タイトルなどは存在するが階層的な論理構造を持たない。
- 非構造化文書は特定のコンピュータ環境とソフトウェア環境に依存する。

表 1 に SGML/XML 文書と非構造化文書の比較を示す。

表 1：SGML/XML 文書と非構造化文書の比較

SGML/XML 文書	非構造化文書
明確な論理構造	不明確な論理構造
特定コンピュータ環境とソフトウェアに依存しない	特定コンピュータ環境とソフトウェアに依存する

非構造化文書との違いの他に、構造化文書の重要な利点としては、次にあげるものがある[8,12]。

- 執筆規則や評価基準を明確に定めることができるので、執筆担当者個人の力量への依存度を軽減し、品質の安定した文書を作成できる。
- 複数の担当で分担作業ができるので、大規模な文書を短期間で作成できる。
- 文書の特定部分の修正や変更を、他の部分に大きく影響を与えずに行えるため、保

守にかかる費用と期間を削減できる。

- 文書の書式を統一できる。

このような構造化文書により情報の検索や再利用が容易になった。

実際に最近では文書をより効率的に作成・管理するために、章、節、項、図表などにパーツ化し、パーツ単位で検索、再利用、履歴管理が行える SGML や XML などの構造化言語が利用されており、これらの構造化言語を用いて文書の構造を記述している文書も多い。

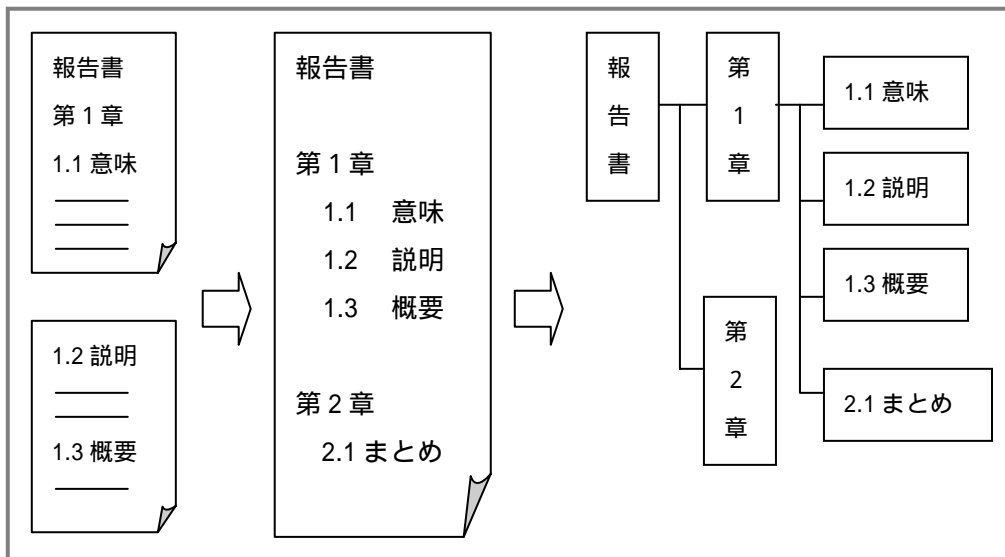
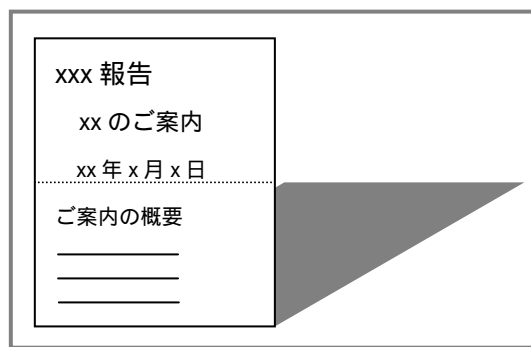


図 1 非構造化文書(上)と構造化文書の概念図(下)

2.2 XML

XML とは eXtensible Markup Language の略であり、日本語で言うと「拡張可能なマークアップ言語」である[18].

Web 上で SGML(Standard Generalized Markup Language) が利用できるように設計されている。

XML はあらかじめ定義されたマークアップ言語ではなく、多言語を定義するための言語という意味でメタ言語である。それによって多くの人がマークアップ言語をデザインすることが可能になる。HTML のようなあらかじめ定義されたマークアップ言語は、情報のある 1 つの決まったクラスの文書によって書き表す方法を定義する。それに対し XML は異なったクラスの文書のより、目的に応じたマークアップ言語を定義することができる。XML によって自分自身のマークアップ言語を定義することが可能なのは、XML が国際標準メタ言語である SGML で書かれているからである。

2.2.1 要素

ほとんどの文書(本や雑誌など)は構成要素(章や記事)に分解することができる。そして、これらの構成要素もさらに別の構成要素(タイトルや段落、図など)に分解することができる。XML ではこうした構成要素を要素(element)と言う。また、要素の中に別の要素を含めることが可能で、階層的に文書を記述することができる。こうした階層構造を木構造(tree structure)と言う。

2.2.2 文書型

文書の種類を指す。例えば実験レポート、ソフトウェア要求仕様書、ソフトウェアデザイン仕様書、電話帳、マニュアルなどは文書型の例となる。つまり、特定文書の種類の応じてマークアップ言語を設計する。

XML はデータの記述にも用いられるので、データの種類もまた文書型と呼ぶことができる。

2.2.3 DTD

DTD は Document Type Definition の略であり、日本語では「文書型定義」と訳される[18]。構造化文書を記述するための設計書とも言うべきものである。SGML や HTML 文書をつくるには必須である(ただし、HTML ではあらかじめ決められたお仕着せの DTD が存在し、ブラウザはそれを解釈できるようになっているので、SGML 文書と異なり文書を交換する際に DTD を意識しないことが多い)。XML では、この DTD を必要としない「Well Formed XML」という形式がある。この場合、XML のタグの位置は「入れ子構造」が守られるだけでよい。DTD では、文書の構造を表現するための「要素」の定義(要素の名前、親子関係、要素の出現順序・出現回数など)、それら要素に持たせたい「属性」の定義などを行なう。DTD の例を図 2 に示す。

2.2.4 Valid XML 文書と Well-formed XML 文書

文書は、処理上の観点から 2 種類に分けられる。「整形式の (Well-formed) XML 文書」と「妥当な (Valid) XML 文書」である[4,6,18]。

「Well-formed XML 文書」とは、開始タグと終了タグの対応が取れており、かつ親要素と子要素がきちんと入れ子になっているなど形式的に「形が整った (Well-formed)」XML 文書を指して言う。つまり、DTD で規定されたタグ名や、構造に従っているかどうかは関係なく、形式的に整っているかどうかという観点からチェックを受けた XML 文書のことである。XML 文書は、必ず Well-formed XML 文書でなくてはならない。

「Valid XML 文書」とは、XML 文書を作成後チェックする際に、整形式かどうかという観点からだけでなく、DTD で定義されているタグ名や構造などに従っているかどうかという点まで含めてチェックした XML 文書を指して言う。表 2 に「Well-formed XML 文書」と「Valid XML 文書」の関係を示す。

表 2 : Well-formed XML 文書・Valid XML 文書の関係

XML 文書のタイプ	XML 宣言	DTD	XML インスタンス
Well-formed XML 文書			
Valid XML 文書			

：あってもなくてもよい

：必須

2.2.5 DOM

DOM (Document Object Model)[7,18,19,20]は、XML パーサにアクセスするための API の 1 つ。W3C が公式に公開した唯一の API である。DOM は、XML 文書を「DOM ツリー」と呼ばれるツリー構造として扱う。そのため、XML パーサが XML 文書全体を読み込んだ後でなければ、文書内のデータにアクセスすることができない。また、DOM ツリーは通常、メモリ上に展開されるため、大規模な XML 文書进行处理する場合には、その分メモリ容量も要求される。その代わりに、DOM を利用すれば、XML 文書内のデータの順番に関係なくアクセスできる。

2.2.6 スタイル指定

XML はタグがあらかじめ定義されておらず、設計者がデザインしなければならない。故に、ユーザが文書を閲覧するためには、XML 文書をそのまま見るか特殊なプログラムを用いて閲覧する必要がある。このような複雑な状況を解決するために、XSL (eXtensible Style

Sheet)[4,18]と呼ばれる XML スタイル定義言語が提案されている。これによりユーザは、専用のソフトウェアをしなくてもHTMLのような概観の文書を閲覧することが可能になった。

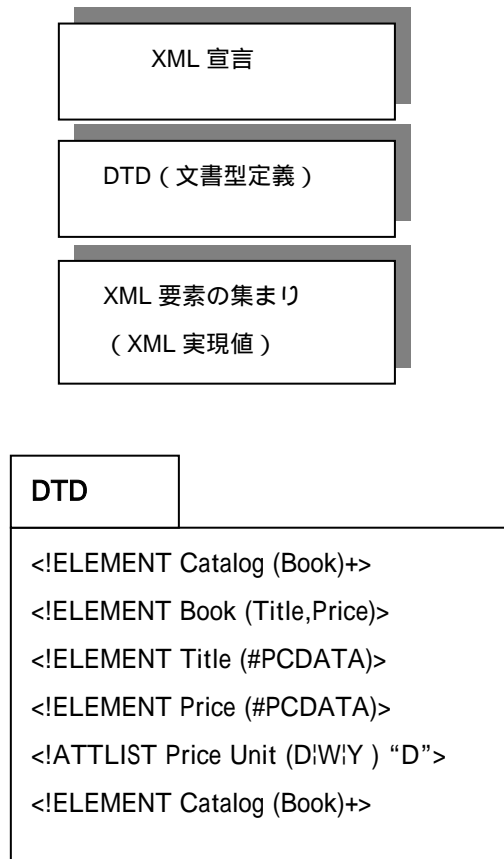


図2 XML 文書の構造(上)と DTD の例(下)

2.3 XML , SGML , HTML の比較

XML や SGML , HTML は全く同じというわけではない。SGML は XML や HTML のいわゆる「母語」であり、人間の様々な活動分野における何千という文書型を記述するのに用いられる。

HTML はそうした多数の文書型のたった一つに過ぎないが、WWW では最も広く用いられているものである。HTML は単一の、固定されたマークアップを定義し、それによって単純な共通クラスを記述する。スタイルには、見出し、段落、箇条書き、図などが備わっており、さらにハイパーテキストやマルチメディアのための機能が付け加わっている。

XML は SGML のサブセットである。したがって、独自の文書型を簡単に定義することが

でき、プログラマがそのような文書の処理プログラムを容易に書くことができる。つまり、複雑過ぎる部分やあまり利用されない部分を削除することによって、容易にアプリケーションが書け、理解しやすく、インターネットを通じて配布や相互運用に適するようにしたものである。

3 構造化文書の評価

情報の共有や、文書の再利用性、検索性などの向上を目的として電子文書は一般的によく用いられる。しかし、全ての電子文書がこれらの性質を満たしているとは限らないため、電子文書の構造を調査し、実際にこれらの目的に即するかを評価する必要がある。

現在、電子文書は、読者のアンケートをもとにして、主にデザインや内容についての評価は行われているが、構造に対する評価はほとんど行われていない。そこで、我々の研究グループでは、HTML で記述された電子マニュアルを対象として、文書構造の品質に対する評価を行ってきた[2]。本章では、本研究と関連しているこれらの評価手法について簡単に紹介する。

3.1 HTML 文書の評価

私たちの研究グループの過去の研究では、HTML で記述された電子マニュアルを対象として、文書構造の品質に対する定量的な評価を行ってきた[2]。この研究では、HTML 文書の構造を把握するためにモジュールや、情報ブロック、階層、参照の4つの構成要素の概念を導入してHTML 文書へ適用している。ここでは、以下4つの概念と評価基準を述べる。

- モジュール
モジュールは、文書中の見出しによって分割可能な一連の情報とする。HTML 文書の場合見出しタグ (<H1>....., <H6>) が存在するため、文書中で見出しタグによって分割可能な情報をモジュールと定義する(図 3)。また、各モジュールは情報ブロックと呼ばれるいくつかの構成要素から成り立っている。
- 情報ブロック
各モジュール内に含まれる段落を情報ブロックと定義とする。HTML の場合<P>タグが用意されているため、HTML 文書における段落タグを情報ブロックとみなす。
- 階層
階層は、見出しタグ (<Hn>;n は 1 から 6 の整数) の大小と定義する(図 3)。
- 参照
HTML における参照をページリンクとして定義する。

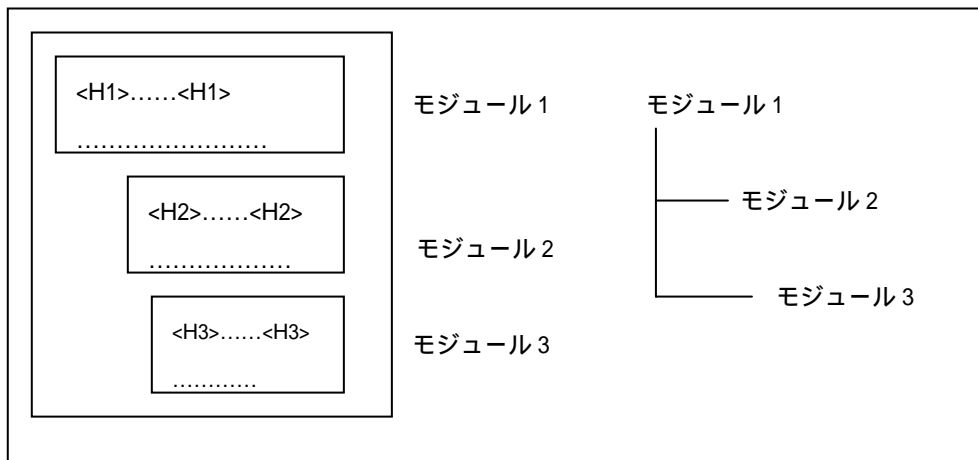


図3 モジュールと階層

この4つの構成要素をHTMLで記述されたマニュアルの構造に適用させ、HTMLマニュアルの構造に対する品質を評価するための8つの評価基準を提示した[2]。ここで、評価基準の一部を説明する。

- モジュールのサイズは1ウィンドウ程度
 モジュールの具体的なサイズとしては、紙媒体で記述された文書であれば見開き2ページ程度、画面上で閲覧する電子化された文書あれば1ウィンドウ分程度がよい。これは、ユーザがページをめくらずに情報を一覧できる情報量として、読みやすさが向上すると定義している。
- モジュールは複数の情報ブロックから構成
 内容的な理由から一覧性を保つサイズに切り分けられないモジュールに関しては、モジュール内を適度な大きさの情報ブロックに分けることによって、ユーザに対する一覧性を提供すると定義している。
- 各モジュールの子供は適切な数にする
 必要以上に長い階層は、ユーザが階層関係を理解を妨げる上に、ユーザの文書を読む意欲を減少させるため、文書を構成するモジュールが持つ子は適度な数に抑える必要があると定義している。
- 1モジュールにつき1つのページ内参照リンク
 一つのファイルに複数のモジュールが含まれるなどファイルに記述されている内容をウィンドウで一覧できない場合、ページ内参照リンクを利用することで、モジュール間の移動がやりやすくなると定義している。

3.2 XML 文書への適用

本研究は，XML で記述された文書構造の品質に対する定量的な評価を目的としている．前述のように，HTML において文書構造を決定するモジュール，情報ブロック，階層，参照の4つの要素をXML文書に対しそのまま適用することは不可能である．その理由は，XMLでは，HTMLのような見出しタグ(<H>)や，段落タグ(<P>)などが存在しないからである．したがって，次の章ではXML文書の構造を利用した品質評価手法を提案する．

4 XML 文書の品質評価手法

ここでは、XML で記述された様々な文書構造の品質に対する定量的な評価を行なうために提案する手法について説明する。本手法では、多様な文書および評価基準に対応するため、DOM モデルより評価用のデータ構造記述を導出し、導出された構造に対する評価を行なう。以下では、まず前提条件、及び XML 文書評価手法の手順について述べ、次に DOM 変換ルール、及び、評価用 DOM ツリーについて説明する。

4.1 前提条件

2章で述べたが、XML 文書は、処理上の観点から 2 種類に分けられる。「Well-formed XML 文書」と「Valid XML 文書」である。

「Well-formed XML 文書」は、DTD が定義されていないので、任意の要素を追加することができる。しかし、「Well-formed XML 文書」はいくつか重要な制限がある。例えば、すべての属性値は必ず指定されなければならない、エンティティを参照することができないなどである。また、XML 文書を自動生成し、交換を行なうような現実的なアプリケーションでは、DTD が定義されている「Valid XML 文書」を利用するのが一般的である。本研究で評価対象とする XML 文書は、「Valid XML 文書」とする。

4.2 XML 文書の品質評価手法の手順

XML 文書を、DTD 定義に基づいて以下の 3 つの手順により評価を行なう。

Step1 DOM ツリーの構築

XML 文書と対応する DTD 定義を読み込んで DTD に従って文書の正しさをチェックし、DOM ツリーを生成する(図 4)。

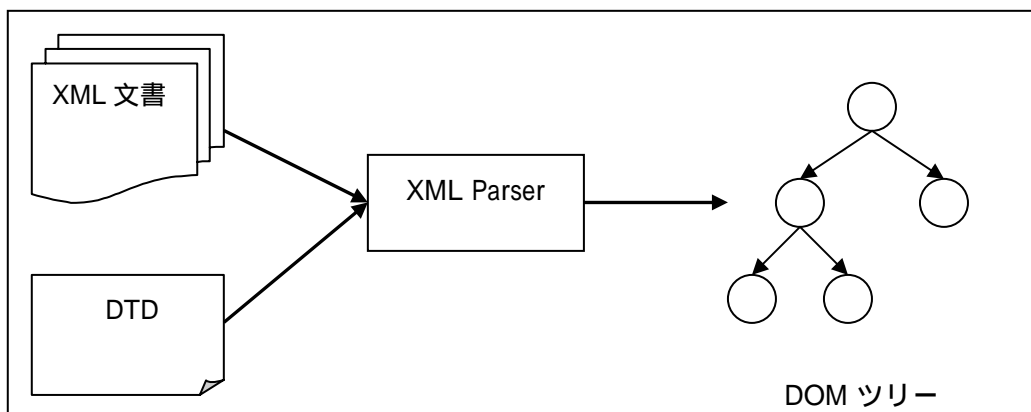


図 4 DOM ツリー構築

Step2 評価用 DOM ツリー作成

Step1 で作成した DOM ツリーを，評価基準と連携した DOM 変換ルールを用いて評価用 DOM ツリーに変換する(図 5)．DOM 変換ルール，評価用 DOM ツリーの詳細は後述する．

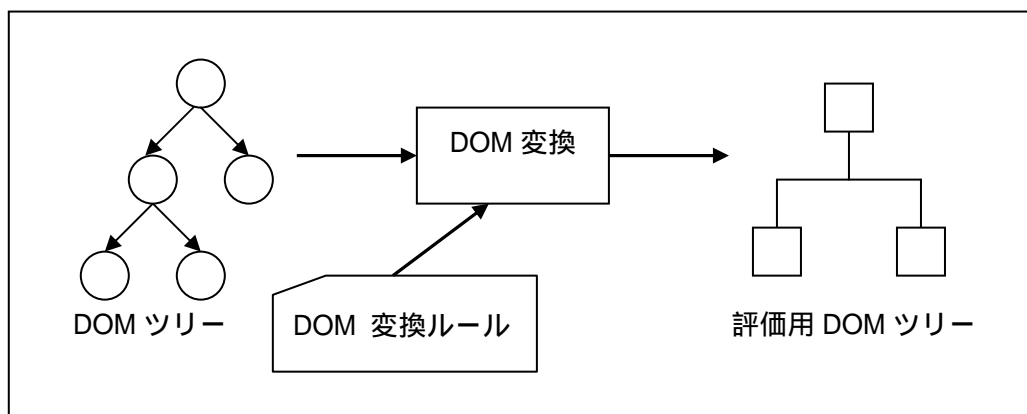


図 5 評価用 DOM ツリーに変換

Step3 評価用 DOM ツリーの構造評価

あらかじめ定めた評価目的に応じた基準に基づき，評価用 DOM ツリーの評価を行なう(図 6)．

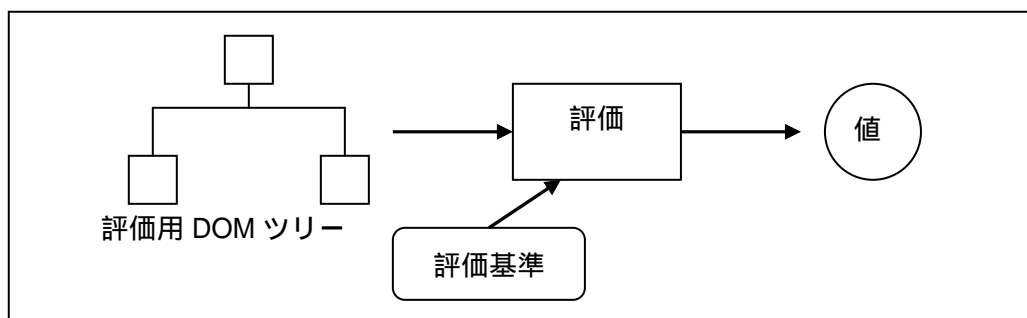


図 6 評価用 DOM ツリーの構造評価

4.3 評価用 DOM ツリー

XML では利用者が自由に DTD を定義でき，文書中の文字列に意味付けができる．それによって，様々な DOM ツリーが生成される．また，評価する目的によって，着目する DOM ツリーの構成要素が異なるので，あらゆる XML 文書をそのままの形で一律に評価することが難しい．そこで，評価したい目的に応じて DOM 変換ルールを用いて，共通の構造パターンを持った評価用 DOM ツリーに変換する．評価用 DOM ツリーは，品質を評価しやすくす

のために導入されたものであり、文書構造を元にしてモデルに変換したツリーである。

4.4 DOM 変換ルール

DOM 変換ルールは、評価対象となる XML 文書を評価用 DOM ツリーに変換するためのルールである。

DOM 変換ルールはモジュールと構造リンク 2 つの概念を用いて記述する。ここでは、以下 2 つの概念について述べる。

4.4.1 モジュール

モジュールは、評価を行なう際の基本単位で、DOM ツリー上で 1 つ以上の要素の集合として、ユーザに対して一度に提供することが可能な情報量を表す。各モジュール内は実際に情報を伝達する要素を含む。

4.4.2 構造リンク

構造リンクは、モジュール間の関係を表すものとして、階層と参照が存在する。階層は、モジュール単位に分割された XML 文書を階層化することによって、情報のまとまりや上下関係を明確に表現することが可能となる。参照は、現在読んでいる文書の中で、その文書内の別の場所や他の文書を指し示すためのものである。

4.4.3 DOM 変換ルールの解析

親子関係において、子供となる内容がテキストを直接含まず、要素のみで成り立っている場合、内容モデルにテキストが出るまで子供の要素を調べていく。

子供となる内容がテキストを直接含んでいる場合、その要素とテキストをモジュールとみなす。

モジュールの親となる要素がそのモジュール以外の要素をもたない場合は、親となる要素もモジュールに含める。

もし、モジュールの親となる要素が別な要素をもつ場合は、兄弟関係となる要素のモジュールの合計サイズが一画面以下である場合には、兄弟関係となる要素と親の要素をモジュールとして一つにまとめる。

もし、兄弟関係となる要素のモジュールの合計サイズが一画面以上になる場合は、兄弟関係にあるモジュールをそれぞれ別のモジュールとする。また、親となる要素は子供と区別し、一つのモジュールとして扱う。

変換ルールの解析に対する定理を図 7 に示す。

M : モジュール	E : 要素のみで構成
S : モジュールサイズ	Y : 混在 (要素, テキスト)
P : 親要素	N : 兄弟モジュールの合計
X : 内容モデル	C : 子要素
T : テキスト	R : ルート要素
O : モジュール以外の要素をもたない場合	

1. Sを入力する .
2. PのXを調べる .
3. if X = E then 6へ
4. if X = T then その要素はMとみなす
MのPを調べる .
If P = O then PをMみなす . (図8(左))
else P = C then 7へ
6. XにTが出るまでCを調べていく . 4へ
7. if N = S or N < S then Pを一つのMとみなす . (図8(右))
else 兄弟要素を別のMとする

図7 変換ルールの解析

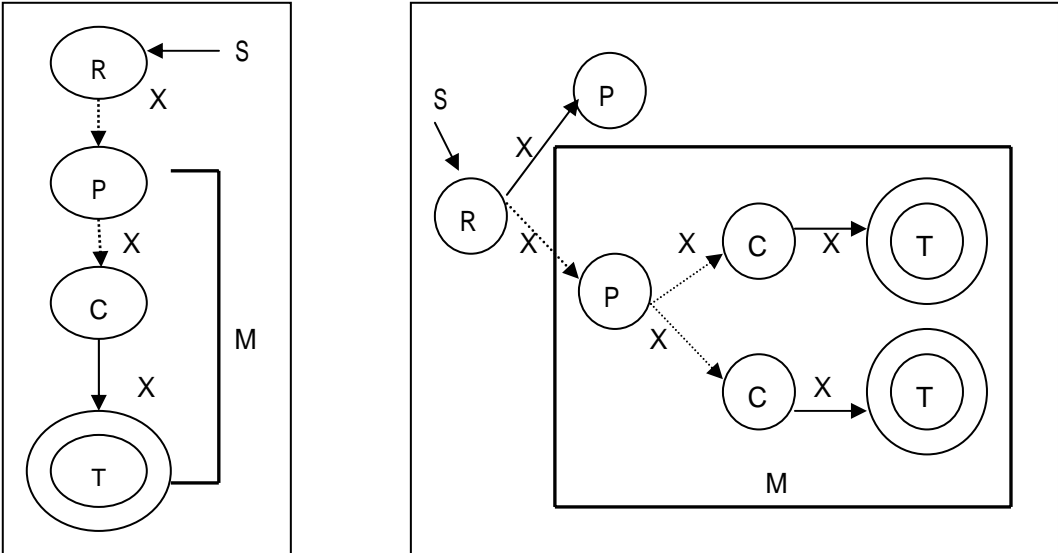


図8 モジュールの例

4.5 品質特性

本研究は、XML 文書の構造を利用した、読みやすさ、文書作成の柔軟性、メンテナンス性、再利用性の4つの品質特性に対して評価を行なう。

4.5.1 読みやすさ

紙文書は、一般に本というモデルがあり、ユーザのもつモデルイメージが確立されている。また、目の前に実体がありユーザにとって読みやすい。しかしながら、XML で記述された電子文書の実体は一見では判断しづらく、ユーザにとって全体がわかりづらい。さらに複雑な構造や、数多くのリンクが張ってある文書など、わかりにくい構成の文書であればなおさらである。また、XML で記述された電子文書は紙文書に比べ文書内のどの位置を読んでいるのかをユーザが把握しづらい。特に階層が深い場合は、ユーザが現在読んでいる位置を直感的に認識することを困難にするため、読みやすさに対する特性を低下させる要因の一つである。

4.5.2 文書作成の柔軟性

取扱説明書やマニュアルなどのページ数は実際には相当な数になるため、複数のファイルに分割して、複数の担当で分担作業ができるようにすることにより大規模な文書を短時間で作成できるので、文書作成の柔軟性を向上させることができる。

4.5.3 再利用性

XML 文書における再利用性は2つに定義することができる。コンテンツ再利用と構造再利用である。

- コンテンツ再利用
コンテンツ再利用は、文書のコンテンツ情報の再利用を意味する。
- 構造再利用
構造再利用は、文書の一般的な構造 (DTD) の再利用を言う。構造再利用性を高めるためには、よい DTD を設計することである。よい DTD とは、修正することなしに長期間使用できる DTD である。粗悪な設計をすると DTD、およびアプリケーションに頻繁な修正を要求することになる。よい DTD を設計する一つのアプローチは、すでに広く使われている DTD を利用することである。

4.5.4 メインテナンス性

ソフトウェア工学におけるメインテナンスには、テスト容易性や、移植性、変更容易性などの副特性が存在するが、本研究でメインテナンス性はテスト容易性と変更容易性とする。XML 文書においてメインテナンス性を向上させるための一つの手法は、パラメータエンティティをうまく使うことである。例えば、次のような 診療記録表のための DTD(hospital-orig.dtd)を考えてみる。

```
<!ELEMENT  診療記録表  (患者, 医者, 病院, 診療日時)>
<!ATTLIST  診療記録表  status  CDATA  #REQUIRED>
<!ELEMENT  患者        (名前, 住所)>
<!ELEMENT  医者        (名前, 住所)>
<!ELEMENT  名前        (#PCDATA)>
<!ELEMENT  住所        (#PCDATA)>
<!ELEMENT  病院        (#PCDATA)>
<!ELEMENT  診療日時    (#PCDATA)>
```

患者と医者は、「名前」と「住所」という共通の内容モデルを持っている。したがって、それぞれ異なるものとして定義しないで、エンティティとして共通のものとして定義すると、次のような DTD として定義することができる。

```
<!ENTITY   Partner      " name, address"  >
<!ELEMENT  診療記録表  (患者, 医者, 病院, 診療日時)>
<!ATTLIST  診療記録表  status  CDATA  #REQUIRED>
<!ELEMENT  患者        (%Partner ; )>
<!ELEMENT  医者        (%Partner ; )>
<!ELEMENT  名前        (#PCDATA)>
<!ELEMENT  住所        (#PCDATA)>
<!ELEMENT  病院        (#PCDATA)>
<!ELEMENT  診療日時    (#PCDATA)>
```

もし患者や医者の内容モデルに「年齢」という要素を追加したい場合、それぞれ要素を定

義しないで、Partner エンティティに「年齢」要素を挿入するとよい。このようにエンティティをうまく使うとメンテナンス性を向上させることができると考えられる。

4.6 評価基準

読みやすさ、再利用性、文書作成の柔軟性、メンテナンス性の4つの品質特性を評価するために用いられる基準をオブジェクト指向ソフトウェアの複雑度を計測するメトリクス及び参考文献を参考にして定義した。

4.6.1 XML 文書の評価構造

評価基準に基づいているモジュールと構造リンクの2つの構成要素についてより明確に定義をした上で、評価基準を定義する。

- モジュール

モジュールは、評価を行なう際の基本単位で、DOM ツリー上で1つ以上の要素の集合として、ユーザに対して一度に提供することが可能な情報量を表す。ここで、一度に提供することが可能な情報量とは、1画面に収める程度の文量を言う。そして、各モジュール内は実際に情報を伝達する要素を含んでいる。

- 構造リンク

構造リンクには、階層と参照がある。

階層は、モジュール単位に分割されたXML文書を階層化することによって、情報のまとまりや上下関係を明確に表現することである

参照は、現在読んでいる文書の中で、その文書内の別の場所や他の文書を指し示すためのものとして、XML文書における参照をエンティティ、リンクとして定義する。

4.6.2 評価基準の定義

読みやすさ、文書作成の柔軟性、メンテナンス性、再利用のしやすさを評価するための評価基準を次のように定義した。

読みやすさ

読みやすさに対する品質特性を評価するために3つの評価基準を定義する。

1. モジュールのサイズ

1画面に収める程度の文量のモジュールサイズが読みやすさを最も向上させることができる。モジュールのサイズはスクロール操作に影響を与える。スクロール操作は、ページ内における視点の位置を把握し、どの部分を読んでいるかが容易に分かるとい

う利点がある一方、視点から離れた部分の内容を把握し難くするという欠点もある。従って、モジュールのサイズを1画面に収まる文量とすると、ユーザがページをスクロールせずに情報を一覧できる情報量となり、ユーザが情報を把握しやすくなるため、読みやすいといえる。

2. モジュール内のリンク数

モジュール内に数多くのリンクが張っていると、ユーザがページの行き来をよく行なうことになるため、ユーザに階層関係の理解を妨げる。

3. 階層

XML 文書が階層化され情報を把握しやすいように整理されていた場合でも、階層が深いと、ユーザが現在読んでいる位置を直感的に認識することを困難にするため、ユーザの文書を読む意欲を減少させる。

文書作成の柔軟性

文書作成の柔軟性という品質特性を評価するために2つの評価基準を定義する。2つの基準を以下に示す。

1. エンティティ

文書の量が多い場合、複数のファイルに分割して、複数の担当で分担作業ができるようにすると大規模な文書を短期間で作成できる。例えば、マニュアルの各章を別のファイルに格納し、それぞれエンティティ名をつけて参照すると、短期間に作成が可能である。

2. 1ファイルにつき1モジュール

この基準は、過去 HTML 文書評価において提案した評価基準と同じである。これは文書作成の分担や、再利用性の向上などを考慮すると、1ファイル上に1モジュールがあることがよい。

再利用性

再利用性に対する品質特性を評価するために2つの評価基準を定義する。

1. DTD

既存のデータ・フォーマット(DTD)を再利用すると、理解しやすく、柔軟性が高くなる。その結果、開発や保守のコストを削減できる。例えば、新たに文書を作成する場合、同類のマニュアルがすでに存在すれば、DTD の変更は少なくなる。したがって、DTD を再利用することによって開発コストを削減できる。

2. エンティティ

最も初期のソフトウェア再利用のアイデアの一つは、サブルーチンの考え方である。同じ処理を繰り返しコーディングするよりも、1つにまとめてサブルーチンとしてコールできるようにすれば、それだけでプログラムの生産性が飛躍的に高まる。現在では多くの再利用可能なソフトウェアが、サブルーチンライブラリの形で提供されている。一方XMLではこのような機能をエンティティが提供している。XMLにおけるエンティティは文書中の中で同じ内容の文字列が繰り返し現れる場合、その文字列データを登録しておき、置換文字列として参照できる。

メンテナンス性

4.5節で述べたようにメンテナンス性をテスト容易性と変更容易性とする。そして、提案する評価基準はこれらに基づいている。メンテナンスに関する評価基準はオブジェクト指向ソフトウェアの複雑度を計測するメトリクスを参考にしている。まず、2つの複雑度を計測するメトリクス[14]について簡単に紹介し、評価基準を示す。

- DIT (Depth of inheritance tree)
あるクラスのス-パクラスの数計測する。DITが高いほど、複雑でメンテナンスのコストがかかる。
- NOC (Number of Children)
あるクラスのサブクラスの数計測する。NOCが高いほど、サブクラスへの影響力が強いので、メンテナンスのコストがかかる。

この2つのメトリクスを着目して3つの評価基準を定義する。

1. 階層

DITの定義に着目する。階層が深いと複雑度が高い。したがって、複雑度が高いと、変更しにくいことが一般的に言える。さらに、XMLでは入れ子が存在するので、階層が深い場合テストや変更が非常に難しい。

2. モジュール内のリンク数

NOCの定義に基づいて、モジュール内に存在するリンクを子とみなす。そうするとモジュール内に数多くのリンクが張ってある文書なら、NOC定義により複雑度が高い。したがって、メンテナンスのコストがかかる。また、ユーザがモジュールを変更する際にそのモジュール内のリンクまでテストする必要があるので、コストがかかる。

3. エンティティ

複雑で規模の大きい DTD になると、要素型宣言の内容モデルなどに頻繁に同じ宣言パターンが現れることがある。このようなときに、そのパターンをエンティティとして定義しておけば、DTD 記述全体がすっきりする。例えば、複数の要素がある共通の内容モデルを持っている場合、それぞれ異なるものとして定義しないで、エンティティを用いると、同じパターンを何度も書く手間が省ける。また内容モデルに追加変更があったときでも、エンティティの宣言部を変更するだけで済む。

DTD の拡張、変更などを行なうときエンティティを用いると容易である。

5 本手法の実装

5.1 ツールの概略

先に述べた，品質評価手法に基づいた XML 文書の構造を利用した品質評価システムの試作を行った．開発環境は以下の通りである．

- CPU : Pentium3 1GHz
- RAM : 246MB
- OS : Windows98
- 言語 : JAVA

コード量は 4000 行程度である．

5.2 ツールの構成

本システムの構成を図 9 に示す．この図からわかる通り，本システムは構文解析部，DOM ツリー変換部，構造計測部の 3 つから構成される．

- 構文解析部

構文解析部では，入力として XML 文書のファイルを受け取る．そして，XML 文書を読み込んで構文解析する過程において，XML パーサは文書型定義に従って DTD を処理し，XML 文書に対する DTD 名前，要素リストおよび属性情報などの様々な情報(表 3)と DOM ツリーを出力する．

表 3 構文解析部で出力される情報

- encoding	- 要素総数
- DTD 名前	- 要素リスト
- Root 名前	- 属性情報
- 要素関係 (階層)	

- DOM ツリー変換部

DOM ツリー変換部は，入力として構文解析部で生成された DOM ツリーとモジュールサイズ，行数を受け取り，DOM 変換ルールを用いてルートモジュールやモジュール，要素などを選択して評価用 DOM ツリーに変換する．

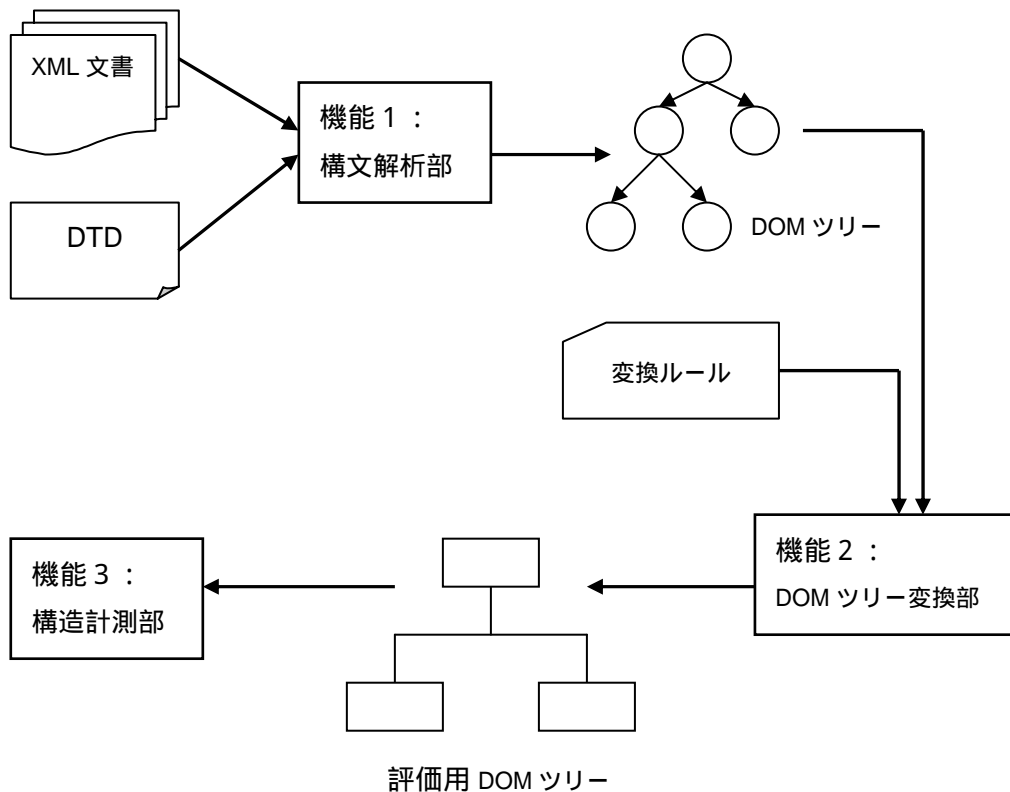


図 9 XML 文書評価システムの構成

- 構造計測部

構造計測部は前述の DOM ツリー変換部の出力を受け取り，XML 文書の各種計測項目の結果を出力する．すなわちリンクの数やモジュールごとの情報などが含まれる．

5.3 ツールの実行例

ここでは，試作した XML 文書評価ツールについて，構文解析部，DOM ツリー変換部，構造計測部の機能別にツールの実行過程の画面を用いて説明する．

5.3.1 ユーザインタフェース

本プログラムは，XML 文書構文解析部と XML 文書評価部から構成されている．

ユーザは XML ツリーから XML 文書構文解析部を選択する(図 10)．そして，「Run」ボタンを押すと「Execution」ダイアログ(図 11)が表示される．

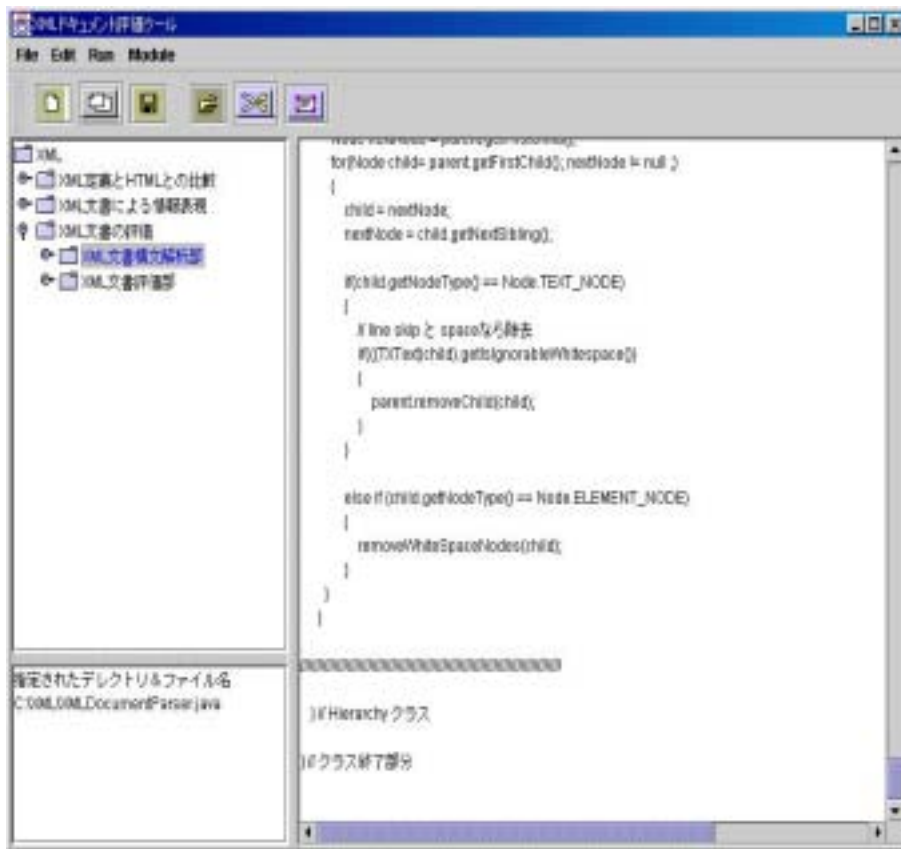


図 10 XML 文書構文解析プログラムの選択画面

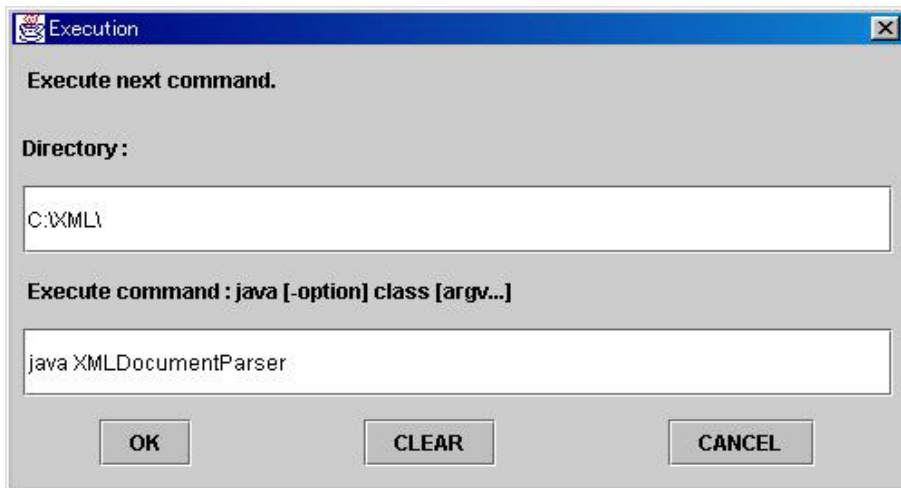


図 11 Execution ダイアログ画面

5.3.2 構文解析部

構文解析部は、「Execution」ダイアログの「Execute Command」に評価対象となる XML 文書を入力する(図 12)。その後「OK」ボタンを押すと構文解析の実行が行われ、XML 文書の情報が出力される(図 13)。また、「階層情報」ボタンを押すと XML 文書を「DOM ツリー」と呼ばれるツリー構造(図 14)として出力する。

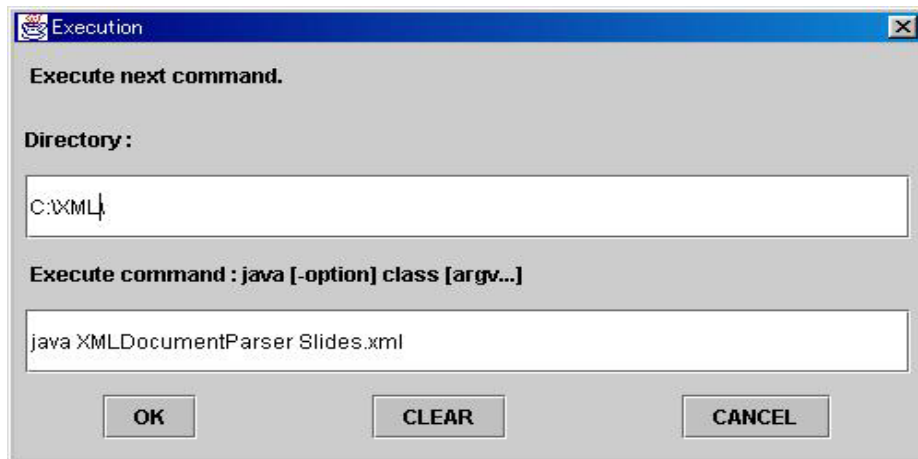


図 12 構文解析の入力画面

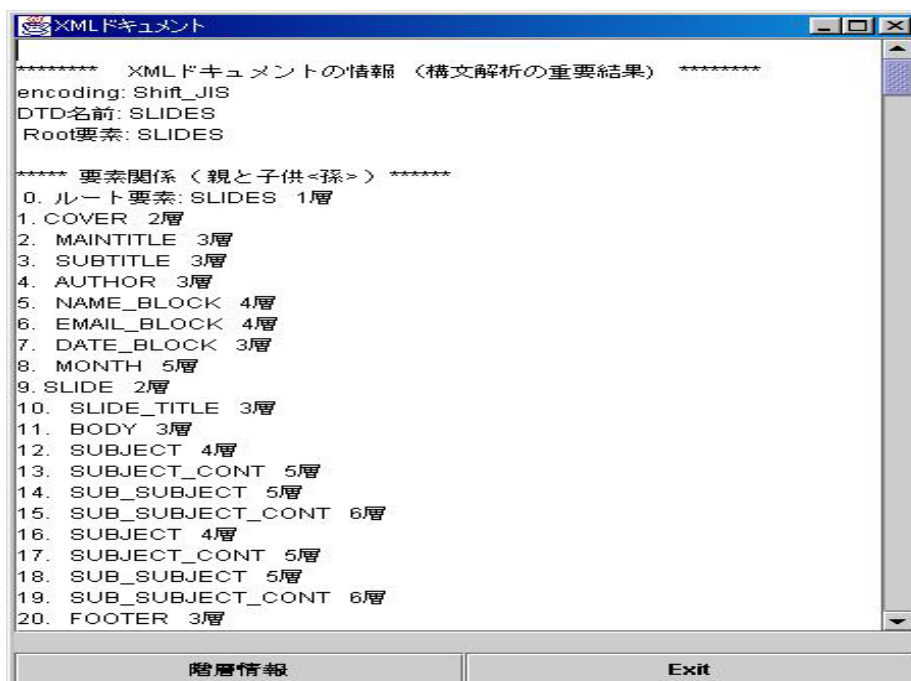


図 13 構文解析の結果画面

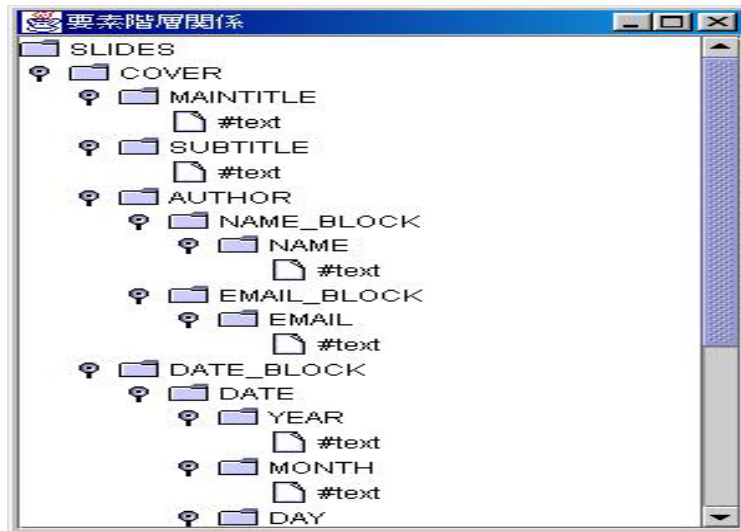
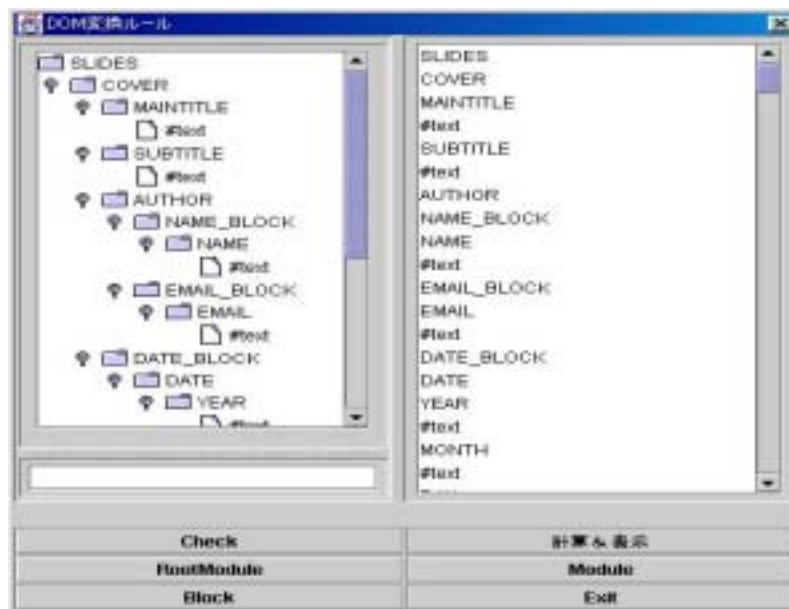


図 14 DOM ツリー画面

5.3.3 DOM ツリー変換部

DOM ツリー変換部では、まず DOM ツリーを入力し、その後出力ファイル名、モジュールサイズ、行数を順々に入力すると DOM ツリー変換部の画面が出力される。

DOM ツリー変換部の出力例を以下に示す。



ここで、評価用 DOM ツリーに変換させるために、ユーザは「Check」、「RootModule」、「Module」、「Block」ボタンの操作を行なう。例えば、SLIDE ルート要素を指定し、

「RootModule」ボタン押すと図 16 のような評価用 DOM ツリーのルート要素が作られる。以下では、5 つのボタンについて簡潔に説明する。

- Check
ツリーやリストの項目を選択し、「Check」ボタンを押すと、選択された要素名とその要素のサイズ、行数が表示される(図 15)。
- RootModule
DOM ツリー上でのルート要素を指す。「RootModule」には、XML 文書全体の文字列サイズを持っている。
- Module
ユーザに対して一度に提供することが可能な情報量を持っている。ここでモジュールのサイズは、おおよそ 1 画面程度であることが望ましい。
- Block
実際に情報を表す要素を持っている。
- Exit
プログラムを終了する。

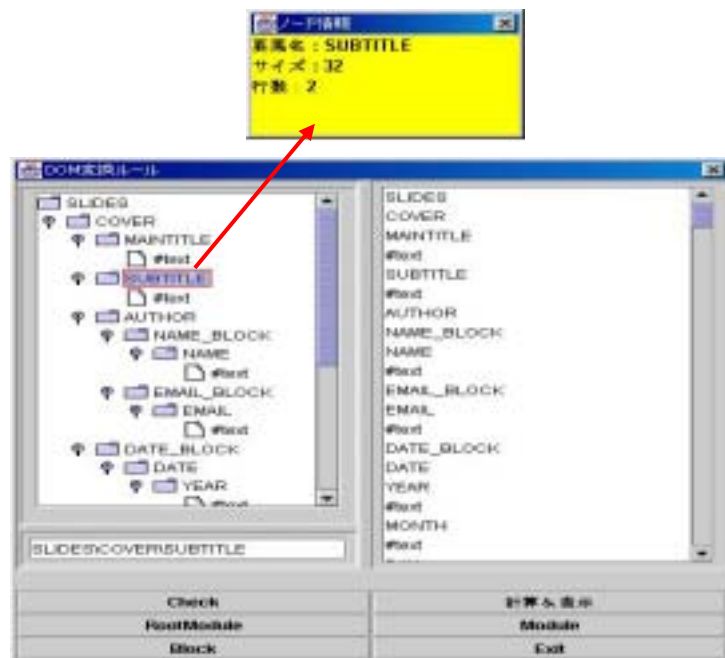


図 15 SUBTITLE 要素に対する結果画面

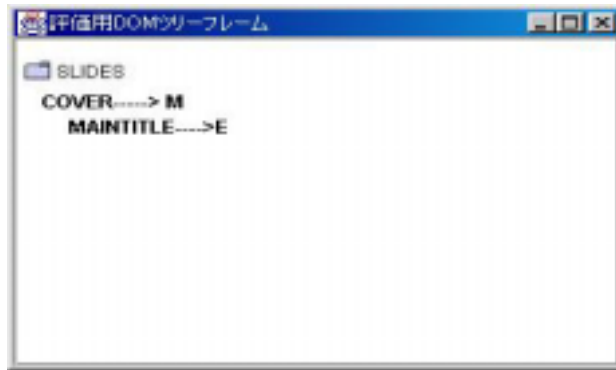


図 16 評価用 DOM ツリー画面

5.3.4 構造計測部

構造計測部は前述の DOM ツリー変換部から生成された評価用 DOM ツリーの各評価項目を計測する．各種計測項目の結果は指定されたファイルに出力される．

計測部の出力例を以下に示す．

```

C:\XMLResult\XML.dat - 九九
ファイル 編集 検索 ウンドゥ 印刷 その他 10:10
1 入力したモジュールのサイズは : 2401
2 このドキュメントの全体サイズは : 16491
3 入力した階層 : 31
4 XMLドキュメント名 : slides.xml
5 The number of file(hyperlink包含) = 14
6 The number of link = 01
7 子モジュール数 : 101
8 Entity数 : 01
9 このドキュメントのDTD(Document Type Definition)は : SLIDES1
10 DTDが存在します(外部) ↓
11 ↓
12 COVER----> Module ↓
13 MAINTITLE1
14 #text1
15 SUBTITLE1
16 #text1
17 AUTHOR1
18 NAME_BLOCK1
19 NAME1
20 #text1
21 Module size : 981
22 Module block = 191
23 Module link count = 01
24 : ↓
25 : ↓
26 (省略) ↓
27 Document height = 21
28 このドキュメントの全体文字数 : 16491
29 Total Module Block: 3141
30 総モジュールの個数: 141
31 実際モジュール数(ルートモジュール除外) : 131
32 ↓
九九九九 | 下準備 | 単語も対訳の心算の基は | 別の付け替の心算の基は |

```

6 評価実験

本論文で述べた XML 文書の品質評価手法により，3.2 節で述べた問題点の解決が可能である．このことを確認するため，試作したツールを利用して実験を行った．

6.1 評価対象

評価対象となるデータとして，論文，スライド，マニュアル 3 種類，18 個の XML 文書を用意した．論文は，ICSE と ICSR に投稿された 2 つの論文と，全国大会に投稿された 1 つの論文である．スライドについては，学生が作成した 4 つのスライドを対象とした．また，マニュアルについては，学生が作成した 5 つのマニュアルと WWW 上で入手した 6 つ文書を対象とした．

<評価用データ>

- ・論文：ICSE(International Conference on Software Engineering. 2001)
ICSR(International Conference on Software Reuse. 1998)
全国大会(電子情報通信論文集)
 - ・スライド：4 つ(学生が作成)
 - ・マニュアル：5 つ(学生が作成)
- Web：<http://www.ogis-ri.co.jp/otc/hiroba/technical/xml/article2.xml>
<http://xml.ibsinc.co.jp/honyaku/>
http://xml.ibsinc.co.jp/scripts/xlnisapi.dll/honyaku/data/idoc_demo_ja.xml
<http://myhome.netsgo.com/jiyuns/docs>

6.2 実験方法

提案手法を用いて XML 文書の品質評価を行なう際，ユーザが指定する必要があるものとして，出力ファイル名，文書形式，モジュールサイズ，行数，階層がある．これらの指定は以下の方針で行なった．

- 出力ファイル名の指定
評価対象となる XML 文書の各種計測項目の結果値を保存するためのファイル名を指定する．
- 文書形式
スライドか，論文か，一般文書か，など形式を指定する．

- モジュールサイズ
1画面に収める程度の文量のサイズを指定する。
- 行数
1画面に割り当てる行数を指定する。
- 階層
文書が構成する階層を指定する。

6.3 実験結果

論文，スライド，マニュアルを実際に評価を行なった．ここで，論文とスライド，マニュアル実験結果について説明する．

<実験 1>

ICSE と ICSR に投稿された論文と全国大会に投稿された論文に対する評価を行なった．ここで，ICSE に投稿された論文(図 17)に対する実験例を説明する．まず，ユーザの入力として，出力ファイル名，文書形式，モジュールサイズ，行数，階層がある．

本実験では，モジュールサイズは 1画面程度であるという基準について 2100 を入力し，階層を 3，行数を 28 とした．そして，それぞれのモジュールに対し，計測ツールを利用して各計測項目値を算出した．その結果，実際に階層やリンク数などの評価項目について異常な特徴は得られなかった．しかし，モジュールサイズにおいて，指定したモジュールサイズより大きなモジュールが 6 つ検出された．これらは，1画面で一覧できないサイズであるため，1画面に収める程度の文量に分けて，再び計測した．計測した結果，内容のまとまりを確認しやすくなり，モジュール単位で読むことが可能となった(表 4)．



図 17 XML 形式の論文

表 4 ICSE 論文の計測結果

入力したモジュールサイズ	2100	Reserchs	モジュール
この文書の全体サイズ	29401	モジュールサイズ	4622
入力した階層	3	要素数	16
要素最大階層	4	モジュールリンク	0
XML 文書名	Process.xml	エンティティ	0
The number of file	1	モジュールサイズ	NO
The number of link	0	階層	OK
予想モジュール数	9	Header	モジュール
Entity	0	モジュールサイズ	3863
DTD 名	Process	要素数	11
DTD 有無	有	モジュールリンク	0
Process	ルートモジュール	エンティティ	0
Paper	モジュール	モジュールサイズ	NO
モジュールサイズ	652	階層	OK
要素数	19	Header	モジュール
モジュールリンク	0	モジュールサイズ	1456
エンティティ	0	要素数	3
モジュールサイズ	OK	モジュールリンク	0
階層	OK	エンティティ	0
Introduction	モジュール	モジュールサイズ	OK
モジュールサイズ	3273	階層	OK
要素数	10	Direction	モジュール
モジュールリンク	0	モジュールサイズ	1557
エンティティ	0	要素数	14
モジュールサイズ	NO	モジュールリンク	0
階層	OK	エンティティ	0
Overview	モジュール	モジュールサイズ	OK
モジュールサイズ	4564	階層	OK
要素数	28	Conclusion	モジュール
モジュールリンク	0	モジュールサイズ	1289
エンティティ	0	要素数	9
モジュールサイズ	NO	エンティティ	0
階層	OK	Reference	モジュール
Header0	モジュール	モジュールサイズ	2958
モジュールサイズ	2928	要素数	41
要素数	27	エンティティ	0
モジュールリンク	0	Header0	モジュール
エンティティ	0	モジュールサイズ	2060
モジュールサイズ	NO	要素数	13
階層	OK	エンティティ	0
		モジュールリンク	0

結果	
全体階層	3
Total module	12
Total element	191
Total Entity	0
全体サイズ	29401
指定値より多過ぎるモジュール	6
指定値より深すぎるモジュール	0
実際モジュール(ルート除外)	11

<実験 2>

学生に作成してもらったスライドに対して実験を行なった．ここで，スライドに対する実験例を説明する．

本実験では，モジュールサイズは 1 画面程度あるという基準に対して 180 を入力し，階層を 3，行数を 7 とした．そして，それぞれのモジュールに対し，計測ツールを利用して各計測項目値を算出した．その結果，実験 1 と同様にモジュールサイズ以外には異常な特徴は発見できなかった．しかし，ある学生が作成した 1 つのスライドにおいて指定したモジュールサイズより大きなモジュールが 1 つ検出された．そして，異常値を示したモジュールは行数においても異常値を示した(図 18)．

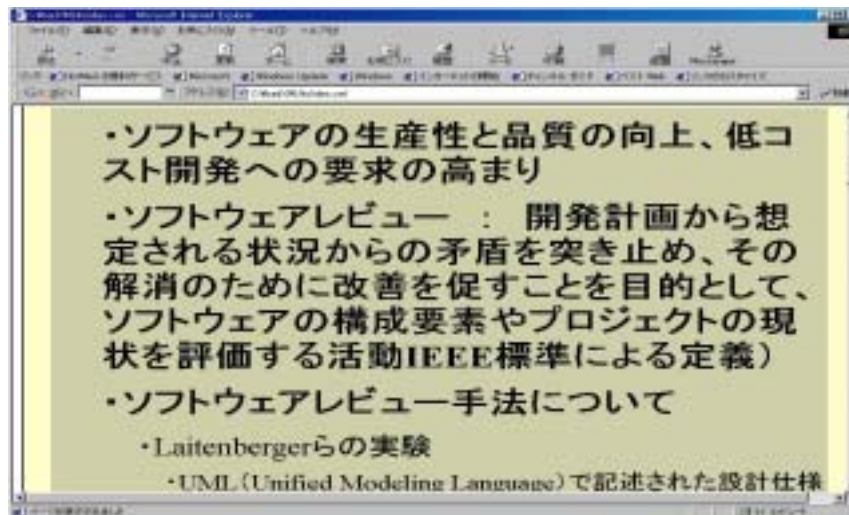


図 18 XML 形式のスライド (修正前)

このモジュールは 1 画面上に情報がすべて表示されていないため，非常に見にくいもの

となっており、内容を把握する場合、スクロールなどの操作が必要となる。しかし、スクロール操作は、ページ内における視点の位置を把握し、どの部分を読んでいるかが容易に分かるという利点がある一方、視点から離れた部分の内容を把握し難くするという欠点もある。従って、モジュールのサイズを 1 画面に収まる分量にすると、ユーザがページをスクロールせずに情報を一覽できる情報量となり、ユーザが情報を把握しやすくなるため、読みやすさを向上させることができる。スクロール操作なく 1 画面に閲覽できるように 2 つのページに分けて、もう一度モジュールサイズ 180、行数 7 を入力した。この修正の結果、ユーザはモジュール単位で読むことが可能となった(図 19)。

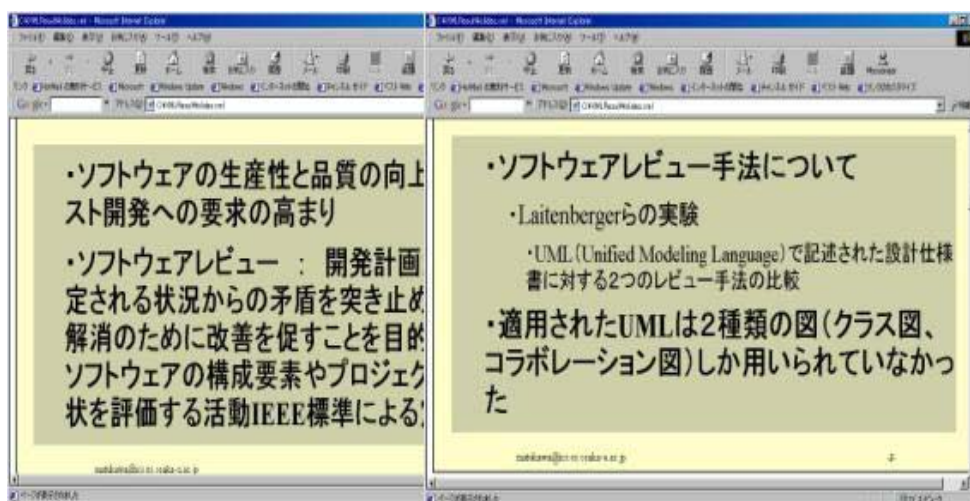


図 19 XML 形式のスライド (修正後)

モジュール	SLIDE
モジュールサイズ	305
要素数	35
モジュールリンク	0
エンティティ	0
階層	2
モジュールサイズ	OK
省略	
指定した階層	3
指定したモジュールサイズ	180
ファイル数	1
全体サイズ	1508
モジュール数	13

表 5 異常値を示したモジュールの計測結果

<実験 3>

学生に作成してもらったマニュアルと Web 上で入手した文書に対して実験を行なったが、異常な特徴を示したモジュールは得られなかった。その理由は、実験用データのマニュアルのサイズが小さいためと思われる。

一方、学生に作成してもらったマニュアルの DTD はかなり類似していた。このように同類のマニュアルは DTD が似ているので、新しくマニュアルを作る場合、同類のマニュアルの DTD を再利用すれば開発コストを削減できると思われる。

6.4 考察

XML 文書の評価基準は、評価しようとしている項目や評価を行なう人によって異なる。そこで、本手法ではモジュールサイズや階層など評価基準値をユーザが指定して評価できるようにした。そして、実際にスライドや論文、マニュアルに対して評価を行なった。

実験 1 と実験 2 の結果では、「モジュールサイズ」の評価項目について異常値を示していた。特に実験 1 は、読みやすさのために指定したモジュールサイズ以上のモジュールが多数であった。

この問題点の対処方法として、1 画面に収める程度の文量に分けた。その結果、内容のまとまりを確認しやすくなり、モジュール単位で読むことが可能となった。

実験 3 において異常値を持つデータは存在しなかった。原因として、学生が作成したマニュアルと Web 上で収集した文書のサイズが小さいためと思われる。また、学生が作成したマニュアルの DTD はほぼ一致しているということを確認したので、次にマニュアルを作成する際にこの DTD を再利用して、コストを削減できると考えられる。

7 まとめ

本研究では，XML で記述された電子文書を対象として，DTD 定義に基づいて評価するための手法を提案した．本手法では，利用目的に応じて多様な XML 文書および評価基準に対応するため，DOM モデルより評価用のデータ構造記述を導出し，導出された構造に対する評価を行った．そして，実際に本手法に基づく評価ツールを実装して，3 種類の XML 文書に対して評価を行なった．その結果，読みやすさの向上と再利用性を確認した．

今後の課題として，以下のような点がある．

- 本研究では，18 件のデータを対象に評価を行なったが，より大量の XML 文書及び大きいサイズの XML 文書を収集して正確な評価を行なう．
- 今回 3 種類の XML 文書に対して評価を行なったが，その以外の XML 文書に対して評価を行なう．
- 本研究で提案した評価基準以外の他の評価基準があれば計測ツールを改良する．
- 過去の HTML 文書評価ツールと統合する．

謝辞

本論文を作成するにあたり、常に適切な御指導を賜わりしました大阪大学大学院基礎工学研究科情報数理系専攻 井上 克郎 教授に心より深く感謝致します。

本論文の作成において、適切な御指導およびご御助言を頂きました大阪大学大学院基礎工学研究科情報数理系専攻 楠本 真一 助教授に深く感謝致します。

本論文の作成において、適切な御指導およびご御助言を頂きました大阪大学大学院基礎工学研究科情報数理系専攻 松下 誠 助手に深く感謝致します。

最後に、その他様々な御指導、御助言等を頂いた大阪大学大学院基礎工学研究科情報数理系専攻 井上研究室の皆様に深く感謝致します。

参考文献

- [1] Guerrieri,Ernesto: “ Software Document Reuse with XML ”, International Conference on Software Reuse , pp246-254 , (1998)
- [2] 谷口慎也, 川口真司, 松下誠, 井上克郎: “ 電子マニュアルの文書構造に対する評価メトリクス ”, 電子情報通信学会ソフトウェアサイエンス研究会, (2001)
- [3] Hoekyung Jung: “ XML Guide ”, Green 出版局, (1998)
- [4] 日本ユニシス情報技術研究会編: “ WWW における XML の活用, 東京電気大学出版局, (2000)
- [5] 金京煥, 川口真司, 谷口慎也, 松下誠, 井上克郎: “ XML ドキュメントデータの構造を利用した品質評価手法の提案 ”, 電子情報通信学会 情報・システムソサイエティ大会論文集, D-3-3, (2001)
- [6] “Extensible Markup Language (XML)”, <http://www.w3c.org/XML> , (2000)
- [7] Michael C.Daconta , Albert J.Saganich Jr: “ Java2 による XML 開発技法 ”, ピアソン・エデュケーション, (2001)
- [8] 富士ゼロックス: “ デジタルドキュメントの作成・管理技法 ”, 日本経済新聞社 ,(1998)
- [9] ニューメディア開発協会: “ 電子マニュアル評価ガイドラインの適正標準化に関する調査研究 ”, <http://www.jtca.org/empg/report98/index.html>, (1999)
- [10] “XML 最新情報 ”, XML PRESS Vol.2 , (1999)
- [11] Simon North, Paul Hermans: “ XML21 日完成 ”, Information Publishing , (1999)
- [12] 横河電気 CyberDoc プロジェクト: “ デジタル時代のドキュメント企画と設計 ”, 日本理工出版会, (2000)
- [13] 長野宏宣: “ ソフトウェア CALS の狙いと実証実験について ”, 情報処理 ,Vol .37 , NO 12 , (1996)
- [14] “Object-Oriented Metrics”: [http:// www.sra.co.jp/people/aoki/OOM006/OomDoc](http://www.sra.co.jp/people/aoki/OOM006/OomDoc)
- [15] HyoungDo Kim: “ An XML -based modeling language for the open interchange of decision models ”, Decision Support System31 pp429-441 , (2001)

- [16] Eckhart Koppen , Gusatf Neumann: “ A Practical approach toward active hyperlinked documents” , Computer Network and ISDN System 30 pp251-258 , (1998)
- [17] P.Ciancarini ,A.Rizzi ,F.Vitali:“ An Extensible Rendering Engine for XML and HTML” , Computer Network and ISDN System , 30(1-7) , pp225-238 , (1998)
- [18] 中山幹敏 , 奥井康弘: “ 改正版標準 XML 完全解説 (上)” , 技術評論社 (2001)
- [19] 丸山宏 , 田村建人 , 浦本直彦: “ XML と Java による Web アプリケーション開発 ” , ピアソン , (1999)
- [20] L.Bompani ,P.Ciancarini ,F.Vitali:“ Active Document in XML ” ,ACM SigWeb Newsletter , 8(1) , pp27-32 , (1999)

