

## 4C-4

構造化文書(ODA)の処理:  
G木を用いた文書割付け処理

齊藤和雄、林 直樹

富士ゼロックス(株) システム技術研究所

## 1. はじめに

ODA<sup>[1]</sup>では、割付けのための制約が文法規則によって記述されるため、割付け処理の効率化には従来とは異なった処理を必要とする。今回、著者等は効率的に構造化文書の割付けを実現するためにG木と呼ぶ処理モデルを新たに考案し、それを用いたODA文書割付け処理を実現したので報告する。

## 2. 割付け処理の効率化

ODAでは割付けに対する制約は二つの形式で記述される。一つはジェネリック割付け構造内に文法規則によって記述されるもので、オブジェクトクラスごとに下部構造の制約が構造生成式と呼ばれる式を用いて記述される。構造生成式とは6種類の構造生成子とオブジェクトクラスを組み合わせて表現される式で、構造生成子には、決まった順序を示すSEQ、複数の選択肢の内のどれかを示すCHO、あるかないかの選択を示すOPT、1個以上の繰り返しを示すREPなどがある。SEQ以外は下部構造に複数の可能性があることを意味している。

もう一つは割付け指示属性と呼ばれ、論理オブジェクトに付随し、それを割付ける割付けオブジェクトクラスを指定している。論理構造には順序があり、それぞれの内容量は実際に割付けを行なってみないと、割付けオブジェクトの領域計算を行なうことができない。そのため、割付け指示属性は論理構造の順序にしたがって順次満たしていかなければならない。つまり、選ばれた構造が割付け可能であるかは、実際に割付けを行ってみないと分からない。従って、生成された構造が誤りであることが分かった時は、他の構造の生成を行なうというバックトラック処理を行う必要がある。

ODAの割付け処理には、左導出により文法規則から一つの構造の生成を行う方法<sup>[2][3]</sup>があるが、この方法は比較的単純なアルゴリズムで割付け処理が構成できる反面、構造を生成する際に割付け指示属性の指定を考慮しないため、不要な構造生成が起こる可能性が高い。不要な構造生成はバックトラックを引き起こすため割付け処理の処理時間を増加させることになる。

従って、割付け構造の生成時に割付け指示属性による指定を考慮すれば、不要なバックトラックの発生

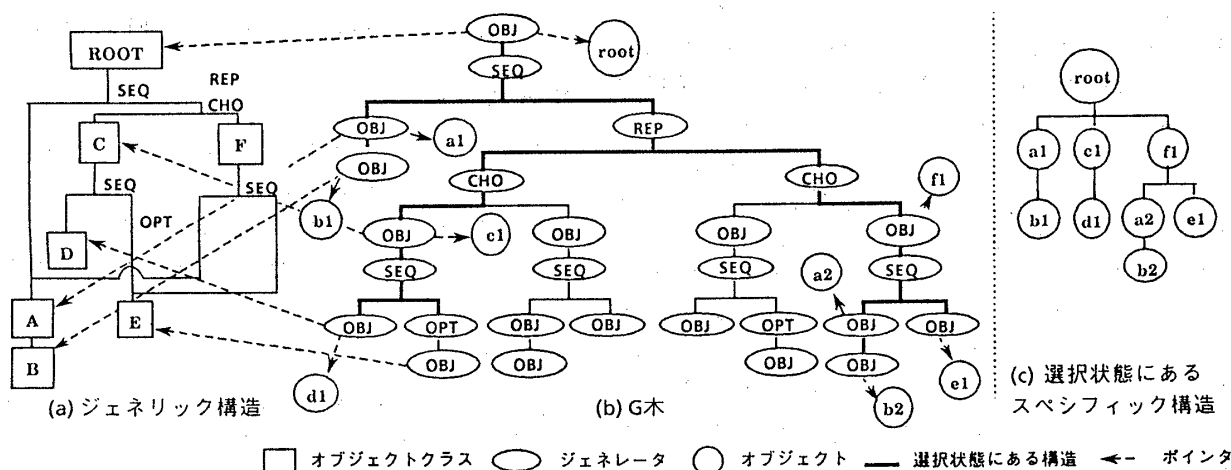


図1 G木の構成

"Processing of Structured Documents (ODA): A Document Layout Process Using G-Tree"

Kazuo SAITO and Naoki HAYASHI

System Technology Research Lab., Fuji Xerox Co., Ltd.

を押さえることが可能になる。そのためには、割付け指示属性によって指定された割付けオブジェクトクラスのインスタンスを割付け構造に付加するには、どのようにすればよいのかを割付け処理が知ることができる機構が必要とされる。

### 3. G木

G木はその機構の解の一つとして考案したもので、割付けオブジェクトと操作可能な場所の両者をノードとする木構造で表現される。G木の例を図1に示す。操作可能な場所とは割付け構造中で複数の可能性を持つ場所であり、構造生成子のOPT, REP, CHO, AGG, OPT-REPで表現される場所である。割付けオブジェクトを表現するノードと操作可能な場所を表現するノードが直接に対応づいているので、その操作可能な場所を操作した時の割付け構造に対する影響を直接に知ることができる(以下、G木のノードを特にジェネレータと呼ぶ)。

さらに、ジェネレータ自身がその操作可能な場所の過去および現在の選択履歴、選択可能な選択肢に関する情報を保持している。

ジェネリック構造内では、オブジェクトクラスは複数の箇所から参照される可能性を持っている。しかし、選択履歴は参照されている場所毎に異なる。そこで、G木ではそのようなオブジェクトクラスは参照されている場所毎に異なったジェネレータで表現している。

図中のOBJというジェネレータは割付けオブジェクトを表現している。これらは必ずジェネリック構造中のクラスを指し示しており(図中では一部省略している)選択状態にあるOBJジェネレータは自分が保持するクラスのインスタンスも保持している。図中の太い実線で示される部分が選択状態にある構造であり、この構造のみを参照することで、仮想的に図1(c)に示す構造を表現している。

G木では構造を変更することは選択状態を切替える操作のことである。変更可能な場所であるジェネレータに対しては、基本操作と呼ぶ現在選択している構造とは別の構造を選択させる命令が定義されている。この基本操作は規則が意味する構造の範囲を逸脱しないように定義される。

割付け処理はG木を用いると、基本的には割付け指示属性によって指定された割付けオブジェクトクラスを保持するOBJタイプのジェネレータを探しだし、それがスペシフィック構造に含まれるようにその上位の変更可能なジェネレータに対して変更を加えるという操作で進められることになる。

### 4. 実験による効率比較

G木に基づく割付け処理を実際に試作し、それを参考文献[2]で示す従来の方式による割付け処理と比較を行なってみた。比較対象は、選択肢を増加させるのに伴って、必要とされる選択肢を選ぶのに要する時間の変化比である。図2に選択肢の数が1つの時の処理時間を基準とした処理時間の変化比を示す。図に示されるように、従来のシステムでは選択肢が増加すると割付け処理の効率が大きく低下するのに対し、G木に基づくシステムの場合は効率の低下は少ない。これは、選択肢が増えても不要な構造の選択は行わず、目的とするジェネレータの探索のためのコストが増加するだけであることによる。

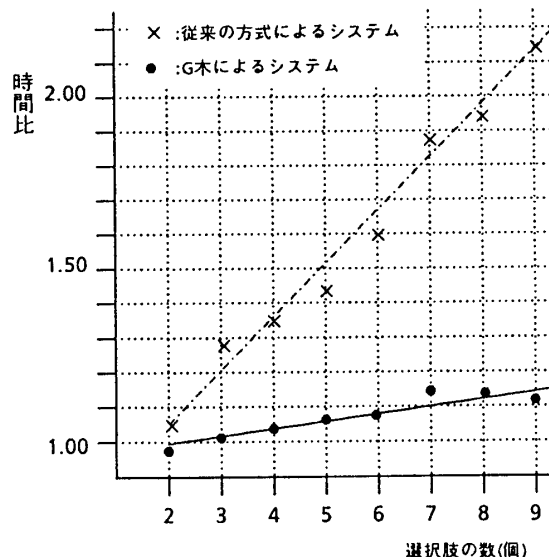


図2 選択肢の増加に伴う処理時間の変化比

### 5. おわりに

G木によって不要な構造生成を起こしにくい割付け処理を実現できることを示した。G木には大きな記憶領域を必要とするという課題が残されているが、これは一部の構造の共有によって、解決できるものと考えている。

### 参考文献

- [1] ISO(ed.): ISO8613: Information Processing - Text and Office Systems - Open Document Architecture(ODA) and Interchange Format(1987).
- [2] 林、斎藤、石田、村田: ODA文書処理システムの試作(3)-割付け処理-, 情報処理学会第37回全国大会論文集、pp1857-1858(1988).
- [3] 村上、山口、松平、上原、鍵政: ODAに基づいた文書割付け処理の実現方式(1)-再試行の課題-, 情報処理学会第40回全国大会論文集、pp.598-599(1990).