

## XML 文書のスタイルシート生成方式

森口 修 今村 誠 鈴木 克志

E-mail:{mog,imamura,suzuki}@isl.melco.co.jp

三菱電機株式会社 情報技術総合研究所 音声・言語インタフェース技術部

〒247 神奈川県鎌倉市大船 5-1-1

XML(Extensible Markup Language)では、文書の構造を業務に合わせて自由に定義できるので、文書を業務システムのデータとして利用することが可能になる。一方、XML 文書はスタイル情報を含まないため、XML を表示するためにはスタイルシートと呼ばれる文書のスタイル定義を作成しなければならない。しかし、表組み等の複雑な割り付けを行なうスタイルシートの作成には多くの労力を要するという問題があった。本稿では、表組みを4つのパターンに分類することにより、XML 文書の表組み表示を行なうスタイルシートを生成する方式を提案する。この方式では、表組みを行なうスタイルシートの作成が容易となり、仕様書等のように項目の繰返しを含む複雑な文書にも対応できる。

## A Method of Generating Style Sheet for XML documents

Osamu Moriguchi Makoto Imamura Katsushi Suzuki

E-mail:{mog,imamura,suzuki}@isl.melco.co.jp

Mitsubishi Electric Corporation  
Information Technology R&D Center  
Human Media Technology Dept.

5-1-1 Ofuna, Kamakura, Kanagawa 247, JAPAN

XML(Extensible Markup Language), which express the document structure for each applications, enables us to use document information effectively as data in application systems. On the other hand, making document style definitions called "stylesheet" needs for showing XML documents, which is not include style information, but we found difficult to make stylesheet for complicated styles like a table. In this paper, we propose a method of generating stylesheet for making XML documents into tables by classifying 4 table patterns. In this method, stylesheet for table expression is easy to make and can be applied complicated documents include repeating items such as a specification document.

## 1. はじめに

WWW(World Wide Web)の普及により、誰でも簡単にインターネット上の情報にアクセスできるようになった。WWW 上のほとんどの文書は HTML(HyperText Markup Language)で記述されているが、HTML 文書に含まれているタグはレイアウトやハイパーリンクの情報しか表わさないため、HTML 文書をデータとして扱うことは困難である。このような HTML の欠点を補うものとして文書記述言語 XML(Extensible Markup Language)[1]が近年脚光を浴びつつある。XML は、HTML の基となった文書記述言語の規格 SGML(Standard Generalized Markup Language, ISO 8879)に対して、処理が複雑な割に効果が少ない仕様(タグの省略や SGML 宣言)を取り除き、WWW での利用を考慮して URL(Uniform Resource Locator)の仕様を追加し[2]、ハイパーリンク機能を HTML より拡張したものであり[3]、W3C(World Wide Web Consortium)によって仕様が勧告された。

XML 文書が HTML 文書と大きく異なるのは、使用するタグや属性を自由に定義できることである[4]。したがって、XML 文書はアプリケーションのデータとみなすことができる。例えば、製品を注文する時に用いる仕様書を XML とすれば、タグや属性を指定するだけで仕様書から個々の項目の数値が取出せるので、仕様内容のチェック[5]やデータベース登録[6]が可能となる。

一方、XML 文書はレイアウトの情報を含まないため、HTML 文書のようにブラウザで気軽にレイアウトを確認することはできなくなった。XML 文書を表示するためには CSS2(Cascading Style Sheets, level 2)[7]、XSL(Extensible Stylesheet Language)[8] や DSSSL(Document Style Semantics and Specification Language)[9]等のスタイル情報を記述したスタイルシートを作成する必要

がある。従来のスタイルシートの作成方法は2つあった。1つはワープロやDTPで作成した文書から変換する方法であり、もう1つはスタイルシート編集用のエディタを利用する方法[10]である。いずれの方法も、タグ毎に文字サイズや配置等のスタイルを指定するという点に変わりはなく、表組みのように複数のタグとの関係を考慮すべき複雑なスタイル付けを行なう場合にはスタイルシートの作成に多くの労力を要するという問題があった。

そこで、XML 文書のスタイルシートの作成を支援するためのツールを提案する。まず2章で本ツールの目的を述べ、3章で操作と処理の方式を説明する。次に、本方式で用いた表組みパターンについて4章で説明する。最後に5章でスタイルシート作成支援ツールの効果と今後の課題をまとめる。

## 2. 目的

XML で記述した文書は、データとして扱うことができるため、インターネットをインフラとした企業間の業務システムの構築が容易になる。例えば、海外からインターネット経由で XML で記述した製品発注仕様書が送られてくるようなシステムを考えると、仕様書の内容には製品の仕様が項目毎にタグで区切られているため、設計システムや生産管理システムとの連携が容易に実現できる。ただし、XML はレイアウトの記述を含まないため、仕様書を画面に表示したり紙に印刷したりするためには、レイアウト情報を記述したスタイルシートを製品毎にあらかじめ作成しておく必要がある。

このスタイルシートの作成は、仕様書フォーム作成業務の1つの作業と見ることができる。仕様書フォーム作成業務とは、新製品が開発された場合や製品の仕様が変更された場合に、その製品に対応する仕様書のフォームを作成または変更するというシステム管理者の業務である。仕様書

フォーム作成業務は以下に示す作業からなり、このうちスタイルシートの作成は(4)の作業である。

- (1) 仕様項目の分析  
対象製品の全ての仕様項目を洗い出し、仕様項目の階層分類を行なう。
- (2) 仕様書フォームの作成  
仕様項目のおおまかなレイアウト、すなわち表組みパターンを決める。表組みパターンの詳細については4章で説明する。
- (3) DTDの作成  
表組みパターン毎に決められた文書構造の条件を満足するように、仕様書のDTD(Document Type Definition)を作成する。
- (4) スタイルシートの作成  
この作業を本ツールにより支援する。本ツールはDTDファイルを入力としてスタイルシートファイルを出力する。仕様書フォーム作成者は(2),(3)で決めた表組みの範囲と表組みパターンを指定する。詳細は3.1節で説明する。
- (5) スタイルシートの登録  
(4)で作成したスタイルシートファイルをシステムに登録する。登録後、新たな製品の仕様書がシステムで扱えるようになる。

しかし、(4)のスタイルシートの作成作業は、仕様書のように繰り返し項目を含む複雑な文書を表組みレイアウトで表示する場合、従来はタグ毎に表組みの開始と終了、行の区切りや列の区切り等のスタイル付けを要するため、多大な労力を要しかつ作成ミスも多かった。

本ツールの目的は、XML文書のスタイルシート作成作業の労力と作成ミスを軽減することである。

### 3. スタイルシート作成支援ツールによる表組み処理方式

前述の目的を達成するための本ツールによる表組み処理方式について説明する。

#### 3.1 スタイルシート作成の操作手順

仕様書フォーム管理ユーザによる本ツールを用いたスタイルシート作成の操作手順は以下ようになる。

- (1) DTDの読込み(入力)  
あらかじめ作成したDTDファイルを指定する。図1に示すスタイル表画面となる。なお、図1は自動販売機の発注仕様書の例である。
- (2) 表組み範囲と表組みパターンの指定  
スタイル表画面上で、表組み範囲と表組みパターンを指定する。  
図1では、COMMON.ITEMSタグ以下の部分構造を表組みパターン4に指定している。「表組み」欄の「TAB4」が表組みパターン4の開始を、「/TAB4」が終了を表わす。「TLF」はその位置で表を改段する指定である。「親見出し1~3」により隣り合った複数のタグをまとめる上位の見出しを指定することができる。
- (3) 表組みプロパティの設定  
スタイル表画面の「表組み」欄の右に位置する「設定」ボタンを押すと図2に示すダイアログボックスが表示され、表組み方向や枠線の太さ等の表組みプロパティを設定できる。
- (4) スタイルシートの保存(出力)  
(2)(3)の設定結果をスタイル表とスタイルシートの2つのファイルに保存する。スタイル表ファイルは、本ツール独自の形式であり、スタイル表を再編集する場合に用いる。

タグ名	L	見出し	親見出し1	親見出し2	親見出し3	表組み	
COMMON ITEMS	3	COMMON ITEMS				TAB4	設定
VERSION	4	VERSION					
VER. PUBLIC	#	公式	副版				
VER. INNER	#	内部	副版				
SPEC. ID	4	SPEC. ID					
PROJECT. NO	#	番号	プロジェクト				
PROJECT. NAME	#	名	プロジェクト				
COUNTRY	#	国名				TLF	
DISPLAY	4	DISPLAY					
DSP. ROW	#	行数	商品陳列				
DSP. COL	#	列数	商品陳列				
DSP. SUM	#	合計数	商品陳列			TLF	
SUPPLY	4	SUPPLY					
SERIES	#	機種					
AGENT	#	販社					
FACTORY	#	工場				TLF	
LOCAL	4	LOCAL					
LS. COUNTRY	#	国名	現地調達	国別仕様			
LS. CODE	#	コード	現地調達	国別仕様			
REGULATION	#	法規		国別仕様			
COLD. DISTRICT	#	寒冷地仕様		国別仕様		TLF	
OS	4	OS					
OS. NAME	#	名前	OS				
OS. VERSION	#	バージョン	OS			/TAB4	

図1. スタイル表画面での表組みパターンの指定

表パターン	
方向	<input checked="" type="radio"/> 水平 <input type="radio"/> 垂直
列数	4
枠線の色	1
セル間スペース	0
セル内マージン	1
見出し背景色	
その他背景色	
行番号	<input checked="" type="radio"/> あり <input type="radio"/> なし
角見出し	
キャプション	
共通項目	
配置	<input checked="" type="radio"/> 左 <input type="radio"/> 中 <input type="radio"/> 右
サイズ	+1

OK      キャンセル

図2. 表組みプロパティの設定

### 3. 2 スタイルシート生成方式

本ツールにおいてスタイル表からスタイルシートを生成する処理方式を説明する。

- (1) XML 文書の構造木を走査し、ユーザが表組みパターンを指定したタグを見つける。表組みパターンが指定されたタグ以下の部分構造木を表組みの範囲とする(図3)。
- (2) 表組み範囲の部分構造木と DTD 部品テンプレートとのパターンマッチを行なう(図4)。DTD 部品テンプレートとは、表組みパターン毎に定められた文書構造の部分パターンである。パターンマッチに失敗する場合、本ツールはスタイル表画面(図1)で表組みパターンの指定ができないことをユーザに通知する。
- (3) DTD 部品テンプレートに対して定められたスタイルシート部品テンプレートを表組み範囲の部分構造木に適用する(図5)。

スタイルシート部品テンプレートとは、DTD 部品テンプレートの各タグに対して定められたスタイル定義ルールである。したがって、各スタイル定義ルールを(2)のパターンマッチにおいて対応付けした表組み範囲の各タグに適用する。

スタイルシートは図6に示すようにスタイル定義ルールの集合からなり、スタイル定義ルールはパターンとアクションからなる。パターンは、タグ名や出現位置などを指定したルールの起動条件である。アクションは、ルールが起動した場合にそのルールが実行すべき処理の内容である。本方式では画面表示に Web ブラウザを利用するため、アクションの処理内容は HTML データの出力である。また、アクションはタグ開始時に実行する処理とタグ終了時に実行する処理を区別する。タグ開始時とは、木構造を縦型に走査した場合にそのタグに初めて到達した時である。タグ終了時とは、木構造を縦型に走査した場合にそのタグの下位構造の走査が完了した時である。

例えば、図5のAはタグ開始時、終了時にそれぞれ<TABLE>タグ、</TABLE>タグを出力する図7に示すスタイル定義ルールとなる。同様に、B、Cのタグ開始時と終了時のアクションが、それぞれ<TR>、</TR>、<TD>とタグ内容および</TD>の出力であるとする、A、B、Cのスタイル定義ルールからなるスタイルシートを用いて、表組み範囲の部分構造木をスタイルシートインタプリタが処理した出力結果は図8に示す画面データとなる。ここでは、図5の太枠で示す繰返しタグの繰返し数を2とし、

X(n)はn番目に出現するタグXを表わすものとする。

- (4) 図2の画面で設定した表組みプロパティにしたがって、(3)で適用した各スタイル定義ルールを適宜調整する。例えば、線の太さを指定しなかった場合の処理結果は<TABLE>タグとなるのに対して、線の太さを 3 に指定した場合は<TABLE BORDER=3>タグとなる。
- (5) 全てのスタイル定義ルールを出力することにより、スタイルシートファイルを作成する。

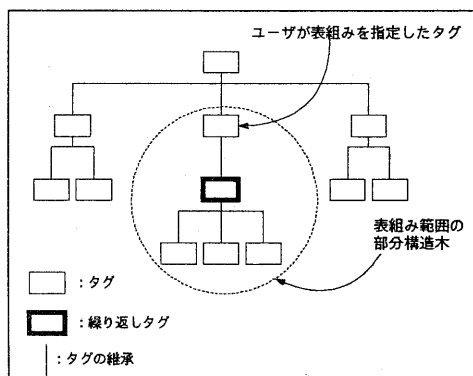


図3. 表組み指定と表組み範囲

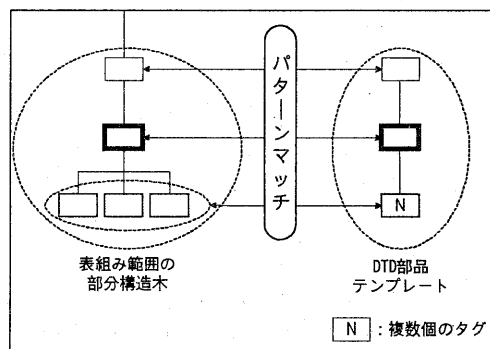


図4. DTD 部品テンプレートとのパターンマッチ

スタイルシート ::= スタイル定義ルールの集合  
 スタイル定義ルール ::= パターン アクション  
 パターン ::= スタイル定義ルールの起動条件  
 アクション ::= スタイル定義ルールの処理内容

図6. スタイルシートの構成

```

<rule>                                ←スタイル定義ルールの開始
  <target-element type="X">           ←パターン指定      ↓以下はアクション指定
  <TABLE>                             ←タグ開始時の処理
  <children/>                         ←下位要素についてもスタイル処理を行なう指定
  </TABLE>                             ←タグ終了時の処理
</rule>                                ←スタイル定義ルールの終了

```

図7. スタイル定義ルールの例

```

<TABLE>
  <TR><TD>X (1)の内容</TD><TD>Y (1)の内容</TD><TD>Z (1)の内容</TD></TR>
  <TR><TD>X (2)の内容</TD><TD>Y (2)の内容</TD><TD>Z (2)の内容</TD></TR>
</TABLE>

```

図8. スタイルシート処理結果の例

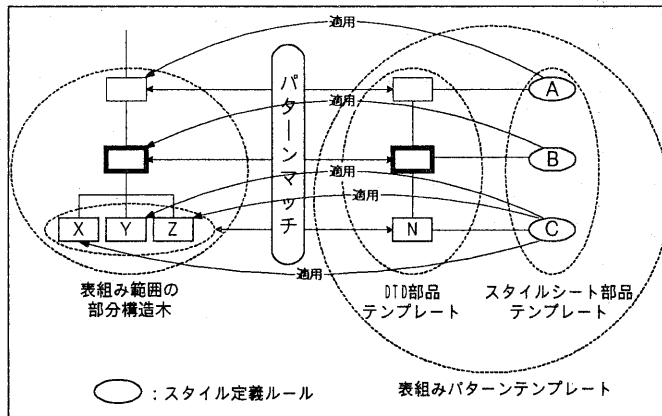


図5. スタイルシート部品テンプレートの適用

#### 4. 表組みパターン

仕様書に出現する表組みパターンを分析した結果、表組みパターンを表1に示す4つに分類した。分類の基準は、表組み範囲の部分構造に対して各階層の継承関係やタグ数等に制約があるかどうか、およびその部分構造が繰返しタグ (DTDで\*または+の出現標識が指定されている要素) を含むかどうかである。表組みが可能となる部分構造は、行と列とにそれぞれ対応付けができればよいため、階層制約は2階層以内、繰

返しは2次元以内に限定できる。したがって、この4つのパターンにより表組みの文書構造を網羅できると考えられる。

表組みパターン1~4の DTD 部品テンプレートとパターンマッチする部分構造木、およびスタイルシート部品テンプレートを適用して生成したスタイルシートによるレイアウト表示の関係を、それぞれ図9~図12にまとめる。図中では、部分構造木のタグに番号を付与し、各タグが表組みレイアウトのどのセルに対応するかを示す。

表1. 表組みパターンの説明と分類基準

表組みパターン	説明	分類の基準	
		階層制約	繰返し
1	行または列が繰り返してある。	2階層かつ一方が繰り返し	1次元
2	行および列が繰り返してある。	2階層かつ両方が繰り返し	2次元
3	2階層の部分構造木の各階層が行と列を形成する。	2階層かつ末端のタグ数が同一	なし
4	項目の見出しと内容の対を列挙する。	なし	なし

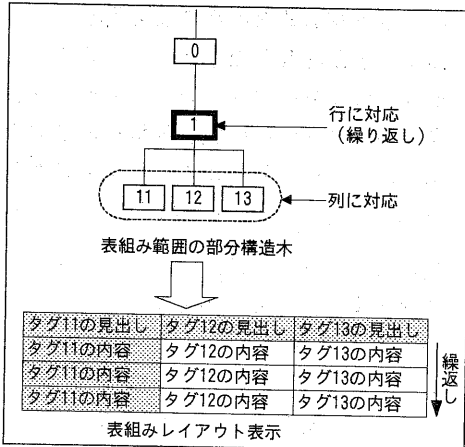


図9. 表組みパターン1の部分構造木とレイアウト表示

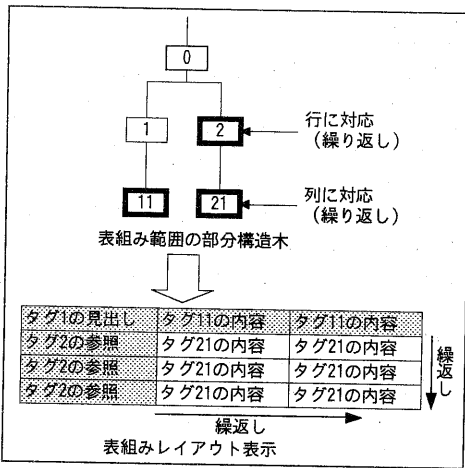


図10. 表組みパターン2の部分構造木とレイアウト表示

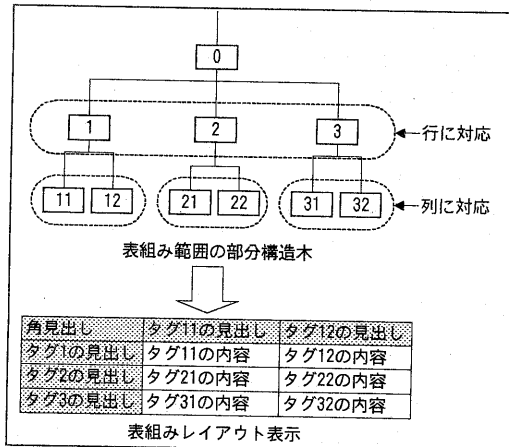


図11. 表組みパターン3の部分構造木とレイアウト表示

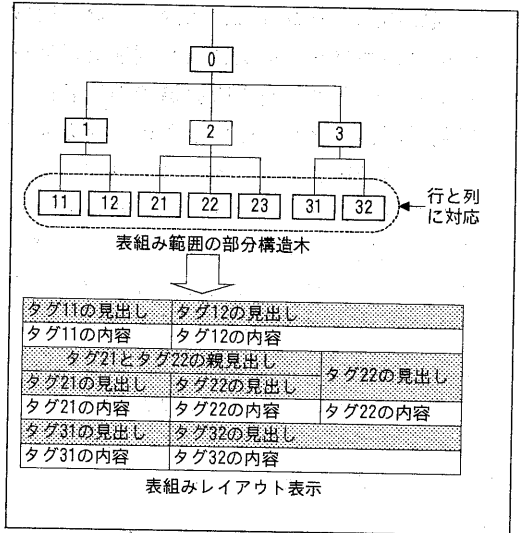


図12. 表組みパターン4の部分構造木とレイアウト表示

表組みパターン1は、1次元の繰返しタグが表の行に対応する。例えば、自動販売機の商品サンプル陳列部の寸法を仕様書に記述する場合、タグ1の繰返しを段としてタグ11で各段の名前を記述し、タグ12で各段の高さを記述することで、商品サンプル陳列部の各段の名前と寸法を表組みして表示できる。

表組みパターン2は、2次元の繰返しタグが表の行と列に対応する。例えば、自動販売機の場合、タグ11の繰返しで列の名前を記述し、タグ2とタグ21からなる2次元の繰返しで各段各列の商品名を記述することで、商品サンプル陳列部の各段各列の商品名を表組みして表示できる。

表組みパターン3は、2つの階層が行と列に対応する。例えば、自動販売機の部品を仕様書で指定する場合、タグ1, 2, 3をそれぞれコイン投入口、商品選択ボタン、商品取り出し口という部品の名前とし、タグ11, 21, 31で部品番号を記述し、タグ12, 22, 32で色を記述することで、各部品名に対する部品番号と色を表組みして表示できる。

表組みパターン4は、繰返しのない複数のタグの列挙を、指定された列数で行に分割する。例えば、自動販売機の防犯装置、自動照明、防水やくじ引き等の種々のオプションの列挙を、3列ずつに分割することでオプションを3列の表組みで表示できる。また、表組みパターン4では末端のタグに対してのみスタイル定義ルールを適用することで、文書構造の階層制約をなくした。これは繰返しを含まない一般的な帳票等が表組みパターン1, 2, 3が適用できない文書構造であったとしても対応できるようにするためである。

## 5. まとめ

本方式では、ユーザはXML文書の複数のタグからなる部分構造に対して1つの表組みパターンを指定すれば良いため、タグ毎にスタイルを定義するのに比べてスタイルシートの作成が容易である。特に、仕様書等のタグ数が多く、繰返しを含む構造を持った文書に対して表組み等の複雑なスタイル付けを行なう場合に有効である。さらに個々のタグ毎のスタイル付けで生じていたミスを防ぐことができる。

また、表組み範囲の部分構造に対して階層関係の制約と繰返しタグの有無に着目することにより、表組みを4つのパターンで網羅した。さらに、表組みの階層制約を満たさない文書構造を有する一般的な帳票から、繰返しを含む仕様書まで広く対応できる処理方式とした。

以下は今後の課題である。

- (1) 出力をHTMLとしたため、表組みの方向等はHTMLによる制限を受けている。HTMLの動向やXSLのレンダリングエンジン等、Webブラウザの動向に沿った対応が必要である。
- (2) 現状では、スタイルシート毎に表示レイアウトが固定であるため、複数の表示レ

アウトが必要な場合には複数のスタイルシートが必要となるという点でメンテナンス性が悪い。1つのスタイルシートを用いてユーザ毎に表示画面を変更するなどのように、外部から表示レイアウトを制御可能とする必要がある。

- (3) 現状のデータベース連携は、データベースから文書内容の一部を受け取るという一方の連携となっている。スタイルシートを処理した結果データベースを更新する機能を実装することによる双方向の連携が必要である。

## 参考文献

- [1] Extensible Markup Language (XML) 1.0 W3C Recommendation 10-February-1998, <http://www.w3.org/TR/REC-xml>
- [2] 津田 宏, 浦本 直彦, 上田 隆也, 佐藤 研治, 野村 浩郷: XMLとその周辺の標準化動向の概要, 情報処理学会研究会報告 DBS-115-3-FI-49-3 pp.17-24, 1998
- [3] 後藤 正智: XLinkを利用したXMLドキュメントブラウザ, 情報処理学会研究会報告 DD-14-1 pp.1-8, 1998
- [4] 道本 健二, 中道 理, 高本 優: 究極のデータ表現XML, 日経バイト 1999.1月号 pp.112-131, 1999
- [5] 今村 誠, 森口 修, 鈴木 克志, 辻 秀一: SGML文書の内容検証方式とその評価, 情報処理学会研究会報告 DD-11-1 pp.1-8, 1998
- [6] 矢島 正樹, 藤津 真一, 大野 邦夫: SGMLデータカートリッジによる文書管理システムの構築, 情報処理学会研究会報告 DD-14-2 pp.9-16, 1998
- [7] Cascading Style Sheets, level 2 CSS2 Specification W3C Recommendation 12-May, 1998, <http://www.w3.org/TR/REC-CSS2/>
- [8] Extensible Stylesheet Language (XSL), <http://www.w3.org/Style/XSL/>
- [9] ISO/IEC 10179:1996 Document Style Semantics and Specification Language (DSSSL), <http://www.jclark.com/dsssl/>
- [10] 岩垂 誠, 屋代 禎: SGML/XML文書の作成と表示に関する取組み, 情報処理学会研究会報告 DD-12-1 pp.1-9, 1998