構造化文書における文法によって表現された割り付け指定の処理手法

齊藤和雄† 林 直樹†

本稿では，構造化文書の割り付け処理における，文法規則によって記述される割り付け指定の処理手法に関して報告する．構造化文書においては，割り付けのための制約が文法規則によって記述されるので，割り付け処理の効率化のためには，従来と異なった処理が必要とされる．本論文では，まず，割り付けに対する制約が文法規則によって表現されているということが，割り付け処理に対してどのような処理を要求するのかを整理することで，割り付け処理実現の課題を明らかにする．効率的な割り付け処理は任意の構造の生成・破棄を行わねばならないので，そのためには任意の時点における構造に対してどのような変更を行うことができるのかを管理できる機構が必要となることを述べる．次にこの機構の一つの解である“G木”と呼ぶ処理モデルを提案する．G木は一種の木構造で表現され，その最も大きな特徴は割り付けのための文法規則とそれが生成する構造を一つに統合したことである．さらに，実際のシステムとしての性能を調べるため，構造化文書の国際規格として知られている ODA (Open Document Architecture) に適用し，G木に基づく文書割り付け処理を実現した．その結果，選択肢の多い場合の有効性を確かめることができた．

New Processing Method for Grammatical Layout Specifications in Structured Documents

KAZUO SAITO† and NAOKI HAYASHI†

In this paper we describe an efficient method for layout processing of structured documents which have layout constraints represented by a grammar. The grammar represents all possibilities of the layout structures. Hence, the layout process has to derive one layout structure from the given grammar rules. The key of this method is introducing a new data representation which is called “G-Tree”. The G-Tree represents all layout structures which satisfy the layout constraints, and keeps historical information of the derivation. It helps the layout process to know where and how the layout structures can be modified. In order to verify its effectiveness, we implemented a prototype ODA (Open Document Architecture) layoutor using G-Tree. Through this experiment, its efficiency was confirmed. Especially it is effective when the layout specifications represent many possibilities.

1. はじめに

計算機ハードウェアと入出力装置の発達は，計算機を用いた文書処理をより高度なものにしてきた．特に近年，電子文書を表示あるいは印刷したものが，活版印刷の品質に近づきつつある．これは，非常に高精細の出力装置が利用可能になったことだけでなく，割り付け処理技術[1),2)]が発達したことによる．

割り付け処理とは，電子文書の内容を，出力装置が利用可能なページレイアウト情報に変換する処理を言う．この変換の際に，どのようなレイアウトにするかを記述した割り付け指定が割り付け処理システムに与えられる[3)]．割り付け処理により，計算機が割り付け指定に応じて自動的に文字や図のページ上の配置を決めるので，美しいレイアウトをもつ文書をユーザは手軽に得られるようになった．例えば，LaTeX[4)]ソースファイルから DVI 形式のファイルを生成することは割り付け処理であり，LaTeX の command は割り付け指定にあたる．

現代の多くの割り付け処理システムでは，処理の入出力情報として，構造化文書[5)]を用いている．構造化文書では，文書は章や見出しといった要素から構成される．

割り付け指定は，文書の構成要素に対して規定される．この指定方法により，ユーザがレイアウトを把握しやすくなり，レイアウト変更も容易となる．例えば，文書内容中の一部の文字だけ大きさと字体が変わっているより，見出しという要素があってそれに文

† 富士ゼロックス(株)システム・コミュニケーション研究所
Systems & Communications Lab., Fuji Xerox Co., Ltd.

字の大きさと字体が指定されている方が，ユーザがどのようなレイアウトを考えているかわかりやすい．また，すべての見出しの字体を変更する，という操作もユーザに提供しやすい．

構成要素に規定される割り付け指定は，大きく二つの種類に分類できる．

一つは，章や節といった文書の論理の構成要素が，どのように割り付けられるかを指定するものである．以下，この種の割り付け指定をタグ表現と呼ぶ．タグ表現には，文字の大きさといった内容の表示法に対する指定と，改ページや特定ページへのレイアウトといった割り付け構造の構成要素を制限する指定がある．SGML[6]における tag，Grif[7]における presentation rules，LaTeX における sectioning commands，J Star[8]におけるテキストプロパティー，ODA[9]における layout style はタグ表現にあたる．

もう一つは，ページや枠といった文書のレイアウトの構成要素が，どのようなページだてを形成するかを示すものである．カラム数やその大きさ，あるいはページ番号の位置といったページ固有の割り付け情報はこの種類の割り付け指定により表される．以下，この種の割り付け指定をページ雛型表現と呼ぶ．

ページ雛型表現は，さらに二つに分類できる．一つは，ユーザが指定可能なパラメータをシステムがあらかじめ用意しているものである．この方式では，生成し得るレイアウトは限定されるが，ユーザはパラメータの値を設定するだけでよいという利点がある．この例として，LaTeX の documentstyle command や pagestyle command，J Star のページ書式プロパティーがあげられる．

もう一つは，ページやフレームといった割り付けの構成要素の出現のしかたを文法規則で記述したものである．これは前者の方式を拡張したものと言える．この方式では，文法規則をユーザが定義する必要があるが，さまざまなレイアウトフォームをユーザが規定できるという利点がある．また，個々の割り付け処理システムに依存したパラメータを用いないため，異機種間での文書交換に向いている．この表現形式は，Interscript[10]によって初めて提案され，電子文書交換の国際規格である ODA にその考えが採用されている．

タグ表現とページ雛型表現の両方を用いることで，非常に高度なレイアウトを自動生成できるようになる．これを可能とするシステムの多くはページ雛型表現としてパラメータを用意するものであり，文法規則を用いるものはまだ少数の報告[11),12)]しかない．これらの報告例は，実際に動作するシステムを実現した点で先駆的なものであるが，ここで提案された文法規則から構造を導出する方式は効率的な割り付け処理を実現するには不向きといえる．

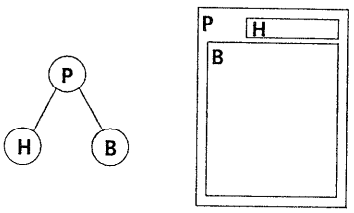
本論文では，ページ雛型表現として文法規則を用いた割り付け処理の効率的な実現手法について述べる．まず，現在までの報告例で提案された方式の持つ問題点と，より効率的な割り付け処理実現のために要求される機能を明らかにする．次に，その機能を備えた新たなデータモデルを提案する．さらに，このデータモデルに基づく割り付け処理を試作した結果から，このモデルの有効性について述べる．

2. 割り付け処理の効率化における課題

2.1 文法を用いた割り付け指定の特徴

ページ雛型表現として用いられる文法は，割り付け構造のある構成要素に対して，その一階層下にどのような構成要素の並びをとりうるかを記述するものである．階層の上下関係は，基本的に，上位の構成要素の表示上の領域が下位のものの領域を内包する*，ということを示している．構造の最下層の構成要素が持つ領域には，文書の内容が割り付けられる．

例えば，ページに相当する構成要素P，ページ見出しに相当する構成要素H，本文領域に相当する構成要素Bが定義されているとする．Pに対して文法規則として HB の並びが与えられるとすれば，PとHとBは**図1**(a)に示す木構造を形成し，図1(b)に示され



(a) 導出される構造 (b) 構造の持つ矩形領域
(a) Derived layout structure. (b) Rectangular areas represented by the layout structure.

図 1 割り付け構造と矩形領域
Fig. 1 Layout structure and rectangular areas.

* 当然ながら，ページより上位の構造については当てはまらない．また，ページから下位においても，割り付け処理によって各構成要素の領域が計算される場合にはこの原則が成り立つが，ユーザが固定の値としてそれらを指定する場合にはこの限りではない．

るような領域をもつ．Hが最下層の要素であれば，この領域にページ見出しの内容が割り付けられる．

この文法では，ある構成要素が繰り返し現れ得ること（繰り返し）や，いくつかの構成要素のうち一つだけが現れ得ること（選択），構成要素が指定した並びで現れ得ること（並び）が記述できる．「選択」，「繰り返し」によって，一つの文法規則から複数の異なった割り付け構造が導出できる．つまり，ある文法規則が異なった論理構造の割り付けに適用できる．

構成要素の持つ領域は，文法規則の「選択」，「繰り返し」に対応して，可変の位置または寸法を持てる．この場合，導出時に割り付け処理が計算を行い，適切な固定の値を構成要素の位置または寸法として設定する．可変位置の場合，領域が内包関係を満たしながら一方向に順番に並ぶように，構成要素の位置が設定される．その方向はその構成要素の上位の要素に指定される．また可変寸法の場合，構成要素の寸法は下位の要素が持つ寸法を加算した値になる．可変の位置と寸法を用いることで，図表など内容の大きさに応じて，適切なページレイアウトをもった割り付け構造を導出できるようになる．

文法規則を用いる割り付け指定には，ストリームという概念が導入されている．これは，割り付け構造の構成要素の繋がりが文法により明示的に記述できることに対応して，より複雑なページレイアウトを導出可能とするために導入された．

ストリームとは，論理構造の構成要素の集合である．あるストリームにおいては，各構成要素の論理的な順序と，それらが割り付けられた割り付け要素のページレイアウト上の順序とが一致する．例えば，論理の構成要素として「段落」があるとき，複数の段落はその論理的な順序とページレイアウト上の順序が一致するので，これらは一つのストリームとみなせる．

一つの文書には複数のストリームが存在しうる．例えば，一つの文書に本文段落と本文から参照される図がある場合を考える．複数の本文段落はその論理的な順序にしたがって割り付けられ，図も同様である．この順序関係は，本文段落の集合と図の集合では各々一意に保たれる．しかしながら，本文段落と図の割り付けの位置関係は一意に決まらない．雑誌のように図が本文から参照されたページに現れる場合もあれば，特許のように図が文書の後ろにまとめて割り付けられる場合もある．この例では，本文段落と図は異なるストリームとみなせる．複数のストリームの例として他にも，本文と注，二ケ国語対訳文書における各言語，本文記事とサイドストーリーを示すコラム記事がある．

個々のストリームがどのように割り付けられるかは，タグ表現を用いて指定する．より具体的には，論理要素がどのストリームに属しているかを記述し，割り付け要素に対してどのストリームが割り付け可能かを記述する．この指定によって，ストリームごとに適切な割り付け要素が割り付け処理により選択される．

2.2 割り付け処理と従来の実現方式の問題点

割り付け処理は，与えられた論理構造と文法規則群から，一つの割り付け構造を導出する．導出される構造は，論理構造が持つ内容とタグ表現から制約を受ける．

文書に存在するすべての内容は，割り付け構造の最下層の要素が持つ，有限の大きさの領域に収まる必要がある．各構成要素の持つ領域は寸法と位置に関して可変値をとりうるが，少なくともページ寸法はＡ４というように固定値であり，領域はページに内包される位置と寸法を持つ必要がある．内容が領域に収まりきれない場合，導出された構造は，その論理構造を割り付けるには不適となる．

タグ表現は，ある論理の構成要素の持つ内容が，特定のフレームやページに割り付けられるために用いられる．例えば，章のタイトルという構成要素はページの先頭に割り付けられる，図表は全段抜きの枠内に割り付けられる，といった用途がある．また，文書に複数のストリームが存在する場合，ストリームごとに適切な割り付け要素に割り付けられるために必ず用いられる．導出された構造にはタグ表現で規定された割り付け要素が含まれ，その割り付け要素にはタグ表現が付随する論理要素の持つ内容が割り付けられねばならない．

割り付け処理は，以上述べた制約を満たす割り付け構造を導出する．内容が収まるかについては割り付け要素の持つ領域の寸法に依存するため，実際に内容を割り付けてみてはじめて導出した構造が適切であるかどうかがわかる．したがって割り付け処理では，導出した構造が適切ではないとわかったときに，他の構造を導出し再割り付けする処理が必要となる．この処理はバックトラック処理と呼ばれる．

現在までに報告されている実現方式は，固定の導出順序にしたがって逐次的に導出を行って割り付け構造を得る．文法規則に「選択」「繰り返し」がある場合，割り付け処理はあらかじめ決まった順序にしたがって

選択肢を選ぶ．割り付け処理の過程で導出した構造が誤りであるとわかった時点で，バックトラック処理を行う．従来の方式でのバックトラック処理は，その時点までで最後に選んだ選択肢を，導出順序上次にあたる選択肢で置き換える．そして新たな選択肢にあわせて割り付け構造を変更する．したがって従来の方式は，概念的には，割り付け処理の過程で選択した選択肢の情報を一つのスタックに積み上げ，バックトラック処理時にスタックの上に積まれたものから順番に他の選択肢を試すものといえる．

従来の方式は，固定の順序にしたがって構造を導出するため，1) 導出する構成要素の選択，2) 構成要素の付加／削除を行う木構造上の場所の選択に問題がある．

1) 構成要素の選択

割り付け処理の過程で，ある論理要素に対してタグ表現により割り付け要素が制限されているとする．従来の方式では，選択肢を選ぶ順序が固定であるため，その割り付け要素を直接選んで導出することはできない．スタックに積まれた選択肢から代替の選択肢を順次得て構造を導出することを，必要な要素が得られるまで繰り返す．このため，不要な構造の導出とバックトラックが頻繁に起こる．

不要な構造の導出は，割り付け処理の処理量を増加させる．まず，割り付けの構成要素の導出と破棄により，割り付け構造の管理のための処理量が増える．さらに，領域の位置と寸法が可変の要素が破棄された要素を内包していれば，位置と寸法の再設定と，この領域に割り付けられていた内容の再割り付けが発生する．可変値を持った領域が多重の内包関係を持つと，値の再設定と再割り付けが広い範囲に及ぶ．多段組や，本文が図を囲むように割り付けられるなど，複雑なページレイアウトは多重の内包関係を持つ傾向があるため，値の再設定と再割り付けが頻繁に起こるのは望ましくない．

2) 付加／削除を行う場所の選択

文書に複数のストリームがある場合，割り付け処理はストリームごとに異なった割り付け要素に割り付ける必要が生じる．

例えば，本文段落と参考文献の二つのストリームが存在する場合を考える．文献への参照を含む本文段落と，参照された参考文献は，論理的な順序は連続しているとする．そして，本文段落は先頭のページから割り付け，参考文献は後ろのページに割り付けるとする．本文段落と参考文献が割り付けられるページである構成要素を各々 Pb，Pr とすると，割り付け構造の根である構成要素 Z に対して，文法規則は Pb の繰り返しと Pr の繰り返しとの並びで与えられる．本文段落と参考文献の内容量に応じて，Pb と Pr は独立に増えることになる．

従来の方式では，概念的にはスタックを利用して処理を進めるため，木構造のどの場所に新たな要素を付加するかはただ一点で管理されている．割り付け要素の付加または削除が木構造上で複数の異なった場所で起こる場合，単純にスタック操作を行えばその時点までの導出履歴が失われる．前の例で本文段落，参考文献，本文段落の順序で論理要素を割り付けるとすれば，2番目の本文段落を割り付けるにはスタックで下に積まれた Pb の繰り返しの選択履歴が必要である．しかし，単純にスタック操作をすれば，スタックの上に積まれていた Pr の繰り返しの選択履歴が失われる．したがって何らかの特殊なスタック操作が必要と考えられるが，従来の報告ではこのような場合の選択履歴の管理についてはふれられていない．

2.3 割り付け処理を効率化する機構

割り付け処理システムが適切な選択肢を選んで構造を導出できるようになれば，より効率的な割り付け処理が実現できる．このためには，ある割り付けの構成要素を必要とする場合に，どの選択肢を選べばよいかを割り付け処理システムが把握できる機構が必要となる．以下，この機構に要求される機能について述べる．

割り付け構造の選択肢を表現する文法規則は，「繰り返し」と「選択」の二つである．「繰り返し」は，ある構成要素が付加あるいは削除ができる場所を表現しており，「選択」は，ある構成要素を選び直して代替要素に入れ換えられる場所を表現している．これらは共に変更を行うことのできる場所として扱うことが可能である．

選択肢を選ぶために，割り付け処理は任意にこれらの場所を操作する．したがって，機構は場所毎の選択履歴を保持する必要がある．選択履歴として必要な情報は，過去に選んだことのある選択肢である．「選択」における選択履歴はどの選択肢を選んだことがあるかであり，「繰り返し」における選択履歴は削除した，あるいは付加したという操作の履歴である．

また，操作可能な場所には階層的な依存関係が存在する．なぜなら割り付け構造は木構造であり，ある

構成要素の存在はその上位の構成要素の存在に依存する．つまり，ある場所の選択肢が選べるためには，その場所が存在するように上位の場所の選択肢が選ばれていることが前提になる．

したがって，この機構は構成要素の階層的な依存関係を取り扱える必要がある．個々の文法規則はある構成要素の直下の構造に関してのみ記述するものであるから，多階層に渡って選びうる構造を把握できることが重要となる．

以上まとめると，1）操作可能な場所が明確に保持され，その操作可能な場所毎に，過去の選択履歴と選択可能な選択肢が保持されていること，2）構成要素の階層的な関係，操作可能な場所の階層的な関係が表現されていることが機構に必要である．

3. G 木

G木は前章で述べた課題を解決し，効率的な割り付け処理を実現するために，著者等が新たに考案した機構である．

G木は割り付け構造の構成要素，操作可能な場所，依存関係，を表現する三種のノードからなる木構造である．各ノードは下部構造に関する情報を保持する．ノードのうち，特に操作可能なノードをジェネレータと呼ぶ．ジェネレータは下部構造の情報に加え，過去の選択履歴，現在選択している選択肢に関する情報も保持している．

各ジェネレータには，規則が意味する構造の範囲を逸脱しないよう下部構造を変更するための「基本操作」と呼ぶ命令が定義されている．G木上での構造の変更は，すべて各ジェネレータに対して「基本操作」を適用することで行われる．

G木が表現できる文法の能力は，木構造を生成する文法である“T-grammar”[13]，あるいは“Bigrammaire Reguliere”[14]と呼ばれているものと同等である．厳密に言えば，現在のG木は再帰構造を扱えないので，これらの文法の再帰を除いた範囲を扱うことができる．

G木の特徴は以下のようにまとめられる．

- 割り付け構造に対する操作可能な場所がジェネレータとして表現されているので，どこを操作できるかが明解（2.3節の課題1）に対応）
- 操作可能な場所間の依存関係，すなわち，ある操作可能な場所はその上位の操作可能な場所の状態に依存するということを，木構造によって表現しているので，操作そのものの依存関係を容易に知ることが可能

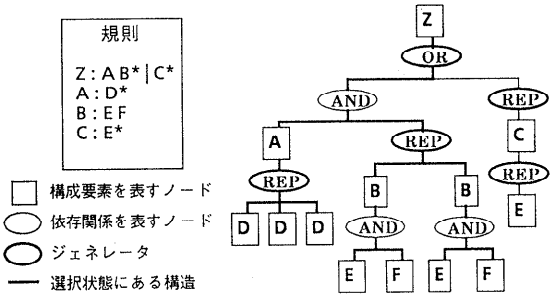


図 2 G木の構成
Fig. 2 G-Tree structure.

（2.3節の課題2）に対応）

- 操作可能な場所自身の過去の選択履歴および選択可能な選択肢に関する情報を保持できるので，選択履歴を局所化，構造化して管理することが可能
- 構成要素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関連づけられるので，その操作可能な場所を操作した時の構成要素の状態，すなわち割り付け構造を容易に導くことが可能

図2に例を示す．この例では，「選択」を意味するORタイプと0個以上の「繰り返し」を意味するREPタイプの二つの種類のジェネレータがある．ANDを意味するノードは構造の依存関係を表現している．また，文法規則上は構成要素Bは二箇所から参照されているが，選択履歴等を別々に管理するため，G木では別の構造として扱っている．

図中の太い実線で示される部分は選択状態にある構造である．この構造のみを参照することによって，割り付け構造を得ることができる．この例の場合，ジェネレータの基本操作には，ORタイプの場合は「他の選択肢を選択せよ」や「ある指定した選択肢を選択せよ」，REPタイプの場合は「下部構造を一つ増やせ」などが定義されている．

ジェネレータおよびノードのデータ構造の例を示す．

```
Gen1 : type=OR
  elements=(Node2 Gen2)
  selected=()
  subordinate=Node2
Node2 : type=AND
  subordinates=(NodeA Gen3)
Gen3 : type=REP
  element=NodeB
  subordinates=(NodeB NodeB')
```

ノードの名前が Gen で始まるものがジェネレータである．Gen1，Gen3 はそれぞれ図2のZの下部のOR，B の上位の REP ジェネレータに相当する．Node2 は Z の下部構造の左側の AND ノードである．

各ノードは幾つかのデータフィールドから成っている．type はノードの種類を意味する．elements は下部構造として取り得る選択肢を保持する．OR の場合は複数であるからリストで保持される．selected はこの選択肢リストのうち，過去に選んだことのある要素のリストである．subordinate は現在選択している選択肢を保持している．REP タイプでは一つの要素の繰り返しを表現するので，element は一つの要素から成る．基本操作によって，下部構造を表現する subordinates に，この element の要素が複写され，追加されていく．“’”が付いたノード名は，それが複写されたものであることを意味している．

4. G木の ODA への適用

本章ではG木を構造化文書の国際規格として知られる ODA に適用した例について述べる．

4.1 Open Document Architecture

ODA では一つの文書を大きく分けて，論理構造と割り付け構造の二つの構造で表現する．そして，そのそれぞれについて構造的な性質を制約する共通構造と，特定の文書を表現する特定構造の計四つの構造で一つの文書を表現している．

割り付けに対する文法規則による制約は，共通割り付け構造内に記述される．共通割り付け構造はオブジェクトクラスの組合せで表現され，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クラスオブジェクト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，任意の順序を示す AGG，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，0個以上の繰り返しを示す OPT-REP がある．SEQ を除く構造生成子には下部構造に複数の可能性があることを示している．

一方，ODA におけるタグ表現は割り付け指示属性と呼ばれる．これは論理構造側に記述され，論理オブジェクトをどのように割り付けるのかを指定する．割り付け指示属性は，特定割り付け構造を選択する際の制約として機能する．

ODA ではストリームの概念は割り付け指示属性の一種である “layout category” によって実現されている．これは基本論理オブジェクトに指定され，割り付けオブジェクトの属性 “permitted categories” によって，割り付け先の指定が行われる．

4.2 ODA のためのG木の構成

ODA では，任意性を持つ構造生成子をジェネレータとして扱うことができる．すなわち，OPT，REP，CHO，AGG の四つ*がそれにあたる．SEQ は前章で述べた AND タイプに対応するので，ジェネレータではなく，単に構造上の依存関係を表現しているノードである．

図3 に ODA に適用した場合のG木の例を示す．構成要素を表現するノード（以下 OBJ ノードと呼ぶ）は必ず共通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 ノード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．この選択されている構造のみを参照することで，仮想的に図の右に示したような特定構造を表現している．

以下，それぞれのジェネレータの保持情報およびその基本操作に関して簡単に述べる．

（1） OPT タイプ　下部構造のノードと共に，現在そのノードが選択状態にあるか否かの状態を保持する．基本操作は，初期状態に復元する（選択されていない状態にする），選択された状態にする，の二つである．

（2） REP タイプ　下部構造のノードを，繰り返された数だけ複写して保持する．基本操作は，初期状態に復元する（REP の下部構造が一つである状態にする），下部構造を一つ増やす，下部構造を一つ削る，の三つがある．

（3） CHO タイプ　下部構造として選択できるノード，過去に既に選択したことのあるノード，現在選択しているノードの三つの情報を保持する．基本操作は，初期状態に復元する（選択履歴を消去する），これまでに選択したことのない下部構造に切替える，指定した下部構造に切替える，の三つがある．

（4） AGG タイプ　現在選択している下部ノードの並び，過去に選択したことのある下部ノードの並びのパターンを保持する．基本操作は，初期状態に復元

* 構造生成子には OPT-REP もあるが，これは OPT と REP の両者の性質を合わせ持っており，OPT と REP の組合せと等価に扱うことができるので，ここでは独立して扱わない．

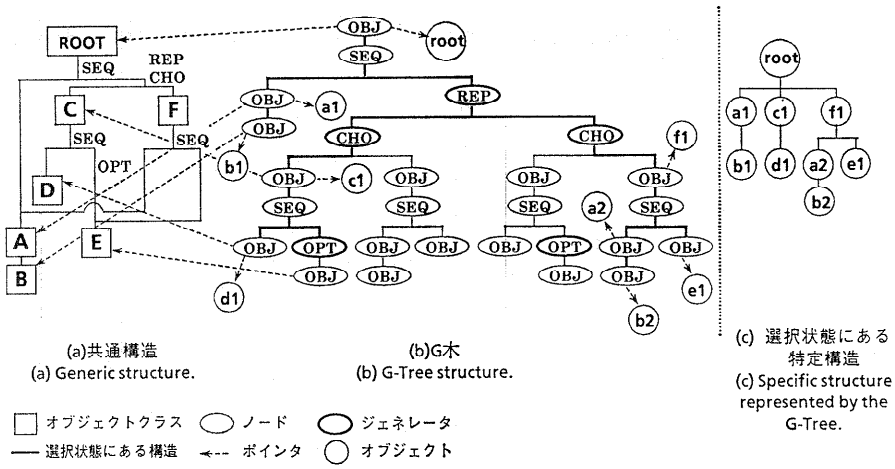


図 3 ODA に適用したG木の構成
Fig. 3 Example of G-Tree applied to ODA.

する（選択履歴を消去する），別の下部構造の並びを生成する，ある条件（一部の下部構造のみを順序づけるなど）を満たす並びを生成する，の三つがある．

4.3 G木を用いた場合の ODA 文書割り付け処理

G木を利用した場合，ODA の割り付け処理のおおまかな流れは以下のようになる．

（1） 共通割り付け構造から必要最小限のノードから構成されるG木を生成する．ここで，必要最小限とは，CHO であればいずれかの下部ノード，REP では一つの下部ノード，AGG は与えられた下部ノードの並び，OPT は選択されていない状態のことを指す．

（2） 以下の操作を論理順位に従って，すべての論理オブジェクトの処理を終了するまで行う．

(2a) 論理オブジェクトが複合論理オブジェクトである（下部構造が内容部でない）とき
割り付け指示属性が指定されていれば，それを満たす割り付けオブジェクトクラスを保持する OBJ ノードを捜し出す．みつからなければバックトラック処理を行う．何も指定されていなければ何もしない．

(2b) 論理オブジェクトが基本論理オブジェクトである（下部構造が内容部である）とき

(2b-1) 同じ “layout category” の値を持つ直前の論理オブジェクトを参照し，それが割り付けられている割り付けオブジェクトを対象割り付けオブジェクトとする．

(2b-2) “layout category” 以外の割り付け指示属性が指定されている場合は，対象割り付けオブジェクト以降で，割り付け指示属性を満たす割り付けオブジェクトクラスを保持する OBJ ノードを捜し出す．見つかればそれを対象割り付けオブジェクトとする．見つからない場合，あるいは発見できても，それを選択することによってこれまでに評価してきた割り付け指示属性を満足できなくなるなどの場合はバックトラック処理を行う．

(2b-3) 対象割り付けオブジェクトに内容を割り付ける．

(2b-4) 内容が溢れたなどにより，内容割り付けに失敗した場合*，現在の対象割り付けオブジェクト以降で，条件を満たす割り付けオブジェクトを探すために，(2b-2)から(2b-4)までと同様の処理を行う．

(2a)，(2b-1)，(2b-2)は，これまでの一般的な割り付け処理における，（ある条件を満たす）割り付けオブジェクトを生成し，現在生成途上にある割り付け構造に付加する処理に相当する．G木を用いた場合には，条件を満たす割り付けオブジェクトを保持するOBJ ノードを探索し，それが選択状態になるようにその上位に存在するジェネレータに対して基本操作を適用するという操作で実現される．上位に存在するジェネレータは複数存在する可能性があるが，どれを操作するかはその時の割り付けの進行状況から判断される．一般には，対象割り付けオブジェクトに近いものほど，割り付け構造に及ぼす影響の範囲が小さくて済むと言えるので，評価の優先順位は高い．

また，(2a)，(2b-2)におけるバックトラックの処

* テキストの場合は分割可能な場合があるが，ここでは説明の簡単のためにそれを省略する．

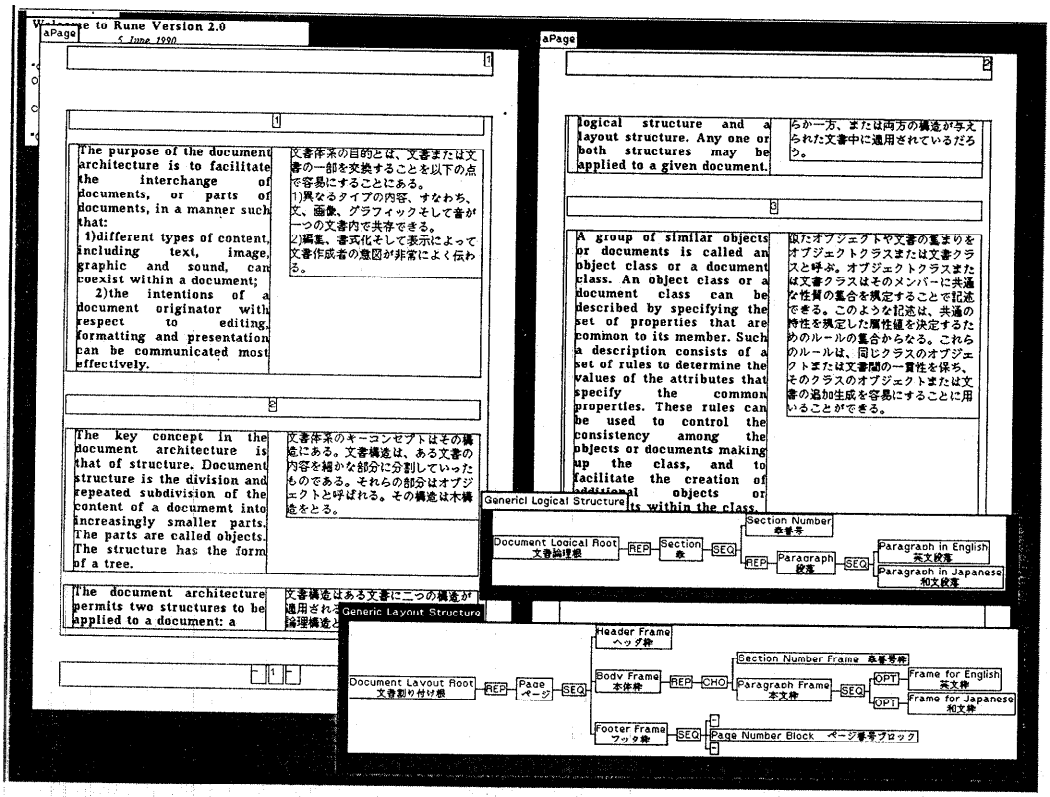


図 4 RUNE 2 上の ODA 文書例
Fig. 4 Example of an ODA document on RUNE 2.

理では，割り付けの失敗原因に応じて，バックトラックの範囲を変化させることで，効率が向上する．例えば，論理オブジェクトが残っているのに，割り付け構造がなくなってしまったときは，バックトラックが発生した直前に処理していた（バックトラックの直接の原因となった）論理オブジェクトを割り付けることのできる割り付けオブジェクトを優先的に捜し出すという手法が有効である．

これはG木を用いると，ある条件を満足する OBJ ノードを探し出す処理によって実現できる．その条件を満たす OBJ ノードとは，失敗の原因となった論理オブジェクトの有する割り付け指示属性を満たし，かつ構造上付加可能であるような割り付けオブジェクトを保持するノードである．構造上付加可能であるか否かは，割り付けオブジェクトの上位に REP，あるいは未評価の OPT タイプのジェネレータが存在するかどうか，あるいは CHO タイプの選択されていない選択肢の下部構造にその割り付けオブジェクトが属しているかどうか，で判定することができる．

5. 実験による効率比較

この章では，G木に基づく割り付け処理と一意の順序に基づいて構造を導出する割り付け処理とを，実験により処理効率を比較した結果について述べる．

処理効率の比較は，割り付け構造の選択肢の増加による，バックトラック回数と処理時間の変化をみる．選択肢の増加によって処理効率が低下しない割り付け処理方式が，実用的なシステムの実現に望ましい．なぜなら，一つの雛型が多くの選択肢を含み，論理構造に応じてさまざまなレイアウトを作りだせることが，割り付け指定として文法規則を用いることの効用として望まれるからである．

選択肢の増加は，割り付け指示の点から特定割り付け構造に現れることはない，幅固定高さ可変の最下位フレームクラス（以下これをダミーフレームと呼ぶ）を加えることとする．実験に用いた ODA 文書を**図 4** に示す．

実験には，著者等が実装した二つの割り付け処理シ

ステムを用いた．一つはG木に基づく割り付け処理システムで，これは RUNE 2 と呼ばれる．もう一つは，一意の順序に基づいて構造を導出する割り付け処理システムで，これは RUNE 1 と呼ばれる．RUNE 1 は文献11)で報告した割り付け処理システムであり，スタックを用いるのと同等の固定の導出順序を持つ．RUNE 2, RUNE 1 はともに Smalltalk-80☆により実装されている．

実験結果を**表1**，**表2**，**図5**に示す．ここで，割り付け処理の実行には SPARCstation☆☆2 上で動作する Smalltalk-80 バージョン 2.5 を用いた．表1にバックトラック発生回数を，表2に割り付け処理に要した時間を示す．ここで，処理時間の測定はミリ秒単位で行い，7回測定して最大値と最小値を除いた5回を平均したものを処理時間とした．図5に，表2の結果でダミーフレーム数が0の時に対する，ダミーフレーム数の増加にともなう処理時間変化比を示す．

表1 バックトラック発生回数
Table 1 Number of occurrence of backtracking.

ダミーフレーム数（個）	バックトラック発生回数（回）	
	RUNE 1	RUNE 2
0	7	3
1	17	3
2	29	3
3	41	3
4	53	3
5	65	3
6	77	3
7	89	3
8	101	3

表2 割り付け処理時間
Table 2 Processing time to layout.

ダミーフレーム数（個）	割り付け処理時間（秒）	
	RUNE 1	RUNE 2
0	3.16	2.45
1	3.33	2.43
2	4.10	2.48
3	4.25	2.50
4	4.55	2.62
5	5.03	2.64
6	5.91	2.82
7	6.18	2.74
8	6.81	2.79

☆ Smalltalk-80 は米国 Parc Place Systems 社の登録商標です．
☆☆ SPARCstation は SPARC International の登録商標です．

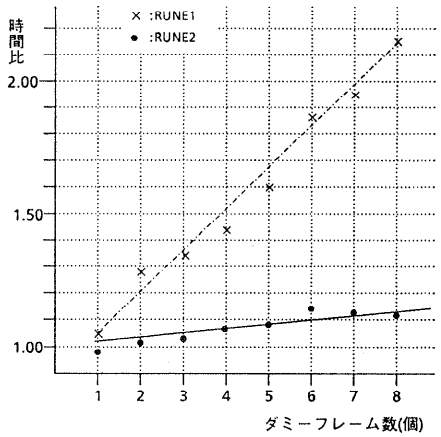


図5 選択肢の増加に伴う処理時間変化比
Fig. 5 Number of possibilities vs. relative processing time change.

表1に示されるように，構造の導出が一意の順序による RUNE 1 の割り付けの場合，選択肢が増えることにともなってバックトラックの発生が増加する．これに対して，G木に基づく RUNE 2 の場合，割り付け指示の評価時にどの選択肢を選べば良いかがわかるので，選択肢の増加によるバックトラック発生の増加はない．RUNE 2 におけるすべてのバックトラックは，割り付け指示の点から許される構造生成を行ったが，フレームの寸法の点から構造が付加できないためこれを削除したものである．

図5に示されるように，RUNE 1 では選択肢が増加すると割り付け処理の効率が大きく低下する．これは，バックトラックの増加と，これにともなって起こるフレームの持つ位置と寸法の再設定の増加の影響である．この例ではダミーフレームが単純な構造であるが，選択肢が入れ子の枠であれば処理効率はより低下すると考えられる．これに対して，RUNE 2 の場合は選択肢の増加にともなう割り付け処理の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増加せず，G木の探索および，REP タイプのジェネレータの下部構造のコピーのコストが増加することのみが割り付け処理の効率に影響を与えるためである．

6. 考　　察

6.1 再帰的規則

文法規則の大きな特徴の一つとして再帰構造が挙げ

られる☆.

文法規則が再帰を含んでいる場合には，G木ではすべての可能性を木構造として展開しようとするため，無限に木構造を生成しようとする．しかし，割り付けを正常に行うことができる文法規則では，再帰構造を含んでいても必ず終端に至る分岐点が存在するはずである．

このことから，終端に至らない再帰構造は未展開のまま残しておき，必要な時に再帰に至らないパスを一段階だけ展開するという一種の遅延評価の機構によって再帰構造を扱うことが可能になると考える．ここで必要な時とは，未展開の下部構造に存在するオブジェクトクラスを指定しているような割り付け指示属性が論理構造側に指定されている時，などのことである．

6.2 データ量

G木は一つの構造を表現するために，すべての可能性の構造を含めた数のジェネレータを必要とするので，必然的にデータ量が多くなる．実際のデータ量をODA の文書処理実験システム RUNE 2 の場合について測定してみた．

必要なメモリ領域は処理実行中にダイナミックに変化するものであるが，これは割り付け処理のアルゴリズムにも依存するので，今回は単純に，同じ文書を表現するために必要なデータ量を比較対象とした．つまり，割り付けが終了した時点での割り付け構造のデータ量を，純粋に特定割り付け構造のみで表現したものと，G木で表現したものとの両者の差を測定した．また，計測はバイト数で行っている．

結果を**図 6** に示す．この図は，G木を表現するために必要なデータ量が文書全体に対してどのくらいの割合を占めているかを，82 の ODA 文書例に関して測定したものである．

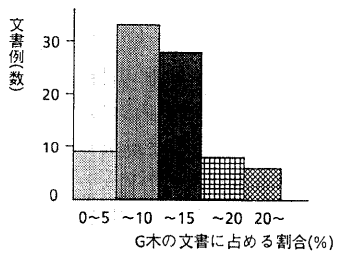


図 6 文書中にG木の占める記憶領域の割合
Fig. 6 Space amount ratio of G-Tree in documents.

☆ 再帰構造が割り付け構造の指定に必要か否かは議論の分かれるところであるが，ここでは触れない．

G木の文書に対する占める割合は，文書の共通割り付け構造の特徴に大きく依存する．共通割り付け構造が，複数の選択肢を持たない構造生成子 REP と SEQ のみで構成されるような文書では，選択されていない構造が全く存在しないので，その割合は数%のオーダである．しかし，OPT や CHO 構造を多く含み，かつその構造生成子以下に大きな構造が存在している場合には，選択状態でない構造の割合が増加するため，最低でも 10% 程度，多い場合は 40% 以上にも達する．G木は構造を大域的に扱うことを可能にするため，CHO の構造が多い場合に速度の向上が著しいがその反面データ量も増加すると言える．

実用化にはこのデータ量をおさえる手法を考案することが必要となると考える．これまでの ODA 文書の作成の経験上，共通構造では CHO の下部構造は複数の箇所から共有されたオブジェクトクラスで構成される場合が多い．例えば，CHO(SEQ(A, B), SEQ(OPT (B), C)) のような構造である．このような場合，CHO の二つの下部構造は同時に存在する可能性はないので，最初の SEQ とBと二つ目の SEQ のBを表現する OBJ ノードは共有できる可能性を持つ．この性質を利用して操作可能な場所の下部の同時に選択される可能性の無いジェネレータを共有させることで，構造を圧縮できる可能性がある．しかし，履歴情報の管理方法の再考が必要であるため，これは今後の検討課題である．

7. おわりに

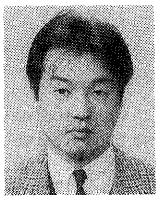
本論文では，効率的な割り付け処理の実現のためには，文法規則から構造を選択的に導出できる必要があり，このためには任意の時点における構造に対してどのような変更が行えるかを管理できる機構が必要であることを示した．そして，その機構で管理すべきものに対する検討結果から，G木という文法規則と構造とを統一的に扱えるデータモデルを提案した．最後に，G木を用いた割り付け処理システムを実際に作成した結果から，G木によって不要な構造生成を起こしにくい効率的な割り付け処理システムが実現できることを示した．

謝辞 日頃よりご指導をいただいている上林憲行主幹研究員，本研究の初期より熱心なご指導と有益なご助言をいただいた村田 真副主任研究員，ならびに本論文の執筆にあたりご意見をいただいたシステム・コミュニケーション研究所の皆様に深謝いたします．

参考文献

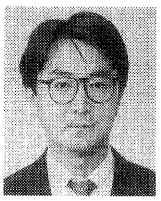
1) Furuta, R., Scofield, J. and Shaw, L.: Document Formatting Systems: Survey, Concepts, and Issues, *Comput. Surv.*, Vol. 14, No. 3, pp. 417-472 (1982).
2) Joloboff, V.: Document Representation: Concepts and Standards, *Structured Documents* (Andre, J., Furuta, R. and Quint, V., eds.), Cambridge University Press, New York (1989).
3) Johnson, J. and Beach, J.: Styles in Document Editing Systems, *Computer*, Vol. 21, No. 1, pp. 32-43 (1988).
4) Lamport, L.: *LaTeX: A Document Preparation System*, Addison-Wesley, Reading, Massachusetts (1986).
5) Furuta, R.: Concepts and Models for Structured Documents, *Structured Documents* (Andre, J., Furuta, R. and Quint, V., eds.), Cambridge University Press, New York (1989).
6) Foldfarb, C. F.: *The SGML Handbook*, Clarendon Press, Oxford (1990).
7) Furuta, R., Quint, V. and Andre, J.: Interactively Editing Structured Documents, *Electron. Publish.*, Vol. 2, Issue 3, pp. 19-44 (1988).
8) 富士ゼロックス(株)編：8080/8083 J Star Ⅱハンディリファレンス第1版，富士ゼロックス(株)，東京 (1988).
9) International Organization for Standardization (ed.): Information Processing—Text and Office Systems—Open Document Architecture (ODA) and Interchange Format, ISO 8613 (1989).
10) Xerox Corp. (ed.): Document Interchange Standard "Interscript", ISO/TC 97/SC 18/WG 3 N 439 R, International Organization for Standardization (1985).
11) 林 直樹，斎藤和雄，石田真美，村田 真：ODA 文書処理システムの試作(3)—割り付け処理—，第37回情報処理学会全国大会論文集，pp. 1857-1858 (1988).
12) 村上晴夫，山口 琢，松平秀樹，上原鉄三，鍵政秀子：ODA に基づいた文書割り付け処理の実現方式(1)—再試行の課題—，第40回情報処理学会全国大会論文集，pp. 598-599 (1990).
13) Takahashi, M.: Generalizations of Regular Sets and Their Application to a Study of Context-Free Languages, *Information and Control*, Vol. 27, No. 1, Academic Press, New York and London (1975).
14) Marchand, P.: Grammaires Parenthesees et Bilanguages Reguliers, *R. A. I. R. O. Informatique Theorique/Theoretical Informatics*, Vol. 14, No. 1, pp. 3-38 (1980).

(平成3年12月16日受付)
(平成5年4月8日採録)



齋藤 和雄（正会員）

1962年生．1985年豊橋技術科学大学工学部情報工学科卒業．1987年同大学院工学研究科修士課程修了．同年富士ゼロックス(株)入社．以来，文書の構造を利用した文書処理，文書アーキテクチャ等に関する研究に従事．



林 直樹（正会員）

1962年生．1985年神戸大学工学部システム工学科卒業．1987年同大学院修士課程修了．同年富士ゼロックス(株)入社．以来，文書構造を利用した編集システムの研究に従事．