

K\_064

## 自動生成されたメタデータに基づく視覚障害者のための

## プレゼンテーション文書アクセスインタフェース

An interface for blind person using meta-data of presentation documents

石原辰也† 高木啓伸† 伊藤隆‡ 浅川智恵子†

Tatsuya Ishihara Hironobu Takagi Takashi Itoh Chieko Asakawa

## 1. はじめに

近年プレゼンテーション文書はビジネス、教育等幅広い分野で利用されてきている。プレゼンテーション文書は、文字情報と画像や図形が混在し、さらにこれらのオブジェクトを自由に配置してビジュアルなページを作成できるという特徴がある。そのため視覚障害者にとって、音声のみでプレゼンテーション文書の内容を理解することが本質的に困難である。現在市販されている主要なスクリーンリーダは、プレゼンテーション文書をオブジェクトの奥行き方向の順序(Z軸方向の順序)で読み上げている[6]。この順序は、オブジェクトが作成された順番によって決まる。晴眼者にとっては、Z軸方向の順序はオブジェクトが重なっていない限り意味を持たないため、正しく読み上げ順序が設定されていない文書が多い。

例として図1のスライドをスクリーンリーダによって読み上げた場合の読み上げ例を図2の左図に示す。このようにZ軸方向の順序では、プレゼンテーション文書の作成者が意図した順番で読まれないことがある。



図1: 構造化するスライド例

ハードウェア	* ソフトウェア
データベース	- データベース
グループウェア	- グループウェア
ソフトウェア	* ハードウェア
サーバー	- サーバー
プリンター	- プリンター
ストレージ	- ストレージ

図2: スクリーンリーダの読み上げテキスト(左)と構造化されたテキスト(右)

また、たとえ読み上げ順序が正しく設定されていても、Z軸方向の順序に従ってテキスト情報を順番に読むだけで

†日本アイ・ピー・エム 東京基礎研究所

‡日本アイ・ピー・エム ソフトウェア開発研究所

はスライドの視覚的な構造を表現するには不十分である。図1には四角形の入れ子関係によって表現された視覚的な階層構造が存在しているため、図2右図のようにテキストを構造化することが、文書の構造を理解するために有効である。また矢印などのオブジェクトを用いて視覚的に構造が表現されている場合にも、その構造を非視覚的に表現する必要がある。

そこで本研究では、プレゼンテーション文書の視覚的な構造を解析することにより、オブジェクトの関係を表すメタデータを自動生成し、それに基づいて文書の構造を非視覚的に提示することのできるスクリーンリーダに最適化されたインタフェースを提案する。

## 2. 関連研究

既存のガイドラインにおいては、アクセシビリティの高いプレゼンテーション文書を作成するために、以下の3項目が定められている。[6]。

1. スライド中のオブジェクトの読み上げ順序を正しく設定する
2. 非テキストオブジェクトに対して代替テキストを設定する
3. オブジェクトを意味的にまとまった単位でグルーピングする。特にポリゴンの集合などで描かれた図はグルーピングが必要となる

上記の注意点1に関しては、Microsoft PowerPointでは1つ1つのオブジェクトを読み上げたい順番に選択して、オブジェクトを最背面に送るまたは最前面に送ることを繰り返していくことによって設定する。また、Z軸方向の順番というのは見た目だけでは確認することができないため、タブ移動を繰り返して順番を調べるという方法でしか確認できなかった。またポリゴンの集合等で描かれた図に対しては適切な単位でグルーピングして、グルーピング結果に対して代替テキストを挿入することが必要となる。

上記の注意点の1と2についての修正を支援するためのツールとして、PPT2HTML[2]がある。このツールは、プレゼンテーション文書をHTMLへ変換する際に、各オブジェクトの読み上げ順序を変更することができる。また非テキストオブジェクトに対して代替テキストを付加することができる。しかし、このツールでは文書構造を意味的にまとまった単位で、自動的にグルーピングすることは扱っていない。そのため1つ1つのオブジェクトに対して代替テキストの設定、読み上げ順序の設定をする必要があり、複雑な文書に対しては手間のかかる修正作業が必要であった。

また自動的に文書構造を理解するという点で関連する研究として、OCR(Optical Character Recognition)における文書のレイアウトの推定がある。これらの研究では文書のレイ

アウトを自動で推定すると共に、それに対する読み順も推定する試みが従来から行われていた[1]。しかし、これらの研究はプレゼンテーション文書のようなグラフィカルな文書を対象にしたものではなく、ある程度レイアウトが決まっている雑誌記事や、学术论文等を対象としていた。

これらのことから[3]において、プレゼンテーション文書のオブジェクト間の関係を自動で解析し、スクリーンリーダにとってアクセスしやすいデータに変換するためのメタデータの生成手法を提案した。本稿では[3]で提案した解析手法から得られるメタデータを活用して、アクセシブルなインタフェースを実現する。

### 3. アダプティブインタフェースの提案

スクリーンリーダに最適化された代替インタフェースのことを、本論文ではアダプティブインタフェースと定義する。[5]で提案されたアダプティブインタフェースは、Webページをツリー構造のUIに変換してアクセスすることでアクセシビリティの向上を実現した。しかし、このアダプティブインタフェースはWebページの読み上げ機能を提供するのみで、文書の編集機能については考慮していなかった。本論文で提案するアダプティブインタフェースの利点は以下の2つである。

1. シンプルなインタフェースで、プレゼンテーション文書の読み上げと編集が可能
2. アダプティブインタフェースと変換元のプレゼンテーションアプリケーションが同期することにより、晴眼者のアシストにより文書の編集を行うことが可能

開発したプレゼンテーション文書のためのアダプティブインタフェース(DocExplorer)のスクリーンショットを図3に示す。DocExplorerでは文書の読み上げは左側のツリービューを用いて行われる。ツリービューには後述する文書構造の解析の結果、木構造に変換されたデータが表示される。ツリービューに変換することのメリットは、シンプルなユーザーインタフェースで、文書の構造を表現できることである。また、一般的なインタフェースであるため、スクリーンリーダーユーザーが、新しいキー操作を覚えることなく操作できる。

また、ツリービューに表示される順番を、X軸方向、またはY軸方向の順番に並び替えることで、Z軸方向以外のナビゲーションも可能にしている。このように複数のナビゲーションモデルを提供することにより、ユーザーは必要なナビゲーションモデルを組み合わせることで文書の視覚的な構造を理解することができる。

文書の編集は右側のテキストボックスで行うことができる。テキストボックスには、ツリービューで選択されているオブジェクトから得られるテキスト情報が表示される。テキストボックスへの入力、DocExplorerのツリービューと変換元のアプリケーションの両方に同時に反映される。

また図中のテキスト属性ビューには、ツリービューで選択されているノードのリッチテキストの情報が表示される。メニューバーの詳細なメニューのコマンドを使用して、リッチテキストの属性の変更等ができる。メニューバーにはこの他に、スライドのレイアウトの変更や、新しいスライドの挿入などの詳細なメニューがある。

ノートビューには、現在ツリービューで選択されているページのプレゼンテーションノートの情報が表示される。

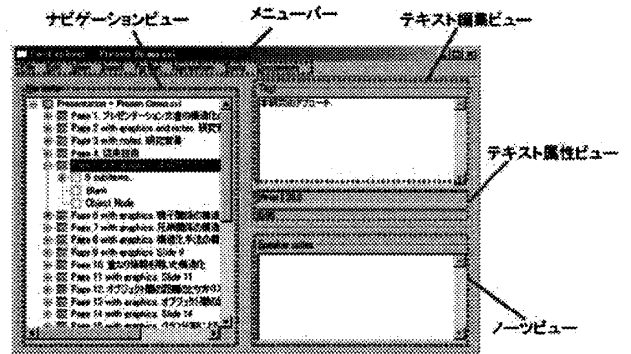


図3: DocExplorerのスクリーンショット

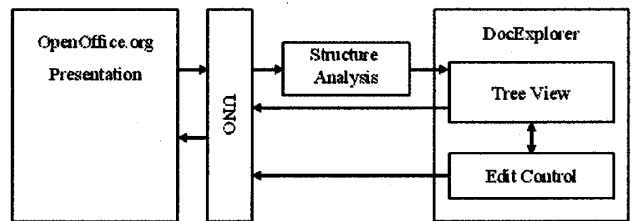


図4: DocExplorerのアーキテクチャ

提案するアダプティブインタフェースを、オフィスアプリケーションであるOpenOffice.org[4]のプレゼンテーションソフトに対して実装した。アーキテクチャを、図4に示す。プレゼンテーション文書のオブジェクトモデルは、OpenOffice.orgのAPI(UNO)を用いて得ることができる。得られたオブジェクトモデルに対して構造解析を行うことにより、メタデータを付加して、構造化したデータをツリービューに表示する。テキスト編集ビューへの入力があったときには、DocExplorerのツリービューとOpenOffice.orgの両方に反映することで、DocExplorerとOpenOffice.orgとの同期が行われる。

### 4. メタデータの自動生成

本稿で用いるメタデータとは、プレゼンテーション文書のオブジェクト間の視覚的な関係性を表現するものである。OpenOffice.orgのオブジェクト構造と共に、メタデータをアダプティブインタフェースへの入力とすることで、視覚的な構造をツリービューで表現することができる。

本稿ではメタデータとして、オブジェクトの重なりによって得られるグループ化情報、オブジェクト間の距離によって得られるグループ化情報、矢印の情報によって得られるリンク情報を扱う。これらのメタデータを自動的に得る方法として、[3]において提案した手法を用いる。この自動解析手法においては現在のところ距離情報、重なり情報のみを用いているので、オブジェクトの数が多く、密に並んでいるような極端に複雑なスライドに関しては正しく自動解析することができない。以下自動的に得られた構造化情報を、アダプティブインタフェースに表示するためにどのようなメタデータとして扱うかを述べる。

#### 4.1 重なり情報に基づくメタデータ

オブジェクトの重なり情報に基づくグループ化情報は、ある2つのオブジェクトが重なっていたときに、重なっている面積が元のオブジェクトの面積に対してある程度以上大きい場合にこれらをグループ化することによって得られる。

例として、図 5(a)のようなスライドを構造化することを考える。図 5(a)中に円で示したラベルは、各オブジェクトに対する ID である。このスライドを重なり情報に基づいて自動的に構造解析することにより図 5(b)のようなツリー構造が得られる。このようにして得られるグループ化情報を親子関係のグルーピングと呼ぶ。

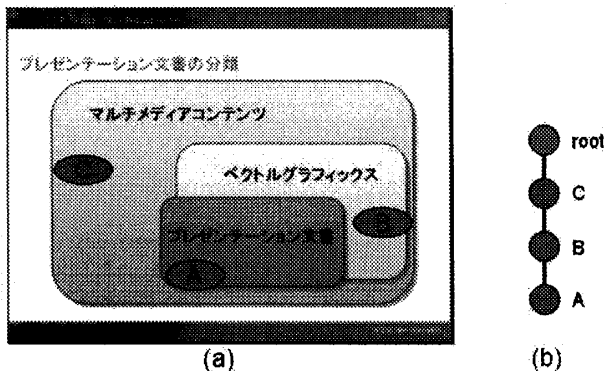


図 5: 重なり情報による構造化の例

また重なり情報に基づいてグルーピングした結果と Openoffice.org のグルーピング機能を用いて作成されたグループ構造は、DocExplorer のツリービューでは同じように表現される。

このように自動的なグルーピング機能を用いて階層的な構造を作ることで、アダプティブインタフェースのツリービューでも階層化した結果を表示することができ、音声アクセスでも文書の構造を理解することができる。

#### 4.2 距離情報に基づくメタデータ

オブジェクト間の距離情報に基づくグループ化情報は、あるオブジェクト間の距離が他のオブジェクト間の距離と比較して近く、他のオブジェクトと比較して整列されていたらグルーピングすることによって得られる。距離に基づくグループ化の情報は、スライド中のオブジェクト全体の関係から決定される。

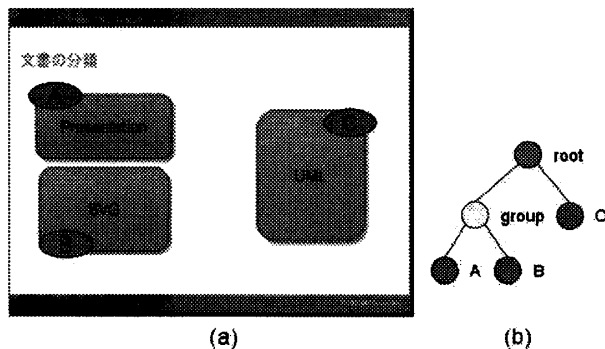


図 6: 距離情報による構造化の例

例として図 6(a)に示すスライドを構造化することを考える。このスライドを距離情報に基づいて構造化することにより、図 6(b)のようなツリー構造が得られる。このようにして得られるグループ化情報を兄弟関係のグルーピングと呼ぶ。

重なり情報に基づいて得られたメタデータと異なる点は、グループ化される対象となるオブジェクトが同一の階層になることである。

#### 4.3 矢印情報に基づくメタデータ

プレゼンテーション文書においては、矢印などのオブジェクト間の関係を表すのに用いられるものがある。これらのオブジェクトは通常スクリーンリーダーでは読むことができず、プレゼンテーションの内容を理解することを難しくする原因の1つとなっている。本研究では[3]で行われているように、矢印オブジェクトが指すオブジェクトを得る。また、矢印の上下に配置されたテキストオブジェクトが、他のオブジェクトと比較して近くに配置されていれば、矢印に対するラベルとする。本稿で定義するメタデータにも、矢印の指すオブジェクトと、矢印のラベルの両方を加えることとした。

例として図 7(a)のようなスライドを考える。このスライドをツリー構造で表すと図 7(b)のようになる。図 7(b)中の点線はノード間のハイパーリンクを表す。ハイパーリンクの上下に書かれている ID は、リンクラベルとして検出されたテキストオブジェクトを表す。

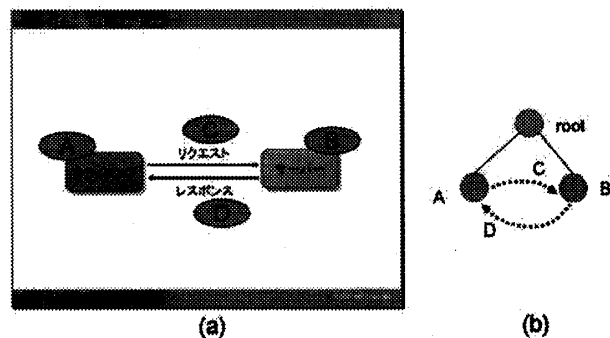


図 7: リンク解析の例

### 5. メタデータに基づくアダプティブインタフェースの表示

前節で定義したメタデータを用いて、プレゼンテーション文書を、ツリービューに変換して提示した結果を示す。図 5(a)のスライドを自動解析することにより得られたメタデータに基づいてツリー表示すると、図 8 のように階層化されて表示される。また、図 6(a)のスライドに対して同様にメタデータに基づいてツリー表示すると図 9 のように表示される。このように親子関係のグルーピング、兄弟関係のグルーピングの構造を表すメタデータを付加することで、アダプティブインタフェースに構造を持った形で出力することができる。

また、図 7(a)のスライドから得られたリンク情報を、アダプティブインタフェースに表示した結果を図 10 に示す。提案するアダプティブインタフェースでは、リンク情報はリンクノードとして提示する。ユーザーが、ツリービューのリンクノードの上でリターンキーを入力すると、ツリー

ビューのフォーカスが対応するリンク先のノードへと移る。このようなインタフェースにすることで、スクリーンリーダーユーザーも少ないキー操作で文書の内容を理解することができる。またリンクのラベル情報は、リンクノードの説明文に表示されることとした。このことはリンクの意味について理解するのに役立つ。

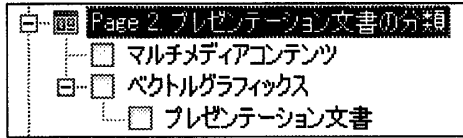


図 8: 図 2 のスライドに対するツリー表示結果



図 9: 図 3 のスライドに対するツリー表示結果

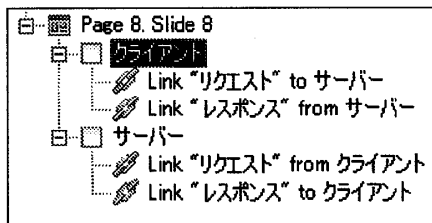


図 10: 図 4 のスライドに対するツリー表示結果

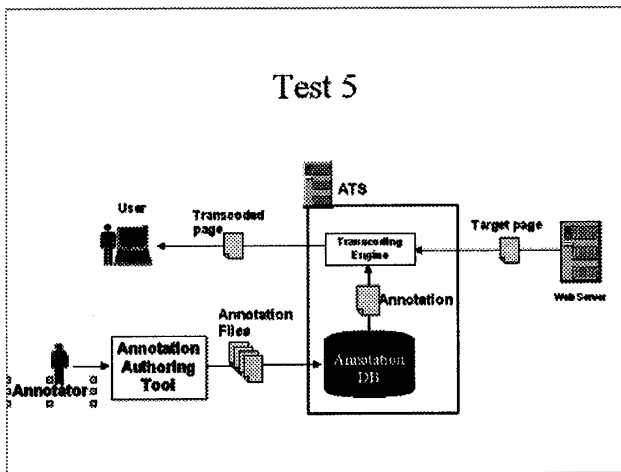


図 11: 構造化するスライドの例

より複雑なグラフィックスをツリー表示した例を図 11 と図 12 に示す。図 11 の "ATS" というテキストオブジェクトに対して、兄弟関係のグルーピングがされていることがわかる。さらに "Annotation DB" というオブジェクトは、親子関係のグルーピングの結果として階層化されていることがわかる。またこれらの階層化された結果にたいしても、リンク情報が正しく解析されてツリービューに表示されていることがわかる。

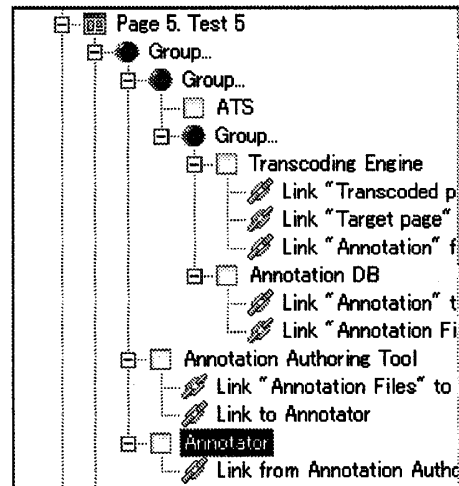


図 12: 図 6 のツリー表示結果

## 6. まとめ

本研究ではプレゼンテーション文書の視覚的な構造を表すメタデータを自動で生成し、それに基づくアクセシブルなアダプティブインタフェースを実現した。本稿のアダプティブインタフェースを用いることにより、スクリーンリーダーユーザーは、変換元のオフィスアプリケーションと同期しながら、プレゼンテーション文書を編集することが可能である。今後の課題としては、提案したユーザーインタフェースを用いて、自動解析によって得られたメタデータがどの程度スクリーンリーダーユーザーにとって有用なものであるのかということを検証する必要がある。

## 参考文献

- [1]. M. Aiello, C. Monz, L. Todoran, and M. Worring, "Document understanding for a broad class of documents", IJDAR, 5(1), pp. 1-16, 2002
- [2]. PPT2HTML.  
<http://www.rdpslides.com/pptools/ppt2html/index.html>
- [3]. 石原辰也, 高木 啓伸, 伊藤 隆, 浅川 智恵子, "プレゼンテーション文書の構造化のためのメタデータの自動付加手法" 福祉情報工学研究会技術報告, SP2005-84, WIT2005-46, 2005
- [4]. OpenOffice.org. <http://www.openoffice.org/>
- [5]. Accessing Web Based Documents Through a Tree Structural Interface, E. Walshe, B. McMullin, ICCHP, pp. 560-563, 2004
- [6]. Creating accessible Microsoft PowerPoint documents  
<http://www-306.ibm.com/able/guidelines/documentation/docmsppt.htm>