

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6977155号
(P6977155)

(45) 発行日 令和3年12月8日 (2021. 12. 8)

(24) 登録日 令和3年11月12日 (2021. 11. 12)

(51) Int. Cl.

F I

G O 6 F 3/0488 (2013.01)

G O 6 F 3/0488

請求項の数 20 (全 43 頁)

(21) 出願番号	特願2020-519079 (P2020-519079)	(73) 特許権者	502208397
(86) (22) 出願日	平成30年9月7日 (2018. 9. 7)		グーグル エルエルシー
(65) 公表番号	特表2021-504774 (P2021-504774A)		G o o g l e L L C
(43) 公表日	令和3年2月15日 (2021. 2. 15)		アメリカ合衆国 カリフォルニア州 94
(86) 国際出願番号	PCT/US2018/049977		043 マウンテン ビュー アンフィシ
(87) 国際公開番号	W02019/190587		アター パークウェイ 1600
(87) 国際公開日	令和1年10月3日 (2019. 10. 3)		1600 Amphitheatre P
審査請求日	令和2年5月26日 (2020. 5. 26)		arkway 94043 Mounta
(31) 優先権主張番号	15/938, 990		in View, CA U. S. A.
(32) 優先日	平成30年3月28日 (2018. 3. 28)	(74) 代理人	100108453
(33) 優先権主張国・地域又は機関	米国 (US)		弁理士 村山 靖彦
		(74) 代理人	100110364
			弁理士 実広 信哉
		(74) 代理人	100133400
			弁理士 阿部 達彦

最終頁に続く

(54) 【発明の名称】 コンテンツアイテムからの関数呼出しのトリガリングの制御

(57) 【特許請求の範囲】

【請求項 1】

コンテンツアイテム上で実施される対話を検証する方法であって、

1つまたは複数のプロセッサを有するデータ処理システムによって、コンテンツアイテムの第1の部分のコードにおいて、対話検証スクリプトを挿入するステップを含み、前記対話検証スクリプトが、

ユーザによる前記コンテンツアイテムとの第1の対話に対応する第1の対話イベントを検出することであって、前記第1の対話イベントが、前記コンテンツアイテムを情報リソース上に提示するように構成されたアプリケーションによって前記スクリプトに渡され、前記アプリケーションによる、前記第1の対話イベントの検出に応答して、前記情報リソース上に前記コンテンツアイテムが提示される、検出すること、

10

前記コンテンツアイテムの第2の部分のコードの実行を介して、あらかじめ決定されたアクションを実施するための関数呼出しの登録を識別することであって、前記関数呼出しが、前記アプリケーションが前記コンテンツアイテムに第2の対話イベントを渡すことに応じてトリガされ、前記第2の対話イベントが、ユーザによる前記コンテンツアイテムとの第2の対話に対応する、識別すること、

前記第1の対話イベントの検出と前記関数呼出しの前記登録の識別との間の時間があらかじめ決定されたしきい値未満であると決定すること、および

前記第1の対話イベントの検出と前記関数呼出しの前記登録の識別との間の前記時間が前記あらかじめ決定されたしきい値未満であるとの決定に応じて、前記アプリケーション

20

ンに前記あらかじめ決定されたアクションを実施させること
を行うように構成される、方法。

【請求項2】

前記対話検証スクリプトが、
touch-startイベントおよびtouch-endイベントの後で、第1のon-clickイベントを含む
前記第1の対話イベントを検出することであって、前記第1のon-clickイベントが、前記コ
ンテンツアイテムのon-clickイベントハンドラによって処理される、検出すること、

前記第1のon-clickイベントの後で、第2のon-clickイベントを含む前記第2の対話イベ
ントを検出することであって、前記第2のon-clickイベントが、前記コンテンツアイテム
の前記on-clickイベントハンドラによって処理される、検出すること、および

前記第2のon-clickイベントと前記第1のon-clickイベントとの間の前記時間が、前記あ
らかじめ決定されたしきい値未満であると決定すること
を行うようにさらに構成される、請求項1に記載の方法。

【請求項3】

前記対話検証スクリプトが、
touch-endイベントを含む前記第2の対話イベントを検出することであって、前記touch-
endイベントが、前記コンテンツアイテムのtouch-endイベントハンドラによって処理され
、前記touch-endイベントのための前記touch-endイベントハンドラが、前記あらかじめ決
定されたアクションを実施するための前記関数呼出しとともに構成される、検出すること

、
前記touch-endイベントの検出に応答して、前記あらかじめ決定されたアクションを実
行することなしに、前記あらかじめ決定されたアクションを実施するための前記関数呼出
しを登録すること、および

前記touch-endイベントの後で、on-clickイベントを含む前記第1の対話イベントを検出
することであって、前記on-clickイベントが、前記コンテンツアイテムのon-clickイベ
ントハンドラによって処理される、検出すること
を行うようにさらに構成される、請求項1に記載の方法。

【請求項4】

前記対話検証スクリプトが、
前記第1の対話イベントの前記検出に
前記第1の対話イベントの前記検出が、
前記関数呼出しの前記登録に先行すると決定すること、および

前記第1の対話イベントの前記検出が、前記関数呼出しの前記登録に先行するとの前記
決定に基づいて、前記あらかじめ決定されたしきい値を設定すること
を行うようにさらに構成される、請求項1に記載の方法。

【請求項5】

前記対話検証スクリプトが、
前記第1の対話イベントに対応する第1のタイプの対話を処理するための第1のイベント
ハンドラを介して、前記第1の対話イベントを検出すること、および

前記第1のタイプの対話、および前記第2の対話イベントに対応する第2のタイプの対話
のうちの少なくとも1つを処理するための、第2のイベントハンドラを介して、前記第2の
対話イベントを検出すること

を行うようにさらに構成される、請求項1に記載の方法。

【請求項6】

前記対話検証スクリプトが、前記アプリケーションに、アプリケーションプログラムイ
ンターフェース(API)関数呼出しを介して、前記あらかじめ決定されたアクションを実施
させるようにさらに構成される、請求項1に記載の方法。

【請求項7】

前記方法が、
前記データ処理システムによって、前記あらかじめ決定されたアクションを実施するた
めの複数のあらかじめ決定されたイベントハンドラのうちの1つに一致する、前記コンテ

ンツアイテムの前記第2の部分のコードにおける、前記第2の対話イベントを処理するためのイベントハンドラを識別するステップであって、前記イベントハンドラが、前記あらかじめ決定されたアクションを実施するようにあらかじめ構成される、ステップをさらに含む、

前記対話検証スクリプトを挿入するステップが、前記コンテンツアイテムの前記第1の部分のコードにおける、あらかじめ決定された複数のイベントハンドラに、前記イベントハンドラをアタッチするステップであって、前記第1の部分のコードが、前記第2の部分のコードの実行より前に、クライアント上でランしている前記アプリケーションによって実行されることになる、ステップをさらに含む、請求項1に記載の方法。

【請求項8】

コンテンツアイテム上で実施される対話を検証するためのシステムであって、

コンテンツアイテムの第1の部分のコードにおいて符号化され、1つまたは複数のプロセッサ上で実行可能であり、ユーザによる前記コンテンツアイテムとの第1の対話に対応する第1の対話イベントを検出するように構成された、イベント検出器であって、前記第1の対話イベントが、前記コンテンツアイテムを情報リソース上に提示するように構成されたアプリケーションによってスクリプトに渡され、前記アプリケーションによる、前記第1の対話イベントの検出に応答して、前記情報リソース上に前記コンテンツアイテムが提示される、イベント検出器と、

前記第1の部分のコードにおいて符号化され、前記1つまたは複数のプロセッサ上で実行可能であり、前記コンテンツアイテムの第2の部分のコードの実行を介して、あらかじめ決定されたアクションを実施するための関数呼出しの登録を識別するように構成された、登録ロガーであって、前記関数呼出しが、前記アプリケーションが前記コンテンツアイテムに第2の対話イベントを渡すことに応じてトリガされるように構成され、前記第2の対話イベントが、ユーザによる前記コンテンツアイテムとの対話に対応する、登録ロガーと、

前記第1の部分のコードにおいて符号化され、前記1つまたは複数のプロセッサ上で実行可能な、関数アクティベータであって、

前記第1の対話イベントの検出と前記関数呼出しの前記登録の識別との間の時間があらかじめ決定されたしきい値未満であると決定すること、および

前記第1の対話イベントの検出と前記関数呼出しの前記登録の識別との間の前記時間が前記あらかじめ決定されたしきい値未満であるとの決定に応じて、前記アプリケーションに前記あらかじめ決定されたアクションを実施させることを行うように構成された、関数アクティベータとを備えるシステム。

【請求項9】

前記イベント検出器が、

touch-startイベントおよびtouch-endイベントの後で、第1のon-clickイベントを含む前記第1の対話イベントを検出すること、

前記第1のon-clickイベントの後で、第2のon-clickイベントを含む前記第2の対話イベントを検出することであって、前記第2のon-clickイベントが、前記コンテンツアイテムのon-clickイベントハンドラによって処理され、前記第2のon-clickイベントのための前記イベントハンドラが、前記あらかじめ決定されたアクションを実施するための前記関数呼出しとともに構成される、検出することを行うようにさらに構成され、

前記登録ロガーが、前記第2のon-clickイベントを含む前記第2の対話イベントの前記検出に応じて、前記関数呼出しの前記登録を識別するようにさらに構成される、請求項8に記載のシステム。

【請求項10】

前記イベント検出器が、

touch-endイベントの後で、on-clickイベントを含む前記第1の対話イベントを検出することであって、前記on-clickイベントが、前記コンテンツアイテムのon-clickイベントハ

10

20

30

40

50

ンドラによって処理される、検出すること

前記touch-endイベントを含む前記第2の対話イベントを検出することであって、前記touch-endイベントが、前記コンテンツアイテムのtouch-endイベントハンドラによって処理され、前記touch-endイベントのための前記touch-endイベントハンドラが、前記あらかじめ決定されたアクションを実施するための前記関数呼出しとともに構成される、検出すること

を行うようにさらに構成され、

前記登録ロガーが、前記touch-endイベントハンドラを含む前記第2の対話イベントの前記検出に応じて、前記関数呼出しの前記登録を識別するようにさらに構成される、請求項8に記載のシステム。

10

【請求項11】

前記イベント検出器が、前記第1の対話イベントを、複数のアタッチされた対話イベントからの前記第2の対話イベントとペアにされるものとして識別することであって、前記複数のアタッチされた対話イベントが、on-clickイベントおよびtouch-endイベントのうちの少なくとも1つを含む、検出することを行うようにさらに構成され、

前記関数アクティベータが、前記第1の対話イベントを前記第2の対話イベントとペアにされるものとする識別に応答して、前記第1の対話イベントの検出と、関連付けられた前記関数呼出しの前記登録の識別との間の前記時間が、前記あらかじめ決定されたしきい値未満であると決定するようにさらに構成される、請求項8に記載のシステム。

20

【請求項12】

前記関数アクティベータが、

前記第1の対話イベントの前記検出が、前記関数呼出しの前記登録に先行すると決定すること、

前記第1の対話イベントの前記検出が、前記関数呼出しの前記登録に先行するとの前記決定に基づいて、前記あらかじめ決定されたしきい値を設定すること、および

前記第2の対話イベントの前記検出の後で、前記時間が前記あらかじめ決定されたしきい値未満であるとの決定に応じて、前記アプリケーションに前記あらかじめ決定されたアクションを実施させること

を行うようにさらに構成される、請求項8に記載のシステム。

30

【請求項13】

前記関数アクティベータが、

前記第1の対話イベントの前記検出が、前記関数呼出しの前記登録に後続すると決定すること、

前記第1の対話イベントの前記検出が、前記関数呼出しの前記登録に後続するとの前記決定に基づいて、前記あらかじめ決定されたしきい値を設定すること、および

前記第1の対話イベントの前記検出の後で、前記時間が前記あらかじめ決定されたしきい値未満であるとの決定に応じて、前記アプリケーションに前記あらかじめ決定されたアクションを実施させること

を行うようにさらに構成される、請求項8に記載のシステム。

40

【請求項14】

前記コンテンツアイテムの前記第1の部分のコードにおいて符号化された、前記イベント検出器、前記登録ロガー、および前記関数アクティベータのうちの少なくとも1つが、前記コンテンツアイテムの前記第2の部分のコードの前記実行より前に、前記アプリケーションによって実行可能である、請求項8に記載のシステム。

【請求項15】

コンテンツアイテム上で実施される対話を検証する方法であって、

コンテンツアイテムの第1の部分のコードにおいて符号化され、1つまたは複数のプロセッサ上で実行している、対話検証器によって、ユーザによる前記コンテンツアイテムとの第1の対話に対応する第1の対話イベントを検出するステップであって、前記第1の対話イベントが、前記コンテンツアイテムを情報リソース上に提示するように構成されたアプリ

50

ケーションによってスクリプトに渡され、前記アプリケーションによる、前記第1の対話イベントの検出に応答して、前記情報リソース上に前記コンテンツアイテムが提示される、ステップと、

前記対話検証器によって、前記コンテンツアイテムの第2の部分のコードの実行を介して、あらかじめ決定されたアクションを実施するための関数呼出しの登録を識別するステップであって、前記関数呼出しが、前記アプリケーションが前記コンテンツアイテムに第2の対話イベントを渡すことに応じてトリガされるように構成され、前記第2の対話イベントが、ユーザによる前記コンテンツアイテムとの第2の対話に対応する、ステップと、

前記対話検証器によって、前記第1の対話イベントの検出と前記関数呼出しの前記登録の識別との間の時間があらかじめ決定されたしきい値未満であると決定するステップと、

前記対話検証器によって、前記第1の対話イベントの検出と前記関数呼出しの前記登録の識別との間の前記時間が前記あらかじめ決定されたしきい値未満であるとの決定に応じて、前記アプリケーションに前記あらかじめ決定されたアクションを実施させるステップを含む方法。

【請求項16】

前記第1の対話イベントを検出するステップが、touch-startイベントおよびtouch-endイベントの後で、第1のon-clickイベントを含む前記第1の対話イベントを検出するステップをさらに含み、前記方法が、

前記対話検証器によって、前記第1のon-clickイベントの後で、第2のon-clickイベントを含む前記第2の対話イベントを検出するステップであって、前記第2のon-clickイベントが、前記コンテンツアイテムのon-clickイベントハンドラによって処理され、前記第2のon-clickイベントのための前記イベントハンドラが、前記あらかじめ決定されたアクションを実施するための前記関数呼出しとともに構成される、ステップをさらに含み、

前記関数呼出しの前記登録を識別するステップが、前記第2のon-clickイベントを含む前記第2の対話イベントの前記検出に応じて、前記関数呼出しの前記登録を識別するステップをさらに含み、請求項15に記載の方法。

【請求項17】

前記第1の対話イベントを検出するステップが、touch-endイベントの後で、on-clickイベントを含む前記第1の対話イベントを検出するステップであって、前記on-clickイベントが、前記コンテンツアイテムのon-clickイベントハンドラによって処理される、ステップをさらに含み、前記方法が、

前記対話検証器によって、前記touch-endイベントを含む前記第2の対話イベントを検出するステップであって、前記touch-endイベントが、前記コンテンツアイテムのtouch-endイベントハンドラによって処理され、前記touch-endイベントのための前記touch-endイベントハンドラが、前記あらかじめ決定されたアクションを実施するための前記関数呼出しとともに構成される、ステップをさらに含み、

前記関数呼出しの前記登録を識別するステップが、前記touch-endイベントハンドラを含む前記第2の対話イベントの前記検出に応じて、前記関数呼出しの前記登録を識別するステップをさらに含み、請求項15に記載の方法。

【請求項18】

前記方法が、

前記対話検証器によって、前記第1の対話イベントを、複数のアタッチされた対話イベントからの前記第2の対話イベントとペアにされるものとして識別するステップであって、前記複数のアタッチされた対話イベントが、on-clickイベントおよびtouch-endイベントのうちの少なくとも1つを含む、ステップをさらに含み、

前記時間が前記あらかじめ決定されたしきい値未満であると決定するステップが、前記

10

20

30

40

50

第1の対話イベントを前記第2の対話イベントとペアにされるものとする識別に応答して、前記第1の対話イベントの検出と、関連付けられた前記関数呼出しの前記登録の識別との間の前記時間が、前記あらかじめ決定されたしきい値未満であると決定するステップをさらに含む、請求項15に記載の方法。

【請求項19】

前記方法が、

前記対話検証器によって、前記第1の対話イベントの前記検出が、前記関数呼出しの前記登録に先行すると決定するステップと、

前記対話検証器によって、前記第1の対話イベントの前記検出が、前記関数呼出しの前記登録に先行するとの前記決定に基づいて、前記あらかじめ決定されたしきい値を設定するステップと

10

をさらに含む、

前記アプリケーションに前記あらかじめ決定されたアクションを実施させるステップが、前記第2の対話イベントの前記検出の後で、前記時間が前記あらかじめ決定されたしきい値未満であるとの決定に応じて、前記アプリケーションに前記あらかじめ決定されたアクションを実施させるステップをさらに含む、請求項15に記載の方法。

【請求項20】

前記方法が、

前記対話検証器によって、前記第1の対話イベントの前記検出が、前記関数呼出しの前記登録に後続すると決定するステップと、

20

前記対話検証器によって、前記第1の対話イベントの前記検出が前記関数呼出しの前記登録に後続するとの前記決定に基づいて、前記あらかじめ決定されたしきい値を設定するステップと

をさらに含む、

前記アプリケーションに前記あらかじめ決定されたアクションを実施させるステップが、前記第1の対話イベントの前記検出の後で、前記時間が前記あらかじめ決定されたしきい値未満であるとの決定に応じて、前記アプリケーションに前記あらかじめ決定されたアクションを実施させるステップをさらに含む、請求項15に記載の方法。

【発明の詳細な説明】

【技術分野】

30

【0001】

関連出願の相互参照

本出願は、その全体が参照により本明細書に組み込まれる、2018年3月28日に提出した米国特許出願第15/938,990号の利益および優先権を主張する。

【背景技術】

【0002】

インターネットなどのコンピュータネットワーク化された環境では、コンテンツプロバイダは、クライアントデバイス上で実行しているアプリケーション(たとえば、ウェブブラウザ)によって処理およびレンダリングされた情報リソース(たとえば、ウェブページ)に挿入されることになる、コンテンツアイテムを提供することができる。そのようなコンテンツアイテムおよび情報リソースとの対話は、クライアントデバイスに接続された入出力(I/O)デバイスを使用して行われ得る。

40

【発明の概要】

【課題を解決するための手段】

【0003】

少なくとも1つの態様は、コンテンツアイテム上で実施される対話を検証する方法を対象とする。1つまたは複数のプロセッサを有するデータ処理システムは、コンテンツアイテムの第1の部分のコードにおいて、対話検証スクリプトを挿入することができる。対話検証スクリプトは、コンテンツアイテムに起因する第1の対話イベントを検出することができる。第1の対話イベントは、コンテンツアイテムを提示するように構成されたアプリ

50

ケーションによってスクリプトに渡され得、アプリケーションによる、コンテンツアイテムとの対話の検出にตอบสนองして情報リソース上にコンテンツアイテムが提示される。対話検証スクリプトは、コンテンツアイテムの第2の部分のコードの実行を介して、あらかじめ決定されたアクションを実施するための関数呼出しの登録を識別することができる。関数呼出しは、アプリケーションがコンテンツアイテムに第2の対話イベントを渡すことに応じてトリガされ得る。第2の対話イベントは、コンテンツアイテムに起因し、コンテンツアイテムとの対話に応じて生成され得る。対話検証スクリプトは、第1の対話イベントの検出と、関数呼出しの登録の識別との間の時間が、あらかじめ決定されたしきい値未満であると決定することができる。対話検証スクリプトは、第1の対話イベントの検出と、関数呼出しの登録の識別との間の時間が、あらかじめ決定されたしきい値未満であるとの決定に応じて、アプリケーションにあらかじめ決定されたアクションを実施させることができる。

10

【0004】

いくつかの実装形態では、対話検証スクリプトは、touch-startイベントおよびtouch-endイベントの後で、コンテンツアイテムに起因する第1のon-clickイベントを含む第1の対話イベントを検出することができる。第1のon-clickイベントは、コンテンツアイテムのon-clickイベントハンドラによって処理され得る。いくつかの実装形態では、対話検証スクリプトは、第1のon-clickイベントの後で、コンテンツアイテムに起因する第2のon-clickイベントを含む第2の対話イベントを検出することができる。第2のon-clickイベントは、コンテンツアイテムのon-clickイベントハンドラによって処理され得る。いくつかの実装形態では、対話検証スクリプトは、第2のon-clickイベントと第1のon-clickイベントとの間の時間が、あらかじめ決定されたしきい値未満であると決定することができる。

20

【0005】

いくつかの実装形態では、対話検証スクリプトは、コンテンツアイテムに起因するtouch-endイベントを含む第2の対話イベントを検出することができる。touch-endイベントは、コンテンツアイテムのtouch-endイベントハンドラによって処理され得る。touch-endイベントのためのtouch-endイベントハンドラは、あらかじめ決定されたアクションを実施するための関数呼出しとともに構成され得る。いくつかの実装形態では、対話検証スクリプトは、touch-endイベントの検出にตอบสนองして、あらかじめ決定されたアクションを実行することなしに、あらかじめ決定されたアクションを実施するための関数呼出しを登録することができる。いくつかの実装形態では、対話検証スクリプトは、touch-endイベントの後で、コンテンツアイテムに起因するon-clickイベントを含む第1の対話イベントを検出することであって、on-clickイベントが、コンテンツアイテムのon-clickイベントハンドラによって処理される、検出することを行うことができる。

30

【0006】

いくつかの実装形態では、対話検証スクリプトは、第1の対話イベントの検出にตอบสนองして、第1の対話イベントの検出が、関数呼出しの登録に先行すると決定することができる。いくつかの実装形態では、対話検証スクリプトは、第1の対話イベントの検出が、関数呼出しの登録に先行するとの決定に基づいて、あらかじめ決定されたしきい値を設定することができる。いくつかの実装形態では、対話検証スクリプトは、第1の対話イベントに対応する第1のタイプの対話を処理するための第1のイベントハンドラを介して、コンテンツアイテムに起因する第1の対話イベントを検出することができる。いくつかの実装形態では、対話検証スクリプトは、第1のタイプの対話、および第2の対話イベントに対応する第2のタイプの対話のうちの少なくとも1つを処理するための、第2のイベントハンドラを介して、コンテンツアイテムに起因する第2の対話イベントを検出することができる。いくつかの実装形態では、対話検証スクリプトは、アプリケーションに、アプリケーションプログラムインターフェース(API)関数呼出しを介して、あらかじめ決定されたアクションを実施させることができる。

40

【0007】

いくつかの実装形態では、データ処理システムは、あらかじめ決定されたアクションを

50

実施するための複数のあらかじめ決定されたイベントハンドラのうちの1つに一致する、コンテンツアイテムの第2の部分のコードにおける、第2の対話イベントを処理するためのイベントハンドラを識別することができる。イベントハンドラは、あらかじめ決定されたアクションを実施するようにあらかじめ構成され得る。いくつかの実装形態では、対話検証スクリプトを挿入することは、コンテンツアイテムの第1の部分のコードにおける、あらかじめ決定された複数のイベントハンドラに、イベントハンドラをアタッチすることを含み得る。第1の部分のコードは、第2の部分のコードの実行より前に、クライアント上でランしているアプリケーションによって実行され得る。

【0008】

少なくとも1つの態様は、コンテンツアイテム上で実施される対話を検証するためのシステムを対象とする。システムは、コンテンツアイテムの第1の部分のコードにおいて符号化され、1つまたは複数のプロセッサ上で実行可能な、イベント検出器を含み得る。イベント検出器は、コンテンツアイテムに起因する第1の対話イベントを検出することができる。第1の対話イベントは、コンテンツアイテムを提示するように構成されたアプリケーションによってスクリプトに渡され得、アプリケーションによる、コンテンツアイテムとの対話の検出にตอบสนองして、情報リソース上にコンテンツアイテムが提示される。システムは、第1の部分のコードにおいて符号化され、1つまたは複数のプロセッサ上で実行可能な、登録ロガーを含み得る。登録ロガーは、コンテンツアイテムの第2の部分のコードの実行を介して、あらかじめ決定されたアクションを実施するための関数呼出しの登録を識別することができる。関数呼出しは、アプリケーションがコンテンツアイテムに第2の対話イベントを渡すことに応じてトリガされるように構成され得る。第2の対話イベントは、コンテンツアイテムに起因し、コンテンツアイテムとの対話に応じて生成され得る。システムは、第1の部分のコードにおいて符号化され、1つまたは複数のプロセッサ上で実行可能な、関数アクティベータを含み得る。関数アクティベータは、第1の対話イベントの検出と、関数呼出しの登録の識別との間の時間が、あらかじめ決定されたしきい値未満であると決定することができる。関数アクティベータは、第1の対話イベントの検出と、関数呼出しの登録の識別との間の時間が、あらかじめ決定されたしきい値未満であるとの決定に応じて、アプリケーションにあらかじめ決定されたアクションを実施させることができる。

【0009】

いくつかの実装形態では、イベント検出器は、touch-startイベントおよびtouch-endイベントの後で、コンテンツアイテムに起因する第1のon-clickイベントを含む第1の対話イベントを検出することができる。いくつかの実装形態では、イベント検出器は、第1のon-clickイベントの後で、コンテンツアイテムに起因する第2のon-clickイベントを含む第2の対話イベントを検出することができる。第2のon-clickイベントは、コンテンツアイテムのon-clickイベントハンドラによって処理され得る。第2のon-clickイベントのためのイベントハンドラは、あらかじめ決定されたアクションを実施するための関数呼出しとともに構成され得る。いくつかの実装形態では、登録ロガーは、コンテンツアイテムに起因する第2のon-clickイベントを含む第2の対話イベントの検出に応じて、関数呼出しの登録を識別することができる。

【0010】

いくつかの実装形態では、イベント検出器は、touch-endイベントの後で、コンテンツアイテムに起因するon-clickイベントを含む第1の対話イベントを検出することができる。on-clickイベントは、コンテンツアイテムのon-clickイベントハンドラによって処理され得る。いくつかの実装形態では、イベント検出器は、コンテンツアイテムに起因するtouch-endイベントを含む第2の対話イベントを検出することができる。touch-endイベントは、コンテンツアイテムのtouch-endイベントハンドラによって処理され得る。touch-endイベントのためのtouch-endイベントハンドラは、あらかじめ決定されたアクションを実施するための関数呼出しとともに構成され得る。いくつかの実装形態では、登録ロガーは、touch-endイベントハンドラを含む第2の対話イベントの検出に応じて、関数呼出しの登

録を識別することができる。

【0011】

いくつかの実装形態では、イベント検出器は、第1の対話イベントを、複数のアタッチされた対話イベントからの第2の対話イベントとペアにされるものとして識別することであって、複数のアタッチされた対話イベントが、on-clickイベントおよびtouch-endイベントのうちの少なくとも1つを含む、識別を行うことができる。いくつかの実装形態では、関数アクティベータは、第1の対話イベントを第2の対話イベントとペアにされるものとする識別にตอบสนองして、第1の対話イベントの検出と、関連付けられた関数呼出しの登録の識別との間の時間が、あらかじめ決定されたしきい値未満であると決定することができる。

10

【0012】

いくつかの実装形態では、関数アクティベータは、第1の対話イベントの検出が、関数呼出しの登録に先行すると決定することができる。いくつかの実装形態では、関数アクティベータは、第1の対話イベントの検出が、関数呼出しの登録に先行するとの決定に基づいて、あらかじめ決定されたしきい値を設定することができる。いくつかの実装形態では、関数アクティベータは、第2の対話イベントの検出の後で、時間があらかじめ決定されたしきい値未満であるとの決定に応じて、アプリケーションにあらかじめ決定されたアクションを実施させることができる。

【0013】

いくつかの実装形態では、関数アクティベータは、第1の対話イベントの検出が、関数呼出しの登録に後続すると決定することができる。いくつかの実装形態では、関数アクティベータは、第1の対話イベントの検出が、関数呼出しの登録に後続するとの決定に基づいて、あらかじめ決定されたしきい値を設定することができる。いくつかの実装形態では、関数アクティベータは、第1の対話イベントの検出の後で、時間があらかじめ決定されたしきい値未満であるとの決定に応じて、アプリケーションにあらかじめ決定されたアクションを実施させることができる。いくつかの実装形態では、コンテンツアイテムの第1の部分のコードにおいて符号化された、イベント検出器、登録ロガー、および関数アクティベータのうちの少なくとも1つが、コンテンツアイテムの第2の部分のコードの実行より前に、アプリケーションによって実行可能であり得る。

20

【0014】

少なくとも1つの態様は、コンテンツアイテム上で実施される対話を検証する方法を対象とする。コンテンツアイテムの第1の部分のコードにおいて符号化され、1つまたは複数のプロセッサ上で実行している、対話検証器は、コンテンツアイテムに起因する第1の対話イベントを検出することができる。第1の対話イベントは、コンテンツアイテムを提示するように構成されたアプリケーションによってスクリプトに渡され得、アプリケーションによる、コンテンツアイテムとの対話の検出にตอบสนองして、情報リソース上にコンテンツアイテムが提示される。対話検証器は、コンテンツアイテムの第2の部分のコードの実行を介して、あらかじめ決定されたアクションを実施するための関数呼出しの登録を識別することができる。関数呼出しは、アプリケーションがコンテンツアイテムに第2の対話イベントを渡すことに応じてトリガされるように構成され得る。第2の対話イベントは、コンテンツアイテムに起因し、コンテンツアイテムとの対話に応じて生成され得る。対話検証器は、第1の対話イベントの検出と、関数呼出しの登録の識別との間の時間が、あらかじめ決定されたしきい値未満であると決定することができる。対話検証器は、第1の対話イベントの検出と、関数呼出しの登録の識別との間の時間が、あらかじめ決定されたしきい値未満であるとの決定に応じて、アプリケーションにあらかじめ決定されたアクションを実施させることができる。

30

40

【0015】

いくつかの実装形態では、第1の対話イベントを検出することは、touch-startイベントおよびtouch-endイベントの後で、コンテンツアイテムに起因する第1のon-clickイベントを含む第1の対話イベントを検出することを含み得る。いくつかの実装形態では、対話検

50

証器は、第1のon-clickイベントの後で、コンテンツアイテムに起因する第2のon-clickイベントを含む第2の対話イベントを検出することができる。第2のon-clickイベントは、コンテンツアイテムのon-clickイベントハンドラによって処理され得、第2のon-clickイベントのためのイベントハンドラが、あらかじめ決定されたアクションを実施するための関数呼出しとともに構成される。いくつかの実装形態では、関数呼出しの登録を識別することは、コンテンツアイテムに起因する第2のon-clickイベントを含む第2の対話イベントの検出に応じて、関数呼出しの登録を識別することを含み得る。

【0016】

いくつかの実装形態では、第1の対話イベントを検出することは、touch-endイベントの後で、コンテンツアイテムに起因するon-clickイベントを含む第1の対話イベントを検出することであって、on-clickイベントが、コンテンツアイテムのon-clickイベントハンドラによって処理される、検出することを含み得る。いくつかの実装形態では、対話検証器は、コンテンツアイテムに起因するtouch-endイベントを含む第2の対話イベントを検出することができる。touch-endイベントは、コンテンツアイテムのtouch-endイベントハンドラによって処理され得る。touch-endイベントのためのtouch-endイベントハンドラは、あらかじめ決定されたアクションを実施するための関数呼出しとともに構成され得る。いくつかの実装形態では、関数呼出しの登録を識別することは、touch-endイベントハンドラを含む第2の対話イベントの検出に応じて、関数呼出しの登録を識別することを含み得る。

10

【0017】

いくつかの実装形態では、対話検証器は、第1の対話イベントを、複数のアタッチされた対話イベントからの第2の対話イベントとペアにされるものとして識別することであって、複数のアタッチされた対話イベントが、on-clickイベントおよびtouch-endイベントのうちの少なくとも1つを含む、識別することを行うことができる。いくつかの実装形態では、時間があらかじめ決定されたしきい値未満であると決定することが、第1の対話イベントを第2の対話イベントとペアにされるものとする識別に応答して、第1の対話イベントの検出と、関連付けられた関数呼出しの登録の識別との間の時間が、あらかじめ決定されたしきい値未満であると決定することを含み得る。

20

【0018】

いくつかの実装形態では、対話検証器は、第1の対話イベントの検出が、関数呼出しの登録に先行すると決定することができる。いくつかの実装形態では、対話検証器は、第1の対話イベントの検出が、関数呼出しの登録に先行するとの決定に基づいて、あらかじめ決定されたしきい値を設定することができる。いくつかの実装形態では、アプリケーションにあらかじめ決定されたアクションを実施させることは、第2の対話イベントの検出の後で、時間があらかじめ決定されたしきい値未満であるとの決定に応じて、アプリケーションにあらかじめ決定されたアクションを実施させることを含み得る。

30

【0019】

いくつかの実装形態では、対話検証器は、第1の対話イベントの検出が、関数呼出しの登録に後続すると決定することができる。いくつかの実装形態では、対話検証器は、第1の対話イベントの検出が、関数呼出しの登録に後続するとの決定に基づいて、あらかじめ決定されたしきい値を設定することができる。いくつかの実装形態では、アプリケーションにあらかじめ決定されたアクションを実施させることは、第1の対話イベントの検出の後で、時間があらかじめ決定されたしきい値未満であるとの決定に応じて、アプリケーションにあらかじめ決定されたアクションを実施させることを含み得る。

40

【0020】

これらおよび他の態様および実装形態について、以下で詳細に説明する。上記の情報および以下の詳細な説明は、様々な態様および実装形態の例示的な例を含み、請求される態様および実装形態の本質および性質を理解するための概観またはフレームワークを提供する。図面は、様々な態様および実装形態の例示およびさらなる理解を提供し、本明細書に組み込まれ、本明細書の一部を構成する。

50

【0021】

添付の図面は、一定の縮尺で描かれることを意図していない。様々な図面における同様の参照番号および指示は、同様の要素を示す。明快のために、あらゆる構成要素があらゆる図面においてラベルを付けられるとは限らないことがある。図面においては、以下の通りである。

【図面の簡単な説明】

【0022】

【図1】例示的な実装形態による、コンテンツアイテム上で実施される対話を検証するためのコンピュータネットワーク化された環境を示すブロック図である。

【図2】例示的な実装形態による、コンテンツアイテム上で実施される対話を検証するためのシステムを示すブロック図である。

【図3A】例示的な実装形態による、コンテンツアイテム上で実施される対話を検証するプロセスを示すシーケンス図である。

【図3B】例示的な実装形態による、コンテンツアイテム上で実施される対話を検証するプロセスを示すシーケンス図である。

【図3C】例示的な実装形態による、コンテンツアイテム上で実施される対話を検証するプロセスを示すシーケンス図である。

【図3D】例示的な実装形態による、コンテンツアイテム上で実施される対話を検証するプロセスを示すシーケンス図である。

【図4】例示的な実装形態による、コンテンツアイテム上で実施される対話を検証する方法の流れ図である。

【図5A】例示的な実装形態による、コンテンツアイテム上で実施される対話を検証する方法の流れ図である。

【図5B】例示的な実装形態による、コンテンツアイテム上で実施される対話を検証する方法の流れ図である。

【図6】例示的な実装形態による、本明細書で説明および例示するシステムおよび方法の要素を実装するために採用され得る、コンピュータシステムのための全体的なアーキテクチャを示すブロック図である。

【発明を実施するための形態】

【0023】

以下は、コンテンツアイテム上で実施される対話を検証する方法、装置、およびシステムに関する様々な概念、およびそれらの実装形態のより詳細な説明である。上記で紹介し、以下でより詳細に説明する様々な概念は、多数の方法のいずれかにおいて実装され得、その理由は、説明する概念が、実装形態のいかなる特定の様式にも限定されないからである。

【0024】

コンテンツアイテムは、情報リソース(たとえば、ウェブページ)のコンテンツスロット(たとえば、インラインフレーム要素)に挿入され得、クライアントデバイス上で実行しているアプリケーション(たとえば、ウェブブラウザ)によって読み取られることになる実行可能コードを含み得る。コンテンツアイテムの実行可能コードは、特に、カスケーディングスタイルシート(CSS)およびJavaScript(商標)を伴う、ハイパーテキストマークアップ言語(HTML)(たとえば、HTML5)の形態であり得る。実行可能コードは、アプリケーションを介して、デバイスに接続された入出力(I/O)デバイスから渡された、様々な対話イベントに応答するために、様々なイベントハンドラ(イベントリスナーと呼ばれることがある)を含み得る。各イベントハンドラは、特定のタイプの対話イベントに対応し得、対話イベントに異なるように応答するように構成され得る。いくつかのイベントハンドラは、対応する対話イベントに応じて、特定の関数(たとえば、別の情報リソースにナビゲートするためのアプリケーションプログラミングインターフェース(API)関数)をトリガするように構成され得る。他のイベントハンドラは、それぞれの対話イベントに応じて、いかなるアクションも実施しないように構成され得る。

【0025】

異なるコンテンツアイテムにわたって一貫性のないイベントハンドラの構成によって、それぞれのイベントハンドラのために構成された関数呼出しのトリガリングのタイミングにおいてなど、挙動における不一致が生じ得る。この問題は、ユーザによるI/Oデバイスとの単一の物理的対話に応じて、複数のイベントハンドラが呼び出されるとき、悪化し得る。たとえば、モバイルスマートフォンのタッチスクリーン上のユーザによる単一のタッチによって、touch-startイベントハンドラ、touch-endイベントハンドラ、scrollイベントハンドラ、およびon-clickイベントハンドラの呼出しが生じ得る。いくつかのコンテンツアイテムは、touch-endイベントハンドラに応じて、関数呼出しをトリガするように構成され得るのに対して、他のコンテンツアイテムは、on-clickイベントハンドラに応じて、関数呼出しをトリガするように構成され得る。単一の物理的対話に応じてすべてが呼び出され得る、これらのイベントハンドラの構成における変動は、エンドユーザの観点からすれば、不要な対話、または不注意による対話につながり得る。さらに、そのような対話は、エンドユーザによって期待された時間に対して早すぎるかまたは遅れた関数呼出しのトリガリングにつながり得る。したがって、コンテンツアイテムの挙動および機能における非整合が、コンテンツアイテムおよび情報リソースとの人間とコンピュータの対話(HCI)の品質の低下において、結果として起こり得る。加えて、生じる関数呼出しは、コンピュータ処理リソースの浪費、ネットワーク帯域幅の過剰な消費、および不要な電力の利用につながり得る。不注意による関数呼出しのそのような悪影響は、エンドユーザがクライアントデバイス上で対話より前の情報リソースの以前の状態を訂正し、取り出すための手段を講じるとき、悪化し得る。

10

20

【0026】

コンテンツアイテムの期待された挙動からの逸脱に対処するための既存の手法は、コンテンツアイテムの他の機能における低下または停止を生じ得る。ある既存の技法は、コンテンツアイテムのイベントハンドラからトリガされることが可能にされる関数呼出しのセットを制限することであり得る。この手法の下では、コンテンツアイテムのイベントハンドラからの各関数呼出しが、許可されたパラメータを伴う特定のコンテキストにおいて使用されることを検証するために、関数呼出しが確認され得る。しかしながら、このような関数呼出しの確認は、困難であることがあり、コンテンツアイテムの他の機能に対する著しい遅延を生じ得る。別の既存の技法は、最初にプログラムされたイベントハンドラを新しいイベントハンドラに置換することであり得る。この手法では、新しいイベントハンドラが、対応する対話イベントの検出に応じて、指定された関数呼出しをトリガするように構成され得る。しかしながら、最初にプログラムされたイベントハンドラを置き換えることは、特に、最初にプログラムされたイベントハンドラとともに呼び出されることになっていた関数呼出しが考慮されないとき、誤りを起こしやすくなることがあり、予期しない結果につながる可能性がある。

30

【0027】

これらおよび他の技術的課題に対処するため、情報リソース上でサービスされることになる各コンテンツアイテムは、コンテンツアイテムのイベントハンドラからの予期しない関数呼出しのトリガリングを防止するために、対話検証スクリプトを含むように修正され得る。アプリケーションが、イベントハンドラなどの後続の実行可能コードより前にスクリプトを実行するように、対話検証スクリプトは、コンテンツアイテムのための実行可能コードの上部分に挿入され得る。クライアントデバイス上のアプリケーションによる実行時に、対話検証スクリプトは、アプリケーションに、指定されたイベントハンドラのための対話イベントの検出と、指定された関数の登録とを待機させることができる。対話検証スクリプトは、アプリケーションに、検出および登録のタイミングに基づいて、指定された関数のトリガリングを修正させることができる。対話検証スクリプトは、アプリケーションに、指定されたイベントハンドラのためのコンテンツに起因する対話イベントを検出させることができる。イベントハンドラは、指定された関数とともに構成され得る。イベントハンドラは、指定された関数なしで構成され得、あらかじめ指定されたイベントハン

40

50

ドラのセットのうちの1つであり得る。対話イベントの検出は、関数の登録より前、または関数の登録の後で発生し得る。

【0028】

指定されたイベントハンドラのためのコンテンツアイテムに起因する対話イベントが最初に発生するとき、対話検証スクリプトは、アプリケーションに、あらかじめ決定された時間制限の間、特定の関数の登録を待機させることができる。関数呼出しの登録が、あらかじめ決定された時間制限内に発生しない場合、対話検証スクリプトは、アプリケーションに、検出されたイベントを無視させることができる。一方、関数の登録が、あらかじめ決定された時間制限内に発生する場合、対話検証スクリプトは、アプリケーションに、登録された関数が遂行および実行されることを可能にさせることができる。逆に、指定されたイベントハンドラのための対話イベントが、指定された関数の登録の後で発生するとき、対話検証スクリプトは、アプリケーションに、検出および登録があらかじめ決定された時間制限内に発生したか否かを決定させることができる。検出および登録が、あらかじめ決定された時間制限内に発生すると決定される場合、対話検証スクリプトは、登録された関数が遂行および実行されることを可能にすることができる。一方、検出および登録が、あらかじめ決定された時間制限外で発生すると決定される場合、対話検証スクリプトは、アプリケーションに、対話イベントの検出を無視させ、登録された関数の実行を防止させることができる。このようにして、コンテンツアイテムは、異なるように挙動することができ、ユーザが期待したときとは異なる、早すぎるかまたは遅れた応答のトリガリングが防止され、それによって、コンテンツアイテムおよび下にある情報リソースとの対話の品質が向上し得る。

【0029】

次に図1を参照すると、図1は、コンテンツアイテム上で実施される対話を検証するためのコンピュータネットワーク化された環境100の一実装形態を示すブロック図である。環境100は、少なくとも1つのデータ処理システム110を含む。1つのデータ処理システム110のみが図示されているが、多数の実装形態では、データ処理システム110は、複数のデータ処理システムまたはコンピューティングデバイスのファーム、クラウド、クラスタ、または他のグループ化であり得る。データ処理システム110は、処理回路と呼ばれることがある、少なくとも1つのプロセッサとメモリとを含み得、その各々について、図6に関して以下でより詳細に説明する。メモリは、プロセッサによって実行されると、本明細書で説明する動作のうちの1つまたは複数を実行させるプロセッサに実施させる、プロセッサ実行可能命令を記憶する。プロセッサは、マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)など、またはそれらの組合せを含み得る。メモリは、限定はしないが、プロセッサにプログラム命令を提供することが可能な、電子、光学、磁気、または任意の他の記憶もしくは送信デバイスを含み得る。メモリは、フロッピーディスク、CD-ROM、DVD、磁気ディスク、メモリチップ、ASIC、FPGA、読取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、電氣的消去可能ROM(EEPROM)、消去可能プログラマブルROM(EPROM)、フラッシュメモリ、光メディア、または、それからプロセッサが命令を読み取ることができる任意の他の好適なメモリをさらに含み得る。命令は、任意の好適なコンピュータプログラミング言語からのコードを含み得る。

【0030】

ネットワーク105は、インターネットなどのコンピュータネットワーク、ローカルエリアネットワーク、ワイドエリアネットワーク、メトロエリアネットワーク、または他のエリアネットワーク、イントラネット、衛星ネットワーク、音声もしくはデータモバイルフォン通信ネットワークなどの他のコンピュータネットワーク、およびそれらの組合せを含み得る。システム100のデータ処理システム110は、少なくとも1つのコンテンツプロバイダ115、少なくとも1つのコンテンツパブリッシャ120、少なくとも1つのクライアントデバイス125などと、ネットワーク105を介して通信することができる。ネットワーク105は、クライアントデバイス125と、データ処理システム110と、コンテンツプロバイダ115またはコンテンツパブリッシャ120など、1つまたは複数のコンテンツソースとの間で情報を中

継する、任意の形態のコンピュータネットワークであり得る。たとえば、ネットワーク105は、インターネット、および/またはローカルエリアネットワーク(LAN)、ワイドエリアネットワーク(WAN)、セルラーネットワーク、衛星ネットワーク、もしくは他のタイプのデータネットワークなど、他のタイプのデータネットワークを含み得る。ネットワーク105はまた、ネットワーク105内でデータを受信および/または送信するように構成される、任意の数のコンピューティングデバイス(たとえば、コンピュータ、サーバ、ルータ、ネットワークスイッチなど)を含み得る。ネットワーク105は、任意の数のハードワイヤードおよび/またはワイヤレス接続をさらに含み得る。クライアントデバイス125は、ネットワーク105における他のコンピューティングデバイスに(たとえば、光ファイバーケーブル、CAT5ケーブルなどを介して)ハードワイヤードされているトランシーバと、(たとえば、Wi-Fi、セルラー、無線などを介して)ワイヤレス通信し得る。

10

【0031】

コンテンツプロバイダ115は、クライアントデバイス125において情報リソース上で表示するためのコンテンツアイテム155を提供するために、コンテンツプロバイダエンティティによって動作させられるサーバまたは他のコンピューティングデバイスを含み得る。コンテンツプロバイダ115によって提供されるコンテンツは、任意の好都合な形態をとることができる。たとえば、サードパーティコンテンツは、他の表示されたコンテンツに関連するコンテンツを含み得、たとえば、表示されたコンテンツに関連するウェブサイトのページであり得る。コンテンツは、コンテンツパブリッシャ120によって提供された1次コンテンツを含む、情報リソース150などの情報リソース上で表示するための、サードパーティコンテンツアイテムまたはクリエイティブ(たとえば、広告)を含み得る。コンテンツアイテム155はまた、検索結果ウェブページ上で表示され得る。たとえば、コンテンツプロバイダ115は、ウェブページの1次コンテンツが会社によって提供される、会社のウェブページなど、情報リソース150のコンテンツスロット(たとえば、インラインフレーム要素)において表示するための、または検索エンジンによって提供された検索結果ランディングページ上で表示するための、コンテンツアイテム155のソースを提供し得るか、またはそのソースであり得る。コンテンツプロバイダ115に関連付けられたコンテンツアイテム155は、スマートフォンまたは他のクライアントデバイス125上のアプリケーションの実行の一部として表示されるコンテンツなど、ウェブページ以外の情報リソース150上で表示され得る。

20

30

【0032】

コンテンツパブリッシャ120は、ネットワーク105を介して表示するための1次コンテンツを含む情報リソース150を提供するために、コンテンツ発行エンティティによって動作させられるサーバまたは他のコンピューティングデバイスを含み得る。たとえば、コンテンツパブリッシャ120は、情報リソース150上で表示するための1次コンテンツを提供する、ウェブページオペレータを含み得る。情報リソース150は、コンテンツパブリッシャ120によって提供されるもの以外のコンテンツを含み得、情報リソース150は、コンテンツプロバイダ115からのコンテンツアイテム155の表示のために構成されたコンテンツスロットを含み得る。たとえば、コンテンツパブリッシャ120は、会社のウェブサイトを動作させることができ、ウェブサイトのウェブページ上で表示するための、その会社についてのコンテンツを提供することができる。ウェブページは、コンテンツプロバイダ115によって、またはコンテンツパブリッシャ120自体によって提供されたコンテンツアイテム155の表示のために構成された、コンテンツスロットを含み得る。いくつかの実装形態では、コンテンツパブリッシャ120は、検索エンジンウェブサイトを動作させる検索エンジンオペレータの検索エンジンコンピューティングデバイス(たとえば、サーバ)を含む。検索エンジンウェブページ(たとえば、結果またはランディングウェブページ)の1次コンテンツは、検索の結果、ならびに、コンテンツプロバイダ115からのコンテンツアイテム155など、情報リソース150のコンテンツスロットにおいて表示されるサードパーティコンテンツアイテム155を含み得る。いくつかの実装形態では、コンテンツパブリッシャ120は、ビデオコンテンツを提供するための1つまたは複数のサーバを含み得る。

40

50

【0033】

クライアントデバイス125は、コンテンツパブリッシャ120によって提供されたコンテンツ(たとえば、情報リソース150)、およびコンテンツプロバイダ115によって提供されたコンテンツ(たとえば、情報リソース150のコンテンツスロットにおいて表示するためのコンテンツアイテム155)などのデータを表示するために、ネットワーク105を介して通信するように構成されたコンピューティングデバイスを含み得る。クライアントデバイス125、コンテンツプロバイダ115、およびコンテンツパブリッシャ120は、デスクトップコンピュータ、ラップトップコンピュータ、タブレットコンピュータ、スマートフォン、携帯情報端末、モバイルデバイス、コンシューマコンピューティングデバイス、サーバ、クライアント、デジタルビデオレコーダ、テレビジョン用のセットトップボックス、ビデオゲームコンソール、またはネットワーク105を介して通信するように構成された任意の他のコンピューティングデバイスを含み得る。クライアントデバイス125は、それを通してエンドユーザがコンテンツを受信するための要求を出すことができる、通信デバイスであり得る。要求は、検索エンジンに対する要求であり得、要求は、検索クエリを含み得る。いくつかの実装形態では、要求は、ウェブページにアクセスするための要求を含み得る。

10

【0034】

コンテンツプロバイダ115、コンテンツパブリッシャ120、およびクライアントデバイス125は、処理回路の一部として、プロセッサとメモリとを各々含み得る。メモリは、プロセッサによって実行されると、本明細書で説明する動作のうちの1つまたは複数をプロセッサに実施させる、マシン語命令を記憶する。プロセッサは、マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)など、またはそれらの組合せを含み得る。メモリは、限定はしないが、プロセッサにプログラム命令を提供することが可能な、電子、光学、磁気、または任意の他の記憶もしくは送信デバイスを含み得る。メモリは、フロッピーディスク、CD-ROM、DVD、磁気ディスク、メモリチップ、ASIC、FPGA、読取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、電気的消去可能ROM(EEPROM)、消去可能プログラマブルROM(EPROM)、フラッシュメモリ、光メディア、または、それからプロセッサが命令を読み取ることができる任意の他の好適なメモリをさらに含み得る。命令は、任意の好適なコンピュータプログラミング言語からのコードを含み得る。

20

【0035】

コンテンツプロバイダ115、コンテンツパブリッシャ120、およびクライアントデバイス125はまた、1つまたは複数のユーザインターフェースデバイスを各々含み得る。一般に、ユーザインターフェースデバイスは、感覚情報(たとえば、ディスプレイ上の視覚化、1つまたは複数のサウンドなど)を生成することによって、ユーザにデータを伝達し、かつ/またはユーザから受信された感覚情報を電子信号に変換する、任意の電子デバイス(たとえば、キーボード、マウス、ポインティングデバイス、タッチスクリーンディスプレイ、マイクロフォンなど)を指す。1つまたは複数のユーザインターフェースデバイスは、様々な実装形態によれば、コンテンツプロバイダ115、コンテンツパブリッシャ120、およびクライアントデバイス125のハウジングの内部(たとえば、内蔵ディスプレイ、マイクロフォンなど)、またはコンテンツプロバイダ115、コンテンツパブリッシャ120、およびクライアントデバイス125のハウジングの外部(たとえば、クライアントデバイス125に接続されたモニタ、クライアントデバイス125に接続されたスピーカなど)にあり得る。たとえば、コンテンツプロバイダ115、コンテンツパブリッシャ120、およびクライアントデバイス125は、1つもしくは複数のコンテンツソースから、および/またはネットワーク105を介して、データ処理システム110から受信されたウェブページデータを使用して、ウェブページを視覚的に表示する、電子ディスプレイを含み得る。いくつかの実装形態では、コンテンツプレースメントキャンペーンマネージャ、またはサードパーティコンテンツプロバイダは、コンテンツプロバイダ115を介して、データ処理システム110と通信することができる。いくつかの実装形態では、コンテンツプロバイダは、コンテンツプロバイダ115のユーザインターフェースデバイス上に表示されたユーザインターフェースを介して、データ処

30

40

50

理システム110と通信することができる。

【0036】

クライアントデバイス125は、アプリケーション145を実行することができる。アプリケーション145は、たとえば、インターネットブラウザ、モバイルアプリケーション、あるいは、情報リソース150およびコンテンツアイテム155中に含まれたコンピュータ実行可能命令など、クライアントデバイス125によって処理されるコンピュータ実行可能命令を実行またはさもなければ呼び出すことが可能な、任意の他のコンピュータプログラムを含み得る。その中に含まれた情報リソース150は、スクリプト、論理、マークアップ、または命令(たとえば、ハイパーテキストマークアップ言語(HTML)、拡張可能ハイパーテキストマークアップ言語(XHTML)、拡張可能マークアップ言語(XML)、カスケーディングスタイルシート(CSS)、およびJavaScript、またはそれらの任意の組合せ)に対応し得る。情報リソース150は、ヘッダ162と本文164とを含み得る。コンテンツアイテム155は、情報リソース150に挿入され得る。

10

【0037】

情報リソース150のうちの1つをフェッチするために、クライアントデバイス125のアプリケーション145は、リソースの要求をコンテンツパブリッシャ120に送ることができる。要求は、データの中でも、情報リソース150のためのアドレス、デバイス識別子、デバイスタイプ、およびアプリケーションプロファイルを含み得る。アドレスは、ユニバーサルリソースロケータ(URL)であり得、パラメータの中でも、プロトコル、ホスト名、およびファイルパス名を含み得る。ホスト名は、コンテンツパブリッシャ120に対応し得る。ファイルパス名は、コンテンツパブリッシャ120によってホストされた情報リソース150のうちの1つに対応し得る。デバイス識別子は、クライアントデバイス125に対応し得る。デバイスタイプは、クライアントデバイス125のシステム構成に対応し得る。アプリケーションプロファイルは、情報リソース150を実行およびランさせるために、どのアプリケーション(たとえば、ウェブブラウザタイプおよびバージョンなど)がクライアントデバイス125上でランしているかを示し得る。要求が送られると、ネットワーク105は、ホスト名において識別されるコンテンツパブリッシャ120に、要求を転送し得る。

20

【0038】

続いて、コンテンツパブリッシャ120は、アプリケーション145から要求を受信することができる。次に、コンテンツパブリッシャ120は、要求をパースして、クライアントデバイス125に提供するための情報リソース150を識別することができる。いくつかの実装形態では、コンテンツパブリッシャ120は、データベースにアクセスして、情報リソース150を取り出すことができる。データベースは、コンテンツパブリッシャ120の一部であるか、コンテンツパブリッシャ120と通信しているか、またはさもなければコンテンツパブリッシャ120によってアクセス可能であり得る。コンテンツパブリッシャ120は、要求中に含まれたホスト名およびファイルパス名に基づいて、提供するための情報リソース150を識別することができる。情報リソース150を識別すると、コンテンツパブリッシャ120は、クライアントデバイス125のアプリケーション145に、情報リソース150を送信するか、または送ることができる。いくつかの実装形態では、情報リソース150は、コンテンツプロバイダ115またはデータ処理システム110によって提供されたコンテンツアイテム155を挿入するための、1つまたは複数のコンテンツスロットを有し得る。いくつかの実装形態では、情報リソース150は、コンテンツパブリッシャ120からのコンテンツアイテム155とともに、1つまたは複数のコンテンツスロットを有し得る。いくつかの実装形態では、コンテンツアイテム155は、あらかじめ挿入され得るか、またはクライアントデバイス125上でランしているアプリケーション145に、情報リソース150とともに提供され得る。

30

40

【0039】

情報リソース150を処理するために、アプリケーション145は、ドキュメントツリージェネレータ、レンダリングツリージェネレータ、およびディスプレイモジュールなど、任意の数の構成要素を含み得る。情報リソース150を読み取る際に、ドキュメントツリージェネレータは、情報リソース150をパースして、階層的に配置された情報リソース150のドク

50

メントオブジェクトモデル(DOM)ツリーを生成することができる。DOMツリーを使用して、レンダリングツリージェネレータは、情報リソースのコンテンツアイテムの視覚的プロパティまたは属性を指定するレンダリングツリーを生成することができる。ディスプレイモジュールは、レンダリングツリーに基づいて、クライアントデバイス125のディスプレイ上に情報リソースをレンダリングおよび表示することができる。アプリケーション145はまた、情報リソース150のCSSおよびJavaScriptをパースすることもできる。

【0040】

情報リソース150が処理される間、アプリケーション145は、情報リソース150のコンテンツスロットをパースすることができる。コンテンツスロットは、コンテンツアイテム155に対応するアドレスを含み得る。アドレスは、ユニバーサルリソースロケータ(URL)であり得、パラメータの中でも、プロトコル、ホスト名、およびファイルパス名を含み得る。ホスト名は、コンテンツプロバイダ115(たとえば、ファーストパーティコンテンツの場合)、コンテンツパブリッシャ120(たとえば、サードパーティコンテンツの場合)、またはデータ処理システム110に対応し得る。ファイルパス名は、コンテンツプロバイダ115、コンテンツパブリッシャ120、またはデータ処理システム110によって維持された、コンテンツアイテム155のうちの1つに対応し得る。いくつかの実装形態では、アプリケーション145は、情報リソース150に挿入するための追加のコンテンツを取り出すために、アドレスを使用して、コンテンツパブリッシャ120にアクセスすることができる。いくつかの実装形態では、アドレスは、コンテンツプロバイダ115に対するコンテンツの要求の生成において使用され得る。いくつかの実装形態では、アドレスは、コンテンツ選択プロセスにおいて、コンテンツアイテム155がデータ処理システム110によって選択されるためのプレースホルダーであり得る。コンテンツスロット中に含まれたアドレスを使用して、アプリケーション145は、情報リソース150に挿入するためのコンテンツの要求を生成することができる。要求は、データの中でも、コンテンツアイテム155のためのアドレス、デバイス識別子、デバイスタイプ、およびアプリケーションプロファイルを含み得る。デバイス識別子は、クライアントデバイス125に対応し得る。デバイスタイプは、クライアントデバイス125のシステム構成に対応し得る。アプリケーションプロファイルは、情報リソース150を実行およびランさせるために、どのアプリケーション(たとえば、ウェブブラウザタイプおよびバージョンなど)がクライアントデバイス125上でランしているかを示し得る。要求が送られると、ネットワーク105は、ホスト名において識別されるコンテンツプロバイダ115またはデータ処理システム110に、要求を転送し得る。

【0041】

続いて、コンテンツプロバイダ115またはデータ処理システム110は、アプリケーション145から、コンテンツの要求を受信することができる。次に、コンテンツプロバイダ115またはデータ処理システム110は、要求をパースして、クライアントデバイス125に提供するためのコンテンツアイテム155を識別することができる。コンテンツアイテム155を識別すると、コンテンツプロバイダ115またはデータ処理システム110は、情報リソース150に挿入するために、クライアントデバイス125のアプリケーション145に、コンテンツアイテム155を送信するか、または送ることができる。コンテンツアイテム155は、対話検証スクリプト160(対話検証器とも呼ばれる)を含み得る。対話検証スクリプト160の機能に関する詳細については、本明細書の以下で提供する。

【0042】

データ処理システム110は、少なくとも1つのサーバを含み得る。いくつかの実装形態では、データ処理システム110は、少なくとも1つのデータセンターまたはサーバファームに位置する、複数のサーバを含み得る。いくつかの実装形態では、データ処理システム110は、コンテンツプレースメントシステムを含む。データ処理システム110は、少なくとも1つのコンテンツアイテム評価器130と、少なくとも1つのスクリプト挿入器135と、少なくとも1つのデータベース140とを含み得る。コンテンツアイテム評価器130およびスクリプト挿入器135は、データベース140と、およびネットワーク105を介して他のコンピューティングデバイス(たとえば、コンテンツプロバイダ115、コンテンツパブリッシャ120、ま

たはクライアントデバイス125)と通信するように構成された、少なくとも1つの処理ユニット、サーバ、仮想サーバ、回路、エンジン、エージェント、アプライアンス、またはプログラマブル論理アレイなどの他の論理デバイスを各々含み得る。

【0043】

コンテンツアイテム評価器130およびスクリプト挿入器135は、データベース140と、およびネットワーク105を介して他のコンピューティングデバイス(たとえば、コンテンツプロバイダ115、コンテンツパブリッシャ120、およびクライアントデバイス125)と通信するように構成された、少なくとも1つの処理ユニット、サーバ、仮想サーバ、回路、エンジン、エージェント、アプライアンス、またはプログラマブル論理アレイなどの他の論理デバイスを各々含み得る。コンテンツアイテム評価器130およびスクリプト挿入器135は、少なくとも1つのコンピュータプログラムまたは少なくとも1つのスクリプトを各々含むか、または実行することができる。コンテンツアイテム評価器130およびスクリプト挿入器135は、別個の構成要素、単一の構成要素、またはデータ処理システム110の一部であり得る。コンテンツアイテム評価器130およびスクリプト挿入器135は、1つまたは複数のスクリプトを実行するように構成された1つまたは複数のプロセッサなど、ソフトウェアおよびハードウェアの組合せを含み得る。

【0044】

データ処理システム110はまた、1つまたは複数のコンテンツリポジトリまたはデータベース140を含み得る。データベース140は、データ処理システム110のローカルであり得る。いくつかの実装形態では、データベース140は、データ処理システム110に対してリモートであり得るが、ネットワーク105を介して、データ処理システム110と通信することができる。データベース140は、コンテンツプロバイダ115によって提供されたコンテンツアイテム155、または各コンテンツアイテム155を識別する基準アドレスを含み得る。いくつかの実装形態では、データベース140は、コンテンツパブリッシャ120によって提供された情報リソース150、または各情報リソース150を識別する基準アドレスを含み得る。データベース140はまた、コンテンツプロバイダ115、コンテンツパブリッシャ120、またはクライアントデバイス125に提供するために、対話検証スクリプト160を含み得る。対話検証スクリプト160の詳細については、本明細書の以下で提供する。

【0045】

コンテンツアイテム評価器130は、コンテンツプロバイダ115の1つまたは複数のコンテンツアイテム155を識別することができる。いくつかの実装形態では、コンテンツアイテム評価器130は、クライアントデバイス125からのコンテンツの要求の受信に応じて、1つまたは複数のコンテンツアイテム155を識別することができる。いくつかの実装形態では、コンテンツアイテム評価器130は、クライアントデバイス125からのコンテンツの要求とは無関係に、1つまたは複数のコンテンツアイテム155を識別することができる。いくつかの実装形態では、コンテンツアイテム評価器130は、コンテンツプロバイダ115のコンテンツアイテム155のために、データベース140にアクセスすることができる。いくつかの実装形態では、コンテンツアイテム評価器130は、コンテンツプロバイダ115からコンテンツアイテム155を受信することができる。コンテンツプロバイダ115は、コンテンツアイテム評価器130に、コンテンツアイテム155を修正するための要求を送信することができる。修正するための要求は、コンテンツアイテム155を含み得る。各コンテンツアイテム155は、コンテンツアイテム識別子(たとえば、アドレス、または英数字の文字セット)によってインデックス付けされ得る。コンテンツアイテム評価器130は、要求をパースして、コンテンツアイテム155を識別することができる。識別された各コンテンツアイテム155について、コンテンツアイテム評価器130は、コンテンツアイテム155の実行可能コードをパースすることができる。コンテンツアイテム155の実行可能コードは、マークアップ(たとえば、HTMLおよびCSS)およびスクリプト言語コード(たとえば、JavaScript(商標))の形態におけるコードを含み得る。コンテンツアイテム155の実行可能コードをパースすることから、コンテンツアイテム評価器130は、1つまたは複数のイベントハンドラ(イベントリスナーとも呼ばれる)を識別することができる。いくつかの実装形態では、コンテンツアイテム評

価器130は、コンテンツアイテム155のスク립ト言語コードをパースして、1つまたは複数のイベントハンドラを識別することができる。コンテンツアイテム155の各イベントハンドラは、クライアントデバイス125のアプリケーション145から情報リソース150に、またはコンテンツアイテム155に渡された、対応する対話イベントを処理するためのルーチン(たとえば、非同期コールバックサブルーチン)であり得る。たとえば、コンテンツアイテム155は、クリック対話イベントを処理するためのon-clickイベントハンドラと、キーボード押下を処理するためのkeypressイベントハンドラとを含み得る。

【0046】

コンテンツアイテム155のイベントハンドラの中で、コンテンツアイテム評価器130は、対応する対話イベントの検出に応答して、あらかじめ決定されたアクションを実施するための関数とともに構成された、1つまたは複数のイベントハンドラを識別することができる。イベントハンドラのうちの少なくとも1つは、あらかじめ決定されたアクションを実施するための関数とともに構成され得る。関数は、コンテンツアイテム評価器130によって検索されることになる、あらかじめ指定された関数のセットのうちの1つであり得る。いくつかの実装形態では、関数、および関数のトリガリングに応じて実施されることになる、あらかじめ決定されたアクションが、アプリケーションプログラミングインターフェース(API)によって定義され得る。たとえば、関数呼出しは、アプリケーション145によってアクセスまたはロードされた現在の情報リソースから、そのアドレスが関数呼出しによって指定される別の情報リソースにナビゲートするための、「ExitAPI.exit()」であり得る。いくつかの実装形態では、コンテンツアイテム評価器130は、コンテンツアイテム155の各イベントハンドラの実行可能コードをパースして、あらかじめ決定されたアクションを実施するための関数とともに構成される、1つまたは複数のイベントハンドラを識別することができる。

【0047】

コンテンツアイテム評価器130は、コンテンツアイテム155のために識別されたイベントハンドラをトラバースして、あらかじめ決定されたアクションを実施するための関数を識別することができる。コンテンツアイテム評価器130は、クライアントデバイス125からのコンテンツの要求の受信に応じて、または要求の受信とは無関係に、コンテンツアイテム155のためのイベントハンドラをトラバースすることができる。各イベントハンドラの実行可能コードをパースする際に、コンテンツアイテム評価器130は、あらかじめ決定されたアクションを実施するための関数のためのあらかじめ定義された文字列に一致する、実行可能コードの文字列を識別することができる。イベントハンドラのための実行可能コードのための文字列が、関数のための文字列を含まない場合、コンテンツアイテム評価器130は、イベントハンドラを、関数とともに構成されていないものとして識別することができる。逆に、イベントハンドラのための実行可能コードのための文字列が、関数のための文字列を含む場合、コンテンツアイテム評価器130は、イベントハンドラを、あらかじめ決定されたアクションを実施するための関数とともに構成されるものとして識別することができる。

【0048】

いくつかの実装形態では、コンテンツアイテム評価器130は、コンテンツアイテム155が、コンテンツアイテム155中に含まれることが期待される、あらかじめ定義されたイベントハンドラのリストのうちの1つまたは複数のイベントハンドラを含むか否かを決定することができる。コンテンツアイテム155はまた、いくつかのイベントハンドラを欠くか、または除外することがある。たとえば、コンテンツアイテム155のための実行可能コードは、on-clickイベントハンドラのための明確な定義を有し得るが、touch-startハンドラのためのいかなるコードも欠くことがある。あらかじめ定義された期待されるイベントハンドラのリストは、関数とともに構成されるものとして識別されたイベントハンドラにアタッチされるか、バインドされるか、またはペアにされることになる1つまたは複数のイベントハンドラを含むか、または指定することができる。あらかじめ定義された期待イベントハンドラのリストは、データベース140上で維持され得る。いくつかの実装形態では

、コンテンツアイテム評価器130は、コンテンツアイテム155中に含まれたイベントハンドラの中をトラバースすることができる。コンテンツアイテム155中に含まれたイベントハンドラがトラバースされるとき、コンテンツアイテム評価器130は、イベントハンドラが、あらかじめ定義された期待されるイベントハンドラのリストのうちの1つに一致するかどうかを決定することができる。コンテンツアイテム155のイベントハンドラの中をトラバースすることによって、コンテンツアイテム評価器130は、あらかじめ定義された期待されるイベントハンドラのリストのうちのどのイベントハンドラが、コンテンツアイテム155中に含まれていないかを決定することができる。コンテンツアイテム評価器130は、あらかじめ定義された期待されるイベントハンドラのリストのうちの、コンテンツアイテム155中に最初に含まれていないイベントハンドラを、コンテンツアイテム155の実行可能コードに挿入するか、または含めることができる。挿入されたイベントハンドラは、最初はヌルであり得、対応する対話イベントの検出に応じて、アクションを実施しないように構成され得る。コンテンツアイテム評価器130はまた、挿入されたイベントハンドラを、関数とともに構成されてい
ないものとして識別することもできる。

【0049】

コンテンツアイテム評価器130によって、関数とともに構成されていないものとして識別された、コンテンツアイテム155の各イベントハンドラについて、スクリプト挿入器135は、イベントハンドラが、あらかじめ定義されたイベントハンドラのリストのうちの1つに一致するかどうかを決定することができる。あらかじめ定義されたイベントハンドラのリストは、関数とともに構成されるものとして識別されたイベントハンドラにアタッチされるか、バインドされるか、またはペアにされることになる1つまたは複数のイベントハンドラを含むか、または指定することができる。あらかじめ定義されたイベントハンドラのリストは、データベース140上で維持され得る。あらかじめ定義されたイベントハンドラのリストは、クライアントデバイス125におけるユーザによる単一の物理的対話とともに呼び出されるようにあらかじめ決定されたイベントハンドラのセットまたはグループを含み得る。たとえば、ユーザが、クライアントデバイス125におけるタッチスクリーン上のある点に接触するとき、物理的接触が比較的短い時間の持続である場合、touch-startイベント、touch-endイベント、およびclickイベントが呼び出され得るか、または物理的接触が比較的長い時間の持続にわたる場合、scrollイベントが呼び出され得る。このことから見て、あらかじめ定義されたイベントハンドラのリストは、touch-endイベントおよびon-clickイベントを含むように定義され得る。

【0050】

スクリプト挿入器135は、関数とともに構成されていないものとして識別された1つまたは複数のイベントハンドラの中をトラバースして、対話検証スクリプト160にアタッチするか、関数とともに構成されるものとして識別されたイベントハンドラにアタッチするかを決定することができる。関数とともに構成されていないものとして識別された1つまたは複数のイベントハンドラをトラバースする間、スクリプト挿入器135は、関数とともに構成されていないものとして識別されたイベントハンドラを、あらかじめ定義されたイベントハンドラのリストのうちの1つと比較することができる。イベントハンドラが、あらかじめ定義されたイベントハンドラのリスト上のいずれにも一致しないものとして識別される場合、スクリプト挿入器135は、イベントハンドラをそのまま維持する。このようにして、イベントハンドラの元の機能が維持され得る。一方、イベントハンドラが、あらかじめ定義されたイベントハンドラのリストのうちの1つに一致するものとして識別される場合、スクリプト挿入器135は、関数とともに構成されていないものとして識別されたイベントハンドラを、(たとえば、JavaScript(商標)関数「addEventListener()」を使用して)関数とともに構成されるものとして識別された1つまたは複数のイベントハンドラにアタッチすることができる。いずれの場合も、スクリプト挿入器135は、関数とともに構成されていないより多くのイベントハンドラがあるかどうかを決定することができる。関数とともに構成されていないものとして識別されたより多くのイベントハンドラがある場合

、スクリプト挿入器135は、関数とともに構成されていないものとして識別された次のイベントハンドラを識別し、関数を繰り返すことができる。さもなければ、関数とともに構成されていないものとして識別されたイベントハンドラがそれ以上ない場合、スクリプト挿入器135は、続けて対話検証スクリプト160をコンテンツアイテム155に挿入することができる。

【0051】

スクリプト挿入器135は、対話検証スクリプト160をコンテンツアイテム155に挿入するか、または含めることができる。いくつかの実装形態では、スクリプト挿入器135は、コンテンツアイテム155をパースして、コンテンツアイテム155が前に対話検証スクリプト160とともに構成されているか、または対話検証スクリプト160をすでに含むか否かを決定することができる。コンテンツアイテム155が、前に対話検証スクリプト160とともに構成されているか、または対話検証スクリプト160をすでに含むと決定される場合、スクリプト挿入器135は、さらなる修正なしに、コンテンツアイテム155を送ることができる。さもなければ、コンテンツアイテム155が、前に対話検証スクリプト160とともに構成されていないと決定されるか、または対話検証スクリプト160を含んでいない場合、スクリプト挿入器135は、対話検証スクリプト160をコンテンツアイテム155に挿入することができる。対話検証スクリプト160は、スクリプト挿入器135によって、1つまたは複数のイベントハンドラを含む実行可能コードの別の部分より前に、実行可能コードの一部に挿入され得る。このようにして、クライアントデバイス125において実行しているアプリケーション145は、後続の実行可能コード中に含まれたイベントハンドラより前に、対話検証スクリプト160を処理および実行することになる。いくつかの実装形態では、スクリプト挿入器135は、イベントハンドラを含む実行可能コードの部分より前に、コンテンツアイテム155の実行可能コードの部分に、イベントハンドラのアタッチメントを挿入するか、または含めることができる。いくつかの実装形態では、スクリプト挿入器135は、あらかじめ定義されたイベントハンドラのリスト上の関数とともに構成されていないものとして識別されたイベントハンドラと、関数とともに構成されるものとして識別されたイベントハンドラとを、対話検証スクリプト160にアタッチすることができる。

【0052】

いくつかの実装形態では、スクリプト挿入器135は、対話検証スクリプト160をコンテンツアイテム155の外部に挿入することができる。対話検証スクリプト160は、コンテンツアイテム155の外部に挿入されたとき、コンテンツアイテム155の1つまたは複数のイベントハンドラとインターフェースすることができる。スクリプト挿入器135は、対話検証スクリプト160を情報リソース150に挿入することができる。いくつかの実装形態では、スクリプト挿入器135は、対話検証スクリプト160を、コンテンツアイテム155のための情報リソース150のコンテンツスロット(たとえば、インラインフレーム)に挿入することができる。いくつかの実装形態では、スクリプト挿入器135は、対話検証スクリプト160をシャドウDOMツリー要素として挿入することができる。シャドウDOMツリー要素は、コンテンツアイテム155のための情報リソース150のコンテンツスロットの一部であり得る。対話検証スクリプト160の機能に関する詳細については、本明細書の以下で提供する。

【0053】

次に図2を参照すると、図示されているものは、コンテンツアイテム上で実施される対話を検証するためのシステム200である。システム200は、環境100の要素の一部または全部を含み得、図1に関して上記で説明したコンテンツアイテム155および対話検証スクリプト160のより詳細な構成を表し得る。概して、システム200は、クライアントデバイス125上でランしているアプリケーション145に提供された情報リソース150中に挿入されたコンテンツアイテム155を含み得る。システム200の各構成要素は、アプリケーション145によって指定された方法で、クライアントデバイス125の1つまたは複数のプロセッサによって実行されることになる、メモリ上の命令として記憶され得る。コンテンツアイテム155は、コンテンツアイテム155のための実行可能コードの第1の部分210Aにおいて、対話検証スクリプト160を含み得る。対話検証スクリプト160は、イベント検出器220と、登録ロガー2

10

20

30

40

50

30と、関数アクティベータ240とを含み得る。コンテンツアイテム155はまた、コンテンツアイテム155のための実行可能コードの第2の部分210Bにおいて、1つまたは複数のイベントハンドラ250A～Nを含み得る。

【0054】

コンテンツプロバイダ115またはデータ処理システム110からコンテンツアイテム155を受信すると、アプリケーション145は、情報リソース150のコンテンツスロット(たとえば、インラインフレーム要素)にコンテンツアイテム155を挿入することができる。クライアントデバイス125上でランしているアプリケーション145は、その中の実行可能コードに従って、コンテンツアイテム155をパース、実行、および提示することができる。コンテンツアイテム155を処理する際に、アプリケーション145は、コンテンツアイテム155の実行可能コードを上から下まで連続的にパースし、読み取ることができる。いくつかの実装形態では、アプリケーション145は、実行可能コードの第2の部分210Bにおける1つまたは複数のイベントハンドラ250A～Nを実行するより前に、実行可能コードの第1の部分210Aにおける対話検証スクリプト160のイベント検出器220、登録ロガー230、または関数アクティベータ240を実行することができる。このようにして、コンテンツアイテム155のための実行可能コードの第1の部分210Aにおいて符号化された対話検証スクリプト160の機能が、実行可能コードの第2の部分210Bにおいて符号化されたイベントハンドラ250A～Nのいずれよりも前に実行され得る。いくつかの実装形態では、対話検証スクリプト160が、コンテンツアイテム155の外部で(たとえば、コンテンツアイテム155のためのコンテンツスロットを定義するインラインフレームにおいて)情報リソース150に挿入されるとき、アプリケーション145は、コンテンツアイテム155のための実行可能コードより前に、対話検証スクリプト160に対応する実行可能コードを実行することができる。

【0055】

イベント検出器220は、コンテンツアイテム155に起因する1つまたは複数の対話イベントを監視することができる。対話イベントは、クライアントデバイス125の任意の入出力(I/O)デバイスを用いた、ユーザによる対話またはジェスチャーに対応し得る。対話イベントは、特に、touch-startイベント、touch-endイベント、touch-move、long pressイベント、on-scrollイベント、on-flingイベント、mouse-upイベント、mouse-downイベント、on-clickイベント、double-clickイベント、key-upイベント、key-downイベント、keypressイベント、mouse-moveイベント、mouse-upイベント、mouse-downイベント、mouse-scrollイベント、およびmouse-wheelイベントを含み得る。対話イベントは、アプリケーション145によって検出され得、クライアントデバイス125のI/Oデバイスから、アプリケーション145を介して、情報リソース150またはコンテンツアイテム155に渡され得る。アプリケーション145は、対話イベントがコンテンツアイテム155に起因するか否かを決定することができる。いくつかの実装形態では、対話イベントの座標が、コンテンツアイテム155に対応する座標の範囲内であるとき、アプリケーション145は、対話イベントがコンテンツアイテム155に起因すると決定することができる。対話イベントがコンテンツアイテム155に起因すると決定される場合、アプリケーション145は、その中の実行可能コードによって処理するために、対話イベントをコンテンツアイテム155に渡すことができる。

【0056】

クライアントデバイス125のI/Oデバイスを用いたユーザによる単一の対話に応じて、複数の対話イベントが渡され得、コンテンツアイテム155の対応するイベントハンドラ250A～Nがトリガされ得る。コンテンツアイテム155のレンダリングに対応するクライアントデバイス125のタッチスクリーン上の点における単一の対話とともに、アプリケーション145は、touch-startイベントおよびtouch-endイベントをコンテンツアイテム155に順次渡すことができる。アプリケーション145は、ファクタの中でも、ユーザ対話の位置および持続時間の変化に基づいて、1つまたは複数の追加イベントを渡すことができる。ユーザ対話の間の位置の変化に基づいて、アプリケーション145はまた、特に、touch-startイベントとtouch-endイベントとの間で、on-scrollイベントまたはtouch-moveイベントをコンテンツアイテム155に渡すこともできる。ユーザ対話の持続時間が、あらかじめ決定された

時間しきい値よりも大きい場合、アプリケーション145はまた、touch-startイベントおよびtouch-endイベントの後で、long pressイベントをコンテンツアイテム155に渡すこともできる。逆に、ユーザ対話の持続時間が、あらかじめ決定された時間しきい値以下である場合、アプリケーション145はまた、touch-startイベントおよびtouch-endイベントの後で、on-clickイベントをコンテンツアイテム155に渡すこともできる。

【0057】

イベント検出器220は、アプリケーション145によって渡された1つまたは複数の対話イベントを受信、識別、または検出することができる。イベント検出器220は、実行可能コードの第1の部分210Aにおいて符号化されるので、イベント検出器220は、実行可能コードの第2の部分210Bにおいて符号化された対話イベントのための対応するイベントハンドラ250A~Nのトリガリングより前に、対話イベントを受信または検出することができる。さらに、アプリケーション145から渡された同じ対話イベントが、第1の部分210Aにおいて符号化されたイベント検出器220によって、および次いで、コンテンツアイテム155の第2の部分210Bにおいて符号化された対応するイベントハンドラ250A~Nによって、複数回検出または登録され得る。いくつかの実装形態では、イベント検出器220は、対話イベントがアプリケーション145によってコンテンツアイテム155に渡されるとき、1つまたは複数の対話イベントを順次検出することができる。ユーザ対話が、クライアントデバイス125のタッチスクリーンを伴うものであるとき、イベント検出器220は、アプリケーション145によって渡されたtouch-startイベントおよびtouch-endイベントを順次検出することができる。いくつかの実装形態では、イベント検出器220は、touch-startイベントとtouch-endイベントとの間で、on-scrollイベントまたはtouch-moveイベントを検出することができる。いくつかの実装形態では、イベント検出器220は、touch-startイベントおよびtouch-endイベントの後で、on-clickイベントを検出することができる。対話検証スクリプト160が検出された対話イベントに応答することになるか否かは、どの対応するイベントハンドラ250A~Nが関数260とともに構成されるかに依存し得る。

【0058】

イベント検出器220による各対話イベントの検出の後で、対話イベントに対応するイベントハンドラ250A~Nは、対話イベントを処理することができる。各イベントハンドラ250A~Nは、特定のタイプの対話イベントを処理することができる。たとえば、touch-startイベントハンドラは、touch-startイベントを処理することができ、on-scrollイベントハンドラは、on-scrollイベントを処理することができ、touch-endイベントハンドラは、touch-endイベントを処理することができ、on-clickイベントハンドラは、on-clickイベントを処理することができる、などとなる。イベントハンドラ250A~Nのサブセットが、対話検証スクリプト160において指定されるようにアタッチされ得る。いくつかの実装形態では、イベントハンドラ250A~Nのサブセットは、互いに、またはイベント検出器220、登録ロガー230、もしくは関数アクティベータ240にアタッチされ得る。イベントハンドラ250A~Nの残りのセットは、対話検証スクリプト160からのいかなる影響もなしに、プログラムされた機能を実施し得る。いくつかの実装形態では、イベント検出器220は、報告モードにおいて作動することができる。報告モードである間、イベント検出器220は、特定のタイプの対話と、イベントハンドラ250A~Nのために構成された関数260とを含む、診断メッセージを生成することができる。イベント検出器は、データ処理システム110に診断メッセージを送信することができる。このようにして、多数のクライアントデバイス125にわたって集められた診断メッセージが、コンテンツアイテム155の挙動を決定または測定するために使用され得る。さらに、報告モードである間、イベント検出器220は、1つまたは複数のイベントハンドラ250A~Nなど、コンテンツアイテム155の最初にプログラムされた機能を維持することができる。イベント検出器220はまた、関数アクティベータ240の機能を無効化することもできる。

【0059】

少なくとも1つのイベントハンドラ250A~N(以下、一般にイベントハンドラ250Aと呼ばれる)は、コンテンツアイテム155から渡された対話イベントに応じて、あらかじめ定義さ

10

20

30

40

50

れたアクションを実施するための関数260とともに構成され得る。他のイベントハンドラ250A~N(以下、一般にイベントハンドラ250Bと呼ばれる)は、関数260なしで構成され得る。したがって、イベントハンドラ250Aは、アプリケーション145からコンテンツアイテム155に渡された対応する対話イベントの検出または受信に応じて、あらかじめ定義されたアクションを実施するために、関数260を呼び出すか、またはトリガすることができる。上記で説明したように、関数260は、コンテンツアイテム評価器130によって指定された、あらかじめ指定された関数のセットのうちの1つを含み得る。いくつかの実装形態では、関数、および関数のトリガリングに応じて実施されることになる、あらかじめ決定されたアクションは、アプリケーションプログラミングインターフェース(API)によって定義され得る(たとえば、アプリケーション145によってアクセスまたはロードされた情報リソース150から、別の情報リソースにナビゲートするための「ExitAPI.exit()」)。イベントハンドラ250Aは、特に、touch-endイベントまたはon-clickイベントであり得る。対照的に、イベントハンドラ250Bは、アプリケーション145からコンテンツアイテム155に渡された対応する対話イベントの検出または受信に応じて、アクションを実施しないことができ、または関数260以外の他の機能呼び出すことができる。イベントハンドラ250Bは、特に、touch-endイベントまたはon-clickイベントであり得る。いくつかの実装形態では、イベントハンドラ250Bによって処理される対話イベントのタイプは、イベントハンドラ250Aによって処理される対話イベントのタイプとは異なり得る。たとえば、イベントハンドラ250Aは、関数260とともに構成されたtouch-endイベントハンドラであり得るが、イベントハンドラ250Bは、on-clickイベントハンドラであり得る。

【0060】

アプリケーション145によって渡された対話イベントを検出すると、イベント検出器220は、対話イベントが、対話検証スクリプト160において指定された、アタッチされたイベントハンドラ250A~Nのうちの1つに対応するか否かを決定することができる。上記で説明したように、アタッチされたイベントハンドラ250A~Nは、データベース140において維持された、あらかじめ定義されたイベントハンドラのリストによって指定されたイベントハンドラのうちのいずれか1つであり得る。いくつかの実装形態では、アタッチされたイベントハンドラ250A~Nは、特に、touch-startイベントハンドラ、touch-endイベントハンドラ、およびon-clickイベントハンドラを含み得る。対話イベントが、アタッチされたイベントハンドラ250A~Nのいずれにも対応しないと決定される場合、対応するイベントハンドラ250A~N(いずれかがコンテンツアイテム155において構成される場合)は、対話イベントを処理することができる。一方、対話イベントが、アタッチされたイベントハンドラ250A~Nのうちの1つに対応すると決定される場合、イベント検出器220は、イベントハンドラ250A~Nに対応する対話イベントの検出の時間を識別することができる。いくつかの実装形態では、イベント検出器220は、時間を追跡するために、タイマーを維持することができる。いくつかの実装形態では、イベント検出器220は、対話イベントが、関数260とともに構成されたイベントハンドラ250Aに対応するか否かを決定することができる。対話イベントが、関数260とともに構成されたイベントハンドラ250Aに対応すると決定される場合、イベント検出器220は、イベントハンドラ250Aに対応する対話イベントの検出の時間を識別することができる。イベント検出器220はまた、アタッチされたイベントハンドラ250A~Nのうちの1つに対応する別の対話イベントの検出を待機するために、イベントハンドラ250Aに対応する対話イベントの検出から、タイマーのカウントを開始することもできる。

【0061】

イベント検出器220が対話イベントを監視するとき、登録ロガー230は、あらかじめ決定されたアクションを実施するための関数260の呼出しの登録を識別または監視することができる。イベントハンドラ250A~Nのうちの1つが呼び出されることの検出に応じて、登録ロガー230は、どのイベントハンドラ250A~Nが呼び出されるかを識別することができる。いくつかの実装形態では、登録ロガー230は、アプリケーション145によって渡された対応する対話イベントの検出に応じて、どのイベントハンドラ250A~Nが呼び出されることに

なるかを識別することができる。どのイベントハンドラ250A～Nがトリガされるかの識別を使用して、登録ロガー230は、関数260とともに構成されたイベントハンドラ250Aがトリガされるか否か、または構成なしのイベントハンドラ250Bがトリガされるか否かを決定する。イベントハンドラ250Aが、対応する対話イベントに応じてトリガされると決定される場合、登録ロガー230は、イベントハンドラ250Aにおいて構成された関数260の呼出しを登録することができる。いくつかの実装形態では、イベントハンドラ250Aに対応する対話イベントを検出すると、登録ロガー230は、関数260の呼出しを登録することができる。登録ロガー230は、関数260の呼出しを追跡し、クライアントデバイス125のローカルのメモリにおいて、関数260の呼出しの登録を記憶することができる。イベントハンドラ250Aのための対応する対話イベントが、アプリケーション145からコンテンツアイテム155に渡されるとき、イベントハンドラ250Aは、関数260を呼び出すか、またはトリガすることができる。いくつかの実装形態では、関数260の呼出しを登録する際に、登録ロガー230は、関数260のためのあらかじめ決定されたアクションの実行を防止または保留することができる。登録ロガー230は、関数260の呼出しの登録の時間を識別することができる。いくつかの実装形態では、登録ロガー230は、時間を追跡するために、タイマーを維持することができる。関数260の呼出しの登録に応じて、登録ロガー230は、アタッチされたイベントハンドラ250A～Nのうちの1つに対応する別の対話イベントの検出を待機するために、登録の時間からタイマーのカウントを開始することができる。

【0062】

一方、イベントハンドラ250Bが、対応する対話イベントに応じてトリガされると決定される場合、登録ロガー230は、イベントハンドラ250Aにおいて構成された関数260の呼出しの登録を識別することができる。いくつかの実装形態では、イベントハンドラ250Bのトリガリングは、イベントハンドラ250Aのトリガリングより前に発生し得る。イベントハンドラ250Bが、イベントハンドラ250Aのトリガリングより前にトリガされるとき、登録ロガー230は、関数の呼出しの登録がないことを識別することができる。登録ロガー230はまた、イベントハンドラ250Aのトリガリングを待機することもでき、関数呼出し260の登録を監視し続けることができる。いくつかの実装形態では、イベントハンドラ250Bのトリガリングは、イベントハンドラ250Aのトリガリングの後で発生し得る。イベントハンドラ250Bが、イベントハンドラ250Aのトリガリングの後でトリガされるとき、イベントハンドラ250Aにおいて構成された関数260は、すでに呼び出されていることがあり、登録ロガー230は、関数260の呼出しを登録していることがある。

【0063】

あらかじめ決定されたしきい値(たとえば、時間ウィンドウ)に対する、アタッチされたイベントハンドラ250A～Nのうちの1つに対応する対話イベントの検出、および関数260の呼出しの登録のタイミングを使用して、関数アクティベータ240は、アプリケーション145に、関数260のあらかじめ決定されたアクションを実施させることができる。関数アクティベータ240は、アタッチされたイベントハンドラ250A～Nに対応する1つまたは複数の対話イベントが検出されるか否かを決定することができる。上記で説明したように、アタッチされたイベントハンドラ250A～Nは、関数260とともに構成されたイベントハンドラ250Aと、構成なしのイベントハンドラ250Bとを含み得る。検出されると決定された、アタッチされたイベントハンドラ250A～Nのうちの1つに対応する各対話イベントについて、関数アクティベータ240は、対話イベントのための時間を識別することができる。対話イベントのいずれも検出されると決定されない場合、関数アクティベータ240は、アプリケーション145によって渡された対話イベントのうちのいずれかの検出を待機することができる。関数アクティベータ240はまた、関数260の呼出しが登録されているか否かを決定することもできる。関数260の呼出しが登録されていると決定される場合、関数アクティベータ240は、関数260の呼出しの登録の時間を識別することができる。関数260の呼出しが登録されていないと決定される場合、関数アクティベータ240は、関数260の呼出しの登録を待機することができる。対話イベントの検出または関数260の呼出しの登録に応じて、関数アクティベータ240は、対話イベントの検出の時間を、登録の時間と比較することができる。

【0064】

関数アクティベータ240は、アタッチされたイベントハンドラ250A～Nのうちの1つに対応する対話イベントの検出が、イベントハンドラ250Aの関数260の呼出しの登録に先行するか、後続するかを決定することができる。対話イベントの検出が関数260の呼出しの登録に先行するか、後続するかを決定するために、関数アクティベータ240は、対話イベントの検出の時間を、登録の時間と比較することができる。関数アクティベータ240は、アタッチされたイベントハンドラ250A～Nのうちのいずれか1つに対応する対話イベントが最初に検出され、関数260の呼出しが登録されていないとき、対話イベントの検出が関数260の呼出しの登録に先行すると決定することができる。対話イベントは、関数260とともに構成されたイベントハンドラ250A、または関数260の構成なしのイベントハンドラ250Bに
10
対応し得る。関数アクティベータ240はまた、対話イベントの検出の時間が関数260の呼出しの登録の時間より前であるとき、対話イベントの検出が関数260の呼出しの登録に先行すると決定することもできる。逆に、関数アクティベータ240は、関数260の呼出しが最初に登録され、アタッチされたイベントハンドラ250A～Nのうちのいずれかに対応する対話イベントのいずれも検出されないとき、対話イベントの検出が関数260の呼出しの登録に後続すると決定することができる。関数アクティベータ240はまた、対話イベントの検出の時間が関数260の呼出しの登録の時間の後であるとき、対話イベントの検出が関数260の呼出しの登録に後続すると決定することもできる。

【0065】

対話イベントの検出が関数260の呼出しの登録に先行するか、後続するかに基づいて、
20
関数アクティベータ240は、検出と登録との間のあらかじめ決定されたしきい値を設定することができる。あらかじめ決定されたしきい値は、アプリケーション145が関数アクティベータ240、および次いで、クライアントデバイス125のI/Oデバイスから受信された対話イベントの処理において、対応するイベントハンドラ250A～Nを実行するとき、経過すると期待される時間を考慮するために設定され得る。対話イベントの検出が関数260の呼出しの登録に先行すると決定される場合、関数アクティベータ240は、あらかじめ決定されたしきい値を、あらかじめ決定された時間の範囲(たとえば、1000ms～2000ms)からの1つの値に設定することができる。関数アクティベータ240は、関数260とともに構成されたイベントハンドラ250Aのタイプに基づいて、あらかじめ決定されたしきい値を設定することができる。たとえば、touch-endイベントハンドラが関数260とともに構成される場合、
30
関数アクティベータ240は、あらかじめ決定されたしきい値を1500msに設定することができる。一方、対話イベントの検出が関数260の呼出しの登録に後続すると決定される場合、関数アクティベータ240は、あらかじめ決定されたしきい値を、異なるあらかじめ決定された時間の範囲(たとえば、50ms～350ms)からの別の値に設定することができる。関数アクティベータ240は、関数260なしで構成されたイベントハンドラ250Bのタイプに基づいて、あらかじめ決定されたしきい値を設定することができる。いくつかの実装形態では、関数アクティベータ240は、対話イベントの検出が関数260の呼出しの登録に先行するか、後続するかにかかわらず、あらかじめ決定されたしきい値を同じ固定された値に設定することができる。

【0066】

関数アクティベータ240は、アタッチされたイベントハンドラ250A～Nのうちの1つに対応する対話イベントの検出と、関数260の呼出しの登録との間の時間が、あらかじめ決定されたしきい値未満であるか否かを決定することができる。イベントハンドラ250A以外のアタッチされたイベントハンドラ250A～Nのうちの1つに対応する対話イベントが、登録より前に検出される場合、関数アクティベータ240は、あらかじめ決定されたしきい値内で、関数260の呼出しの登録を待機することができる。いくつかの実装形態では、関数アクティベータ240は、対話イベントの検出からの時間をカウントするために、タイマーを維持することができる。関数アクティベータ240は、対話イベントの検出から経過した時間を、あらかじめ決定されたしきい値と比較することができる。逆に、関数260の呼出しの登録が、対話イベントの検出より前に発生する場合、関数アクティベータ240は、あらか
50

じめ決定されたしきい値内で、アタッチされたイベントハンドラ250A～Nのうちの1つに対応する対話イベントの検出を待機することができる。いくつかの実装形態では、関数アクティベータ240は、関数260の呼出しの登録からの時間をカウントするために、タイマーを維持することができる。関数アクティベータ240は、関数260の呼出しの登録から経過した時間を、あらかじめ決定されたしきい値と比較することができる。

【0067】

対話イベントの検出および関数260の呼出しの登録が、あらかじめ決定されたしきい値内に発生したと決定される場合、関数アクティベータ240は、アプリケーション145に、関数260のあらかじめ決定されたアクションを実施させることができる。いくつかの実装形態では、関数アクティベータ240は、イベントハンドラ250Aにおいて構成された関数260のトリガリングを許可することができる。いくつかの実装形態では、関数アクティベータ240は、あらかじめ決定されたアクションを実行するために、関数260の呼出しの登録を取り出すことができる。対照的に、対話イベントの検出および関数260の呼出しの登録が、あらかじめ決定されたしきい値内に発生しなかったと決定される場合、関数アクティベータ240は、関数260のあらかじめ決定されたアクションの実行を防止することができる。いくつかの実装形態では、関数アクティベータ240は、対話イベントを無視することができる。

【0068】

このようにして、対話検証スクリプト160は、ユーザが期待したときに対して早すぎるかまたは遅れた関数260のトリガリングを防止するために、コンテンツアイテム155に異なる挙動をさせることができる。したがって、対話検証スクリプト160は、コンテンツアイテム155および下にある情報リソース150との対話の品質を向上させ得る。さらに、対話検証スクリプト160が、コンテンツアイテム155の残りの構成に最小限の影響を及ぼし得るので、コンテンツアイテム155(たとえば、イベントハンドラ250A～N)の最初にプログラムされた機能が保たれ得る。

【0069】

次に図3Aを参照すると、図示されているものは、コンテンツアイテム上で実施される対話を検証するプロセスのためのシーケンス図である。概して、プロセス300Aは、対話検証スクリプト160なしで、アプリケーション145および関数260に関して連続的に示されたコンテンツアイテム155の機能を含み得る。プロセス305Aは、対話検証スクリプト160ありで、アプリケーション145および関数260に関して連続的に示されたコンテンツアイテム155の機能を含み得る。プロセス300Aと305Aの両方において、コンテンツアイテム155のon-clickイベントハンドラが、あらかじめ決定されたアクションを実施するために、関数260をトリガするように構成されていることがある。対話検証スクリプト160は、on-clickイベントハンドラを、コンテンツアイテム155のtouch-endイベントハンドラとともにアタッチしていることがある。

【0070】

プロセス300Aでは、クライアントデバイス125のユーザは、情報リソース150内で提示されたコンテンツアイテム155のレンダリングに対応する、タッチスクリーン上の点へのタッチ302を実施し得る。アプリケーション145は、コンテンツアイテム155に渡すために、タッチ302をtouch-startイベント304に変換することができる。コンテンツアイテム155のtouch-startイベントハンドラが、関数260の構成を欠いていることがあるので、コンテンツアイテム155は、touch-startイベント304を無視するか、または関数260とは異なる別の動作を実施することができる。クライアントデバイス125のタッチスクリーンからのタッチ302のリリース306があると、アプリケーション145は、リリース306をtouch-endイベント308と、touch-endイベント308に後続するon-clickイベント310とに変換することができる。タッチ302およびリリース306は、ユーザによるクライアントデバイス125のタッチスクリーンとの同じ対話の一部であり得る。コンテンツアイテム155のtouch-endイベントハンドラが、関数260の構成を欠いていることがあるので、コンテンツアイテム155は、touch-endイベント308を無視するか、または関数260とは異なる別の動作を実施することがで

きる。しかしながら、コンテンツアイテム155は、関数260をトリガするために、on-click イベントハンドラとともに構成され得る。on-click イベント310の検出に応じて、コンテンツアイテム155は、関数260(たとえば、別の情報リソースにナビゲートするための「ExitAPI.exit()」)の登録312を実施することができる。次いで、アプリケーション145は、関数260の実行314(たとえば、他の情報リソースへのナビゲーション)を実施し得る。

【0071】

プロセス305Aでは、機能は、on-click イベント310まで、プロセス300Aと同様または同じであり得る。この時点で、on-click イベント310は、元のon-click イベントハンドラより前に、最初に対話検証スクリプト160を通過し得る。したがって、登録312は、対話検証スクリプト160による処理の後で発生し得る。対話検証スクリプト160に従って、コンテンツアイテム155は、後続の対話イベントを待機するために、タイマーを開始すること316ができる。アプリケーション145はまた、コンテンツアイテム155のon-click イベントハンドラに渡すために、ユーザによるリリース306を別のon-click イベント318に変換することもできる。on-click イベント318を検出すると、on-click イベントハンドラは、関数260の登録312'を実施することができる。対話検証スクリプト160が、タイマーの開始316および登録312'があらかじめ決定された時間ウィンドウ(たとえば、200ms)内であると決定する場合、対話検証スクリプト160は、アプリケーション145が関数260の実行314'を実施することを許可することができる。

【0072】

次に図3Bを参照すると、図示されているものは、コンテンツアイテム上で実施される対話を検証するプロセスのためのシーケンス図である。概して、プロセス300Bは、対話検証スクリプト160なしで、アプリケーション145および関数260に関して連続的に示されたコンテンツアイテム155の機能を含み得る。プロセス305Bは、対話検証スクリプト160ありで、アプリケーション145および関数260に関して連続的に示されたコンテンツアイテム155の機能を含み得る。図3Aに関して詳述したプロセス300Aおよび305Aの場合のように、プロセス300Bおよび305Bでは、コンテンツアイテム155のon-click イベントハンドラが、あらかじめ決定されたアクションを実施するために、関数260をトリガするように構成されていることがある。加えて、対話検証スクリプト160は、on-click イベントハンドラを、コンテンツアイテム155のtouch-end イベントハンドラとともにアタッチしていることがある。

【0073】

プロセス300Bでは、クライアントデバイス125のユーザは、情報リソース150内で提示されたコンテンツアイテム155のレンダリングに対応する、タッチスクリーン上の点へのタッチ320を実施し得る。アプリケーション145は、コンテンツアイテム155に渡すために、タッチ320をtouch-start イベント322に変換することができる。コンテンツアイテム155のtouch-start イベントハンドラが、関数260の構成を欠いていることがあるので、コンテンツアイテム155は、touch-start イベント322を無視するか、または関数260とは異なる別の動作を実施することができる。続いて、クライアントデバイス125のユーザは、タッチスクリーン上のある点から別の点に移動することによって、スクロール324を実施し得る。クライアントデバイス125のタッチスクリーンからのタッチ320および後に続くスクロール324のリリース326があると、アプリケーション145は、リリース326をtouch-end イベント328に変換することができる。タッチ320およびリリース326は、ユーザによるクライアントデバイス125のタッチスクリーンとの同じ対話の一部であり得る。スクロール324がリリース326より前に発生したので、on-click イベントはトリガされないことがある。コンテンツアイテム155のtouch-end イベントハンドラが、関数260の構成を欠いていることがあるので、コンテンツアイテム155は、touch-end イベント328を無視するか、または関数260とは異なる別の動作を実施することができる。コンテンツアイテム155が、touch-end イベント328ではなく、on-click イベントに応じて、関数260をトリガするように構成されているので、プロセス305Bは、プロセス300Bと同様であり得る。

【0074】

次に図3Cを参照すると、図示されているものは、コンテンツアイテム上で実施される対話を検証するプロセスのためのシーケンス図である。概して、プロセス300Cは、対話検証スクリプト160なしで、アプリケーション145および関数260に関して連続的に示されたコンテンツアイテム155の機能を含み得る。プロセス305Cは、対話検証スクリプト160ありで、アプリケーション145および関数260に関して連続的に示されたコンテンツアイテム155の機能を含み得る。プロセス300Cと305Cの両方において、コンテンツアイテム155のtouch-endイベントハンドラが、あらかじめ決定されたアクションを実施するために、関数260をトリガするように構成されていることがある。対話検証スクリプト160は、touch-endイベントハンドラを、コンテンツアイテム155のon-clickイベントハンドラとともにアタッチしていることがある。

10

【0075】

プロセス300Cでは、クライアントデバイス125のユーザは、情報リソース150内で提示されたコンテンツアイテム155のレンダリングに対応する、タッチスクリーン上の点へのタッチ340を実施していることがある。アプリケーション145は、コンテンツアイテム155に渡すために、タッチ340をtouch-startイベント342に変換することができる。コンテンツアイテム155のtouch-startイベントハンドラが、関数260の構成を欠いていることがあるので、コンテンツアイテム155は、touch-startイベント342を無視するか、または関数260とは異なる別の動作を実施することができる。クライアントデバイス125のタッチスクリーンからのタッチ340のリリース344があると、アプリケーション145は、リリース344をtouch-endイベント346と、touch-endイベント346に後続するon-clickイベント348とに変換することができる。タッチ340およびリリース344は、ユーザによるクライアントデバイス125のタッチスクリーンとの同じ対話の一部であり得る。touch-endイベントハンドラが、touch-endイベント346の検出に応じて関数260をトリガするように構成されていることがあるので、コンテンツアイテム155は、関数260(たとえば、別の情報リソースにナビゲートするための「ExitAPI.exit()」)の登録350を実施することができる。次いで、アプリケーション145は、関数260の実行352(たとえば、他の情報リソースへのナビゲーション)を実施し得る。対照的に、on-clickイベントハンドラは、関数260の構成を欠くことがあるので、コンテンツアイテム155は、後続のon-clickイベント348を無視し得る。ユーザの観点からすれば、関数260のトリガリングが、on-clickイベントハンドラではなく、touch-endイベントハンドラに結びつけられるので、関数260の実行352が早すぎるように見えることがある、予期しないものであり得る。

20

30

【0076】

プロセス305Cでは、機能は、touch-endイベント346まで、プロセス300Cと同様または同じであり得る。この時点で、touch-endイベント346は、対話検証スクリプト160を通過して、コンテンツアイテム155の元のtouch-endイベントハンドラまで行き得る。touch-endイベント346の検出に応じて、コンテンツアイテム155は、関数260の呼出しの登録350'を実施することができ、対話検証スクリプト160に従って、アタッチされたイベントハンドラのうちの1つに対応する別の対話イベントを待機するために、タイマーを開始することができる。後続のon-clickイベント348が受信されるとき、コンテンツアイテム155は、登録350'およびon-clickイベント348の受信のタイミングの比較354を実施することができる。関数260の呼出しの登録350'、およびon-clickイベント348の受信が、あらかじめ決定された時間ウィンドウ(たとえば、1500ms)内に発生すると決定される場合、対話検証スクリプト160は、アプリケーション145が関数260の実行352'を実施することを許可することができる。このようにして、早すぎるかまたは予期しない関数260のトリガリングが防止され得る。

40

【0077】

次に図3Dを参照すると、図示されているものは、コンテンツアイテム上で実施される対話を検証するプロセスのためのシーケンス図である。概して、プロセス300Dは、対話検証スクリプト160なしで、アプリケーション145および関数260に関して連続的に示されたコンテンツアイテム155の機能を含み得る。プロセス305Dは、対話検証スクリプト160ありで

50

、アプリケーション145および関数260に関して連続的に示されたコンテンツアイテム155の機能を含み得る。図3Cに関して詳述したプロセス300Cおよび305Cの場合のように、プロセス300Dおよび305Dでは、コンテンツアイテム155のtouch-endイベントハンドラが、あらかじめ決定されたアクションを実施するために、関数260をトリガするように構成されていることがある。対話検証スクリプト160は、touch-endイベントハンドラを、コンテンツアイテム155のon-clickイベントハンドラとともにアタッチしていることがある。

【0078】

プロセス300Dでは、クライアントデバイス125のユーザは、情報リソース150内で提示されたコンテンツアイテム155のレンダリングに対応する、タッチスクリーン上の点へのタッチ360を実施していることがある。アプリケーション145は、コンテンツアイテム155に渡すために、タッチ360をtouch-startイベント362に変換することができる。コンテンツアイテム155のtouch-startイベントハンドラが、関数260の構成を欠いていることがあるので、コンテンツアイテム155は、touch-startイベント362を無視するか、または関数260とは異なる別の動作を実施することができる。続いて、クライアントデバイス125のユーザは、タッチスクリーン上のある点から別の点に移動することによって、スクロール364を実施し得る。クライアントデバイス125のタッチスクリーンからのタッチ360および後に続くスクロール364のリリース366があると、アプリケーション145は、リリース366をtouch-endイベント368に変換することができる。タッチ360およびリリース366は、ユーザによるクライアントデバイス125のタッチスクリーンとの同じ対話の一部であり得る。touch-endイベントハンドラが、touch-endイベント368の検出に応じて関数260をトリガするように構成されていることがあるので、コンテンツアイテム155は、関数260(たとえば、別の情報リソースにナビゲートするための「ExitAPI.exit()」)の登録370を実施することができる。次いで、アプリケーション145は、関数260の実行372(たとえば、他の情報リソースへのナビゲーション)を実施し得る。スクロール364がリリース366より前に発生したので、on-clickイベントはトリガされないことがある。ユーザの観点からすれば、関数260のトリガリングが、on-clickイベントハンドラではなく、touch-endイベントハンドラに結びつけられるので、関数260の実行372が早すぎるように見えることがあり、予期しないものであり得る。

【0079】

プロセス305Dでは、機能は、touch-endイベント368まで、プロセス300Dと同様または同じであり得る。この時点で、touch-endイベント368は、対話検証スクリプト160を通過して、コンテンツアイテム155の元のtouch-endイベントハンドラまで行き得る。touch-endイベント368の検出に応じて、コンテンツアイテム155は、関数260の呼出しの登録370'を実施することができ、対話検証スクリプト160に従って、アタッチされたイベントハンドラのうちの1つに対応する別の対話イベントを待機するために、タイマーを開始することができる。アプリケーション145から渡された、後に続くon-clickイベントがないので、対話検証スクリプト160は、あらかじめ決定された時間ウィンドウ(たとえば、1500ms)内で検出された他の対話イベントがないと決定することができる。したがって、対話検証スクリプト160は、アプリケーション145が関数260の実行を実施することを制限することができる。このようにして、早すぎるかまたは予期しない関数260のトリガリングが防止され得る。

【0080】

次に図4を参照すると、図示されているものは、コンテンツアイテム上で実施される対話を検証する方法400の流れ図である。方法400に関して本明細書で説明する機能は、図1に示されるようなシステム100(たとえば、データ処理システム110、コンテンツプロバイダ115、コンテンツパブリッシャ120、およびクライアントデバイス125)、もしくは図6に示されるようなコンピューティングデバイス、またはそれらの任意の組合せによって実施されるか、または別段に実行され得る。

【0081】

データ処理システムは、データベースからのコンテンツアイテムを識別することができ

る(405)。データ処理システムは、あらかじめ指定された関数とともに構成されたイベントハンドラを識別することができる(410)。あらかじめ指定された関数は、データ処理システムによって検索されることになる、あらかじめ指定された関数のセットのうちの1つであり得る。いくつかの実装形態では、関数は、APIによって定義され得る。データ処理システムは、関数なしのイベントハンドラを識別することができる(415)。構成なしのイベントハンドラは、関数の構成をもつイベントハンドラとは異なり得る。

【0082】

データ処理システムは、イベントハンドラが、あらかじめ定義されたハンドラのリストのうちの1つに一致するか否かを決定することができる(420)。あらかじめ定義されたリストにおける各イベントハンドラは、コンピューティングデバイスのI/Oデバイス上の単一の物理的対話(たとえば、touch-startイベント、touch-endイベント、およびon-clickイベント)に関連付けられ得る。イベントハンドラが、あらかじめ定義されたハンドラのリストのうちの1つに一致すると決定される場合、データ処理システムは、構成なしのイベントハンドラをアタッチすることができる(425)。いくつかの実装形態では、データ処理システムは、構成なしのイベントハンドラと、構成をもつイベントハンドラとを、コンテンツアイテムに挿入されることになるスクリプトにアタッチすることができる。いずれの場合も、データ処理システムは、より多くのイベントハンドラがあるか否かを決定することができる(430)。解析されることになるより多くのイベントハンドラがある場合、データ処理システムは、ステップ415~430の機能を繰り返すことができる。

【0083】

解析されることになるイベントハンドラがそれ以上ない場合、データ処理システムは、コンテンツアイテムのためのコードの上部分にアタッチメントとともにスクリプトを挿入することができる(435)。このようにして、コンテンツアイテムがコンピューティングデバイスによって読み取られ、実行されるとき、スクリプトは、コンテンツアイテムにおいて最初にプログラムされたイベントハンドラに対応するコードより前に読み取られ得る。データ処理システムは、挿入されたスクリプトをもつコンテンツアイテムを、コンピューティングデバイスに提供することができる(440)。次に、コンテンツアイテムが、コンピューティングデバイス上で提示された情報リソースに挿入され得る。

【0084】

次に図5Aを参照すると、図示されているものは、コンテンツアイテム上で実施される対話を検証する方法500の流れ図である。方法500に関して本明細書で説明する機能は、図1に示されるようなシステム100、図2に示されるようなシステム200、もしくは図6に示されるようなコンピューティングデバイス、またはそれらの任意の組合せによって実施されるか、または別段に実行され得る。図4に関して、方法500は、ステップ440からの方法400の継続であり得る。方法500は、最初に検出された対話イベントのためのコンテンツアイテムのイベントハンドラが、あらかじめ定義されたアクションを実施するための関数の構成を欠くことがある、シナリオに対応し得る。

【0085】

コンピューティングデバイスは、対話イベントを検出することができる(502)。対話イベントのためのコンテンツアイテムのイベントハンドラは、関数の構成を欠くことがある。いくつかの実装形態では、コンピューティングデバイスは、検出された対話イベントが、アタッチされたイベントハンドラのセットのうちの1つに対応すると決定することができる。あらかじめ定義されたセットにおける各イベントハンドラは、コンピューティングデバイスのユーザによる単一の対話(たとえば、touch-startイベント、touch-endイベント、およびon-clickイベント)に関連し得る。コンピューティングデバイスは、関数呼出しが登録されているか否かを決定することができる(504)。関数は、検出された対話イベントに対応するイベントハンドラとは異なる別のイベントハンドラ上で構成され得る。関数呼出しが登録されている場合、コンピューティングデバイスは、時間しきい値を第1の値に設定することができる(506)。第1の値は、50msから350msに及び得る。一方、関数呼出しが登録されていない場合、コンピューティングデバイスは、時間しきい値を第2の値

に設定することができる(508)。第2の値は、1000msから2000msに及び得る。第1の値と第2の値の両方は、関数呼出しの登録と対話イベントの検出との間に経過した時間を考慮するために設定され得る。

【0086】

コンピューティングデバイスは、対話イベントの時間が第1の値だけ関数呼出しの時間より前であるか否かを決定することができる(510)。コンピューティングデバイスは、対話の検出に応じて、対話イベントの時間を識別することができる。コンピューティングデバイスは、関数呼出しの登録に応じて、関数呼出しの時間を識別することができる。コンピューティングデバイスは、現在の時間が第2の値だけ対話イベントの時間より前であるか否かを決定することができる(512)。いくつかの実装形態では、コンピューティングデバイスは、対話イベントの検出の時間から開始して、現在の時間を追跡するために、タイマーを維持することができる。コンピューティングデバイスは、第2の値に設定された時間しきい値内で、対話イベントの検出を待機することができる。現在の時間が第2の値だけ対話イベントの時間より後である場合、コンピューティングデバイスは、関数呼出しの登録を監視することができる(514)。関数呼出しの登録が検出されない場合、コンピューティングデバイスは、ステップ512および514の機能を繰り返し得る。

10

【0087】

対話イベントの時間が第1の値だけ関数呼出しの時間より後であるか、または、現在の時間が第2の値だけ対話イベントの時間の後である場合、コンピューティングデバイスは、対話イベントを無視することができる(516)。コンピューティングデバイスはまた、関数呼出しが登録されている場合、関数呼出しの実行を防止することもできる。一方、対話イベントの時間が第1の値だけ関数呼出しの時間の後であるか、または、現在の時間が第2の値だけ対話イベントの時間より前であるとき、関数呼出しの登録が検出される場合、コンピューティングデバイスは、関数呼出しを実行することができる(518)。いくつかの実装形態では、コンピューティングデバイスは、関数呼出しの実行を許可することができる。

20

【0088】

次に図5Bを参照すると、図示されているものは、コンテンツアイテム上で実施される対話を検証する方法550の流れ図である。方法550に関して本明細書で説明する機能は、図1に示されるようなシステム100、図2に示されるようなシステム200、もしくは図6に示されるようなコンピューティングデバイス、またはそれらの任意の組合せによって実施されるか、または別段に実行され得る。図4に関して、方法550は、ステップ440からの方法400の継続であり得る。方法550は、上記で詳述した方法500と同様であり得、検出対話イベントに対応するイベントハンドラが、あらかじめ決定されたアクションを実施するために、関数呼出しをトリガするように構成される、シナリオのためのものであり得る。

30

【0089】

コンピューティングデバイスは、対話イベントを検出することができる(552)。対話イベントの検出とともに、コンピューティングデバイスは、関数呼出しを登録することができる(554)。対話イベントのためのコンテンツアイテムのイベントハンドラは、あらかじめ定義されたアクションを実施するための関数とともに構成され得る。いくつかの実装形態では、コンピューティングデバイスは、検出された対話イベントが、関数とともに構成されたイベントハンドラに対応すると決定することができる。いくつかの実装形態では、コンピューティングデバイスは、登録時に関数の実行を防止することができる。コンピューティングデバイスは、アタッチされた対話イベントが検出されるか否かを決定することができる(556)。アタッチされた対話イベントは、あらかじめ定義された、アタッチされたイベントハンドラのセットのうちの1つであり得る。あらかじめ定義されたセットにおける各イベントハンドラは、コンピューティングデバイスのユーザによる単一の対話(たとえば、touch-startイベント、touch-endイベント、およびon-clickイベント)に関連し得る。アタッチされた対話イベントが検出される場合、コンピューティングデバイスは、時間しきい値を第1の値に設定することができる(558)。第1の値は、1000msから2000msに

40

50

及び得る。一方、アタッチされた対話イベントが検出されない場合、コンピューティングデバイスは、時間しきい値を第2の値に設定することができる(560)。第2の値は、50msから350msに及び得る。

【0090】

コンピューティングデバイスは、登録の時間が第1の値だけアタッチされた対話イベントの時間より前であるか否かを決定することができる(562)。コンピューティングデバイスは、対話の検出に応じて、対話イベントの時間を識別することができる。コンピューティングデバイスは、関数呼出しの登録に応じて、関数呼出しの時間を識別することができる。コンピューティングデバイスは、現在の時間が第2の値だけ登録の時間より前であるか否かを決定することができる(564)。いくつかの実装形態では、コンピューティングデバイスは、関数呼出しの登録の検出の時間から開始して、現在の時間を追跡するために、タイマーを維持することができる。コンピューティングデバイスは、第2の値に設定された時間しきい値内で、アタッチされた対話イベントの検出を待機することができる。現在の時間が第2の値だけ登録の時間より後である場合、コンピューティングデバイスは、アタッチされた対話イベントを監視することができる(566)。

【0091】

関数呼出しの登録が検出されない場合、コンピューティングデバイスは、ステップ564および566の機能を繰り返し得る。登録の時間が第1の値だけ対話イベントの時間より後であるか、または、現在の時間が第2の値だけ登録の時間の後である場合、コンピューティングデバイスは、対話イベントを無視することができる(568)。コンピューティングデバイスはまた、関数呼出しが登録されている場合、関数呼出しの実行を防止することもできる。一方、対話イベントの時間が第1の値だけ関数呼出しの時間の前であるか、または、現在の時間が第2の値だけ対話イベントの時間より前であるとき、関数呼出しの登録が検出される場合、コンピューティングデバイスは、関数呼出しを実行することができる(570)。いくつかの実装形態では、コンピューティングデバイスは、関数呼出しの実行を許可することができる。

【0092】

図6は、いくつかの実装形態による、(データ処理システム110およびその構成要素と、コンテンツプロバイダ115およびその構成要素と、コンテンツパブリッシャ120およびその構成要素と、クライアントデバイス125およびその構成要素とを含む)本明細書で説明するコンピュータシステムのいずれかを実装するために採用され得る、例示的なコンピュータシステム600の全体的なアーキテクチャを示す。コンピュータシステム600は、表示のために、ネットワーク105を介して情報を提供するために使用され得る。図6のコンピュータシステム600は、メモリ625に通信可能に結合された1つまたは複数のプロセッサ620と、1つまたは複数の通信インターフェース605と、1つまたは複数の出力デバイス610(たとえば、1つまたは複数のディスプレイユニット)と、1つまたは複数の入力デバイス615とを備える。プロセッサ620は、データ処理システム110、またはデータ処理システム110の他の構成要素中に含まれ得る。プロセッサ620は、コンテンツプロバイダ115中に含まれ得る。プロセッサ620は、コンテンツパブリッシャ120中に含まれ得る。プロセッサ620は、クライアントデバイス125、またはクライアントデバイス125の他の構成要素中に含まれ得る。

【0093】

図6のコンピュータシステム600では、メモリ625は、任意のコンピュータ可読記憶媒体を備え得、それぞれのシステムのための本明細書で説明する様々な機能を実施するためのプロセッサ実行可能命令などのコンピュータ命令、ならびにそれに関する、それによって生成された、または通信インターフェースもしくは入力デバイス(存在する場合)を介して受信された任意のデータを記憶し得る。図1のシステム100を再度参照すると、データ処理システム110は、特に、1つまたは複数のコンテンツアイテムの利用可能性に関連する情報を記憶するために、メモリ625を含み得る。メモリ625は、データベース140を含み得る。図6に示されるプロセッサ620は、メモリ625中に記憶された命令を実行するために使用され得、そうする際に、また、命令の実行に従って処理および/または生成された様々な情

報を、メモリから読み取るか、またはメモリに書き込み得る。

【0094】

図6に示されるコンピュータシステム600のプロセッサ620はまた、命令の実行に従って、様々な情報を送信または受信するために、通信インターフェース605に通信可能に結合されるか、または通信インターフェース605を制御するようにされ得る。たとえば、通信インターフェース605は、ワイヤードもしくはワイヤレスネットワーク、バス、または他の通信手段に結合され得、したがって、コンピュータシステム600が他のデバイス(たとえば、他のコンピュータシステム)に情報を送信するか、または他のデバイスから情報を受信することを可能にし得る。図1、図2、または図6のシステムにおいて明示的に示されないが、1つまたは複数の通信インターフェースが、システム600の構成要素の間の情報フローを容易にする。いくつかの実装形態では、通信インターフェースは、コンピュータシステム600の少なくともいくつかの態様へのアクセスポータルとしてのウェブサイトを提供するように(たとえば、様々なハードウェア構成要素またはソフトウェア構成要素を介して)構成され得る。通信インターフェース605の例には、ユーザインターフェース(たとえば、ウェブページ)が含まれ、ユーザインターフェースを通して、ユーザはデータ処理システム110と通信することができる。

10

【0095】

図6に示されるコンピュータシステム600の出力デバイス610は、たとえば、命令の実行に関連して、様々な情報が視聴またはそれ以外で知覚されることを可能にするために提供され得る。入力デバイス615は、たとえば、ユーザが手動の調整を行うこと、選択を行うこと、データを入力すること、または、命令の実行中にプロセッサと様々な方法のいずれかにおいて対話することを可能にするために提供され得る。本明細書で説明する様々なシステムのために採用され得る全体的なコンピュータシステムアーキテクチャに関する追加の情報が、本明細書でさらに提供される。

20

【0096】

本明細書で説明する主題および動作の実装形態は、本明細書で開示する構造およびそれらの構造的等価物を含む、デジタル電子回路において、または有形媒体上で実施されたコンピュータソフトウェア、ファームウェア、もしくはハードウェアにおいて、あるいはそれらのうちの1つまたは複数の組合せにおいて実装され得る。本明細書で説明する主題の実装形態は、1つまたは複数のコンピュータプログラム、すなわち、データ処理装置による実行のための、またはデータ処理装置の動作を制御するために、コンピュータ記憶媒体上に符号化されたコンピュータプログラム命令の1つまたは複数のモジュールとして実装され得る。プログラム命令は、データ処理装置による実行のために、好適な受信機装置に送信するための情報を符号化するために生成される、人工的に生成された伝搬信号、たとえば、マシンにより生成された電気信号、光信号、または電磁信号上で符号化され得る。コンピュータ記憶媒体は、コンピュータ可読記憶デバイス、コンピュータ可読記憶基板、ランダムもしくは順次アクセスメモリアレイもしくはデバイス、またはそれらのうちの1つもしくは複数の組合せであり得るか、またはその中に含まれ得る。その上、コンピュータ記憶媒体は、伝搬信号ではないが、コンピュータ記憶媒体は、人工的に生成された伝搬信号において符号化されたコンピュータプログラム命令のソースまたは宛先を含み得る。コンピュータ記憶媒体はまた、1つまたは複数の別個の物理的構成要素または媒体(たとえば、複数のCD、ディスク、または他の記憶デバイス)であり得るか、またはその中に含まれ得る。

30

40

【0097】

本明細書で開示する特徴は、スマートテレビジョンモジュール(または、接続されたテレビジョンモジュール、ハイブリッドテレビジョンモジュールなど)上で実装され得、スマートテレビジョンモジュールは、インターネット接続性を(たとえば、ケーブル、衛星、オーバージエア、または他の信号を介して受信された)より従来のテレビジョンプログラミングソースと統合するように構成された、処理モジュールを含み得る。スマートテレビジョンモジュールは、テレビ受信機に物理的に組み込まれ得るか、あるいは、セットト

50

ップボックス、Blu-ray(登録商標)もしくは他のデジタルメディアプレーヤ、ゲームコンソール、ホテルテレビジョンシステム、または他のコンパニオンデバイスなど、別個のデバイスを含み得る。スマートテレビジョンモジュールは、視聴者がウェブ上、ローカルケーブルTVチャンネル上、衛星TVチャンネル上の、またはローカルハードドライブ上に記憶された、ビデオ、映画、写真、および他のコンテンツを検索および発見することを可能にするように構成され得る。セットトップボックス(STB)またはセットトップユニット(STU)は、チューナーを含み、テレビ受信機および外部の信号ソースに接続し、信号を、次いでテレビジョン画面または他のディスプレイデバイス上に表示されるコンテンツに変え得る、情報アプライアンスデバイスを含み得る。スマートテレビジョンモジュールは、ウェブブラウザおよび複数のストリーミングメディアサービス、接続されたケーブルまたは衛星メディアソース、他のウェブ「チャンネル」など、複数の異なるアプリケーションのためのアイコンを含む、ホームスクリーンまたはトップレベルスクリーンを提供するように構成され得る。スマートテレビジョンモジュールは、電子プログラミングガイドをユーザに提供するようにさらに構成され得る。スマートテレビジョンモジュールへのコンパニオンアプリケーションは、ユーザがスマートテレビジョンモジュールなどを制御することを可能にするために、利用可能なプログラムについての追加の情報をユーザに提供するために、モバイルコンピューティングデバイス上で動作可能であり得る。いくつかの実装形態では、特徴は、ラップトップコンピュータもしくは他のパーソナルコンピュータ、スマートフォン、他のモバイルフォン、ハンドヘルドコンピュータ、タブレットPC、または他のコンピューティングデバイス上で実装され得る。いくつかの実装形態では、本明細書で開示する特徴は、(たとえば、別のコンピューティングデバイスまたはネットワーク105との)インターネット接続性を統合するように構成された処理モジュールを含み得る、ウェアラブルデバイスまたは構成要素(たとえば、スマートウォッチ)上で実装され得る。

【0098】

本明細書で説明する動作は、1つもしくは複数のコンピュータ可読記憶デバイス上に記憶されたデータにおいて、または他のソースから受信されたデータにおいて、データ処理装置によって実施される動作として実施され得る。

【0099】

「データ処理装置」、「データ処理システム」、「ユーザデバイス」または「コンピューティングデバイス」という用語は、例として、プログラマブルプロセッサ、コンピュータ、システムオンチップ、もしくは複数のシステムオンチップ、または前述の組合せを含む、データを処理するためのすべての種類の装置、デバイス、およびマシンを包含する。装置は、専用論理回路、たとえば、FPGA(フィールドプログラマブルゲートアレイ)、またはASIC(特定用途向け集積回路)を含み得る。装置はまた、ハードウェアに加えて、問題になっているコンピュータプログラムのための実行環境を作成するコード、たとえば、プロセッサファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、クロスプラットフォームランタイム環境、仮想マシン、またはそれらのうちの1つもしくは複数の組合せを構成するコードを含み得る。装置および実行環境は、ウェブサービス、分散コンピューティングおよびグリッドコンピューティングインフラストラクチャなど、様々な異なるコンピューティングモデルインフラストラクチャを実現することができる。コンテンツアイテム評価器130およびスクリプト挿入器135は、1つまたは複数のデータ処理装置、コンピューティングデバイス、またはプロセッサを含むか、または共有することができる。

【0100】

(プログラム、ソフトウェア、ソフトウェアアプリケーション、スクリプト、またはコードとしても知られる)コンピュータプログラムは、コンパイラ型言語またはインタープリタ型言語、宣言型言語または手続き型言語を含む、任意の形式のプログラミング言語において記述され得、スタンドアロンプログラムとして、またはモジュール、コンポーネント、サブルーチン、オブジェクト、もしくはコンピューティング環境において使用するのに好適な他のユニットとしてを含む、任意の形態において展開され得る。コンピュータ

プログラムは、ファイルシステムにおけるファイルに対応し得るが、そうである必要はない。プログラムは、他のプログラムもしくはデータ(たとえば、マークアップ言語文書中に記憶された1つもしくは複数のスクリプト)を保持するファイルの一部分において、問題になっているプログラム専用の単一のファイルにおいて、または、複数の協調されたファイル(たとえば、1つもしくは複数のモジュール、サブプログラム、もしくはコードの部分を記憶するファイル)において記憶され得る。コンピュータプログラムは、1つのコンピュータ上、または、1つのサイトに位置するか、もしくは複数のサイトにわたって分散され、通信ネットワークによって相互接続される、複数のコンピュータ上で実行されるように展開され得る。

【0101】

10

本明細書で説明するプロセスおよび論理フローは、入力データにおいて動作すること、および出力を生成することによって、アクションを実施するために、1つまたは複数のプログラマブルプロセッサが1つまたは複数のコンピュータプログラムを実行することによって実施され得る。プロセスおよび論理フローはまた、専用論理回路、たとえば、FPGA(フィールドプログラマブルゲートアレイ)またはASIC(特定用途向け集積回路)によって実施され得、装置もまた専用論理回路として実装され得る。

【0102】

コンピュータプログラムの実行に好適なプロセッサには、例として、汎用マイクロプロセッサと専用マイクロプロセッサの両方、および任意の種類のデジタルコンピュータの任意の1つまたは複数のプロセッサが含まれる。一般に、プロセッサは、読取り専用メモリもしくはランダムアクセスメモリ、または両方から、命令およびデータを受信することになる。コンピュータの本質的な要素は、命令に従ってアクションを実施するためのプロセッサ、ならびに、命令およびデータを記憶するための1つまたは複数のメモリデバイスである。一般に、コンピュータはまた、データを記憶するための1つまたは複数の大容量記憶デバイス、たとえば、磁気ディスク、光磁気ディスク、または光ディスクを含むか、または、それからデータを受信するため、もしくはそれにデータを転送するため、もしくははその両方のために動作可能に結合されることになる。ただし、コンピュータは、そのようなデバイスを有する必要はない。その上、コンピュータは、別のデバイス、たとえば、たとえば、モバイル電話、携帯情報端末(PDA)、モバイルオーディオもしくはビデオプレーヤ、ゲームコンソール、全地球測位システム(GPS)受信機、またはポータブル記憶デバイス(たとえば、ユニバーサルシリアルバス(USB)フラッシュドライブ)中に埋め込まれ得る。コンピュータプログラム命令およびデータを記憶するのに好適なデバイスには、例として、半導体メモリデバイス、たとえば、EPROM、EEPROM、およびフラッシュメモリデバイス、磁気ディスク、たとえば、内蔵ハードディスクまたはリムーバブルディスク、光磁気ディスク、ならびにCD-ROMおよびDVD-ROMディスクを含む、すべての形態の不揮発性メモリ、媒体、およびメモリデバイスが含まれる。プロセッサおよびメモリは、専用論理回路によって補助されるか、または専用論理回路に組み込まれ得る。

20

30

【0103】

ユーザとの対話を提供するために、本明細書で説明する主題の実装形態は、情報をユーザに表示するためのディスプレイデバイス、たとえば、CRT(陰極線管)、プラズマ、またはLCD(液晶ディスプレイ)モニタと、それによってユーザがコンピュータに入力を与えることができるキーボードおよびポインティングデバイス、たとえば、マウスまたはトラックボールとを有する、コンピュータ上で実装され得る。他の種類のデバイスが、ユーザとの対話を提供するために同様に使用され得る。たとえば、ユーザに与えられるフィードバックは、任意の形態の感覚フィードバック、たとえば、視覚フィードバック、聴覚フィードバック、または触覚フィードバックを含み得、ユーザからの入力、音響入力、音声入力、または触覚入力を含む、任意の形態で受信され得る。加えて、コンピュータは、ユーザによって使用されるデバイスに文書を送ること、およびそのデバイスから文書を受信することによって、たとえば、ウェブブラウザから受信された要求に応じて、ユーザのクライアントデバイス上のウェブブラウザにウェブページを送ることによって、ユーザと対話

40

50

することができる。

【0104】

本明細書で説明する主題の実装形態は、バックエンド構成要素を、たとえば、データサーバとして含むか、あるいは、ミドルウェア構成要素、たとえば、アプリケーションサーバを含むか、あるいは、フロントエンド構成要素、たとえば、それを通してユーザが、本明細書で説明する主題の実装形態と対話することができる、グラフィカルユーザインターフェースもしくはウェブブラウザを有するクライアントコンピュータ、または、1つもしくは複数のそのようなバックエンド構成要素、ミドルウェア構成要素、もしくはフロントエンド構成要素の任意の組合せを含む、コンピューティングシステムにおいて実装され得る。システムの構成要素は、任意の形態または任意の媒体のデジタルデータ通信、たとえば、通信ネットワークによって相互接続され得る。通信ネットワークの例には、ローカルエリアネットワーク(「LAN」)およびワイドエリアネットワーク(「WAN」)、インターネットネットワーク(たとえば、インターネット)、ならびにピアツーピアネットワーク(たとえば、アドホックピアツーピアネットワーク)が含まれる。

10

【0105】

システム600またはシステム100などのコンピューティングシステムは、クライアントおよびサーバを含み得る。たとえば、データ処理システム110は、1つまたは複数のデータセンタまたはサーバファームにおいて、1つまたは複数のサーバを含み得る。クライアントおよびサーバは、一般に互いからリモートであり、典型的には、通信ネットワークを通して対話する。クライアントおよびサーバの関係は、それぞれのコンピュータ上でランしており、互いとクライアントサーバ関係を有する、コンピュータプログラムによって生じる。いくつかの実装形態では、サーバは、(たとえば、クライアントデバイスと対話するユーザにデータを表示し、ユーザからユーザ入力を受信するために)データ(たとえば、HTMLページ)をクライアントデバイスに送信する。クライアントデバイスにおいて生成されたデータ(たとえば、ユーザ対話の結果)が、クライアントデバイスからサーバにおいて受信され得る。

20

【0106】

本明細書は、多数の特定の实装詳細を含むが、これらは、いずれかの発明の範囲の限定、または請求され得るものの範囲の限定として解釈されるべきではなく、むしろ、本明細書で説明するシステムおよび方法の特定の实装形態に固有の特徴の説明として解釈されるべきである。別個の实装形態との関連で本明細書で説明するいくつかの特徴はまた、単一の实装形態において組み合わせられて実装され得る。逆に、単一の实装形態との関連で説明する様々な特徴もまた、複数の実装形態において別個に、または任意の好適な部分組合せにおいて実装され得る。その上、特徴は、いくつかの組合せにおいて作用するとして上記で説明されることがあり、最初にそのようなものとして請求されることさえあるが、請求される組合せからの1つまたは複数の特徴は、場合によっては、その組合せから削除され得、請求される組合せは、部分組合せ、または部分組合せの変形形態を対象とし得る。

30

【0107】

同様に、動作は、特定の順序で図面において図示されるが、これは、望ましい結果を達成するために、そのような動作が、図示された特定の順序で、もしくは連続した順序で実施されること、または、すべての図示された動作が実施されることを必要とするものとして理解されるべきではない。場合によっては、特許請求の範囲に記載されているアクションは、異なる順序で実施され、なお、望ましい結果を達成することが可能である。加えて、添付の図面に図示されたプロセスは、望ましい結果を達成するために、必ずしも図示の特定の順序、または連続した順序を必要とするとは限らない。

40

【0108】

いくつかの状況では、マルチタスキングおよび並列処理が有利であり得る。その上、上記で説明した実装形態における様々なシステム構成要素の分離は、すべての実装形態においてそのような分離を必要とするものとして理解されるべきではなく、説明したプログラム構成要素およびシステムは、一般に単一のソフトウェア製品にともに統合され得るか、

50

または複数のソフトウェア製品にパッケージ化され得ることを理解されたい。コンテンツアイテム評価器130およびスクリプト挿入器135は、データ処理システム110の一部、単一のモジュール、1つもしくは複数の処理モジュールを有する論理デバイス、1つもしくは複数のサーバ、または検索エンジンの一部であり得る。

【0109】

ここでいくつかの例示的な実装形態および実装形態について説明したが、上記は例示的であり、限定ではなく、例として提示されたことは明らかである。具体的には、本明細書で提示した例の多くは、方法の行為またはシステム要素の特定の組合せを伴うが、それらの行為およびそれらの要素は、同じ目的を達成するために、他の方法で組み合わせられ得る。ある実装形態のみに関連して説明した行為、要素、および特徴は、他の実装形態または実装形態における同様の役割から除外されるものではない。

10

【0110】

本明細書で使用する言い回しおよび専門用語は、説明のためのものであり、限定として見なされるべきではない。本明細書での「含む(including)」「備える(comprising)」「有する(having)」「含む(containing)」「伴う(involving)」「によって特徴づけられる(characterized by)」「ことを特徴とする(characterized in that)」、およびそれらの変形形態の使用は、その後に列挙される項目、それらの等価物、および追加の項目、ならびに、排他的にその後に列挙される項目からなる代替実装形態を包含するように意図されている。一実装形態では、本明細書で説明するシステムおよび方法は、説明する要素、行為、または構成要素のうちの1つ、2つ以上の各組合せ、または全部からなる。

20

【0111】

本明細書で単数形において言及したシステムおよび方法の実装形態または要素または行為へのいかなる言及も、複数のこれらの要素を含む実装形態をも包含し得、本明細書のいかなる実装形態または要素または行為への複数形におけるいかなる言及も、単一の要素のみを含む実装形態をも包含し得る。単数形または複数形における言及は、本開示のシステムまたは方法、それらの構成要素、行為、または要素を単一の構成または複数の構成に限定するものではない。いずれかの情報、行為、または要素に基づいている、いずれかの行為または要素への言及は、その行為または要素がいずれかの情報、行為、または要素に少なくとも部分的に基づく実装形態を含み得る。

【0112】

30

本明細書で開示するいかなる実装形態も、任意の他の実装形態と組み合わせられてよく、「実装形態」、「いくつかの実装形態」、「代替実装形態」、「様々な実装形態」、「一実装形態」などへの言及は、必ずしも相互排他的であるとは限らず、実装形態に関連して説明する特定の特徴、構造、または特性が、少なくとも1つの実装形態中に含まれ得ることを示すものである。本明細書で使用するそのような用語は、必ずしもすべて同じ実装形態に言及しているとは限らない。いかなる実装形態も、包含的にまたは排他的に、本明細書で開示する態様および実装形態に一致するいずれかの方法で、任意の他の実装形態と組み合わせられてよい。

【0113】

「または(or)」への言及は、「または(or)」を使用して説明するいかなる用語も、説明する用語のうちの単一、2つ以上、および全部のいずれかを示し得るように、包含的として解釈され得る。

40

【0114】

図面、発明を実施するための形態、またはいずれかの請求項における技術的特徴の後に参照符号が続く場合、参照符号は、図面、発明を実施するための形態、および特許請求の範囲の了解度を増すことを唯一の目的として含まれている。したがって、参照符号もそれらの不在も、いかなる請求項要素の範囲におけるいかなる限定の効果も有するものでもない。

【0115】

本明細書で説明するシステムおよび方法は、それらの特性から逸脱することなしに、他

50

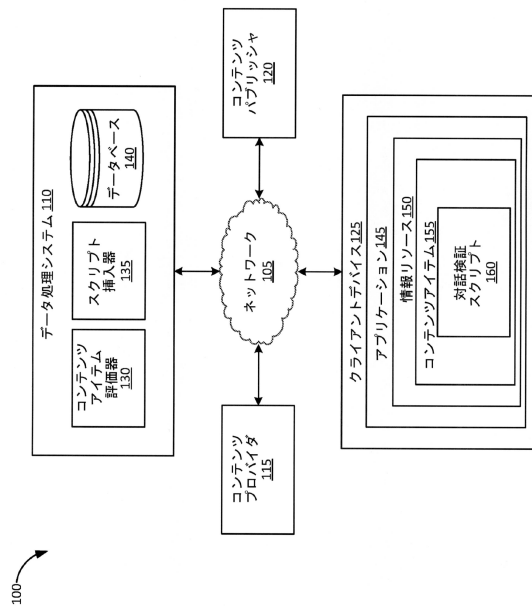
の特定の形態において実施され得る。本明細書で提供する例は、コンテンツアイテム上で実施される対話を検証すること、およびその対応策を展開することに関するが、本明細書で説明するシステムおよび方法は、他の環境に適用され得る。上記の実装形態は、説明したシステムおよび方法の限定ではなく、それらを説明するものである。したがって、本明細書で説明するシステムおよび方法の範囲は、上記の説明ではなく、添付の特許請求の範囲によって示され、特許請求の範囲の等価の意味および範囲内に入る変更は、特許請求の範囲内に包含される。

【符号の説明】

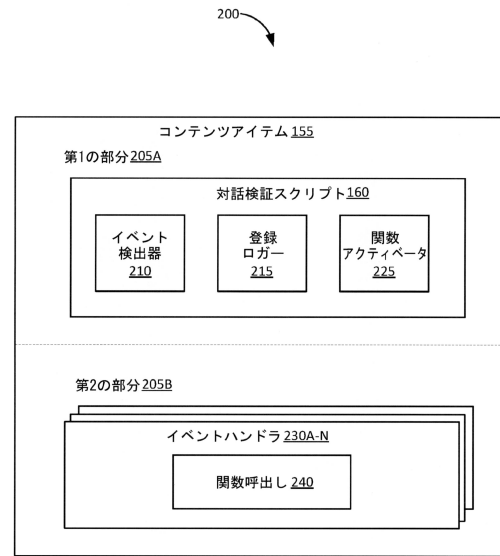
【0116】

100	コンピュータネットワーク化された環境、環境、システム	10
105	ネットワーク	
110	データ処理システム	
115	コンテンツプロバイダ	
120	コンテンツパブリッシャ	
125	クライアントデバイス	
130	コンテンツアイテム評価器	
135	スクリプト挿入器	
140	データベース、コンテンツリポジトリまたはデータベース	
145	アプリケーション	
150	情報リソース	20
155	コンテンツアイテム、サードパーティコンテンツアイテム	
160	対話検証スクリプト	
162	ヘッダ	
164	本文	
200	システム	
210A	第1の部分	
210B	第2の部分	
220	イベント検出器	
230	登録ロガー	
240	関数アクティベータ	30
250A～N	イベントハンドラ	
260	関数、関数呼出し	
302、320、340、360	タッチ	
304、322、342、362	touch-startイベント	
306、326、344、366	リリース	
308、328、346、368	touch-endイベント	
310、318、348	on-clickイベント	
312、312'、350、350'、370、370'	登録	
314、314'、352、352'、372	実行	
316	タイマーを開始すること、タイマーの開始	40
324、364	スクロール	
354	比較	
600	コンピュータシステム、システム	
605	通信インターフェース	
610	出力デバイス	
615	入力デバイス	
620	プロセッサ	
625	メモリ	

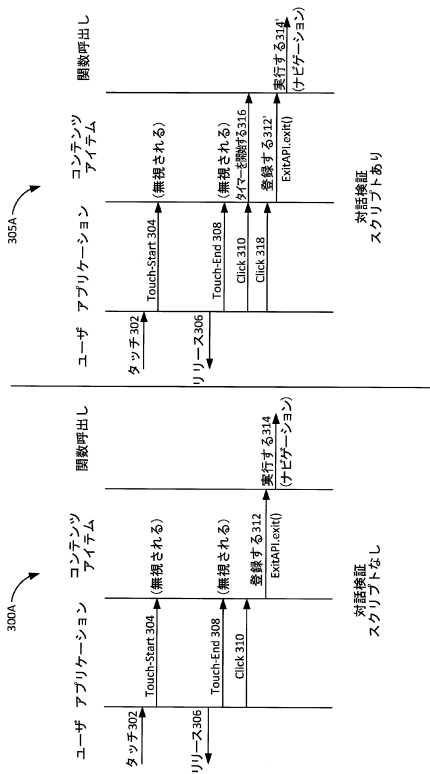
【図 1】



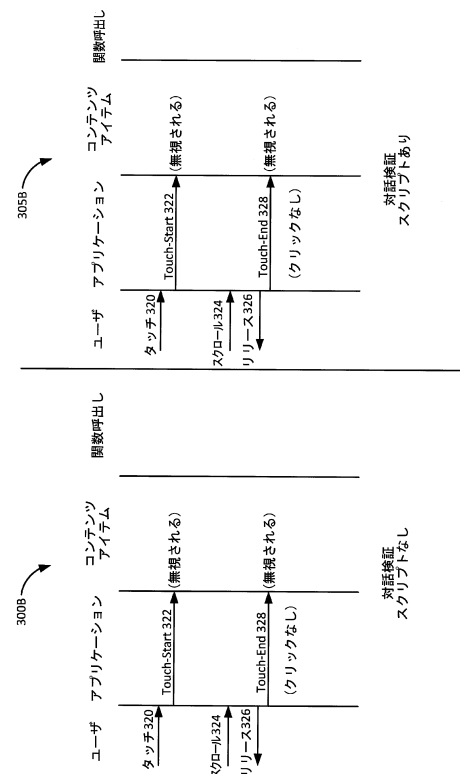
【図 2】



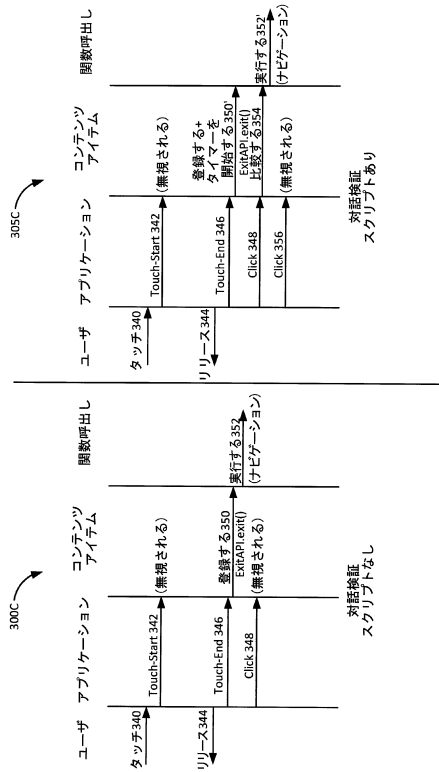
【図 3 A】



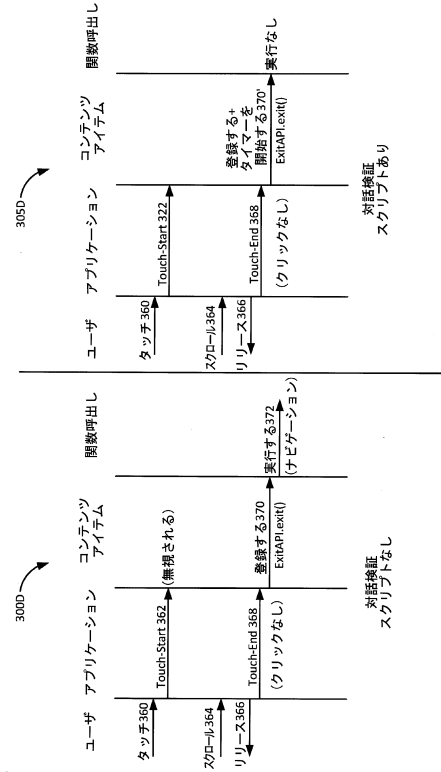
【図 3 B】



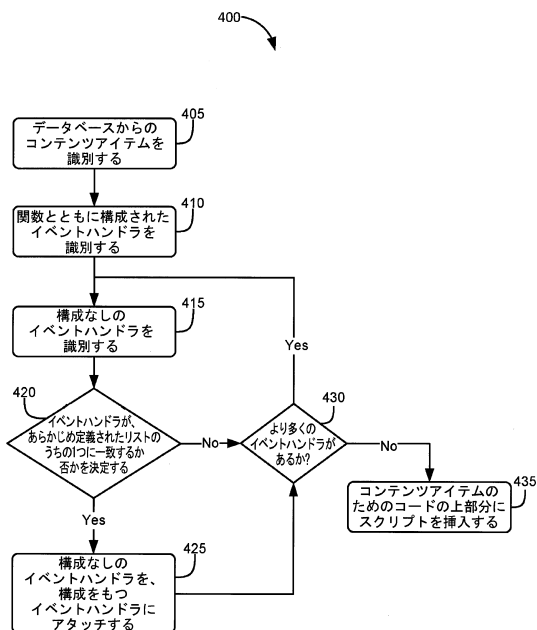
【図 3 C】



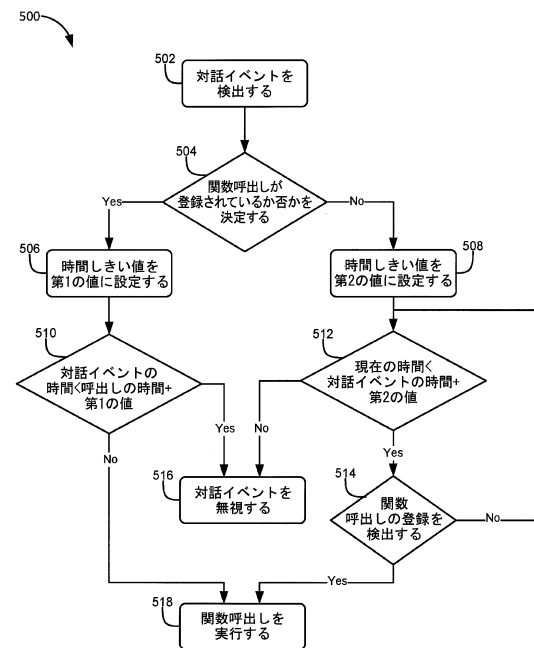
【図 3 D】



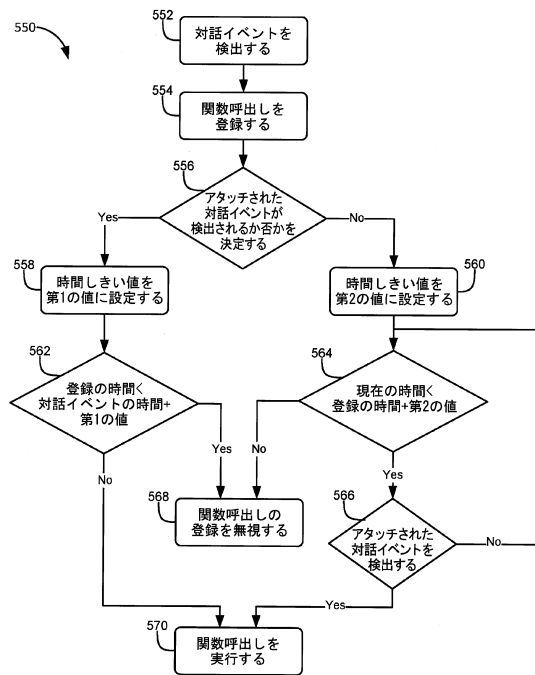
【図 4】



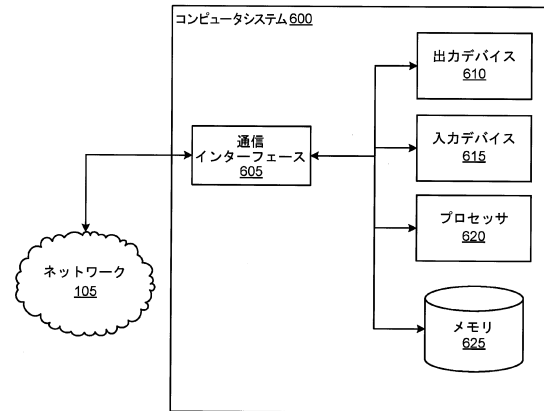
【図 5 A】



【図 5 B】



【図 6】



フロントページの続き

- (72)発明者 エウジェニオ・マルキオーリ
アメリカ合衆国・カリフォルニア・94043・マウンテン・ビュー・アンフィシアター・パーク
ウェイ・1600
- (72)発明者 カート・ローレンス
アメリカ合衆国・カリフォルニア・94043・マウンテン・ビュー・アンフィシアター・パーク
ウェイ・1600
- (72)発明者 リチャード・アーンショー
アメリカ合衆国・カリフォルニア・94043・マウンテン・ビュー・アンフィシアター・パーク
ウェイ・1600

審査官 酒井 優一

- (56)参考文献 特開2017-215983 (JP, A)
米国特許出願公開第2014/0361982 (US, A1)
特表2015-523643 (JP, A)
特開2014-215648 (JP, A)
国際公開第2016/137481 (WO, A1)
国際公開第2017/075033 (WO, A1)
特開2000-076266 (JP, A)
特開2012-014212 (JP, A)
特開2008-225802 (JP, A)
特開2016-201122 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F 3/0488