

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5110640号  
(P5110640)

(45) 発行日 平成24年12月26日 (2012. 12. 26)

(24) 登録日 平成24年10月19日 (2012. 10. 19)

(51) Int. Cl.		F I			
<b>G06F</b>	<b>9/44</b>	<b>(2006.01)</b>	G06F	9/06	620A
<b>G06F</b>	<b>13/00</b>	<b>(2006.01)</b>	G06F	13/00	500H
			G06F	9/44	530P

請求項の数 20 (全 26 頁)

(21) 出願番号	特願2007-265234 (P2007-265234)	(73) 特許権者	390009531
(22) 出願日	平成19年10月11日 (2007. 10. 11)		インターナショナル・ビジネス・マシーンズ・コーポレーション
(65) 公開番号	特開2009-93521 (P2009-93521A)		INTERNATIONAL BUSINESS MACHINES CORPORATION
(43) 公開日	平成21年4月30日 (2009. 4. 30)		アメリカ合衆国10504 ニューヨーク州 アーモンク ニュー オーチャードロード
審査請求日	平成22年6月23日 (2010. 6. 23)	(74) 代理人	100108501 弁理士 上野 剛史
		(74) 代理人	100112690 弁理士 太佐 種一
		(74) 代理人	100091568 弁理士 市位 嘉宏

最終頁に続く

(54) 【発明の名称】 アクセシビリティ情報を得る方法、コンピュータ・プログラム及びアクセシビリティ情報装置

(57) 【特許請求の範囲】

【請求項1】

リッチ・インターネット・アプリケーションのコンテンツにおけるアクセシビリティ情報を得る方法であって、

コンピュータが、表示画面に表示された前記コンテンツのオブジェクトを実験環境で実行するステップと、

コンピュータが、前記オブジェクトを実行したことに応じて、前記オブジェクトから出力された出力情報と前記オブジェクトの状態とに基づき、予め用意された複数のオブジェクトに関する参照モデル情報を用いて前記オブジェクトの役割を推定するステップと、

コンピュータが、推定された前記オブジェクトの役割を、前記アクセシビリティ情報として出力するステップと、

を含む、アクセシビリティ情報を得る方法。

【請求項2】

コンピュータが、前記オブジェクトと、前記オブジェクトの近傍に表示されるテキスト・オブジェクトとの位置関係に基づき、前記テキスト・オブジェクトを用いて前記オブジェクトを表す代替テキストを推定するステップと、

コンピュータが、推定された前記代替テキストを、前記アクセシビリティ情報として出力するステップと、

を含む、

請求項1に記載のアクセシビリティ情報を得る方法。

## 【請求項 3】

前記代替テキストを推定するステップは、前記オブジェクトを前記実験環境で実行したことに応じて前記コンテンツを形成するプログラムから取得した、前記オブジェクトに割当てられたシンボル情報に基づき、前記オブジェクトの前記代替テキストを推定する、請求項 2 に記載のアクセシビリティ情報を得る方法。

## 【請求項 4】

コンピュータが、前記代替テキストが推定されないことに応じて前記オブジェクトのサイズを計測するステップと、

コンピュータが、計測された前記オブジェクトのサイズが所定の大きさを満たしたことを条件に、前記オブジェクトの役割を判断するステップと、

コンピュータが、判断結果に応じて、推定された前記オブジェクトの役割に関する前記推定を取り消すステップと、

を含む、

請求項 2 に記載のアクセシビリティ情報を得る方法。

## 【請求項 5】

コンピュータが、前記オブジェクトのサイズを計測するステップを含み、

前記代替テキストを推定するステップは、

前記オブジェクトのサイズがゼロであることに応じて、

## 【数 1】

$W$ : オブジェクト

$t$ : テキスト・オブジェクトとして

For  $t \in \text{Text}(\text{Succ}(\text{Parent}(W)))$  に対して

$(\text{Dist}(W, t) < K_d \times \text{Diag}(t)) \cap (\text{Left}(W) < \text{Left}(t)) \cap (\text{Top}(W) > \text{Top}(t))$

但し、

$\text{Text}(\text{nodes})$ : ノード集合  $\text{nodes}$  のうち、テキスト・オブジェクトの集合

$\text{Succ}(\text{node})$ : ノード  $\text{node}$  の子孫の集合

$\text{Parent}(\text{node})$ : ノード  $\text{node}$  の親ノード

$K_d$ : 重なりや距離に関する定数

$\text{Dist}(o_1, o_2)$ : オブジェクト  $o_1$  と  $o_2$  との距離

$\text{Diag}(\text{obj})$ : オブジェクト  $\text{obj}$  の対角線の長さ

$\text{Left}(\text{obj})$ : オブジェクト  $\text{obj}$  の左端の座標、左を 0 とし右が正

$\text{Top}(\text{obj})$ : オブジェクト  $\text{obj}$  の上端の座標、下を 0 とし上が正

を満たす前記テキスト・オブジェクトを、前記代替テキストと推定し、前記オブジェクトのサイズがゼロでないことに応じて、

## 【数 2】

$$\left( \frac{\text{Area}(t)}{\text{Int}(t, W)} > T_{\text{text}} \right) \cap \left( \frac{\text{Area}(W)}{\text{Int}(t, W)} > T_{\text{widget}} \right)$$

但し、

$\text{Area}(\text{obj})$ : オブジェクト  $\text{obj}$  の表示エリア

$\text{Int}(\text{obj}_1, \text{obj}_2)$ :  $\text{obj}_1$  と  $\text{obj}_2$  の重なり合ったエリア

$T_t, T_w$ : 重なりや距離に関する定数

を満たす前記テキスト・オブジェクトを、前記代替テキストと推定することを含む、請求項 2 に記載のアクセシビリティ情報を得る方法。

## 【請求項 6】

前記役割を推定するステップは、  
 前記実験環境を初期化し、  
 前記オブジェクトの状態を記録し、  
 記録した前記オブジェクトの状態と、前記参照モデル情報の状態とを照合し、  
 照合された前記参照モデル情報の次にとりうる遷移に対する状態を取得し、  
 取得した前記参照モデル情報の状態における出力情報と、前記オブジェクトを実行した  
 ことに応じて前記オブジェクトから出力された前記出力情報とを検証し、  
 検証結果に応じて、前記参照モデル情報に基づき前記オブジェクトの役割を推定するこ  
 とを含む、

請求項 1 に記載のアクセシビリティ情報を得る方法。

10

【請求項 7】

前記役割を推定するステップは、  
 前記オブジェクトを実行する前に、前記実験環境を保存し、  
 前記オブジェクトの入力の送込に応じて、前記オブジェクトの状態を記録し、  
 前記オブジェクトを実行した後に、前記実験環境を復帰することを更に含む、  
 請求項 6 に記載のアクセシビリティ情報を得る方法。

【請求項 8】

前記役割を推定するステップは、  
 取得した前記参照モデル情報の前記出力情報を、実行した前記オブジェクトの前記出力  
 情報が含むか否かを検証することを更に含む、  
 請求項 6 に記載のアクセシビリティ情報を得る方法。

20

【請求項 9】

前記リッチ・インターネット・アプリケーションは、Flash（登録商標）及び Dynamic HTML の少なくともいずれか一方を用いて作成される、  
 請求項 1 に記載のアクセシビリティ情報を得る方法。

【請求項 10】

請求項 1 から 9 のいずれかに 1 項に記載の各ステップを、コンピュータに実行させるためのコンピュータ・プログラム。

【請求項 11】

リッチ・インターネット・アプリケーションのコンテンツにおけるアクセシビリティ情報装置であって、

30

表示部に表示された前記コンテンツのオブジェクトを実行する実験環境を制御する実験環境制御部と、

予め用意された複数のオブジェクトに関する参照モデル情報を記憶する参照モデル記憶部と、

前記オブジェクトを実行したことに応じて、前記オブジェクトから出力された出力情報と前記オブジェクトの状態とに基づき、前記参照モデル記憶部に記憶された前記参照モデル情報を用いて前記オブジェクトの役割を推定する役割推定部と、

前記役割推定部により推定された前記オブジェクトの役割を、アクセシビリティ情報として出力する役割出力部と、

40

を備える、アクセシビリティ情報取得装置。

【請求項 12】

前記オブジェクトと、前記オブジェクトの近傍に表示されるテキスト・オブジェクトとの位置関係に基づき、前記テキスト・オブジェクトを用いて前記オブジェクトを表す代替テキストを推定する視覚情報解析部と、

前記視覚情報解析部により推定された前記代替テキストを、前記アクセシビリティ情報として出力する代替テキスト出力部と、

を備える、

請求項 11 に記載のアクセシビリティ情報取得装置。

【請求項 13】

50

前記役割推定部は、前記オブジェクトを前記実験環境で実行したことに応じて前記コンテナを形成するプログラムから前記オブジェクトに割当てられたシンボル情報を取得し、

前記視覚情報解析部は、前記役割推定部により取得された前記シンボル情報に基づき、前記オブジェクトの前記代替テキストを推定する、

請求項 1 2 に記載のアクセシビリティ情報取得装置。

【請求項 1 4】

前記視覚情報解析部は、

前記代替テキストが推定されないことに応じて前記オブジェクトのサイズを計測し、計測された前記オブジェクトのサイズが所定の大きさを満たしたことを条件に、前記オブジェクトの役割を判断し、

判断結果に応じて、推定された前記オブジェクトの役割に関する前記推定を取り消す、請求項 1 2 に記載のアクセシビリティ情報取得装置。

【請求項 1 5】

前記視覚情報解析部は、

前記オブジェクトのサイズを計測し、

前記オブジェクトのサイズがゼロであることに応じて、

【数 3】

$W$ : オブジェクト

$t$ : テキスト・オブジェクトとして

For  $t \in \text{Text}(\text{Succ}(\text{Parent}(W)))$  に対して

$(\text{Dist}(W, t) < K_d \times \text{Diag}(t)) \cap (\text{Left}(W) < \text{Left}(t)) \cap (\text{Top}(W) > \text{Top}(t))$

但し、

$\text{Text}(\text{nodes})$ : ノード集合  $\text{nodes}$  のうち、テキスト・オブジェクトの集合

$\text{Succ}(\text{node})$ : ノード  $\text{node}$  の子孫の集合

$\text{Parent}(\text{node})$ : ノード  $\text{node}$  の親ノード

$K_d$ : 重なりや距離に関する定数

$\text{Dist}(o_1, o_2)$ : オブジェクト  $o_1$  と  $o_2$  との距離

$\text{Diag}(\text{obj})$ : オブジェクト  $\text{obj}$  の対角線の長さ

$\text{Left}(\text{obj})$ : オブジェクト  $\text{obj}$  の左端の座標、左を 0 とし右が正

$\text{Top}(\text{obj})$ : オブジェクト  $\text{obj}$  の上端の座標、下を 0 とし上が正

を満たす前記テキスト・オブジェクトを、前記代替テキストと推定し、前記オブジェクトのサイズがゼロでないことに応じて、

【数 4】

$\left( \frac{\text{Area}(t)}{\text{Int}(t, W)} > T_{\text{text}} \right) \cap \left( \frac{\text{Area}(W)}{\text{Int}(t, W)} > T_{\text{widget}} \right)$

但し、

$\text{Area}(\text{obj})$ : オブジェクト  $\text{obj}$  の表示エリア

$\text{Int}(\text{obj}_1, \text{obj}_2)$ :  $\text{obj}_1$  と  $\text{obj}_2$  の重なり合ったエリア

$T_t, T_w$ : 重なりや距離に関する定数

を満たす前記テキスト・オブジェクトを、前記代替テキストと推定することを含む、請求項 1 2 に記載のアクセシビリティ情報取得装置。

【請求項 1 6】

前記役割推定部は、  
前記実験環境を初期化し、  
前記オブジェクトの状態を記録し、  
記録した前記オブジェクトの状態と、前記参照モデル記憶部に記憶された前記参照モデル情報の状態とを照合し、  
照合された前記参照モデル情報の次にとりうる遷移に対する状態を取得し、  
取得した前記参照モデル情報の状態における出力情報と、前記オブジェクトを実行したことに応じて前記オブジェクトから出力された前記出力情報とを検証し、  
検証結果に応じて、前記参照モデル情報に基づき前記オブジェクトの役割を推定することを含む、

10

請求項 11 に記載のアクセシビリティ情報取得装置。

【請求項 17】

前記役割推定部は、  
前記オブジェクトを実行する前に、前記実験環境を保存し、  
前記オブジェクトの入力の送出に応じて、前記オブジェクトの状態を記録し、  
前記オブジェクトを実行した後に、前記実験環境を復帰することを更に含む、  
請求項 16 に記載のアクセシビリティ情報取得装置。

【請求項 18】

前記役割出力部は、内部モデルである、  
請求項 11 に記載のアクセシビリティ情報取得装置。

20

【請求項 19】

前記役割出力部に出力された前記アクセシビリティ情報は、音声情報として出力可能である、  
請求項 11 に記載のアクセシビリティ情報取得装置。

【請求項 20】

前記視覚情報解析部及び前記役割推定部により出力されたログ情報を解析するログ解析部と、  
前記ログ解析部による解析結果を用いて、前記アクセシビリティ情報を出力するアクセシビリティ情報出力部と、  
を備える、

30

請求項 12 に記載のアクセシビリティ情報取得装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、リッチ・インターネット・アプリケーションのコンテンツにおけるアクセシビリティ情報を得る方法、コンピュータ・プログラム及びアクセシビリティ情報装置に関する。

【背景技術】

【0002】

近年、Flash (登録商標) や Dynamic HTML (DHTML) に代表されるようなリッチ・インターネット・アプリケーション (RIA) が大幅に増加してきている。Ajax (Asynchronous JavaScript (登録商標) + XML) の勃興により、DHTML は多くの Web アプリケーションで積極的に活用されるようになってきている。また、1996 年には単純なアニメーションを表現するだけだった Flash も、その後の 10 年間で 98% 以上のコンピュータにインストールされるようになってきている。Flash は、音声出力、スクリプトによるユーザ・インタラクション、GUI (Graphical User Interface) コンポーネントの提供、高度な画像処理機能、ストリーミング動画等に対応し、インタラクティブなコンテンツを提供するためのプラットフォームとしての地位を確立している。

40

【0003】

50

このように、R I AがW e b上で積極的に活用されている一方で、アクセシビリティの重要性も増している。D H T M LやF l a s h等のR I Aでは、( 1 ) 視覚的な効果を重視するために、テキストを用いずにイメージで情報を表現することが多い( その際に、イメージ情報の注釈の役割をする代替テキストが用意されていないことも多い)。また、( 2 ) J a v a s c r i p t ( 登録商標) の標準規格であるE C M A S c r i p tを用いて、ユーザ・インターフェースを設計するために、スクリプトによってプレゼンテーションが制御される。そのため、コンテンツの各オブジェクトがどのようなR o l e ( 役割) を持っているのかを知ることが難しい。

#### 【0004】

D H T M Lにおいては、W A I - A R I A ( <http://www.w3.org/TR/aria-roadmap/>) とよばれる標準制定活動が始まっており、アクセシビリティを確保するためのフレームワークが制定されつつある。しかし、W A I - A R I Aは、コンテンツの作成者がスクリプトを用いて適切にR o l eやS t a t e ( 状態) 等の情報を付与する必要がある。そして、現状において、W A I - A R I Aに対応しているコンテンツは、ほとんど存在しない。

#### 【0005】

F l a s hについても、アクセシビリティの機構が用意されており、R o l eやS t a t eの情報をM S A A ( M i c r o s o f t ( 登録商標) A c t i v e A c c e s s i b i l i t y) とよばれるメカニズムを用いて通知する方法が用意されている。しかし、実際には、F l a s hのアクセシビリティに対する取り組みは少ない。そのため、インターネット上のほとんどのF l a s hコンテンツはアクセシブルではなく、画面の読み上げを行うスクリーン・リーダーを使ってコンテンツにアクセスする視覚障害者のユーザにとっては、ほとんど操作することができない。

【非特許文献1】Microsoft Corporation, "Microsoft Active Accessibility", [online], [2007年9月27日検索]、インターネット<URL:<http://msdn2.microsoft.com/en-us/library/ms697707.aspx>>

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0006】

このように、R I Aでは、多くのコンテンツがアクセシブルでないのは、以下の主に3つの問題から引き起こされている。第1に、オブジェクトに対して、ボタンやリンク等のR o l eやS t a t eの情報が正しく提供されないことである。第2に、代替テキストが提供されていないことである。そして、第3に、スクリプトとしてコンテンツが提供されるため、アクセシビリティの問題を発見し、修復することが難しいことである。

#### 【0007】

本発明は、リッチ・インターネット・アプリケーションのコンテンツにおけるアクセシビリティ情報を得る方法、コンピュータ・プログラム及びアクセシビリティ情報装置を提供することを目的とする。

#### 【課題を解決するための手段】

#### 【0008】

本発明の1つの態様では、以下のような解決手段を提供する。

#### 【0009】

本発明の第1の形態によると、リッチ・インターネット・アプリケーションのコンテンツにおけるアクセシビリティ情報を得る方法を提供する。アクセシビリティ情報を得る方法は、表示画面に表示されたコンテンツのオブジェクトを実験環境で実行する。そして、実行したことによりオブジェクトから出力された出力情報とオブジェクトの状態とに基づき、予め用意された複数のオブジェクトに関する参照モデル情報を用いてオブジェクトの役割を推定する。また、オブジェクトの状態は、推定されたオブジェクトの役割を、アクセシビリティ情報として出力する。出力は内部モデルに対して行われる。なお、リッチ・

10

20

30

40

50

インターネット・アプリケーションは、Flash及びDynamic HTMLの少なくともいずれか一方を用いて作成される。

【0010】

本発明の第2の形態によると、オブジェクトと、オブジェクトの近傍に表示されるテキスト・オブジェクトとの位置関係に基づき、テキスト・オブジェクトを、オブジェクトを表す代替テキストと推定する。位置関係は、オブジェクトのサイズを計測し、オブジェクトのサイズがゼロか否かに応じて、所定の条件式を満たすテキスト・オブジェクトを代替テキストと推定する。また、位置関係に基づき代替テキストが推定されない場合において、オブジェクトを実験環境で実行したことに応じてコンテンツを形成するプログラムから取得した、オブジェクトに割当てられたシンボル情報を、オブジェクトの代替テキストと推定する。

10

【0011】

本発明の第3の形態によると、オブジェクトの役割が推定された場合であって、代替テキストが推定されない場合に、オブジェクトのサイズを計測し、オブジェクトのサイズが所定の大きさを満たしたことを条件に、オブジェクトの役割を判断する。そして、判断結果に応じて、既に推定されたオブジェクトの役割に関する推定を取り消す。

【0012】

本発明の第4の形態によると、オブジェクトの役割は、様々な検証の結果推定される。実験環境を初期化し、オブジェクトの状態を記録し、記録したオブジェクトの状態と、参照モデル情報の状態とを照合し、照合された参照モデル情報の次にとりうる遷移に対する状態を取得し、取得した参照モデル情報の状態における出力情報と、オブジェクトを実行したことに応じてオブジェクトから出力された出力情報とを検証し、検証結果に応じて、参照モデル情報に基づきオブジェクトの役割を推定する。また、オブジェクトの実行に関して、オブジェクトを実行する前に、実験環境を保存し、オブジェクトの入力の送出に応じて、オブジェクトの状態を記録し、オブジェクトを実行した後に、実験環境を復帰することを含むものである。更に、取得した参照モデル情報の出力情報を、実行したオブジェクトの出力情報が含むか否かを検証する。

20

【0013】

また、本発明は、別の態様として、コンピュータにおいて実行されるコンピュータ・プログラムあるいはアクセシビリティ情報取得装置としても提供できる。

30

【0014】

本発明の利点は以下の通りである。

【0015】

第1に、表示画面に表示されたコンテンツのオブジェクトを実験環境で実行したことによりオブジェクトから出力された出力情報と、オブジェクトの状態とに基づき、予め用意された複数のオブジェクトに関する参照モデル情報を用いてオブジェクトの役割を推定する。よって、オブジェクトにアクセシビリティ情報を有していなくても、実行時の出力及び内部状態から、アクセシビリティ情報を得ることができる。また、出力は内部モデルに対して行われるので、内部モデルを用いてアクセシビリティ情報を音声で出力することができ、視覚障害者にとって有益な情報を得ることができる。更に、内部モデルを用いてコンテンツ作成時にアクセシビリティ情報を予め入力することができる。

40

【0016】

第2に、オブジェクトと、オブジェクトの近傍に表示されるテキスト・オブジェクトとの位置関係に基づき、テキスト・オブジェクトを、オブジェクトを表す代替テキストと推定することができる。よって、オブジェクトにアクセシビリティ情報を有していなくても、オブジェクトとテキスト・オブジェクトとによって代替テキストを推定することで、アクセシビリティ情報を得ることができる。また、オブジェクトのサイズがゼロの場合を考慮して、アクセシビリティ情報を得ることができる。更に、オブジェクトを実験環境で実行することにより、プログラムからオブジェクトのシンボル情報を取得することができる。このシンボル情報を、オブジェクトの代替テキストと推定することができるので、オブ

50

ジェクトを表すアクセシビリティ情報を得ることができる。

【0017】

第3に、オブジェクトの役割の推定について、代替テキストが推定されない場合に、オブジェクトのサイズを計測し、オブジェクトのサイズが所定の大きさを満たしたことを条件に、オブジェクトの役割を判断し、判断結果に応じて、既に推定されたオブジェクトの役割に関する推定を取り消す。つまり、誤ってオブジェクトの役割を推定したもののうち、特定のものに関して、所定の処理により役割の推定を取り消すことができる。よって、推定の誤りを少なくすることができる。

【0018】

第4に、オブジェクトの役割の推定を、実験環境でオブジェクトを実行し、参照モデルを用いた検証処理により、オブジェクトの役割を検証することができる。よって、推定されたオブジェクトの役割の正当性を確認することができる。役割をアクセシビリティ情報として出力するので、ユーザにとって有益なアクセシビリティ情報を得ることができる割合を向上させることができる。

【発明の効果】

【0019】

本発明によれば、リッチ・インターネット・アプリケーションのコンテンツにおいて、オブジェクトの代替テキスト及び役割を推定し、推定した代替テキスト及び役割を用いることで、アクセシビリティ情報を得る方法、コンピュータ・プログラム及びアクセシビリティ情報装置を提供することができる。

【発明を実施するための最良の形態】

【0020】

以下、本発明の実施形態について、図面に基づいて以下に説明する。図1は、本発明の実施形態に係るアクセシビリティ情報取得システム100の構成の一例を示す図である。アクセシビリティ情報取得システム100は、RIAのコンテンツにおけるオブジェクト（以下、コンテンツのオブジェクト、又は単にオブジェクトという）のアクセシビリティ情報を取得するアクセシビリティ情報取得装置1と、取得したアクセシビリティ情報を音声データとしてユーザに通知する音声部8とからなるシステムである。アクセシビリティ情報取得装置1は、制御部10として、実験環境制御部11、視覚情報解析部12及び役割推定部13を有する。また、コンテンツ実行エンジン14、参照モデル記憶部15及び内部モデル16を有する。

【0021】

実験環境制御部11は、各ユーザが使用するパソコン等の表示部（図示せず）に、コンテンツ実行エンジン14として、例えば、ブラウザを介して表示されたコンテンツのオブジェクトを実行する実験環境を制御する。他に影響を与えずに実行可能なこのような実験環境を、Sandbox環境という。実験環境にてコンテンツのオブジェクトを実行することで、アクセシビリティ情報となる代替テキスト及びRoleを推定する。視覚情報解析部12は、オブジェクトと、当該オブジェクトの近傍に表示される情報とから、オブジェクトに関連する情報が否かを解析する。より具体的には、オブジェクトと、当該オブジェクトの近傍に表示されるテキスト・オブジェクトとの位置関係に基づき、テキスト・オブジェクトを、オブジェクトを表す代替テキストとして推定する。役割推定部13は、参照モデル記憶部15の複数の参照モデルを用いて、該当のオブジェクトの役割を推定する。参照モデル記憶部15は、予め用意した複数の参照モデルを記憶する。内部モデル16は、制御部10により取得したアクセシビリティ情報を受け付ける。

【0022】

次に、アクセシビリティ情報を代替テキストにより推定する処理について説明する。図2は、オブジェクトの代替テキストを推定する処理のフローチャートを示す。図3は、図2のフローチャートの内容を説明するための図である。

【0023】

まず、図2のステップS11において、視覚情報解析部12は、オブジェクトと、画面

10

20

30

40

50



(図示せず) 上に表示されているテキスト・オブジェクトとの視覚的關係による代替テキストの推定を行う。あるテキスト・オブジェクトがオブジェクトの代替テキストであるか否かは、当該オブジェクトとテキスト・オブジェクトとの視覚的關係を調べることで推定する。あるオブジェクトWに対する代替テキストtとの關係は、以下の数式で決定する。

【数1】

For  $t \in \text{Text}(\text{Succ}(\text{Parent}(W)))$

$$\left\{ \begin{array}{l} \text{if } \text{Size}(W) > 0 \\ \left( \frac{\text{Area}(t)}{\text{Int}(t, W)} > T_{\text{text}} \right) \cap \left( \frac{\text{Area}(W)}{\text{Int}(t, W)} > T_{\text{widget}} \right) \\ \text{otherwise} \\ (\text{Dist}(W, t) < Kd \times \text{Diag}(t)) \cap (\text{Left}(W) < \text{Left}(t)) \cap (\text{Top}(W) > \text{Top}(t)) \end{array} \right. \quad 10$$

但し、

- Text(nodes): ノード集合nodesのうち、テキスト・オブジェクトの集合
- Succ(node): ノードnodeの子孫の集合
- Parent(node): ノードnodeの親ノード
- Area(obj): オブジェクトobjの表示エリア
- Int(obj<sub>1</sub>, obj<sub>2</sub>): obj<sub>1</sub>とobj<sub>2</sub>の重なり合ったエリア
- T<sub>t</sub>, T<sub>w</sub>, K<sub>d</sub>: 重なりや距離に関する定数
- Dist(o<sub>1</sub>, o<sub>2</sub>): オブジェクトo<sub>1</sub>とo<sub>2</sub>との距離
- Diag(obj): オブジェクトobjの対角線の長さ
- Left(obj): オブジェクトobjの左端の座標、左を0とし右が正
- Top(obj): オブジェクトobjの上端の座標、下を0とし上が正
- Size(obj): オブジェクトobjの大きさ、幅×高さ

【0024】

図3を用いて、より具体的に説明する。図3(a)は、オブジェクトWがボタンであり、オブジェクトWのサイズがある(サイズがゼロではなく、一定の範囲を有する)場合を示している。オブジェクトWの範囲であるボタンの領域31は、面積を有するため、ボタンのオブジェクトWの近傍に表示されるテキスト・オブジェクトt1の範囲である領域32と、一定の重なり領域33を有する(図3(a)右図参照)。このように、あるオブジェクトWとテキスト・オブジェクトt1との重なりが一定以上ある場合は、オブジェクトWの近傍に有するテキスト・オブジェクトt1を代替テキストtと推定することができる。

【0025】

他方、図3(b)は、オブジェクトWはボタンであるが、オブジェクトWのサイズがない(サイズがゼロである)場合を示している。例えば、Flashにより作成されたコンテンツのオブジェクトには、サイズがゼロのボタンのオブジェクトが存在する。この場合、オブジェクトWが有する相対位置を用いる。まず、ボタンを示すオブジェクトWの位置34とオブジェクトWの位置34から右下に表示されたテキスト・オブジェクトt1の領域32との位置関係を検討する。そして、オブジェクトWの位置34からテキスト・オブジェクトt1へ、最短となる直線を引いたときの長さDと、テキスト・オブジェクトt1の領域32の対角線の長さdとを比較し、D < dである場合にテキスト・オブジェクトt1を代替テキストtと推定することができる。これは、オブジェクトWについてのテキスト・オブジェクトt1は、一般的に、オブジェクトWに十分に近く、かつオブジェクトWの右下に存在する、との経験則に基づく。

## 【 0 0 2 6 】

図2に戻って、次に、ステップS13では、制御部10は、代替テキストを推定できたか否かを判断する。上述の図3(a)及び(b)のいずれにも該当するテキスト・オブジェクトが存在しない場合には、代替テキストが推定できていない。代替テキストが推定できた場合(ステップS13:Y)には、制御部10は、代替テキストが推定できたため、本処理を終了する。他方、代替テキストが推定できていない場合(ステップS13:N)には、制御部10は、ステップS15において、オブジェクトにコンテンツ作成者が付けた名前(シンボル)による代替テキストの推定を行う。

## 【 0 0 2 7 】

プログラマ等のコンテンツ作成者は、オブジェクトに任意に名前を付けてプログラミングに利用する。プログラムを組む上で操作の対象となるオブジェクトを参照するには、オブジェクトに名前を付ける必要があるためである。コンテンツ作成者は、例えば、ビデオを再生するためのボタンに、「play\_button」という名前を付ける。このようにして付けられた名前は、オブジェクトを端的に表すことが多いという経験則に基づき、名前を利用するのである。名前は、コンテンツ作成者自身の理解のために必要なのは当然のこと、コンテンツ作成者以外の人にも理解の助けになると考えられるため、オブジェクトの代替テキストとして利用する。ただし、プログラム上で操作されないオブジェクトには名前を付ける必要がないため、ランタイムの都合上「instance#」(#は1, 2, ...)等の名前が自動的に付くことがあるが、代替テキストとしては不適切である。そのため、「instance#」等の明らかにオブジェクトの意味を表さない名前は、予め記憶部(図示せず)に、例えばリスト形式で記憶することで、代替テキストから除外する。この名前は、実験環境制御部11による実験環境での実行時に、役割推定部13がインタプリタの助けを借りて取得し、視覚情報解析部12が名前に基づき代替テキストを推定する。

## 【 0 0 2 8 】

ステップS15の処理の後、制御部10は、本処理を終了する。本処理により、代替テキストのないオブジェクトに対して、代替テキストを推定することができる。

## 【 0 0 2 9 】

次に、アクセシビリティ情報をRoleにより推定する処理について説明する。図4は、オブジェクトのRoleを推定する処理のフローチャートを示す。図5から図7は、図4のフローチャートの内容を説明するための図である。

## 【 0 0 3 0 】

まず、図4のステップS31において、制御部10(実験環境制御部11、役割推定部13)は、オブジェクトの実行時の出力及び内部状態からRoleを推定する。

## 【 0 0 3 1 】

ここで、Roleの推定方法について説明する。Flash及びDHTMLは、実行時に型情報が決定されるECMAScriptという言語を用いているため、静的にアクセシビリティ情報を検出することは難しい。そこで、実験環境でスクリプトを実行する際に解析することで、シミュレーションによって参照モデルからアクセシビリティ情報を推測する。シミュレーションは、参照モデルと実行時モデルとの等価性や包含関係を検証するための技法である。本発明では、オブジェクトに対して実際にクリック等の入力を行い、状態がどのように変化するか、また、それが参照モデルとどのように対応するかを検証することによって、アクセシビリティ情報を推定する。

## 【 0 0 3 2 】

より詳細に説明すると、まず、アクセシビリティ情報を推測したいオブジェクトの内部の状態に着目する。ECMAScriptでは、オブジェクトの状態sは、プロパティ(property)とバリュー(value)とのペアで表現されている。これ以外の実行時に用いられる状態は、全て「環境(Environment)」とみなす。図5(a)に、シミュレーションにおける実行モデルの概念図を示す。オブジェクトのプロパティとバリューとのペアが、オブジェクトの状態41であり、オブジェクトの状態41以外が

10

20

30

40

50

環境 4 2 である。

【 0 0 3 3 】

また、本発明では、オブジェクトのアクセシビリティに必要な R o l e や状態を推定するために、参照モデルを用意する。参照モデルは、例えば、図 5 ( b ) 及び ( c ) に示すようなミーリ・マシン ( M e a l y M a c h i n e ) と等価なモデルで定義することができる。図 5 ( b ) 及び ( c ) は、一般的な意味でのボタンの参照モデルの例である。図 5 ( b ) は、オブジェクトが p 1 から「 c l i c k 」されたことによって、別の状態 p 2 に遷移したことを意味している。なお、「 c l i c k 」の次に書かれた「 」は、出力を表すが、「 」自体は外部に何も出力しないことを示す。図 5 ( c ) は、オブジェクトが p 1 から「 c l i c k 」されたことによって外部に何らかの出力「 o 1 」をしたことを意味している。オブジェクトの状態は明示的にモデル中に指定することもできるが、モデル中で別々の状態は別々のオブジェクトの状態を指すとゆるく指定することもできる。

10

【 0 0 3 4 】

実際のオブジェクトに対しても、「 c l i c k 」イベントを送るシミュレーションを行うことで、これらのモデルとの適合関係を見て、ボタンであるかどうかを推測することができる。例えば、図 5 ( b ) のモデルにしたがっているボタンは、クリックするとオブジェクトそのものが何らかの影響を与えることがわかる。図 5 ( c ) のモデルは、クリックすることによって環境に対して何らかの出力を出すものであることがわかる。例えば、出力が「 g e t U R L 」メソッドの呼び出しであったならば、ブラウザ上では何らかのページ遷移が発生することがわかる。あるボタンがページ遷移を引き起こすことは、視覚障害者にとっては非常に有益な情報である。このとき、U R L ( U n i f o r m R e s o u r c e L o c a t o r ) を取得して、読み上げを行うことができれば、視覚障害者にとってどのようなボタンであるかを理解する助けとなる。図 5 ( b ) 及び ( c ) において例として示した参照モデルは、参照モデル記憶部 1 5 ( 図 1 参照 ) に記憶される。

20

【 0 0 3 5 】

図 6 に、参照モデル記憶部 1 5 に記憶される参照モデルテーブル 1 5 0 を示す。参照モデルテーブル 1 5 0 は、R o l e 1 5 1 と参照モデル 1 5 2 とからなり、R o l e 毎に参照モデルが 1 又は複数記憶されている。例えば、R o l e 1 5 1 が「ボタン」である場合は、「 c l i c k 」をするモデルと「 o n P r e s s 」をするモデルの 2 つの参照モデルが定義されている。なお、図 4 のステップ S 3 1 により推定した R o l e の検証方法については、後述する。

30

【 0 0 3 6 】

図 4 に戻って、次に、ステップ S 3 3 では、制御部 1 0 ( 視覚情報解析部 1 2 ) は、オブジェクトとテキスト・オブジェクトとの視覚的關係により、代替テキストが推定されたか否かを判断する。オブジェクトとテキスト・オブジェクトとの視覚的關係により代替テキストを推定する処理は、上述の図 2 のステップ S 1 1 の結果を用いる。代替テキストが推定された場合 ( ステップ S 3 3 の処理で Y E S が判断される場合 ) には、制御部 1 0 は、本処理を終了する。他方、代替テキストが推定されなかった場合 ( ステップ S 3 3 の処理で N O が判断される場合 ) には、制御部 1 0 は、ステップ S 3 5 に処理を移す。

40

【 0 0 3 7 】

ステップ S 3 5 では、制御部 1 0 は、R o l e が、オブジェクトに関するクリックによる操作が必要な R o l e ではないと判断されたか否かを判断する。これは、オブジェクトとテキスト・オブジェクトとの視覚的關係により、代替テキストが推定されなかったオブジェクト W について、次の条件で、オブジェクト W に関するクリックのみで動作する R o l e ( 例えば、ボタンやチェックボックス、ラジオボタン等 ) ではないと判断する。

## 【数 2】

$$(\text{Size}(W) < T_s) \cup (\text{Height}(W) \times K_h > \text{Width}(W)) \cup (\text{Width}(W) \times K_w > \text{Height}(W))$$

但し、

$T_s, K_h, K_w$ : サイズ、大きさに関する定数

$\text{Size}(obj)$ : オブジェクト  $obj$  の大きさ、幅×高さ

$\text{Width}(obj)$ : オブジェクトの幅

$\text{Height}(obj)$ : オブジェクトの高さ

この式の意味するところは、ボタン等の  $\text{Role}$  を持つオブジェクトは、ユーザの操作対象となる。そのため、 $\text{Role}$  が、ある一定以上の大きさを持ち、かつ形状が細長すぎないことを条件としたものである。例えば、図 7 に示すオブジェクトのうち、タイムライン 51、ボリューム 52 及び測定表示 53 は、各オブジェクトを操作するための、 $\text{onPress}$  等のイベントハンドラを持つ。上述のボタン等の  $\text{Role}$  を持つオブジェクトも、同様に  $\text{onPress}$  等のイベントハンドラを持つため、何の考慮もしないと、ボタンではない種類のオブジェクトをボタンと誤認してしまう。そのため、ある一定以上の大きさを持たないもの、又は、ある一定以上細長い形状をしたオブジェクトは、ボタン等の  $\text{Role}$  を持つオブジェクトではないと推定する。

10

## 【0038】

ステップ S35 で、クリックによる操作が必要ではないと判断された場合（ステップ S35 の処理で YES が判断される場合）には、制御部 10 は、処理をステップ S37 に移す。他方、クリックによる操作が必要であると判断された場合（ステップ S35 の処理で NO が判断される場合）には、制御部 10 は、本処理を終了する。

20

## 【0039】

ステップ S37 では、制御部 10 は、ステップ S31 で推定された  $\text{Role}$  の候補から、当該オブジェクトをはずす処理を行う。その後、制御部 10 は、本処理を終了する。本処理により、オブジェクトの  $\text{Role}$  を推定することができ、推定された  $\text{Role}$  をアクセシビリティ情報とすることができる。

## 【0040】

次に、図 4 のステップ S31 で推定した  $\text{Role}$  を検証するフローチャートについて説明する。図 8 から図 10 は、実際のオブジェクトが参照モデルを含んでいるかをチェックするフローチャートを示す。ここで、チェック対象となるオブジェクトを、対象オブジェクトとする。

30

## 【0041】

まず、図 8 のステップ S51 において、制御部 10（実験環境制御部 11）は、対象コンテンツの初期化を行う。具体的には、 $\text{Sandbox}$  環境で、対象オブジェクト  $o$  を含むコンテンツを、検証を行いたい状態に戻す。そして、 $\text{Oex}$  で示される出力リストを空にする。 $\text{Oex}$  は、実際のオブジェクトを実行したことにより出力された情報を格納する領域である。

## 【0042】

40

次に、ステップ S52 では、制御部 10 は、参照モデルの初期化を行う。具体的には、検証を行う参照モデルの現在の状態を初期状態 ( $\text{init}$ ) にし、 $\text{Oref}$  で示される出力リストを空にする。 $\text{Oref}$  は、参照モデルの出力情報を格納する領域である。

## 【0043】

次に、ステップ S53 では、制御部 10（役割推定部 13）は、初期状態である参照モデルの状態  $s$  についての検証を行う。ここで、参照モデルの状態  $s$  についての検証について、図 9 を用いて説明する。

## 【0044】

まず、図 9 のステップ S101 では、制御部 10 は、対象オブジェクト  $o$  の状態が辞書に記録済か否か、つまり、対象オブジェクト  $o$  の状態が記憶部に記憶された辞書に既に記

50

録されているか否かを判断する。記録済である場合（ステップS 1 0 1 : Y）には、制御部 1 0 は、処理をステップS 1 0 2 に移し、他方、記録済でない場合（ステップS 1 0 1 : N）には、制御部 1 0 は、処理をステップS 1 0 3 に移す。

【 0 0 4 5 】

ステップS 1 0 2 では、制御部 1 0 は、対象オブジェクトと参照モデルとを照合し、参照モデルの現在の状態  $s$  と、辞書に記録されている状態  $s'$  とが一致するか否かを判断する。状態が一致する場合（ステップS 1 0 2 : Y）には、制御部 1 0 は、処理をステップS 1 0 4 に移す。他方、状態が一致しない場合（ステップS 1 0 2 : N）には、制御部 1 0 は、処理をステップS 1 0 9 に移し、参照モデルには適合しないと判定 ( f a l s e ) した後に、本検証処理を終了する。

10

【 0 0 4 6 】

ステップS 1 0 3 では、制御部 1 0 は、辞書に対象オブジェクトの新規状態を記録する。具体的には、辞書に新たな項目を追加し、対象オブジェクト  $o$  の状態と、参照モデルの現在の状態  $s$  とを結び付けて記録する。

【 0 0 4 7 】

ステップS 1 0 4 では、制御部 1 0 は、現在の状態から次にとりうる遷移の集合を取得する。そして、各々の遷移に対して、出力  $out$ 、入力  $in$ 、次の状態  $s$  を取得する。これは、次にとりうる遷移が複数ある場合は、その各々に対して処理を行う。次に、ステップS 1 0 5 では、制御部 1 0 は、ステップS 1 0 4 で取得したデータを用いて遷移の評価を行う。ここで、遷移の評価について、図 1 0 を用いて説明する。

20

【 0 0 4 8 】

図 1 0 のステップS 1 2 1 では、制御部 1 0 は、評価しようとする遷移が検証済か否かを判断する。検証済である場合（ステップS 1 2 1 : Y）には、制御部 1 0 は、処理をステップS 1 2 9 に移し、参照モデルに適合すると判定 ( t r u e ) した後に、本評価処理を終了する。他方、検証済ではない場合（ステップS 1 2 1 : N）には、制御部 1 0 は、処理をステップS 1 2 2 に移す。

【 0 0 4 9 】

ステップS 1 2 2 では、制御部 1 0 は、S a n d b o x 環境を保存する。具体的には、O r e f 及びO e x を保存した上で、O r e f に対してステップS 1 0 5 からパラメータである出力  $out$  を追加する。次に、ステップS 1 2 3 では、対象オブジェクト  $o$  への入力の送出行う。具体的には、対象オブジェクト  $o$  に対して入力  $in$  を送出し、出力の集合をO e x に追加する。

30

【 0 0 5 0 】

次に、ステップS 1 2 4 では、ステップS 1 2 3 によって得られる状態  $s$  について、検証を行う。つまり、再度、図 9 を呼び出す。つまり、処理を入れ子とすることで、検証済となるまで評価と遷移を繰り返して、対象オブジェクト  $o$  を検証する。次に、ステップS 1 2 5 では、評価結果が適合 ( t r u e ) か否かを判断する。評価結果が適合すると判断された場合（ステップS 1 2 5 : Y）には、制御部 1 0 は、処理をステップS 1 2 6 に移す。他方、評価結果が適合しないと判断された場合（ステップS 1 2 5 : N）には、制御部 1 0 は、処理をステップS 1 2 7 に移し、参照モデルには適合しないと判定 ( f a l s e ) した後、本遷移処理を終了する。

40

【 0 0 5 1 】

他方、ステップS 1 2 6 では、制御部 1 0 は、O r e f がO e x と等しいか又はO r e f がO e x に含まれるか否かを判断する。O e x がO r e f に含まれるとしたのは、O e x の方が、イベントが発生した際に生じる状態が様々存在することを考慮したからである。上記の条件に合致した場合（ステップS 1 2 6 : Y）には、制御部 1 0 は、処理をステップS 1 2 8 に移す。他方、上記の条件に合致しなかった場合（ステップS 1 2 6 : N）には、制御部 1 0 は、上述のステップS 1 2 7 の処理を行った後、本遷移処理を終了する。

【 0 0 5 2 】

50

ステップS128では、制御部10は、Sandbox環境を復帰する。すなわち、保存されたOref及びOexを復帰する。その後、上述のステップS129の処理を行い、本評価処理を終了する。

【0053】

図9に戻って、ステップS106において、制御部10は、遷移の評価結果が適合(true)か否かを判断する。遷移の評価結果が適合する場合(ステップS106:Y)には、制御部10は、処理をステップS107に移す。他方、遷移の評価結果が適合しない場合(ステップS106:N)には、制御部10は、上述のステップS109の処理を行い、本検証処理を終了する。

【0054】

ステップS107では、制御部10は、全ての遷移について行ったか否かを判断する。全てとは、上述のステップS104にて次にとりうる遷移が複数存在した場合に、その全てをいう。全ての遷移について行った場合(ステップS107:Y)には、処理をステップS108に移す。他方、全ての遷移について行っていない場合(ステップS107:N)には、処理をステップS104に移し、他の遷移についても評価処理を行う。

【0055】

ステップS108では、制御部10は、参照モデルに適合すると判定(true)し、本検証処理を終了する。

【0056】

図8に戻って、ステップS54では、制御部10は、評価結果が適合(true)か否かを判断する。検証の評価結果が適合である場合(ステップS54:Y)には、制御部10は、ステップS55にて参照モデルに適合すると判定(true)した後、本処理を終了する。他方、検証の評価結果が適合でない場合(ステップS54:N)には、制御部10は、ステップS56にて参照モデルに適合しないと判定(false)した後、本処理を終了する。

【0057】

図8から図10で説明したフローチャートをより理解するため、具体例を示す。図11は、checkboxの参照モデルである。この参照モデルについて、図8から図10で説明したフローチャートを適用する。ここで、検証するコンテンツのcheckboxを対象オブジェクトoとする。この対象オブジェクトoは、例えば、checkboxをon/offするたびに、別window側でコンテンツがロードされて表示されたり、表示されなくなったりするものである。

【0058】

1. 対象コンテンツの初期化。これは、対象オブジェクトoがdragやマウス操作等を受け付ける状態にまでコンテンツの状態を戻すことをさす。また、Oexを空にする。

2. 参照モデルの初期化。初期状態はp1であるため、現在の状態をp1とする。また、Orefを空にする。

3. check(o, p1)を評価する。以降、check()の説明に移る。

4. 対象オブジェクトoの状態を記録する。例えば、oにはcheckedプロパティが定義されており、初期値がfalseであったとする。そこで、checked:falseのペアを状態として記録する。

5. 先ほどの状態(checked:false)は、記録済みではないため、辞書に新たに項目を追加し、現在の状態であるp1と結び付けて記録する。

6. p1から遷移できる状態は図11よりp2のみであるため、trans(, click, p2)を評価する(ここでは出力がないことを示す)。以降trans()の説明に移る。

7. (click, p2)はまだ検証されていないため、評価を継続する。

8. Sandbox環境を保存し、Oref、Oexを保存する。Orefに対しては、outがであるため、何も追加しない。

9. 対象オブジェクトoに対して、clickを送出する。コンテンツが動作し、何ら

10

20

30

40

50

かの出力を出す。ここでは、例えば、`getURL`が出力されるものとする。そこで、`Oex = { getURL }`となる。

10 . 再帰的に`check(o, p2)`を評価する。以降、同様に`check()`の説明に移る。

11 . 対象オブジェクトの状態を記録する。ここで、`checked`プロパティの値が`true`であったとする。そこで、`checked : true`のペアを状態として記録する。

12 . 先ほどの状態(`checked : true`)は、記録済みではないため、辞書に新たに項目を追加し、現在の状態である`p2`と結び付けて記録する。

13 . `p2`から遷移できる状態は、図11より`p1`のみであるため、`trans( , click, p1)`を評価する。以降、再度`trans()`アルゴリズムの説明に移る。 10

14 . (`click, p1`)はまだ検証されていないため、評価を継続する。

15 . `Sandbox`環境を保存し、`Oref`、`Oex`を保存する。`Oref`に対しては、`out`が であるため、何も追加しない。

16 . 対象オブジェクトに対して、`click`を送出する。コンテンツが動作し、何らかの出力を出す。ここでは、例えば、`getURL`が出力されるものとする。そこで、`Oex = { getURL , getURL }`となる。

17 . 再帰的に`check(o, p1)`を評価する。以降、同様に`check()`の説明に移る。

18 . 対象オブジェクトの状態を記録する。ここで、`checked`プロパティの値が`false`であったとする。そこで、`checked : false`のペアを状態として記録する。 20

19 . 先ほどの状態(`checked : true`)は、記録済みであり、辞書中に`p1`として記録されており、現在の状態である`p1`と一致するため、評価は継続される。

20 . `p1`から遷移できる状態は、`p2`のみであるため、`trans( , click, p2)`を評価する。以降、再度`trans()`アルゴリズムの説明に移る。

21 . (`click, p2`)は検証済みであるため、検証は適合となる。

22 . 3番目の`check()`アルゴリズムの評価に戻る。結果は適合であり、全ての遷移について行ったため適合となる。

23 . 2番目の`trans()`の評価に戻る。`check()`の結果は適合であるため、`Oref Oex`をチェックする。`Oref =`  であるため、評価は継続される。 30

24 . `Sandbox`環境を元に戻す。`Oref`及び`Oex`を元に戻す。`Oref =`  となり、`Oex = { getURL }`となる。最終的に、`trans()`は適合となる。

25 . 2番目の`check()`アルゴリズムの評価に戻る。結果は適合であり、全ての遷移について行ったため適合となる。

26 . 1番目の`trans()`の評価に戻る。`check()`の結果は適合であるため、`Oref Oex`をチェックする。`Oref =`  であるため、評価は継続される。

27 . `Sandbox`環境を元に戻す。`Oref`及び`Oex`を元に戻す。`Oref =`  となり、`Oex =`  となる。最終的に、`trans()`は適合となる。

28 . 1番目の`check()`アルゴリズムの評価に戻る。結果は適合であり、全ての遷移について行ったため適合となる。 40

29 . 1番目の`check()`アルゴリズムが適合と返してきたため、最終的に対象オブジェクトは`checkbox`参照モデルに適合していると判定する。

#### 【0059】

次に、上述の説明により得ることができるアクセシビリティを、`Flash`のコンテンツの場合を例に説明する。図12は、`Flash`コンテンツにおけるアクセシビリティ情報の取得方法を示す図である。

#### 【0060】

`Flash`は、ファイル・フォーマットのライセンス上、実験環境を作成することが難しい。ここで説明するのは、`Flash`プレーヤである`Flash VM`に実験環境を構 50

築できない場合であっても、Flashコンテンツの内部構造を解析してアクセシビリティ情報を推測する方法である。ここで、Webブラウザは、プロキシを経由してインターネットに接続する。方法の詳細は、以下の通りである。

#### 【0061】

まず、Webブラウザはインターネットに対してHTMLコンテンツの要求を行い、HTMLコンテンツを得る(図中(1))。次に、取得したHTMLコンテンツの<object>タグにより、Flashプレーヤが作成される(図中(2))。そして、Flashプレーヤは、Flashコンテンツに対する要求を行う過程で、プロキシからコンテンツ・ローダ(Content Loader)を得る(図中(3))。コンテンツ・ローダは、リペア・スクリプト(Repair Script)を要求し、プロキシからリペア・スクリプトを得る(図中(4))。また、コンテンツ・ローダが、Flashコンテンツを要求し、インターネットからFlashコンテンツを得る(図中(5))。

#### 【0062】

ここで、図中(4)でのリペア・スクリプトによりObjectのprototype propertyに関数を挿入することで、内部オブジェクト・モデルの情報の取得及び実験環境を構築する。Flashの内部状態と外部への出力を観測するため、置き換えが可能な副作用を持つ関数をフックすることで、値が変更される等の副作用が起こらないようにする。また、プロパティの変更や、視覚情報は、ActionScriptでのwatch関数を利用することによって観測することで、アクセシビリティ情報を推測する。

#### 【0063】

上述の説明により得たアクセシビリティ情報の、Flashコンテンツでの出力について説明する。Flashコンテンツの各オブジェクトには\_\_accPropsや\_\_accImplと呼ばれる属性があり、Roleや代替テキスト等のアクセシビリティのための情報をセットすることができる。したがって、これらの属性を設定することで、推測したアクセシビリティ情報を出力することができる。出力したアクセシビリティ情報は、スクリーン・リーダーを介してユーザに音声で提供する。

#### 【0064】

しかし、Flashプレーヤにおいては、ボタンRoleを適切に設定するために、やや特別な処理が必要となる。それは、例えば、Flashコンテンツでは、Buttonオブジェクトと呼ばれるボタンRoleを表現するためのオブジェクトが存在し、これにおいて必要となる。Buttonオブジェクトは、Up-frameやOver-frameと呼ばれるフレームを持つ。Up-frame及びOver-frameは、それぞれ、マウスがオブジェクトに当たっていない状態及びマウスがオブジェクトに当たった状態のButtonのグラフィクスを保持している。図13に、Buttonオブジェクトと呼ばれるボタンのRoleを表現する例を示す。図13(a)は、Up-frame及びOver-frameのButtonオブジェクト64、テキスト・オブジェクト65、及びテキスト・オブジェクト65にButtonオブジェクト64を重ねた全体63を説明する図である。ここで、テキスト・オブジェクトは、例えば、リンク先情報が書かれた画像データである。Up-frameの場合、つまりマウスがオブジェクトに接していない場合は、矢印61として表示され、Over-frameの場合、つまりマウスがオブジェクトに接している場合は、手型矢印62として表示される。また、Over-frameでは、Buttonオブジェクトに下線が付される。図13(b)は、実際の使用例を示す。図中(A)のButtonオブジェクトは、手型矢印62と接することにより図中(B)で示すように、下線が引かれる。

#### 【0065】

Flashプレーヤは、Up-frameにグラフィックが設定されていない場合には、ボタンのサイズが0x0であると判断し、ボタンとしてアクセシビリティ情報を提供しない。このようなUp-frameの問題を持ったボタンが検出された場合、正常にアクセシビリティ情報が出力されるプロキシとなるButtonオブジェクトを作成し、その

10

20

30

40

50



プロキシとなるボタンに「click」イベントが送られた場合、対象となる（Up - frameの問題を持った）ボタンに対して「click」イベントを送る。

【0066】

次に、DHTMLのコンテンツの場合を例に説明する。DHTMLの環境では、スクリプトの実行は、ECMAScriptを実行するブラウザ上のECMAScript VMによって行われる。そのため、実験環境をVM上に構築することは容易である。すなわち、VM上で、オブジェクトを監視するための機構、及び外部環境への操作を制御できるようにする。図14は、DHTMLのコンテンツにおけるアクセシビリティ情報の取得方法を示す図である。図14に示すように、ECMAScript VMにフック・コードを入れ込み、そこからアクセシビリティ情報を得るようにする。具体的には、部分的なオブジェクト監視機構であれば、Webブラウザの1つであるMozillaで、拡張機能を実装することで行うことができる。また、視覚情報については、各オブジェクトのスタイル・プロパティを参照することによって取得する。

10

【0067】

上述の説明により得た推測したアクセシビリティ情報の、DHTMLコンテンツでの出力について説明する。DHTMLコンテンツでは、alt属性や、title属性を修正することによって代替テキストを出力することができる。また、オブジェクトのRole情報については、WAI - ARIAの仕様にしたがい、Role属性を追加することによって、出力することができる。したがって、これらの属性を設定することで、推測したアクセシビリティ情報を出力することができる。

20

【0068】

次に、図15及び図16を用いて、世界で有名なサイトのうちから、実際に本発明を利用した結果を説明する。図15は、アクセシビリティ情報を得る前（Before）と、アクセシビリティ情報を取得して自動的に修正を実施した後（After）での、代表的なスクリーン・リーダの読み上げの例を示す。また、図16は、（1）修正の前後でのボタンの総数、（2）出力されていないボタンの数、（3）キーボードで操作できず、出力すべきではないボタンの数、（4）代替テキストを持たないボタンの数を示している。これは、「x 1」及び「x 2」のあるWebページ上のコンテンツを用いる。「x 1」では、修正前には4つのボタンが出力されたが、どのボタンにも「Button」とあるだけで、何のボタンであるかを示す代替テキストはついていなかった。また、「x 2」では、19のボタンのうち15のボタンに代替テキストがなく、代替テキストのある4つのボタンのうち1つは、キーボードで操作できないボタンであった。これらのコンテンツを自動修正した結果、「x 1」では、出力されていなかった8つのボタンを出力することができ、それぞれのボタンに、代替テキストを付与することができた。しかし、修正前から出力されていた4つのボタンには適切な代替テキストを付与することができなかった。また、「x 2」では、15の代替テキストがないボタンのうち、12のボタンに代替テキストを付与することができた。「x 2」の場合では、キーボードで操作できないボタンを出力させないように修正することはできなかったが、全体として、自動的な修正によってアクセシビリティを改善させることが可能であることを証明することができた。

30

40

【0069】

本発明によって、自動的にFlashコンテンツやDHTMLコンテンツ等のRIAのコンテンツにおけるアクセシビリティを向上することが可能になり、スクリーン・リーダを使ってそれらのコンテンツにアクセスするユーザに大きなメリットをもたらすことができる。例えば、「x 1」及び「x 2」では、本発明を利用しない場合には、ほとんどのオブジェクトについて代替テキストが存在せず、本来クリックすることができないボタンにすら「ボタン」とスクリーン・リーダが読み上げる。そのため、ユーザは、事実上全く操作することができない。しかし、本発明を適用した場合には、図15及び図16に示すように、多くのアクセシブルではないオブジェクトに対しても、ボタンであることの適切な情報及び代替テキストを付与することに成功している。このため、スクリーン・

50

リーダの利用者も、これらのボタンを操作することができるようになる。

【0070】

本発明によって、全てのオブジェクトに対してアクセシビリティを改善させることができるとは限らない。しかし、現在インターネットで配信されているほとんどのFlashコンテンツやHTMLコンテンツは、アクセシブルではなく、コンテンツの作成者自身がアクセシビリティを改善させることは期待しにくい。本発明によって、そういったコンテンツに対しても、完全ではないにせよ、自動的にアクセシビリティを確保することができる。

【0071】

また、スクリーン・リーダを使うユーザへのメリットだけではなく、コンテンツの作成者にとっても、コンテンツをアクセシブルにするための手間を軽減させることができるというメリットがある。図17は、アクセシビリティ情報取得装置2をコンテンツの作成に用いる例を示す。アクセシビリティ情報取得装置2は、視覚情報解析部22及び役割推定部23から、実行による解析ログ28を取得し、アクセシビリティ情報取得装置2が有するログ解析部27でログを解析することにより、編集時内部モデル26にアクセシビリティ情報を出力する。コンテンツ作成者は、編集ツール9により編集時内部モデル26を用いてコンテンツを作成することにより、WAI-ARIAの仕様に仕上がったコンテンツを作成することができる。

10

【0072】

FlashコンテンツやHTMLコンテンツは、今後とも、リッチ・インターネット・アプリケーション(RIA)において、重要なクライアント技術となることが予想され、アクセシビリティの改善は重要な課題である。本発明は、上述で詳細を説明したように、(1)オブジェクトの視覚的表示情報及び画面上に表示されているテキスト情報、(2)オブジェクトに割り当てられているシンボル情報、(3)オブジェクトの出力及び内部状態の変化、の主に3つの情報を利用してオブジェクトのアクセシビリティ情報を推定できるものである。このような本発明は、自動的に既存コンテンツのアクセシビリティを改善させるとどまらず、今後開発されるコンテンツのアクセシビリティ確保にも貢献することができるため、本発明の重要性は、将来において更に大きくなるものと予測される。

20

【0073】

図18は、本発明の実施形態に係るアクセシビリティ情報取得装置1のハードウェア構成を示す図である。以下は、コンピュータを典型とする情報処理装置として全般的な構成を説明するが、専用機や組み込み型装置の場合、その環境に応じて必要最小限な構成を選択できることはいうまでもない。

30

【0074】

アクセシビリティ情報取得装置1は、CPU(Central Processing Unit)1010、バスライン1005、通信I/F1040、メインメモリ1050、BIOS(Basic Input Output System)1060、パラレルポート1080、USBポート1090、グラフィック・コントローラ1020、VRAM1024、音声プロセッサ1030、I/Oコントローラ1070、ならびにキーボード及びマウス・アダプタ等1100の入力手段を備える。I/Oコントローラ1070には、フレキシブル・ディスク(FD)ドライブ1072、ハード・ディスク1074、光ディスク・ドライブ1076、半導体メモリ1078、等の記憶手段を接続することができる。グラフィック・コントローラ1020には、表示装置1022が接続されている。また、オプションとして、音声プロセッサ1030には、増幅回路1032及びスピーカ1034が接続される。

40

【0075】

BIOS1060は、アクセシビリティ情報取得装置1の起動時にCPU1010が実行するブートプログラムや、アクセシビリティ情報取得装置1のハードウェアに依存するプログラム等を格納する。FDドライブ1072は、フレキシブル・ディスク1071からプログラム又はデータを読み取り、I/Oコントローラ1070を介してメインメモリ

50

1050又はハード・ディスク1074に提供する。

【0076】

光ディスク・ドライブ1076としては、例えば、DVD-ROMドライブ、CD-ROMドライブ、DVD-RAMドライブ、CD-RAMドライブを使用することができる。この際は各ドライブに対応した光ディスク1077を使用する必要がある。光ディスク・ドライブ1076は光ディスク1077からプログラム又はデータを読み取り、I/Oコントローラ1070を介してメインメモリ1050又はハード・ディスク1074に提供することもできる。

【0077】

アクセシビリティ情報取得装置1に提供されるコンピュータ・プログラムは、フレキシブル・ディスク1071、光ディスク1077、又はメモリカード等の記録媒体に格納されて利用者によって提供される。このコンピュータ・プログラムは、I/Oコントローラ1070を介して、記録媒体から読み出され、又は通信I/F1040を介してダウンロードされることによって、アクセシビリティ情報取得装置1にインストールされ実行される。コンピュータ・プログラムが情報処理装置に働きかけて行わせる動作は、既に説明した装置における動作と同一であるので省略する。

【0078】

上述のコンピュータ・プログラムは、外部の記憶媒体に格納されてもよい。記憶媒体としてはフレキシブル・ディスク1071、光ディスク1077、又はメモリカードの他に、MD等の光磁気記録媒体、テープ媒体を用いることができる。また、専用通信回線やインターネットに接続されたサーバシステムに設けたハード・ディスク又は光ディスク・ライブラリ等の記憶装置を記録媒体として使用し、通信回線を介してコンピュータ・プログラムをアクセシビリティ情報取得装置1に提供してもよい。

【0079】

以上の例は、アクセシビリティ情報取得装置1について主に説明したが、コンピュータに、情報処理装置で説明した機能を有するプログラムをインストールして、そのコンピュータを情報処理装置として動作させることにより上記で説明した情報処理装置と同様な機能を実現することができる。したがって、本発明において1つの実施形態として説明した情報処理装置は、方法及びそのコンピュータ・プログラムによっても実現可能である。

【0080】

本発明のアクセシビリティ情報取得装置1は、ハードウェア、ソフトウェア、又はハードウェア及びソフトウェアの組み合わせとして実現可能である。ハードウェアとソフトウェアの組み合わせによる実施では、所定のプログラムを有するコンピュータ・システムでの実施が典型的な例として挙げられる。係る場合、該所定のプログラムが該コンピュータ・システムにロードされ実行されることにより、該プログラムは、コンピュータ・システムに本発明に係る処理を実行させる。このプログラムは、任意の言語、コード、又は表記によって表現可能な命令群から構成される。そのような命令群は、システムが特定の機能を直接実行すること、又は(1)他の言語、コード、もしくは表記への変換、(2)他の媒体への複製、のいずれか一方もしくは双方が行われた後に、実行することを可能にするものである。もちろん、本発明は、そのようなプログラム自体のみならず、プログラムを記録した媒体を含むプログラム製品もその範囲に含むものである。本発明の機能を実行するためのプログラムは、フレキシブル・ディスク、MO、CD-ROM、DVD、ハード・ディスク装置、ROM、MRAM、RAM等の任意のコンピュータ可読媒体に格納することができる。係るプログラムは、コンピュータ可読媒体への格納のために、通信回線に接続する他のコンピュータ・システムからダウンロードしたり、他の媒体から複製したりすることができる。また、係るプログラムは、圧縮し、又は複数に分割して、単一又は複数の記録媒体に格納することもできる。

【0081】

以上、本発明を実施形態に則して説明したが、本発明は上述した実施形態に限るものではない。また、本発明の実施形態に記載された効果は、本発明から生じる最も好適な効果

10

20

30

40

50

を列挙したに過ぎず、本発明による効果は、本発明の実施形態又は実施例に記載されたものに限定されるものではない。

【図面の簡単な説明】

【0082】

【図1】本発明の実施形態に係るアクセシビリティ情報取得システムの構成の一例を示す図である。

【図2】オブジェクトの代替テキストを推定する処理のフローチャートである。

【図3】図2のフローチャートの内容を説明するための図である。

【図4】オブジェクトのRoleを推定する処理のフローチャートである。

【図5】図4のフローチャートの内容を説明するための図である。

10

【図6】図4のフローチャートの内容を説明するための図である。

【図7】図4のフローチャートの内容を説明するための図である。

【図8】実際のオブジェクトが参照モデルを含んでいるかをチェックするフローチャートである。

【図9】実際のオブジェクトが参照モデルを含んでいるかをチェックするフローチャートである。

【図10】実際のオブジェクトが参照モデルを含んでいるかをチェックするフローチャートである。

【図11】checkboxの参照モデルを示す図である。

【図12】Flashコンテンツにおけるアクセシビリティ情報の取得方法を示す図である。

20

【図13】ButtonオブジェクトのRoleを表現する例を示す図である。

【図14】DHTMLのコンテンツにおけるアクセシビリティ情報の取得方法を示す図である。

【図15】本発明を利用した結果を説明するための図である。

【図16】本発明を利用した結果を説明するための図である。

【図17】アクセシビリティ情報取得装置をコンテンツの作成に用いる例を示す図である。

【図18】アクセシビリティ情報取得装置のハードウェア構成を示す図である。

【符号の説明】

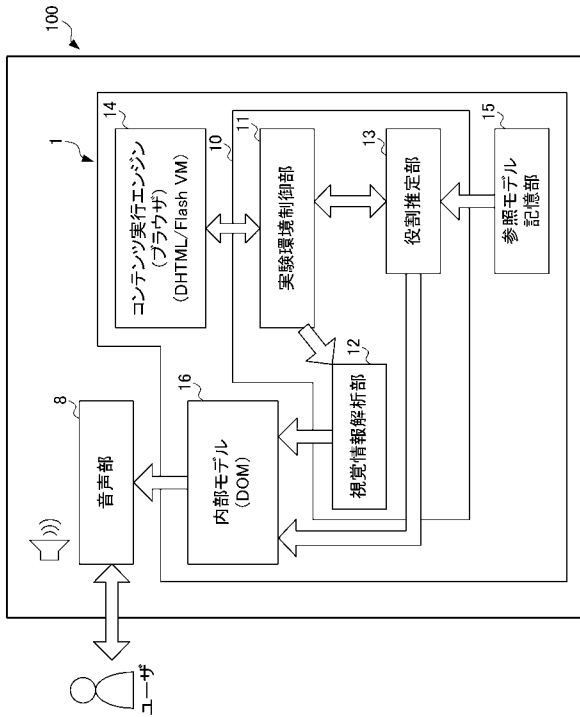
30

【0083】

- 1, 2      アクセシビリティ情報取得装置
- 8          音声部
- 10, 20    制御部
- 11, 21    実験環境制御部
- 12, 22    視覚情報解析部
- 13, 23    役割推定部
- 14, 24    コンテンツ実行エンジン
- 15, 25    参照モデル記憶部
- 16          内部モデル
- 26          編集時内部モデル
- 27          ログ解析部
- 28          解析ログ
- 150        参照モデルテーブル

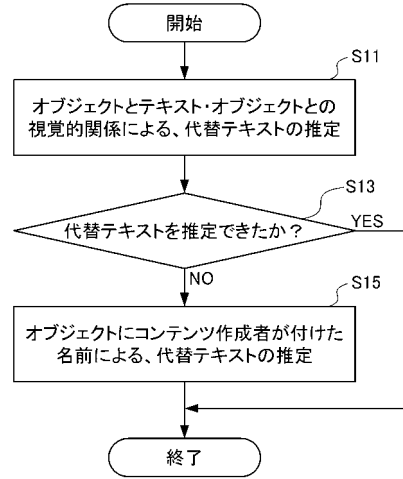
40

【図1】

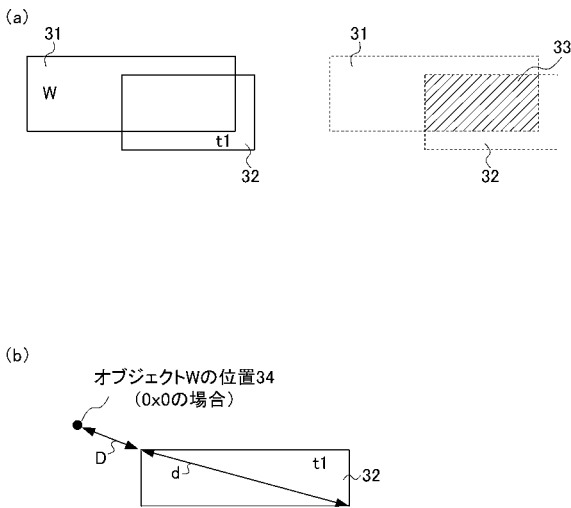


【図2】

オブジェクトの代替テキストを推定する処理のフローチャート

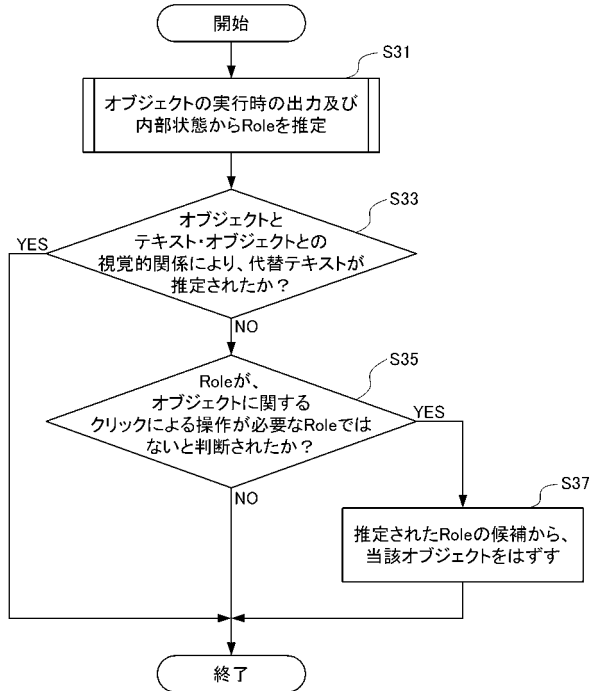


【図3】

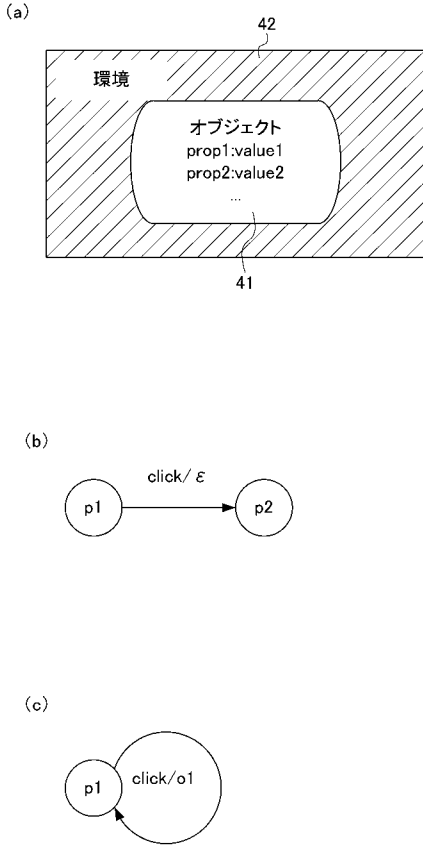


【図4】

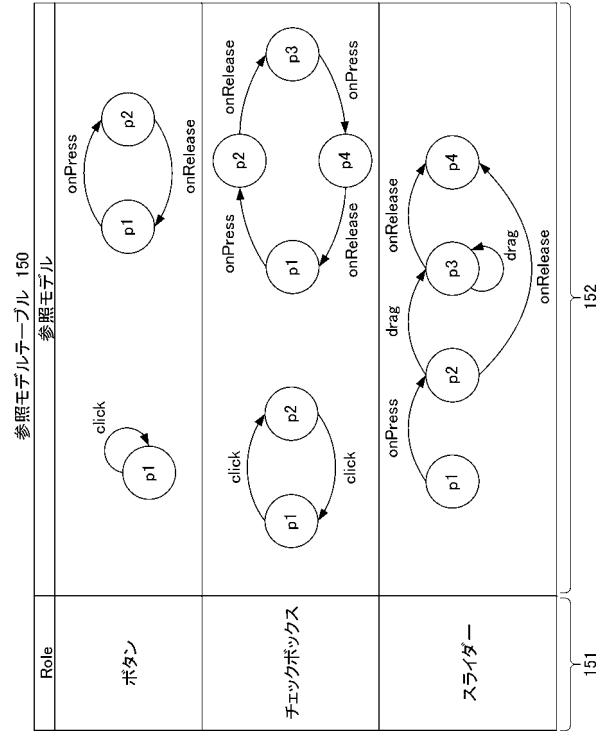
オブジェクトのRoleを推定する処理のフローチャート



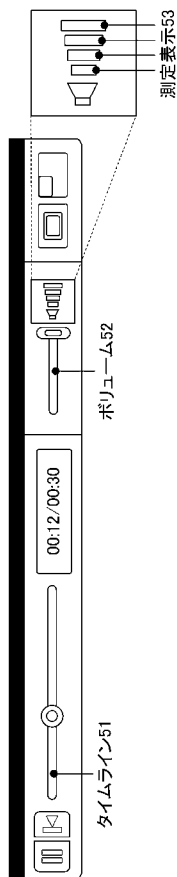
【図5】



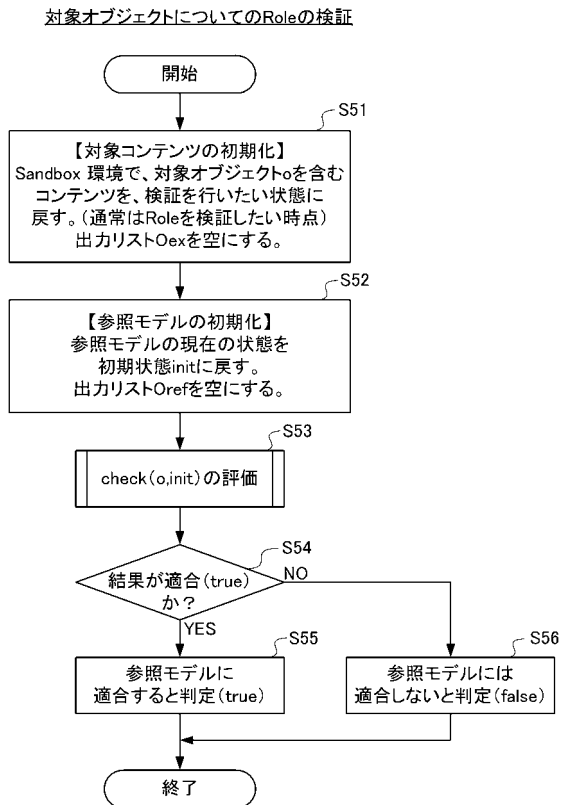
【図6】



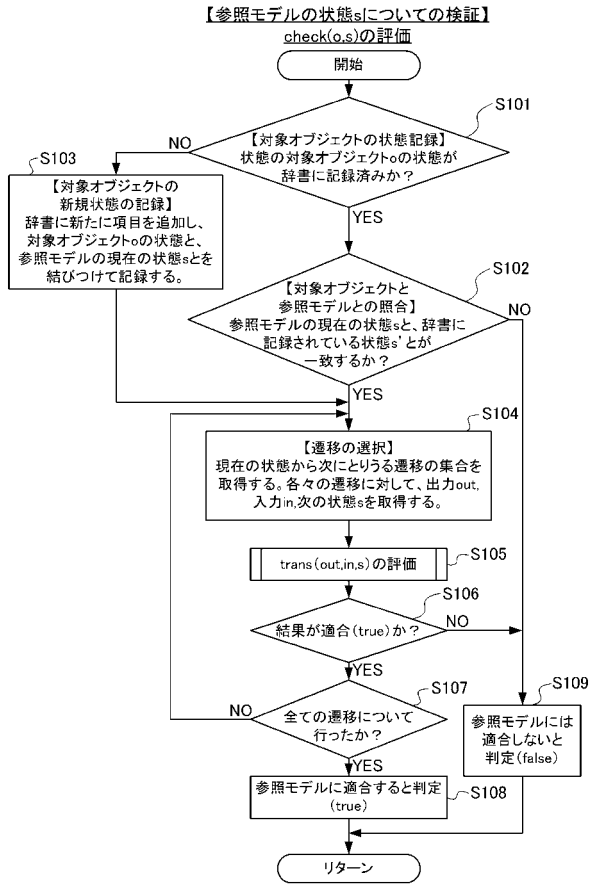
【図7】



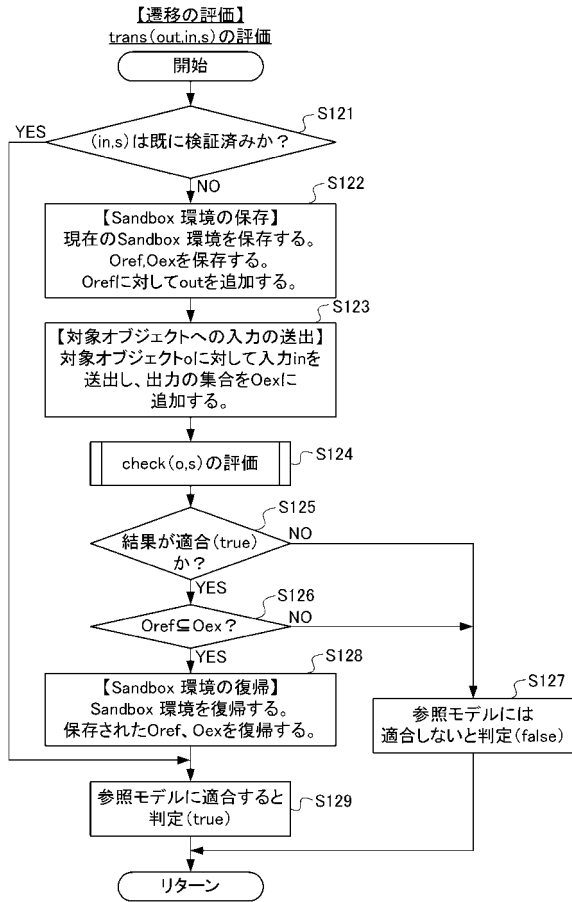
【図8】



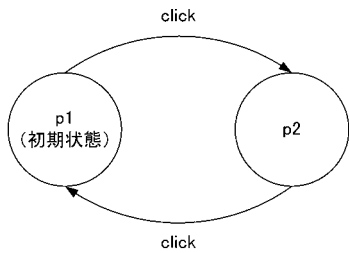
【図 9】



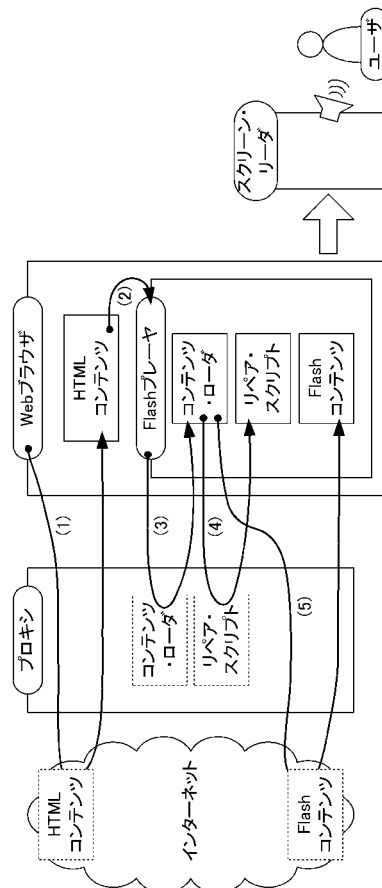
【図 10】



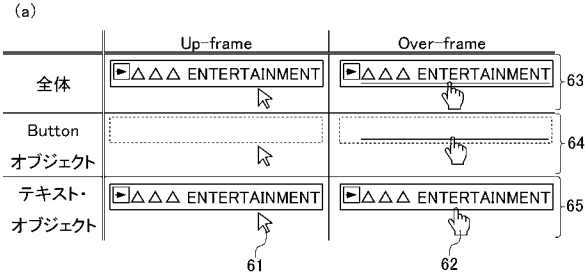
【図 11】



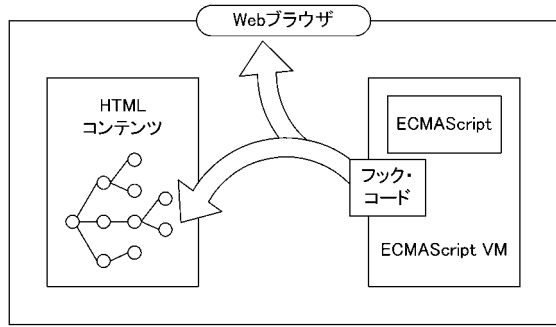
【図 12】



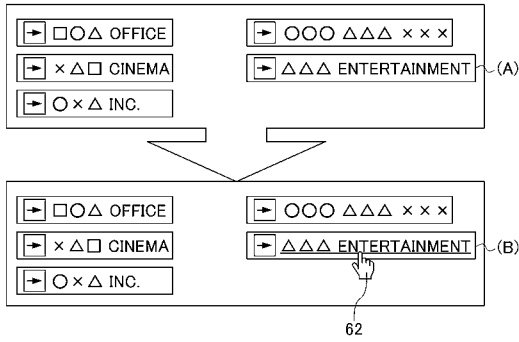
【 図 1 3 】



【 図 1 4 】



(b)



【 図 1 5 】

http://www.xOΔ1.com/corp/		http://OΔ□2.go.com/faires./movies./videos.html		OΔ□2-Videos	
Before	After	Before	After	Before	After
8 Button	View the Company Fact Sheet Button	4 Button	Meet the Fairies Button	T <sub>1</sub>	T <sub>1</sub>
9 Button	thumb2 Button	7 Button	Books Button	T <sub>2</sub>	T <sub>2</sub>
10 Button	thumb3 Button	9 Button	Movies Button	T <sub>3</sub>	T <sub>3</sub>
11 Button	thumb4 Button	24 Button	Games & Activities Button	T <sub>4</sub>	T <sub>4</sub>
-	thumb5 Button	28 Button	Create a Fairy Button	T <sub>5</sub>	T <sub>5</sub>
-	OΔ x Button	Home Button	Parents Button	T <sub>6</sub>	T <sub>6</sub>
-	x O OFFICE Button	Videos Button	Photo Gallery Button	T <sub>7</sub>	T <sub>7</sub>
-	x OΔ1 CABLE Button	Hisotry Button	Pause Button	T <sub>8</sub>	T <sub>8</sub>
-	xΔ□ CINEMA Button	44 Button	handle Button	N <sub>1</sub>	N <sub>1</sub>
-	O O O ΔΔΔ x x x Button	45 Button (Unnecessary)	Rewind Button	N <sub>2</sub>	N <sub>2</sub>
-	O x Δ INC. Button	47 Button	SoundController Button	N <sub>3</sub>	N <sub>3</sub>
-	ΔΔΔ ENTERTAINMENT Button	50 Button	Send to a friend Button	N <sub>4</sub>	N <sub>4</sub>
-		53 Button	item.1 Button	X <sub>1</sub>	X <sub>1</sub>
		56 Button	item.2 Button	X <sub>2</sub>	X <sub>2</sub>
		58 Button	item.3 Button	X <sub>3</sub>	X <sub>3</sub>
				X <sub>4</sub>	X <sub>4</sub>

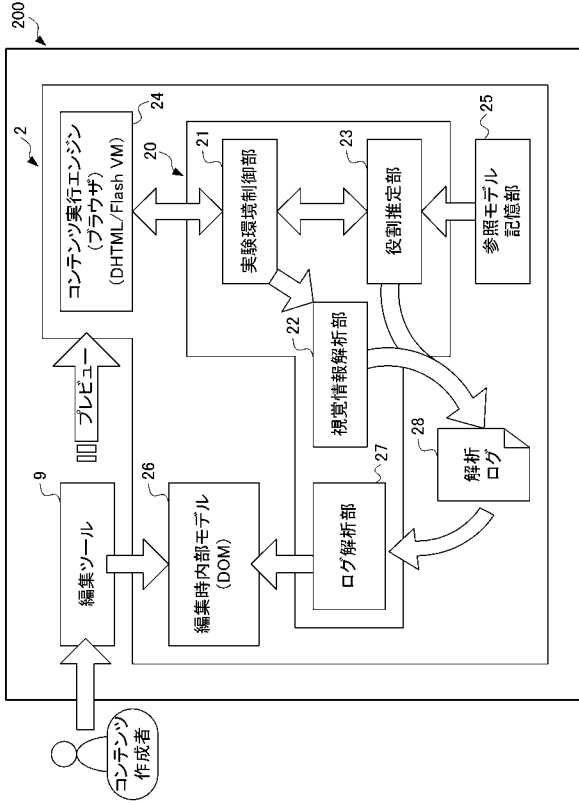
Status
Tn: Repaired with Text
Nn: Repaired with Instance Name
Xn: Not Repaired

【 図 1 6 】

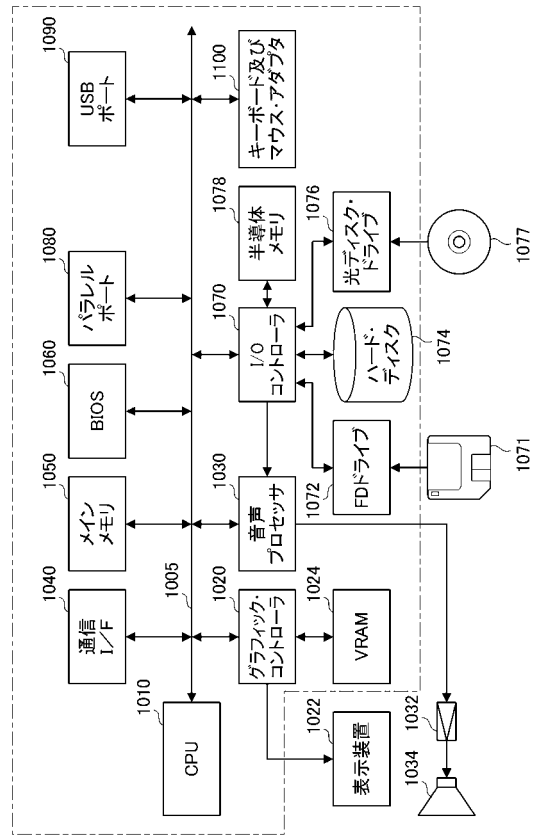
Page Title	xOΔ1: Home		OΔ□2-Videos	
	Before	After	Before	After
(1) Total Buttons	4	12	19	19
(2) Not Exposed Buttons	8	0	0	0
(3) Unnecessary Buttons	0	0	1	1
(4) Buttons with Invalid Alternative Text	12	4	15	3



【図 17】



【図 18】



## フロントページの続き

- (74)代理人 100086243  
弁理士 坂口 博
- (72)発明者 佐藤 大介  
神奈川県大和市下鶴間1623番地14 日本アイ・ピー・エム株式会社 東京基礎研究所内
- (72)発明者 宮下 尚  
神奈川県大和市下鶴間1623番地14 日本アイ・ピー・エム株式会社 東京基礎研究所内
- (72)発明者 高木 啓伸  
神奈川県大和市下鶴間1623番地14 日本アイ・ピー・エム株式会社 東京基礎研究所内
- (72)発明者 浅川 智恵子  
神奈川県大和市下鶴間1623番地14 日本アイ・ピー・エム株式会社 東京基礎研究所内

審査官 林 毅

- (56)参考文献 特開2007-249754(JP,A)  
特開2007-140642(JP,A)  
特開2007-172377(JP,A)  
福田、外4名,インタラクティブソフトウェア Webアクセシビリティ向上支援ツール: a Designer, コンピュータソフトウェア, 日本, 日本ソフトウェア科学会, 2006年10月26日, 第23巻、第4号, 第26-35頁  
福田、外3名, 視覚障害者のためのWebページ構造提示手法, ヒューマンインタフェース学会誌, 日本, ヒューマンインタフェース学会, 2003年11月25日, 第5巻、第4号, 第69-78頁  
森、外3名, Java GUIプログラムに対するアクセシビリティチェック, 電子情報通信学会技術研究報告, 日本, 社団法人電子情報通信学会, 2006年3月15日, 第105巻、第684号(WIT2005-83), 第53-58頁

(58)調査した分野(Int.Cl., DB名)

G06F 9/44

G06F 13/00