A1-5

Colored Bottom-up DataGuide による 半構造データのための差異発見・可視化機構

小 島 岳 史[†] 清 光 英 成[††] 田 中 克 己[†††]

Web ページに記載されているデータは，同じ種類の対象に関するデータでも，データの記述に使われる用語やデータの構造自体が異なっている半構造データである．そのため，機械的な比較が困難であり，ユーザは個々のページを閲覧して内容を分析する必要がある．そこで，Web 上の複数のコンテンツから得られた同種の対象に関する半構造データの値や構造の差異を可視化する Colored Bottom-up DataGuide(CBA-DataGuide) を提案する．CBA-DataGuide は複数の半構造データの共通スキーマをツリー形式 (OEM) で表し，データの構造や値の差異をノードの色の違いで可視化したものである．CBA-DataGuide は既存の DataGuide と異なり，ボトムアップに DataGuide オブジェクトを生成することでデータごとに用語の違いや構造の違いがあっても共通のスキーマを生成することが出来る．

Difference Discovery and Visualization of Semistructured Data by Colored Bottom-up DataGuide

Takeshi Kojima,[†] Hidenari Kiyomitsu[††] and Katsumi Tanaka[†††]

In this paper, we suggest a different discovery and visualization system for semistructured data. We assume that semistructured data with the same or similar property belong to the same category, we call these data “same field data”. Our system discovers differences between same field data and visualizes the property. The system generates common schema of OEM graphs from same field data, same field data can be comparable.

The common schema which is colored OEM generated by bottom-up algorithm in this system. This common schema is Colored Bottm-up DataGuide(CBA-DataGuide).

Difference between same-field data can be discovered from their OEM graphs by using its common schema. Moreover, visualization of same field data’s difference or each data own are represented by the Colored OEM graphs.

1. はじめに

近年，情報検索技術の発展により，ユーザの求める情報を Web 上から効率よく探すことが出来るようになった．しかし，検索エンジンによって示された各 Web ページは，ユーザ自身が一つ一つのページを閲覧して内容を判別しなければならない．なぜなら，Web 上に存在する多くのデータは構造がまちまちな半構造データであり，データ間の差異を機械的に発見することが困難であるからである．そのため各 Web ページの情報の比較・差異発見が自動でできれば有用である．

本論文はこのような機構を実現するために

- 比較対象の半構造データを OEM 形式で表現
- 比較対象の半構造データの共通スキーマをボトムアップに生成
- データを特徴に応じてノードに着色したツリーで表現し，ユーザに提示

する手法を提案する．

Web ページから得た半構造データは，共通の表現方法で記述されているとは限らない．そこで，扱いやすくするために全てのデータを情報交換モデルの一つである OEM で表現する．

半構造データを比較するために，データごとの用語，構造の違いを解消した共通のスキーマを生成する．共通スキーマは，本論文で提案するボトムアップなアルゴリズムで生成される．

生成したスキーマを用いてデータを比較し，差異を発見する．共通のスキーマと個々のデータをツリー形式で表現し，ユーザに提示する．提示されるツリーは，複数のデータに共通な部分，個々のデータの特異

† 神戸大学大学院自然科学研究科
Graduate School of Science and Technology, Kobe University
†† 神戸大学国際文化学部
Faculty of Cross-Cultural Studies, Kobe University
††† 京都大学大学院情報学研究科
School of Informatics,Kyoto University

な部分が色分けされており，データ全体の傾向と個々のデータの特徴をつかむことができる．

2. 関連研究

2.1 SCD

SCD(Semantic Change-Detection)[1] は，Seung-Jin らが提案した，二つの HTML の間の意味的な違いを検出するアルゴリズムである．SCD は比較対象の HTML をツリー形式に変換し，二つのツリーから共通の枝を除去して残った枝をデータ間の差異とする．

SCD は二つのデータの内容の違いよりも，一つの HTML データがどう変化したかを検出するのに向いている．そのため，気象や株式のデータがどのように変化してきたかを，一つのサイトの更新を追うことで調べるようなアプリケーションが提案されている．

本研究で提案する機構は，複数のサイトに記載されているために記述方法が異なっている情報を比較し，その内容の差異の発見を容易にするものである．本研究でも HTML データをツリー形式にしてデータ同士を比較するという手段をとっているが，同じサイトの情報を比較する SCD と違いデータの用語，構造の違いが非常に大きい．そこで，共通のスキーマを生成してデータ間で共通の属性値を明確にし，データを比較する．また，共通のスキーマにメタデータを持たせておくことで，データ間の差異だけでなく比較対象のデータ全体の傾向を把握できるようになっている．

2.2 XWRAP

XWRAP[2] は Ling らが提案した，複数の HTML ソースからユーザが興味のある情報を抜き出し，共通の形式にまとめるためのラッパーである．XWRAP は以下のように生成される．

(1) 複数のサイトからサンプルページを取得
(2) サンプルページの HTML ソースをツリーで表現
(3) ユーザが必要なデータを表現している部分木を判別
(4) 一つのサンプルの部分木の切り取り方をルール化
(5) ルールを他のサンプルに適用して，不都合があればルールを追加

これは，同じサイトに記載されている同種の対象に関するデータは，属性の並び順や HTML タグの記述方法が同じであり，サイトが異なってもある程度の共通性があるという性質を利用したものである．

他に，複数の HTML データから必要な情報を抜き出してまとめなおすための手法は，冨田らの商品検索システム RBIMD[3] や．商品価格比較サイト Excite[4] で使われている Doorenbos らのショッピングサイトの値段比べシステム shopbot[5] がある．

XWRAP などの手法を用いるには，データ元となるサイトがわかっている必要がある．しかし，サーチエンジンなどを用いて必要な情報を検索する場合，当然データ元になるサイトを前もって知ることはできない．そのため，収集できるデータの範囲が限られてしまう．それに対して本研究は，収集したデータについて共通のスキーマを生成するので，初めて閲覧するページに記載されている情報も比較対象にできる．従って，本研究で提案する機構の方がより多くのデータを扱うことができる．

3. 同属情報

3.1 同属情報の定義

同種の対象物を同じ観点から見た属性の集合を同属情報と呼ぶ．同属情報である属性の集合は，対象物を見る観点によって変化する．同属情報の属性は，複数のデータで使われている用語やデータの構造が異なっても，データ間で共通の属性ならば，値のデータ型や，値の単位，値のとりうる範囲という値の性質が共通である．

同属情報の例を挙げる．パーソナルコンピュータという対象に関する情報には，CPU 周波数，メモリ容量，値段，発売日など，様々な属性がある．これらの情報の中で，CPU 周波数やメモリ容量はパーソナルコンピュータの部品構成という観点から見た同属情報で，パーソナルコンピュータの性能，機能について比較するのに必要な同属情報である．そして，CPU の周波数についての属性値は，データ中の表記が “CPU 周波数” であったり “CPU” であったりしても，値のデータ型は数値データで，単位は Hz，値の取りうる範囲は 1 G ～ 2 G と，値の性質は同じである．値段，発売日という属性は，パーソナルコンピュータを部品構成という観点から見た同属情報には含まれないが，パーソナルコンピュータの新製品の情報という観点から見たときはこれらの属性の集合が同属情報となる．

3.2 同属情報の差異

同属情報には属性値の差異と構造の差異という二つの差異がある．

属性値の差異
属性値（数値・文字列）の大小や一致／不一致という違い

構造の差異
データのある属性が他と比べて共通／特有なものであるという違い

同属情報の二つの差異について例を挙げる．

数種類のパーソナルコンピュータという対象について，それぞれの仕様のデータはパーソナルコンピュータの部品構成という観点から見た同属情報である．製品仕様の中で，どのデータも CPU 周波数という属性を持っている．CPU 周波数の大小が属性値の差異である．また，パーソナルコンピュータのある機種は DVD ドライブを持っておりそれに関するいくつかの属性を持つが，DVD ドライブを持たない機種はそれ

らの属性を持たない．このように，ある属性の有無という違いが構造の差異である．

4. 半構造データのスキーマ生成機構

属性値の差異はデータ間で共通の属性を比較することで発見でき，構造の差異は，同じ属性が存在するかどうかを比較して発見できる．そのため，全てのデータの同じ意味の属性が明らかになっている必要がある．

そこで，比較対象のデータに共通するスキーマを生成する手法を提案する．共通スキーマを用いることで，比較するデータに共通の属性が明白になり，比較が可能になる．

共通のスキーマは，Web から得た同属情報を OEM(Object Exchange Model)[6] に変形し，異なったデータ間で共通のノードをボトムアップなアプローチで集約することで得る．

以下に，共通のスキーマを生成するための手法を述べる．

4.1 OEM への変形

OEM は識別子と値を持つオブジェクト（ノード）で構成されているツリー型のデータモデルである．OEM のノードが持つ値には，atomic な値と，complex な値がある．atomic な値は，整数，実数，文字列，画像，プログラムなどそれ以上に分割できない最小の値である．atomic な値はツリーのリーフノードが持つ．リーフノードが持つ値は一般的な属性値である．complex な値とは，0 個以上の子ノードのことである．OEM のノードを結ぶ枝にはラベルがつけられている．また，OEM のノードは複数の親ノードや循環する枝を持つことを許している．

Web から得た同属情報を OEM に変形するには，HTML タグの階層構造を利用する．HTML タグによって HTML データをツリー形式に変換する方法については，Seung-Jin らの研究[1] がある．これは，HTML タグを table 関係のタグとそれ以外のタグに分類してある．table 関係のタグで囲まれた内容はその階層構造をツリーにし，それ以外のタグには優先順位を設け，優先順位に従って囲まれた内容をツリーにする．本研究もこの方法で同属情報を OEM にする．

また，OEM の属性値が文章になる場合がある．このような時は，形態素解析などの自然言語処理を行って，十分に小さく，かつ属性の値としてデータの特徴を示すことが出来るだけのプリミティブな値に分割し，途中の単語によって枝のラベル付けをする．さらに，属性値に単位が含まれているような場合，単位と値を分離して，値を属性値，単位を属性値を持つノードに入力している枝のラベルにする．

表 1，表 2 はパーソナルコンピュータの製品仕様の一部で，外部記憶装置部分のデータを示している．これらのデータはメーカーのホームページから得られた．パーソナルコンピュータの製品仕様は，パーソナルコンピュータの性能を示す属性の集合で，メーカーごとに記述方法が異なっている半構造な同属情報である．これを上で述べたように OEM にしたものが，**図 1** である．これは表 1，表 2 の二つの同属情報を一つの OEM に集約したものである．右側の破線で囲まれた部分が表 1 の OEM で，左側の破線で囲まれた部分が表 2 の OEM である．

表 1 仕様書 1

項目	項目の情報
フロッピー	3.5 型（1.44MB ／ 720KB）× 1
内蔵ＨＤ	80GB Ultra ATA（66）
ＣＤ－RW	読み出し 32 倍速，書き込み 8 倍速

表 2 仕様書 2

項目		項目の情報
外部記憶	ＦＤＤ	3.5 インチ（1.44MB ／ 720KB）× 1
	ＨＤＤ	60GB
	ＣＤ－RW	読み出し 32 倍速，書き込み 12 倍速
ＨＤＤコントローラ		Ultra ATA（66）

4.2 OEM の問題点

取得した同属情報を OEM に変形し，比較しようとすると，次の二点の問題がある．

- データごとのラベルに使われている単語が異なっており，それには同義語や類義語，あるいはデータ独自の言葉が使われている．そのため，値の比較を行う共通の属性を単純にパスのラベルからは発見できない．
- データごとの表現や値の分類方法の違いにより，機械的に OEM に変形した場合は同じ内容でもパスが異なっていたり，あるいは逆にパスが同じでも別のオブジェクトを示している可能性がある．そのため，OEM の構造的な差異を調べることが困難である．

問題点の例を挙げる．図 1 では，記憶装置の名前が “内蔵 HD” と “HDD”，フロッピーディスクの大きさの単位が “型”，“インチ” と異なっているという単語の違いの問題がある．また，各種記憶装置の特徴を示す部分木の前に” 外部記憶” というラベルの枝があることや，HDD コントローラについての属性値が明らかに別の場所にあるというパスの違いの問題がある．

以上の問題点より，半構造データを単純に OEM に変形しただけでは，あるデータの属性のひとつの値について比較を行う前に，他のすべてのデータの属性の値を調査して，比較可能な値であるかどうか調べなければならない．そこで，収集したすべての同属情報の OEM について，共通のスキーマを生成する．共通のスキーマに基づいて指定される値は属性が明確になっているので，比較が可能である．

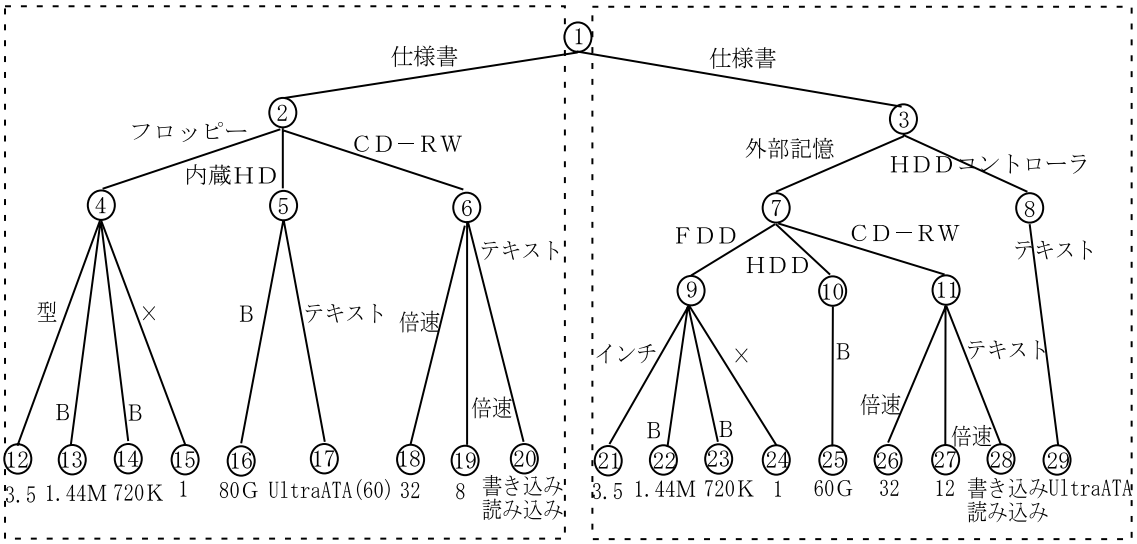


図 1　表 1, **表 2** から得られる OEM

DataGuide

パスのラベルに依存しないで異なったデータ間で共通のノードを発見する手法として，Goldman らの strong DataGuide[7] がある．strong DataGuide は複数のデータ間で共通のノードを，一つの DataGuide オブジェクトと呼ぶノードに集約していくことで OEM を要約した，共通のスキーマである．DataGuide オブジェクトと集約された共通のノードの集合の対応関係はハッシュ表に格納される．共通のスキーマとハッシュ表によって，異なったデータ間で共通の属性を指定できるようになる．

strong DataGuide はターゲットセットに基づいて OEM を要約する．ターゲットセットとは，ある OEM のノードからあるラベルの枝によって到達可能な全てのオブジェクトの識別子の集合である．strong DataGuide のオブジェクトは，ターゲットセットで示される全てのノードに対応している．

strong DataGuide の共通スキーマは以下の手順で生成される．

(1) OEM のオブジェクトからグラフ探索をして，あるパス l のターゲットセット $T(l)$ を得る．

(2) $T(l)$ が今までに発見されたことのないターゲットセットならば，$T(l)$ に含まれる全てのノードを一つの DataGuide オブジェクトに集約し，その対応関係をハッシュ表に格納する．

(3) $T(l)$ が既存のターゲットセットならば，対応する既存の DataGuide へ入力の枝のラベルに l を加える．

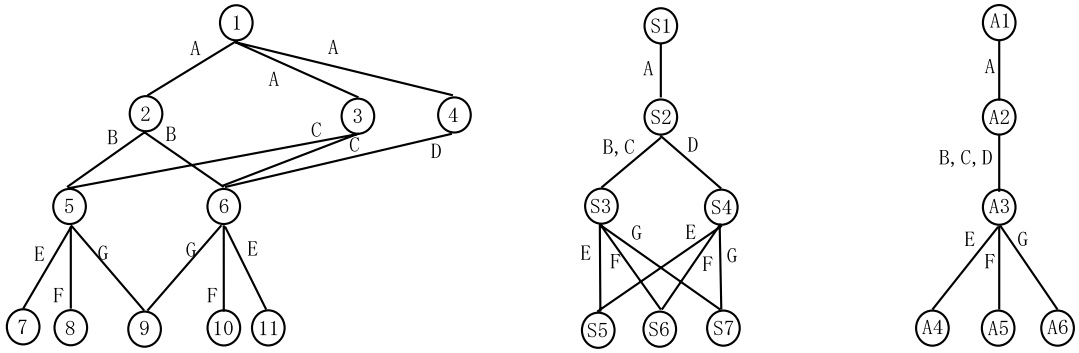
図 2(b) は (a) の strong DataGuide である．ノード S2 は，(a) のパス A でリンクされているターゲットセット 2, 3, 4 に対応する DataGUide オブジェクトである．また，ラベル B とラベル C は同じターゲットセット 5,6 を持つので，同じ枝とみなされて，(b) では一つの枝でターゲットセット 5,6 に対応する DataGuide オブジェクト S3 に入力している．

strong DataGuide 中のターゲットセットが一致するラベルは元の OEM 中のターゲットセットを共有するように定義してあるため，strong DataGuide は OEM の枝のラベルに依存しないで同じ属性の値を発見することが出来る．従って，データごとにラベルが異なっているという OEM の問題点が解決できる．

近似的 DataGuide

strong DataGuide は，ターゲットセットがすべて一致しなければ，異なったラベルの枝を一つにまとめることが出来ない．図 2(a) のノード 4 からのパス D は，ターゲットセットが 6 なので， B,C と D は別のパスとして扱われる．しかし，例えば，ノード 5 以下のノードが 5 インチのフロッピーディスク，オブジェクト 6 以下のノードが 3.5 インチのフロッピーを表現しているような場合，これら二つの部分木は本質的には同じなので，データのスキーマとしては，B,C,D が同じパスであることが望ましい．このような問題を解決するための手法として，近似的 DataGuide(Approximate DataGuide)[8] がある．

近似的 DataGuide を用いることによって，図 2(a) の OEM のパス B, C, D が類似なとき，これら



(a) OEM　　(b) storong DataGuide　　(c) Approximate DataGuide

図 2　ある OEM とその strong DataGuide と近似的 DataGuide

のパスを一つにまとめて，図 2(c) のように要約することが出来る．

パスの類似は，例えば次のように定義できる．strong DataGuide 生成時，ある二つのパス X, Y のターゲットセット T(X) と T(Y) を比較するときに，

$$|T(X) \cap T(Y)|/max(|T(X)|, |T(Y)|)$$

が一定の閾値を超えているときに，これらのターゲットセットを与えるパスは類似である．ただし，これはもっとも単純な定義であり，扱う同属情報の種類によって，構造の類似の定義を考慮する必要がある．

4.3　ボトムアップ型 DataGuide

既存の strong DataGuide や近似的 DataGuide を生成するアルゴリズムでは，図 1 で示したような OEM からスキーマを生成する場合には正確なスキーマを生成できない．なぜならば，これらのアルゴリズムは，異なったラベルの枝が同じノードに入力している場合か，同じラベルの枝が別のノードへ入力している場合に，それぞれの枝またはノードを一つにまとめ，元の OEM を要約することで全てのデータに共通のスキーマを生成している．つまり，図 1 が，ラベルが異なっても同じオブジェクトにリンクされている**図 3** かラベルの用語が統一されている**図 4** のような OEM であれば，**図 5** のように正確に要約した DataGuide が得られる．しかし，現実的には，Web ページから収集したデータは，使われている用語やデータの表現方法の違いがあり，図 3，図 4 のような OEM が得られることは稀である．

4.3.1　Bottom-up な近似的 DataGuide 生成の概要

近似的な DataGuide をボトムアップに生成していくことで，前述の問題は解決できる．

まず，同属情報は同種の対象に関して同じ観点から見たデータであるため，途中の構造や用語が異なっても，値のレベルではデータ間で共通の属性は同じ性質をもつ．ここでは，値のデータ型，単位，とりうる範囲を値の特性と定義する．特性が同じ値を探すことで，同じ属性値を発見する．

更に，ツリー形式のデータは，展開構造である．つまり，子ノードの集合が同じ中間ノード同士は同じ意味を持つといえる．異なったデータ間においても，同じ意味を持つ子ノードの集合を持つノード同士は同じ意味を持っているといえる．

このことから，以下のように共通のノードをリーフからルートまで発見していくことで，データ間に共通な部分を見つけ出すことができる．

(1)　異なったデータのリーフノードで，属性値の特性が一致するノードを共通リーフノードとする
(2)　ある共通リーフノードの集合の親である中間ノードを共通中間ノードとする
(3)　ある共通中間ノードの集合の親である中間ノードを共通中間ノードとする

発見した共通のノードを DataGuide オブジェクトとして一つに集約していけば，近似的な DataGuide となる．

このように，データ間で共通なノードを発見し，DataGuide オブジェクトに集約する操作ををリーフからルートまで繰り返すことで，同じノードへの入力や同じラベルの無い OEM でも近似的な DataGuide が生成できる．

以下に近似的 DataGuide のボトムアップな生成の例を示す．

図 1 のリーフノードに着目する．例えば，“仕様書. フロッピー. 型” というパスで指定されるノード（識別子は 12）と，“仕様書. 外部記憶. ＦＤＤ. インチ” というパスで指定されるノード（識別子は 21）は，ともにデータの型は数値型，値は 3.5 である．単位を示す直前のラベルはそれぞれ型，インチと異なっているが，データ型と価の範囲は同じである．他にも数値型の値はあるが，1.44M，720k と大きく値のとりうる範囲が異なっている．従って，この二つのノードは単

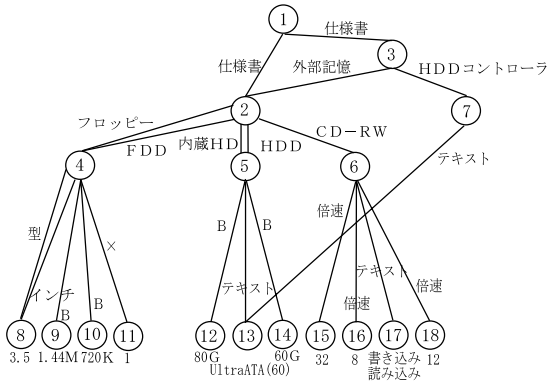


図 3 ラベルが異なっても同じオブジェクトにリンクしている OEM

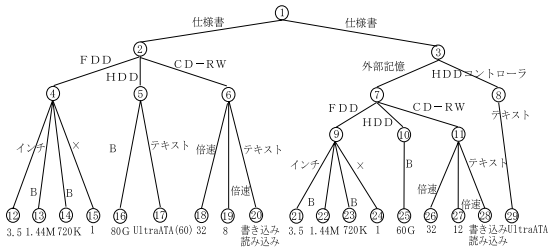


図 4 ラベルが同じ OEM

位の表記が異なっているが同じ属性について表しているリーフノードであると分かる．このように，データ型と値のとりうる範囲，値の単位といった属性値の特性が一致するリーフノードを発見し，それぞれを一つの DataGuide オブジェクトに集約したものが **図 6** である．

次に，図 6 のリーフノードにリンクしているオブジェクトに着目する．図 6 では，共通のリーフノードはすでに一つにまとめられている．子ノードの集合が同じノードは共通のノードであるとみなすので，ともに子ノードが D1, D2, D3, D4 である 4 と 9 のノードは共通のノードであるとみなす．同様に，子ノードが共通する，あるいは類似なノードをそれぞれ一つの DataGuide オブジェクトに集約したものが **図 7** である．

以下同様に子ノードが類似なノードを集約する操作をルートまで繰り返したものが **図 8** である．このように，ボトムアップにノードを集約していくことによって，近似的な DataGuide を構造，用語の違いがあっても生成することが出来る．

以上のように生成する DataGuide をボトムアップに生成する（近似的な）DataGuide ということから Bottm-up (Approximate) DataGuide (BA-DataGuide) と呼ぶこととする．

4.3.2 BA-DataGuide のメタデータ

DataGuide オブジェクトは，それぞれ集約したノー

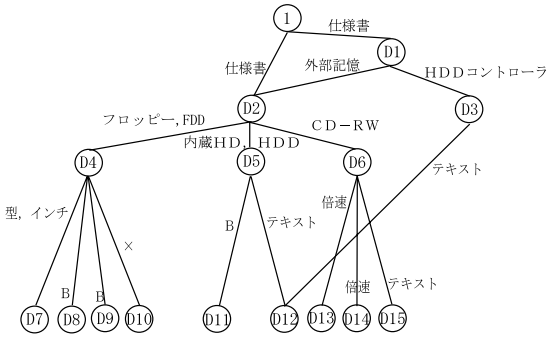


図 5 図 3 を要約した DataGuide

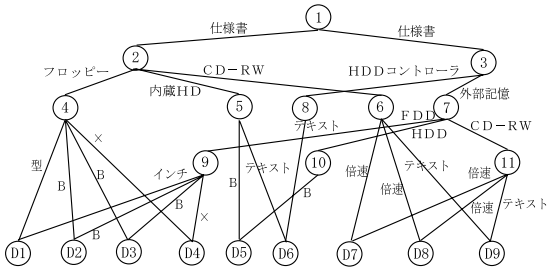


図 6 図 1 の末端のオブジェクトをまとめた OEM

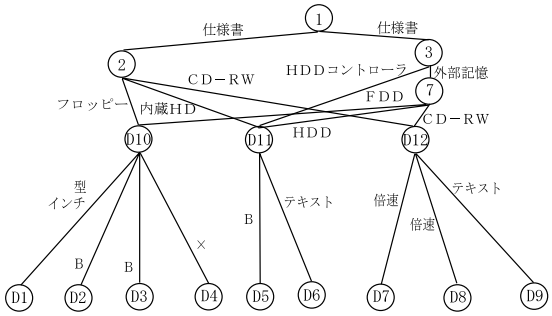


図 7 図 6 の同じオブジェクトの集合にリンクするオブジェクトをまとめた OEM

ドについてのメタデータを持つ．メタデータには以下のものがある．

- 属性値の平均値
- 属性値の取りうる値
- まとめたオブジェクトの数
- 子オブジェクトの数

これらの値は，ノードを集約するときに計算する．

メタデータによって，あるデータの属性値の一つが他と比べて高いものであるかどうかや，同じ属性をもつデータの数がどのくらいあるかなどを知ることができる．

4.3.3 BA-DataGuide 生成アルゴリズム

BA-DataGuide の生成アルゴリズムを述べる．このアルゴリズムは，比較対象の同属情報の OEM の

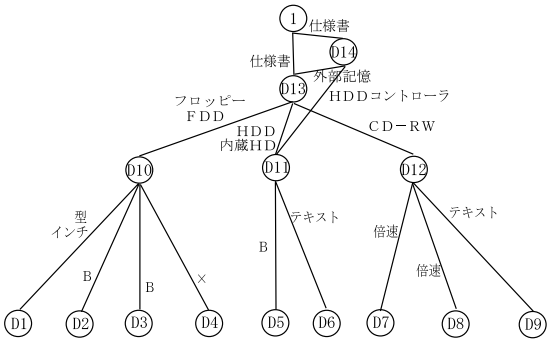


図 8 図 1 の DataGuide

ルートを一つにまとめた集約 OEM を入力とする．例えば，表 1 と表 2 の同属情報を比較するならば，入力は図 1 となる．このアルゴリズムの停止条件はルートの子ノードが全て DataGuide オブジェクトになっているときで．共通スキーマの OEM と DataGuide オブジェクトとノードの対応関係を記したテーブルを出力して停止する．ここで，

- 入力の OEM を G
- $root$ を G のルートオブジェクト
- $prop(l)$ をあるリーフノード l の属性値の特性
- $ts(c)$ をあるノード c の子ノードの集合
- $L = \{l_1, l_2, \ldots, l_m\}$ を入力のリーフノードの集合.

とする．

[Step1]

$$L_k = \{l_i \mid\ l_i, l_j \in L_k,\ l_i \neq l_j,\ prop(l_i) = prop(l_j)\}$$
$$(\text{ただし，}\ L = \bigcup_{k=1}^{n} L_k,\ L_s \not\subseteq L_t,\ 1 \leq s \leq n,\ 1 \leq t \leq n,\ s \neq t)$$

であるような集合に分割し，各 L_k に含まれるノードを一つの DataGuide オブジェクトに集約する．

DataGuide オブジェクトとノードの対応をテーブルに格納する．

G のノードを集約した OEM を共通スキーマ G' として Step2 を行う．

[Step2]

(2-1)$root$ のすべての子ノードが DataGuide オブジェクトであるならば，共通スキーマ G' とテーブルを出力して終了．

(2-2)すべての子ノードが DataGuide オブジェクトであるような $root$ でない中間ノードの集合 $C = \{c_1, c_2, \ldots, c_m\}$ に対して，

$$C_l = \{c_i \mid\ c_i, c_j \in C_l,\ c_i \neq c_j,\ ts(c_i) = ts(c_j)\}$$
$$(\text{ただし，}\ C = \bigcup_{l=1}^{n} C_l,\ C_s \not\subseteq C_t,\ 1 \leq s \leq n,\ 1 \leq t \leq n,\ s \neq t)$$

であるような集合に分割し，各 C_l に含まれるノードを一つの DataGuide オブジェクトに集約する．

DataGuide オブジェクトとノードの対応をテーブルに格納する．

ノードを集約した OEM を共通スキーマ G' として Step2 を行う．

5. 同属情報の差異の可視化

ユーザにデータのインスタンスと共通のスキーマを OEM で提示する．データのインスタンスによってそのデータの特徴を，共通のスキーマによって全体の特徴を知ることが出来る．

提示する OEM のノードは着色されている．共通／特有のノードというような種類の違いは色の違いで表現され，どのくらいの数のノードと共通しているのか，というような程度の違いは色の濃淡で表現されている．

ツリーが大きくなる場合は，一定の深さより深いところにある部分木は表示されない．これらの部分木は，ユーザは任意に展開してデータの詳細を得ることが出来るようになっている．

5.1 インスタンスの表示

データのインスタンスは，以下のように色づけされる．

- 共通のノードが多数のノードと少数のノードには別の色をつける．ただし，子ノードが格納状態のノードは，子ノードに一つでも少数のノードがあれば，多数のノードであっても少数のノードの色をつける．
- リーフノードで数値データなどの大小が比較できる値を持っている場合，平均値と値の差に比例して色の濃さが変わる．平均よりも高い数値ならば色が濃くなり，低い数値ならば色が薄くなる．

インスタンスの表示例を**図 9** に示す．

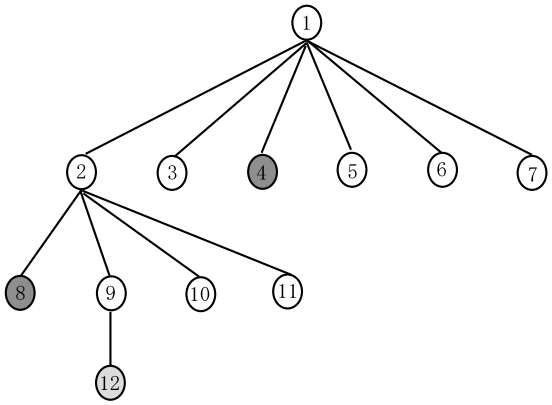


図 9 インスタンス表示例

この図では，オブジェクト 1，2，3，5，6，7，9，10，11 が多数のデータに存在するノードである．ま

た，オブジェクト 4，8 は，それ自体が少数のデータだけがもつノードであるか，子ノードに希少なノードをもつ．

オブジェクト 12 はリーフノードで，色が濃くなっているので，平均値よりも高い数値を持っていることを示している．

5.2 共通スキーマの表示

共通スキーマは，以下のように色づけされる．

- 中間ノードとリーフノードには別の色がつけられる．
- 中間ノードは，インスタンス全体の中で対応するノードを持っているインスタンスの割合に比例して色の濃さが変わる．割合が高い部分は色を淡くし，低い部分は色を濃くする．
- リーフノードはデータの値の分散に比例して色の濃さが変わる．分散が小さいノードは色を淡くし，大きいノードは色を濃くする．

あるスキーマの表示例を**図 10** に示す．

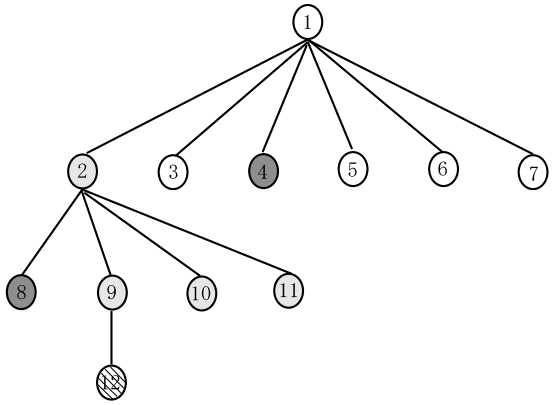


図 10 スキーマ表示例

オブジェクト 1，3，5，6，7 は無着色なので，全てのインスタンスがこれらのオブジェクトに対応するノードを持つことがわかる．また，オブジェクト 2，9，10，11 は淡い色がついているので，これらのオブジェクトに対応するノードを持つインスタンスが多数存在することを示している．オブジェクト 4，8 は色がかなり濃くなっているので，これらのオブジェクトに対応するノードを持つインスタンスが非常に少ないことがわかる．リーフノードであるオブジェクト 12 は，色が濃くなっているので，属性値の分散が大きいことがわかる．

ユーザに提示する共通スキーマは，色付きの BA-DataGuide である．従って，これを Colored Bottom-up (Approximate) DataGuide (CBA-DataGuide) と呼ぶ．

6. おわりに

本論文では，現行の Web システムでは，ユーザ自身が手動で複数のページを閲覧してデータを比較しなければならないという問題を解決するための機構を提案した．この機構は，半構造データの共通のスキーマを生成し，データを比較してそれぞれの共通部分，例外部分を発見する．そして，これらの差異をノードの色の違いによって表現したツリーをユーザに提示する．

提案した機構は，多くのメーカーが製造していて個々の違いが分かりにくいような商品の違い，例えばパーソナルコンピュータの性能の違いを発見するようなアプリケーションのコアエンジンにするというような応用が考えられる．

謝辞 本研究の一部は，文部省科学研究費基盤 (C)「分散型ハイパーメディアからの構造発見とアクセス管理」（課題番号 12680416）及び，日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア／コンテンツの工事処理の研究」（プロジェクト番号 JSPS-RFTF97P00501）によっております．ここに記して謝意を表すものとします．

参考文献

1) Seung-Jin Lim, Yiu-Kai Ng: “An Automated Change-Detection Algorithm for HTML Documents Based on Semantic Hierarchies”,in Proceedings of the 17th International Conference on Data Engineering (ICDE’01), pp. 303-312, Heidelberg, Germany, April 2-6, 2001

2) Ling Liu, Calton Pu, Wei Han: “XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources”,in Proceedings of the 17th International Conference on Data Engineering (ICDE 2000), pp. 611-621,April, 2000, San Diego, CA (IEEE CS Press)

3) 富田一郎，手塚祐一，山本修一郎，長岡満夫: HTML 文書からの商品情報抽出方式の提案，情報処理学会第 56 回全国大会講演論文集 (3)，pp.79-80，1998 年 3 月

4) excite.com: http://www.jango.excite.com/

5) Robert D. Doorenbos, Oren Etzioni, Daniel S. Weld: “A Scalable Comparison-Shopping Agent for the World-Wide Web”, Proceeding of the First International Conference on Autonomous Agents,1997

6) S. Abiteboul, D. Quass, J. McHuge, J. Wiener: “The Lorel Query Language for Semistructured Data”, Jornal of Digital Libraries,1(1), November, 1996

7) Roy Goldman, Jennifer Widom: “DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases”, Proceedings of the Twenty-Third International Conference on Very Large DataBases, pp. 436-445, Athens, Greece, August, 1997

8) Roy Goldman, Jennifer Widom: “Approximate DataGuides”, Technical report, Stanford University, 1998