# 汎用な表のためのXML形式

XML Data Formats for General Tabular Forms

塩野 康徳[†] 有田 友和[††] 切島 忠昭[†] 土田 賢省[†] 夜久 竹夫[†††]
Yasunori SHIONO Tomokazu ARITA Tadaaki KIRISHIMA Kensei TSUCHIDA Takeo YAKU

[†] 東洋大学工学部情報工学科
Dept. of Information and Computer Sciences, Toyo University
{gz0400100, dz020001x} @toyonet.toyo.ac.jp, kensei@eng.toyo.ac.jp

[††] 桜美林大学文学部言語コミュニケーション学科
Dept. of Languages and Infomation Studies, Obirin University
arita@obirin.ac.jp

[†††] 日本大学文理学部情報システム解析学科
Dept. of Computer Science and System Analysis, Nihon University
yaku@cs.chs.nihon-u.ac.jp

我々は，編集と表示の信頼性と効率性を考慮に入れた表処理システムの開発を進めている．既に，非スライス構造も含む汎用な表を効率的に扱える表のグラフ表現と，それに対応したリスト型データ構造 H3-Code を提案している．今回は，XML に対応できるように H3-Code を拡張した XML_H3-Code，さらに XML_H3-Code のファイル保存形式である RD_XML を定義した．RD_XML は，用意した XSL ファイルにより汎用 Web ブラウザでの表示が可能である．そして，非スライス構造を含む汎用な表を対象とした XML 対応の表エディタの開発を進めている．本稿では，これらの中で，XML_H3-Code，RD_XML，そして表の XML 表示を中心に述べる．

## 1 はじめに

様々な表が多くの分野で情報の整理や視覚化のツールとして利用されている．大きさの異なるセルをもつ表等，様々な種類の表がコンピュータインタフェースやドキュメント [1][2] において頻繁に使われている．このような表を扱う既存の表処理モジュールの編集操作では，しばしばユーザが予期しない結果を招くことがある．これは，表の基本データ構造を形式的に定義していない，あるいは，その構造がユーザに公開されていないこと等が一因と考えられる．そこで，我々は，rectangular dissection graph[3] に基づく，信頼性と効率性のある編集と表示を考慮に入れた表処理システムの開発を進めている．その中で，本研究では，インターネット上でのデータ交換フォーマットの標準として認められつつある XML[4] に対応した処理系の開発を目的とする．さらに，オープンソースソフトウェア [5] である OpenOffice.org へのアドオンを目指している．以下では，我々の最近の成果である XML_H3-Code，RD_XML，そして表の XML 表示を中心に説明する．

## 2 Tabular Diagrams[3]

tablar diagram は，表を形式的に定義したものである．ここでは，表に関するいくつかの我々の定義を述べる．そして，[3] で述べられている Condition 2.1 を満たす tabular diagram を考える．この tabular diagram は，表現したい表の周囲に基準となるような perimeter セルを持っていることを特徴とする．

**[定義 2.1]** $(n,m)-table$ は整数の組の集合 $\{(i,j)|1 \leq i \leq n, 1 \leq j \leq m\}$ である．表とは，ある整数 $n$ と $m$ に対する $(n,m)-table$ である．$n$ を行数，$m$ を列数と呼ぶ．*partial table* は $(n,m)-table$ の部分集合 $S$ であり，かつ $S = \{(i,j)|k \leq i \leq l, s \leq j \leq t\}$ となるものである．ただし, $1 \leq k$, $l \leq n$, $1 \leq s$, $t \leq m$ とする．表 $T$ の *partition* $P$ とは，互いに素な $T$ の部分集合の集まりで，$P = \{S_1, S_2, \ldots, S_N\}$ である．ここで，$S_1 \cup S_2 \cup \ldots \cup S_N = T$ であり, 各々の $S_i$ は，セルと呼ばれる．

[**定義 2.2**] $(n,m)-$table $T$ の *row grid* とは，写像 $g_{row}:\{0,1,\ldots,n\}\to\mathbf{R}$ のことである．ただし，$g_{row}(i)\le g_{row}(i+1)$ であり，$0\le i\le n-1$ である．$T$ の *column grid* とは，写像 $g_{column}:\{0,1,\ldots,m\}\to\mathbf{R}$ のことである．ただし，$g_{column}(j)\le g_{column}(j+1)$ であり，$0\le j\le m-1$ である．*grid* とは，ペア $g=(g_{row},g_{column})$ である．

*tabular diagram* は，表 $T$ と $T$ 上の partition $P$，そして $T$ の grid $g$ からなる 3 項組 $D=(T,P,g)$ である．

[**例 2.1**] 図 1 の $D_1$ は tabular diagram $D_1=(T_1,P_1,g_1)$ である．ここで，$P_1=\{\{(1,1),(2,1)\},\{(1,2)\},\{(1,3)\},\{(2,2),(2,3)\}\}$ は $(2,3)-$table $T_1$ 上の partition である．グリッド $g_1$ は $g_{1row}(0)=0, g_{1row}(1)=1$, $g_{1row}(2)=2$ と $g_{1column}(0)=0, g_{1column}(1)=2$, $g_{1column}(2)=3$, $g_{1column}(3)=4$ によって定義される．図 1 の $D_{1p}$ は perimeter セルを持つ tabular diagram $D_{1p}=(T_{1p},P_{1p},g_{1p})$ である．$D_{1p}$ の perimeter セルを縮退して（面積を 0 にして）描いた表が $D_1$ であり，この $D_1$ が実際の処理系の表示となる．

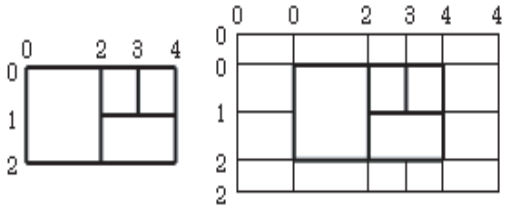


図 1　$D_1$（左）と $D_{1p}$（右）

## 3 Rectangular Dissection Graph[3]

表のグラフ表現である rectangular dissection graph (tessellation graph [1]) は，セルをノードで表し，隣のセルとの接続をエッジによって定義している．そして，表の周囲に基準となる perimeter セルに対応する perimeter ノードを持つ（図 2 参照）．

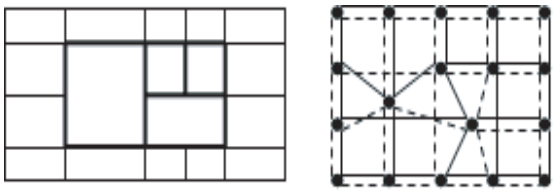

図 2　表構造図とそれに対応する rectangular dissection graph

以下に，このグラフの定義と性質について述べる．

[**定義 3.1**] tabular diagram $D=(T,P,g)$ に対する *rectangular dissection graph* は，6 項組みの属性グラフ $G_D=(V_D,E_D,L,\lambda_D,A,\alpha_D)$ として表される．ここで，$(V_D,E_D)$ は無向多重辺グラフ，$L$ はエッジラベルの集合，$\lambda_D:E_D\to L$ はラベル関数でルール 1-4 を満たし，$A$ は属性の集合で $A=\mathbf{R^4}$，$\alpha_D:V_D\to\mathbf{R^4}$ は $\alpha_D(v_c)=(nw(c),sw(c),ew(c),ww(c))$ により定義される．

[**ルール 1**] もし $nw(c)=nw(d)$，すなわち，$c$ と $d$ が等しい north wall を持ち，かつ $c$ と $d$ の間に等しい north wall をもつセルがないならば，そのとき，$[v_c,v_d]$ は $E_D$ の元であり，$\lambda_D[v_c,v_d]=enw$ である．この場合，$[v_c,v_d]$ は north wall エッジと呼ばれる．

同様に，ルール 2, ルール 3，ルール 4 はそれぞれ south wall エッジ, east wall エッジ, west wall エッジを定義する．

このようにして定義された rectangular dissetion graph では，各頂点の次数は，高々8 である．

[**例 2.2**] 図 2 は，tabular diagram とそれに対応する rectangular dissection graph である．south wall エッジと west wall エッジは，点線で表している．

perimeter セルをもつ tabular diagram を考えたとき，以下の命題が成り立つ．

[**命題 2.1**] $G_D$ を $(n,m)-$table の tabular diagram $D$ に対する rectangular dissection graph とする．$k$ を $G_D$ の inner node とする．$G_D$ のエッジの数 $\#E_D$ に対して，$2\#E_D=6(2n-4)+6(2m-4)+8k+16.$ が成り立つ．

## 4 H3-Code[6]

H3-Code は，rectangular dissection graph による表のモデルを反映したリスト形データ構造である．表の 1 つのセルを表すノードとその接続によって表される．グラフのノードは，H3-Code のノードに対応し，グラフの辺は，H3-Code のノード間をポインタで接続して表現する．H3-Code のノードは，次のような情報を持つ．

(1) ノード番号 (ID).
(2) 他ノードの接続を表す 8 個のポインタ変数.
(3) ノード周りの壁の位置情報.

(4) ノード内に設定されるコンテンツの情報.
(5) 表構造図を多言語に変換するための情報.

図 3 は，ノードの図表現である．図 4 は，図 2 の rectangular dissection graph に対応する H3-Code の概形図である．

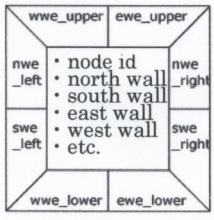


図 3　H3-Code のノードの図表現

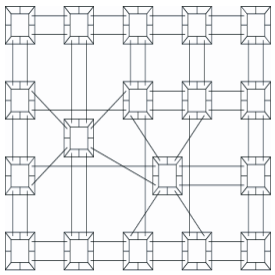

図 4　H3-Code の概形図

このような H3-Code を用いて表処理システムを実現することにより，非スライス構造を含む一般的な表も扱うことが可能となる．さらに，ノードの最大次数が 8 であるので，罫線移動等の編集操作の計算量を小さくできる．

## 5　XML 対応の表処理システム

### 5.1　システム構成

図 5 は，本システムの構成を図示したものである．次節以降に，図 5 の各部分について説明する．

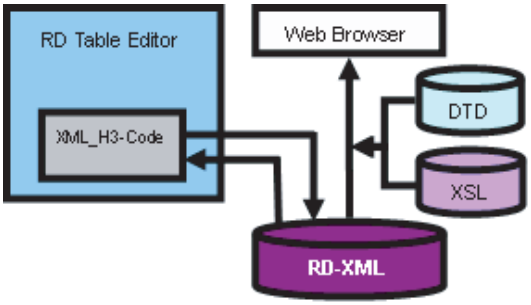


図 5　表処理システムの構成図

### 5.2　XML_H3-Code

XML_H3-Code は，XML に対応した内部データ形式であり，H3-Code を基にして，XML 用に拡張したリスト型データ構造である．図 6 は，XML_H3-Code のクラス図である．

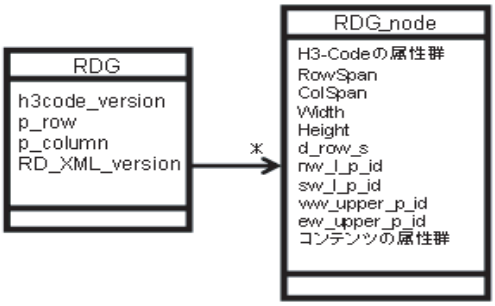


図 6　XML_H3-Code のクラス図

RDG クラスは一つの表を表しており，H3-Code の表情報と，複数のセル情報を持つ．XML への拡張として RD_XML_version 情報を持つ．RDG_node クラスは一つのセルを表しており，H3-Code のノード情報を持ち，XML への拡張として RowSpan, ColSpan, Width, Height 等の情報を持つ．

### 5.3　RD-XML

表の交換言語としての XML 形式である RD-XML を提案する．RD-XML は，XML_H3-Code に基づいている．XML_H3-Code で表現した表に，XSL を適用することにより，IE 等の一般の Web ブラウザで表示することができるファイル形式であり，表の情報を扱うアプリケーション間でのデータ交換のために基礎的なフォーマットを提供する．これにより，アプリケーション間での表構造情報等の交換を容易にする．XML_H3-Code に双方向に変換できる形式であるため，XML_H3-Code のファイル保存形式でもあり， XML_H3-Code の編集と表示に対応できる．

RD-XML では、H3-Code 情報，XSL を用いて HTML に変換するときに使用する情報，セルのコンテンツ情報，RD Table Editor で使用する情報等を扱うことができる．

図 7 は，表の構造図とそれに対応する RD-XML のデータ構造である．図中の数字は XML_H3-Code に基づいたセルの ID を表し，その対応を示す．RD-XML は XML_H3-Code に基づくデータ構造になっており，位置情報の取得が容易であることや，表の罫線構造を長方形分割として捉えたとき，スライス

構造と非スライス構造をすべて表現すること等ができる．

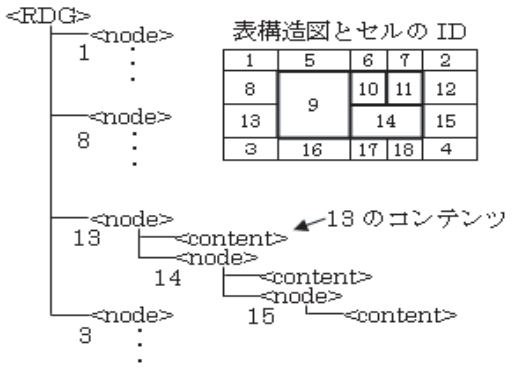


図 7　表構造図とそれに対応する RD-XML のデータ構造

図 8 は，RD-XML に XSL を適用し，実際に Web ブラウザで表示した結果である．

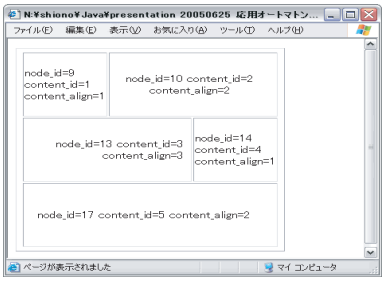


図 8 Web ブラウザでの表示結果

### 5.4　RD Table Editor

本システムにおける RD Table Editor は XML_H3-Code を内部データとして持ち，RD-XML をファイル形式としている．つまり，XML_H3-Code で表現できる表はすべて編集することができ，RD-XML での保存と読み込みが可能である．

図 9 は RD Table Editor の実行画面であり，現在，editor は開発継続中である．

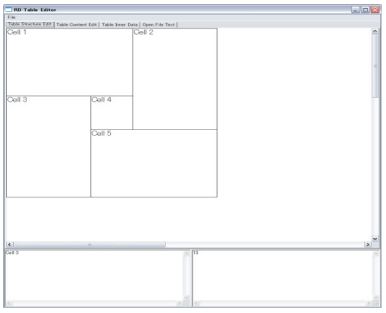

図 9 RD Table Editor の実行画面

RD Table Editor は，次のような特徴を持つ．

- XML_H3-Code を内部データとして持つ，RD-XML 対応の表エディタ．
- Rectangular dissection graph に対する編集操作を実装（一部実装）．
- XML_H3-Code と RD-XML の双方向の変換ができ，RD-XML での保存と読み込みが可能．

## 6　まとめ

今回は，信頼性と効率性のある表処理システムについて検討し，H3-Code を基に，XML 対応の内部データ形式である XML_H3-Code を定義した．そして，表の交換言語としての XML 形式である RD-XML と，それを Web ブラウザで表示するための XSL を定義した．さらに，XML_H3-Code と RD-XML の双方向の変換を実現した．その結果，RD-XML は， Web ブラウザで表示が可能であり，かつ XML_H3-Code への変換により，エディタでの編集が可能である．エディタに関しては，編集機能を一部実装した状態で，現在開発中である．

今後は，編集機能での効率的なアルゴリズムの開発および実装，OpenOffice.org へのアドオンを行う予定である．

### 謝辞

本論文について，貴重な御助言，御協力頂きました関東学院大学の本橋友江講師に感謝致します．

### 参考文献

[1] 出原栄一，吉田武夫，渥美浩章，図の体系　図的思考とその表現，日科技連，1986.

[2] A.J.Wilder and J.V Tucker, System Documentation Using Tables — A Short Course, Communications Research Laboratory, McMaster University, CRL Report 306, 1995.

[3] T. Motohashi, K. Tsuchida and T. Yaku, Attribute Graphs and Their Algorithms for Table Interface, TECHNICAL REPORT OF IEICE, SS2002-1, 2005, pp. 1-6.

[4] XML Consortium, http://www.xmlconsortium.org/

[5] IPA, http://www.ipa.go.jp/software/open/index.html

[6] H3-Code Web Site, http://www.yaku.cs.chs.nihon-u.ac.jp/tech_note/2003/hcc03-001/index.html

[7] T. Kirishima, T. Motohashi, K. Tsuchida and T.Yaku, TABLE PROCESSING BASED ON ATTRIBUTE GRAPHS, Proc. the IASTED International Conference on Software Engineering and Applications, 2002, pp. 317-322.