

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4976561号

(P4976561)

(45) 発行日 平成24年7月18日(2012.7.18)

(24) 登録日 平成24年4月20日(2012.4.20)

(51) Int.Cl.

F I

G 0 6 F 17/30 (2006.01)

G 0 6 F 17/30 4 1 5

G 0 6 F 17/30 1 1 0 E

請求項の数 20 (全 23 頁)

(21) 出願番号	特願2010-544303 (P2010-544303)	(73) 特許権者	500046438
(86) (22) 出願日	平成20年12月29日 (2008.12.29)		マイクロソフト コーポレーション
(65) 公表番号	特表2011-510419 (P2011-510419A)		アメリカ合衆国 ワシントン州 9805
(43) 公表日	平成23年3月31日 (2011.3.31)		2-6399 レッドモンド ワン マイ
(86) 国際出願番号	PCT/US2008/088487		クロソフト ウェイ
(87) 国際公開番号	W02009/094097	(74) 代理人	100077481
(87) 国際公開日	平成21年7月30日 (2009.7.30)		弁理士 谷 義一
審査請求日	平成23年12月2日 (2011.12.2)	(74) 代理人	100088915
(31) 優先権主張番号	12/020,427		弁理士 阿部 和夫
(32) 優先日	平成20年1月25日 (2008.1.25)	(72) 発明者	ジョン デイビッド ドーティ
(33) 優先権主張国	米国 (US)		アメリカ合衆国 98052 ワシントン
早期審査対象出願			州 レッドモンド ワン マイクロソフト
			ウェイ マイクロソフト コーポレーシ
			ョン インターナショナル パテンツ内
			最終頁に続く

(54) 【発明の名称】 ブックマークを用いたストリーミングオブジェクトのインスタンス化

(57) 【特許請求の範囲】

【請求項 1】

ストリーミングトークンのセット (203) をインスタンス化マネージャ (213) に渡すストリーミングトークンのナビゲータ (212) に関する方法であって、該方法は、コンピュータシステムに記憶されたコンピュータ実行可能命令が前記コンピュータシステムによって実行されると実施され、前記ストリーミングトークンのセットは、一群のインスタンス化命令を表し、

インスタンス化命令を表す特定のトークンを受信する (303) ナビゲータの動作と、
前記特定のトークンを前記インスタンス化マネージャに最初に渡す (動作306) ナビゲータの動作と、

前記インスタンス化マネージャからの、前記特定のトークンにおいて前記ストリーミングトークンのセットをブックマークする要求を受信する (401) ナビゲータの動作であって、前記ブックマークは、前記ナビゲータが少なくとも前記特定のトークンから開始される前記ストリーミングトークンのセットをバッファリングすることを示す動作と、

前記ストリーミングトークンのセットの1または複数の後続のトークンを受信する (303) ナビゲータの動作と、

前記特定のトークンおよび前記1または複数の後続のトークンをバッファリングする (305) ナビゲータの動作と、

前記1または複数の後続のトークンを前記インスタンス化マネージャに最初に渡す (306) ナビゲータの動作と、

10

20

前記ナビゲータが前記 1 または複数の後続のトークンを前記インスタンス化マネージャに渡した後、前記インスタンス化マネージャからの、前記特定のトークンから開始される前記ストリーミングトークンのセットを再生する要求を受信する (501) ナビゲータの動作と、

前記インスタンス化マネージャからの再生の要求に応じて、前記特定のトークンおよび前記 1 または複数の後続のトークンを、前記ストリーミングトークンのセットにおいてそれらを適切に順序付けて前記インスタンス化マネージャに対して再生する (502) ナビゲータの動作と、

を備えることを特徴とする方法。

【請求項 2】

前記特定のトークンは、前記ストリーミングトークンのセットの第 1 トークンであることを特徴とする請求項 1 に記載の方法。

【請求項 3】

前記特定のトークンは、前記ストリーミングトークンのセットの中間トークンであることを特徴とする請求項 1 に記載の方法。

【請求項 4】

前記特定のトークンを前記インスタンス化マネージャに最初に渡すナビゲータの動作の前に、

先行するトークンを前記ストリーミングトークンのセット内で受信するナビゲータの動作と、

前記先行するトークンを前記インスタンス化マネージャに最初に渡すナビゲータの動作と、

前記インスタンス化マネージャからの、前記先行するトークンにおいて前記ストリーミングトークンのセットをブックマークする要求を受信するナビゲータの動作であって、前記ブックマークは、ナビゲータが少なくとも前記先行するトークンから開始されてバッファリングすることを示す動作と、

前記先行するトークン、および、トークンの前記ストリーム内で前記先行するトークンと前記特定のトークンとの間に介在する任意のトークンをバッファリングするナビゲータの動作と

をさらに備えることを特徴とする請求項 3 に記載の方法。

【請求項 5】

前記特定のトークンおよび前記 1 または複数の後続のトークンを再生するナビゲータの動作の後に、

前記インスタンス化マネージャからの、前記先行するトークンから開始される前記ストリーミングトークンのセットを再生する要求を受信するナビゲータの動作と、

前記インスタンス化マネージャからの、前記先行するトークンから開始される前記ストリーミングトークンのセットを再生する前記要求に応じて、前記先行するトークン、前記先行するトークンと前記特定のトークンとの間に介在する任意のトークン、前記特定のトークン、および前記 1 または複数の後続のトークンを前記インスタンス化マネージャに対して再生するナビゲータの動作と、

をさらに備えることを特徴とする請求項 4 に記載の方法。

【請求項 6】

前記先行するトークンから開始される前記ストリーミングトークンのセットを再生する動作の後および／または同時に、

前記先行するトークン、前記先行するトークンと前記特定のトークンとの間に介在する任意のトークン、前記特定のトークン、および前記 1 または複数の後続のトークンが含まれたバッファを解放する動作

をさらに備えることを特徴とする請求項 5 に記載の方法。

【請求項 7】

前記特定のトークンから開始される前記ストリーミングトークンのセットを再生する動

10

20

30

40

50

作の後および／または同時に、

前記先行するトークン、前記先行するトークンと前記特定のトークンとの間に介在する任意のトークン、前記特定のトークン、および前記 1 または複数の後続のトークンが含まれたバッファを解放する動作

をさらに備えることを特徴とする請求項 1 に記載の方法。

【請求項 8】

トークンのストリーミング時にトークンのセットを受信するナビゲータ (212) から、前記トークンのセット (203) を受信するインスタンス化マネージャ (213) に関する方法であって、該方法は、コンピュータシステムに記憶されたコンピュータ実行可能命令が前記コンピュータシステムによって実行されると実施され、前記トークンのセット

10

は、一群のインスタンス化命令を表し、
トークンのストリーミング時に前記トークンのセットを受信するように構成されたナビゲータから前記トークンのセットの特定のトークンを受信する (601) インスタンス化マネージャの動作であって、前記特定のトークンは、インスタンス化命令を表す動作と、

前記インスタンス化マネージャが、前記特定のトークンによって表されたインスタンス化命令に対してまだ動作することができないと判断する (602) インスタンス化マネージャの動作と、

前記ナビゲータに前記特定のトークンにおいて前記トークンのセットをブックマークするように命令する (603) インスタンス化マネージャの動作と、

前記ナビゲータから前記トークンのセットの 1 または複数の後続のトークンを受信する (601) インスタンス化マネージャの動作と、

20

前記 1 または複数の後続のトークンの最後のトークンが、前記インスタンス化マネージャに、前記特定のトークンによって表されたインスタンス化命令に対して動作することを可能にすると判断する (602、603、605) 動作と、

前記特定のブックマークされたトークンから開始される前記トークンのセットを再生する (606) 動作と、

を備えることを特徴とする方法。

【請求項 9】

前記トークンのセット内の各トークンは、マークアップ言語文書の一部を備えることを特徴とする請求項 8 に記載の方法。

30

【請求項 10】

前記特定のトークンは、特定のクラスのオブジェクトをインスタンス化する命令を表し、前記インスタンス化マネージャが前記特定のトークンによって表された前記インスタンス化命令に対してまだ動作することができないと判断するインスタンス化マネージャの前記動作は、

前記特定のクラスのオブジェクトを割り当てることができる前に必要な 1 または複数の情報項目があると判断する動作

を備えることを特徴とする請求項 8 に記載の方法。

【請求項 11】

前記特定のクラスは標準クラスであり、前記 1 または複数の情報項目は、少なくとも前記標準クラスの固有のクラスが識別されることを可能にする型引数を備えることを特徴とする請求項 10 に記載の方法。

40

【請求項 12】

前記 1 または複数の情報項目は、少なくとも 1 または複数のコンストラクタ引数を備えることを特徴とする請求項 10 に記載の方法。

【請求項 13】

前記ストリーミングトークンの各セットはインスタンス化のコンテキストを表すが、前記ストリーミングトークンのすべてのセットがインスタンス化命令を表すわけではないことを特徴とする請求項 8 に記載の方法。

【請求項 14】

50

前記特定のブックマークされたトークンから開始される前記トークンのセットを再生する動作は、

前記ナビゲータに前記特定のトークンから開始される前記トークンのセットを再生するように命令するインスタンス化マネージャの動作と、

前記特定のトークンから開始される前記トークンのセットの再生を前記ナビゲータから受信するインスタンス化マネージャの動作と、

前記インスタンス化マネージャが前記特定のトークンから開始される前記トークンのセットの再生を受信すると、前記ナビゲータから受信したインスタンス化命令に従うインスタンス化マネージャの動作と

を備えることを特徴とする請求項 8 に記載の方法。

10

【請求項 15】

前記ナビゲータは、すべての前記トークンのセットよりも少ない数をバッファリングすることを特徴とする請求項 8 に記載の方法。

【請求項 16】

1 または複数のコンピュータ可読記憶媒体 (104) であって、コンピュータシステム (100) の 1 または複数のプロセッサ (102) によって実行されると、前記コンピュータシステムにナビゲータ (212) およびインスタンス化マネージャ (213) をインスタンス化させるコンピュータ実行可能命令を記憶し、

前記ナビゲータは、ストリーミングトークンのセット (112) を受信して、前記ストリーミングトークンのセット (203) を前記インスタンス化マネージャに渡すように構成され、さらに、前記ストリーミングトークンのセットを前記ストリーミングトークンのセットの中にブックマークするように構成され、さらに、ブックマークから開始される前記ストリーミングトークンのセットを再生するように構成され、

20

前記インスタンス化マネージャは、前記ナビゲータから受信した各トークンに対して、前記受信したトークンによって表されたインスタンス化命令を現在実行できるか否かを判断する (602) こと、

前記インスタンス化命令を現在実行できる場合、前記インスタンス化命令を実行する (603) こと、

前記インスタンス化命令を現在実行できない場合、前記ナビゲータに前記トークンをブックマークするように命令 (608) し、前記ナビゲータに前記ブックマークされたトークンによって表された前記インスタンス化命令を実行することを可能にする後続のトークンを受信する (602) と、前記ブックマークされたトークンによって表された前記インスタンス化命令を実行 (603) して、前記ストリーミングトークンのセット内で前記後続のトークンを再生する (606) こと、

30

を行うように構成されることを特徴とするコンピュータ可読記憶媒体。

【請求項 17】

前記 1 または複数のコンピュータ可読記憶媒体は、物理メモリであることを特徴とする請求項 16 に記載のコンピュータ可読記憶媒体。

【請求項 18】

前記ナビゲータは、前記インスタンス化マネージャからのブックマーク命令に応じて、バッファを制御して、前記トークンのセットのすべてのトークンよりも少ない数でバッファリングできるようにさらに構成されることを特徴とする請求項 16 に記載のコンピュータ可読記憶媒体。

40

【請求項 19】

前記インスタンス化マネージャは、一群のクラス定義にアクセスできることを特徴とする請求項 16 に記載のコンピュータ可読記憶媒体。

【請求項 20】

前記クラス定義をさらに記憶し、

前記ストリーミングトークンのセットは、前記ストリーム内で連続して受信され、

前記インスタンス化マネージャが前記インスタンス化命令を現在実行できると判断され

50

たことに応じて前記インスタンス化命令を実行した後に、前記受信したトークンの前のトークンが実行されることを可能にすると判断し、前記受信したトークンの前の前記トークンを再生することを特徴とする請求項16に記載のコンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ストリーミングオブジェクトのインスタンス化に関する。

【背景技術】

【0002】

オブジェクトインスタンスは、ストリーミングデータから生成できる。例えば、オブジェクトのインスタンス化マネージャは、さまざまなクラス定義にアクセスできる。ストリーミングデータは、インスタンス化マネージャに特定のクラスのオブジェクトをインスタンス化させる命令を渡す。ストリーミングデータは、さらに、それらのオブジェクトをインスタンス化する方法およびポピュレート (populate) する方法についての命令を渡す。バッファリングせずにストリーミングデータから直接インスタンス化する場合、その命令は、インスタンス化マネージャが命令を受信した時に、各命令に応答するために必要な情報をすべて有する形で書かれなければならない。あるいは、インスタンス化マネージャは、すべての命令が受信されるまでストリーミングデータをバッファリングできる。例えば、マークアップ言語を用いてインスタンス化命令が表される場合、すべてのマークアップ文書はバッファリングされる。

【発明の概要】

【発明が解決しようとする課題】

【0003】

本発明は、ブックマークを用いたストリーミングオブジェクトのインスタンス化を提供することにある。

【課題を解決するための手段】

【0004】

本明細書で説明する実施形態は、ストリーミングトークンのセット内のトークンをブックマークすることに関し、各トークンはインスタンス化命令を表す。ナビゲータは、ストリーミングトークンのセットを受信して、そのトークンをインスタンス化マネージャに渡す。しかしながら、インスタンス化マネージャが、ナビゲータに特定のトークンをブックマークするように求めた場合、ナビゲータは、少なくともブックマークされるトークンから開始される（および先行するブックマークがあった場合、さらに前のトークンから開始される可能性がある）ストリーミング命令のセットをバッファリングする。ナビゲータは、インスタンス化マネージャからの命令に応じて、ブックマークから開始されるストリーミングトークンのセットを再生 (replay) する。

【0005】

インスタンス化マネージャは、ストリーミングトークンのセットを受信する。インスタンス化マネージャが特定のトークンの命令を現在処理できない場合、インスタンス化マネージャは、ナビゲータにその特定のトークンをブックマークするように命令する。インスタンス化マネージャが、その後、インスタンス化マネージャがブックマークに戻ることを可能にするトークンを受信した時、インスタンス化マネージャは、ブックマークに戻ってそのブックマークから開始されるインスタンス化命令を再生する。

【0006】

これによって、インスタンス化マネージャが、すべてのインスタンス化命令をバッファリングする必要なく、一群のインスタンス化命令に適切に従うことが可能になる。さらに、インスタンス化マネージャは、トークンを受信した時にすべてのトークンを処理することができなくてもインスタンス化トークンのストリームを処理できる。

【0007】

上述した本発明の概要は、特許請求の範囲に記載の事項の重要な特徴または不可欠な特

10

20

30

40

50

徴を特定することを意図せず、また、特許請求の範囲に記載の事項の範囲を決めるのに役立つものとして用いられることも意図しない。

【0008】

本発明の上記およびその他の利点および特徴を得ることができる方法を説明するために、本発明の実施形態のより詳細な説明を添付図面で示す。これらの図面は、単に本発明の典型的な実施形態を示すものであり、本発明の範囲を限定するものとは見なされないという理解のもと、添付図面を用いることによって付加的な詳細とともに本実施形態を説明する。

【図面の簡単な説明】

【0009】

【図1】ストリーミングデータを送信するために動作できる例示的なコンピュータシステムを示す図である。

【図2】ストリーミングデータを用いてオブジェクトをメモリ内でインスタンス化するソフトウェアアーキテクチャを示す図である。

【図3】ナビゲータが、ストリーミングトークンに応じてオブジェクトのインスタンス化を行う方法のフローチャートを示す図である。

【図4】ナビゲータにトークンをブックマークさせる方法のフローチャートを示す図である。

【図5】ナビゲータに、先行するブックマークからトークンストリームを再生する再生モードに入らせることができる方法のフローチャートを示す図である。

【図6】インスタンスマネージャが、ナビゲータからトークンのセットを受信する方法のフローチャートを示す図である。

【発明を実施するための形態】

【0010】

本明細書で説明する実施形態に従って、ストリーミングトークンのセット内のトークンをブックマークすることを説明する。ここで、各トークンは、インスタンス化命令を表すか、または少なくともインスタンス化のコンテキストを渡す。ナビゲータは、ストリーミングトークンのセットを受信して、そのトークンをインスタンス化マネージャに渡す。しかしながら、ナビゲータが特定のトークンをブックマークするように求められた場合、ナビゲータは、少なくともブックマークされるトークンから開始される（および先行するブックマークがあった場合、さらに前のトークンから開始される可能性がある）ストリーミング命令のセットをバッファリングする。ナビゲータは、インスタンス化マネージャからの命令に応じて、ブックマークから開始されるストリーミングトークンのセットを再生する。

【0011】

まず、一般的なコンピュータシステムについての導入的説明をある程度行った後、図1に関連して基本的なコンピュータシステムを説明する。その後、図2から6に関連して、ストリーミングデータに基づいてオブジェクトをインスタンス化するコンテキストにおいてブックマークを用いるさまざまな実施形態を説明する。

【0012】

コンピュータシステムは、多様な形をとりながら広まっている。コンピュータシステムは、例えば、ハンドヘルドデバイス、電化製品、ラップトップコンピュータ、デスクトップコンピュータ、メインフレーム、分散コンピュータシステム、または標準のコンピュータシステムとは見なされていないデバイスであってもよい。本明細書および特許請求の範囲において、用語「コンピュータシステム」は、少なくとも1つのプロセッサと、プロセッサが実行できるコンピュータ実行可能命令を有することができるメモリとを含む任意のデバイスまたはシステム（またはそれらの組み合わせ）を含むものとして広く定義される。メモリは、任意の形をとってよく、コンピュータシステムの性質および形によって決める。コンピュータシステムは、ネットワーク環境に分散してよく、複数の構成要素を含むコンピュータシステム（multiple constituent computing system）を含んでよい。

【0013】

図1を参照すると、その最も基本的な構成において、コンピュータシステム100は、典型的には、少なくとも1つのプロセッサ102とメモリ104とを含む。メモリ104は、揮発性、不揮発性、またはその2つの或る組み合わせからなる物理システムメモリであってよい。用語「メモリ」は、本明細書において物理記憶媒体などの不揮発性の大容量記憶装置を示すのに用いることもできる。コンピュータシステムが分散された場合、その処理、記憶および／または格納能力も分散できる。本明細書で用いられるように、用語「モジュール」または「コンポーネント」は、コンピュータシステム上で実行するソフトウェアオブジェクトまたはルーチンを示すことができる。本明細書で説明する様々なコンポーネント、モジュール、エンジン、およびサービスを、コンピュータシステム上で実行されるオブジェクトまたはプロセスとして（例えば、個々のスレッドとして）実装できる。

10

【0014】

以下の説明において、1または複数のコンピュータシステムによって行われる動作を参照して実施形態を説明する。そのような動作がソフトウェアに実装される場合、その動作を行うことに関連するコンピュータシステムの1または複数のプロセッサは、コンピュータ実行可能命令が実行されたことに応じてコンピュータシステムの動作を導く。そのような動作の例は、データの操作を伴う。コンピュータ実行可能命令（および操作されるデータ）を、コンピュータシステム100のメモリ104に格納できる。

【0015】

コンピュータシステム100は、コンピュータシステム100が、例えば、ネットワーク110などの他のコンピュータシステムとの通信を行うことができるようにする通信チャネル108を含むこともできる。通信チャネル108は、通信媒体の例である。通信媒体は、典型的には、コンピュータ可読命令、データ構造、プログラムモジュール、または他のデータを、搬送波または他の移送機構などの変調データ信号によって具体化し、任意の情報配信媒体を含む。例を挙げると、通信媒体は、有線ネットワークおよび直接有線接続などの有線媒体と、音響、無線、赤外線などの無線媒体および他の無線媒体とを含むが、これらに限定されない。本明細書で用いられる用語コンピュータ可読媒体は、記憶媒体と通信媒体とを含む。

20

【0016】

図1において、コンピュータシステム100は、通信チャネル108を経由するストリーミングデータ112の受信を示す。ストリーミングデータ112は、プロセッサ102に、メモリ104内において1または複数のオブジェクトをインスタンス化させるオブジェクトのインスタンス化命令を表す。具体的には、オブジェクト114は、メモリ104の揮発性部においてインスタンス化されるように示されている。オブジェクト114は、ストリーミングデータ112に応じてインスタンス化されていてもよい。ストリーミングデータは、図1に示すネットワーク経由で受信できるが、その他のデータチャネル経由で受信することもできる。例えば、ローカルディスク、メモリスティック、光ディスクドライブなどからストリーミングデータを得ることができる。図2および後続の図を参照して、このプロセスについてさらに詳細に説明するが、これらの例には限定されない。

30

【0017】

本発明の範囲における実施形態は、格納されたコンピュータ実行可能命令を実行し、またはデータ構造を有するコンピュータ可読媒体も含む。そのようなコンピュータ可読媒体は、汎用または専用のコンピュータによってアクセスすることができる入手可能な任意の媒体とすることができる。例を挙げると、そのようなコンピュータ可読媒体は、RAM、ROM、EEPROM、CD-ROMなどの物理記憶装置および／またはメモリ媒体、もしくは他の光ディスク記憶装置、磁気ディスク記憶装置もしくは他の磁気記憶装置、またはコンピュータ実行可能命令またはデータ構造の形式で所望のプログラムコード手段を実行しまたは格納するのに用いることができ、汎用または専用のコンピュータによってアクセスすることができるその他の媒体を備えることができるが、これに限らない。ネットワークまたは別の通信接続（有線、無線、または有線もしくは無線の組み合わせのいずれ

40

50

か)を経由してコンピュータに情報を転送しまたは渡す時、そのコンピュータは、その接続をコンピュータ可読媒体として適切に見なす。従って、そのような接続はどれもコンピュータ可読媒体と適切に呼ばれる。上記の組み合わせもコンピュータ可読媒体の範囲内に含めなければならない。

【0018】

コンピュータ実行可能命令は、例えば、汎用コンピュータ、専用コンピュータ、または専用処理デバイスに、或る機能または一群の機能を行わせる命令およびデータを備える。発明の対象事項を、構造的特徴および／または方法論的動作に特有の言語によって説明するが、添付の特許請求の範囲に定義された発明の対象事項は、本明細書で説明される具体的な特徴または動作に必ずしも限定されないことを理解されたい。むしろ、本明細書で説明される具体的な特徴および動作は、特許請求の範囲を実装する例示的な形として開示される。

10

【0019】

図2は、ストリーミングデータを用いてメモリ内でオブジェクトをインスタンス化するソフトウェアアーキテクチャ200を示す。物理層において、ストリーミングデータは、電気信号または光信号などの或る物理形式によって表される単純なデジタルビット列である。ストリーミングデータ201は、バイナリ1とバイナリ0とから成るバイナリ列として示される。

【0020】

リーダ211は、ハードウェア、ソフトウェア、ファームウェア、またはそれらを組み合わせたもので実装できる。一実施形態では、リーダ211自体が、メモリ内でインスタンス化されたオブジェクトである。リーダ211は、受信したデータストリーム201をトークン202の列に変換するロジックを含む。それぞれのトークンはインスタンス化のコンテキストを表し、一部がインスタンス化命令を表す。

20

【0021】

この変換プロセスは、いくつかのステップにおいて行われる。例えば、データストリーム201によって表されるインスタンス化命令のセットがマークアップ言語文書(例えば、XML(eXtensible Markup Language)文書)の場合、そのトークンは、特定の要素の先頭、特定の要素の終端、要素の中身、要素の属性などを表すことができる。リーダ211は、従って、最初に各ビット列の区分を特定の文字に変換する。例えば、ASCII文字を表すビット列に対して、各ビット列は、8ビットになり得る。しかしながら、ユニコード文字あるいは他の文字セットの場合のように、文字を他のビット列長によっても表すことができる。リーダ211は、次に文字ストリームを解析して、階層的なマークアップ言語文書の論理部分を表す論理セクション(すなわち、トークン)を見つけることができる。リーダ211は、フロー制御およびバッファリングの制限を任意に行ってトークンストリーム202を渡すこともできる。

30

【0022】

ナビゲータ212も、コンポーネントとしてハードウェア、ソフトウェア、ファームウェア、またはそれらを組み合わせたもので実装できる。一実施形態では、ナビゲータ212自体が、メモリ内でインスタンス化されたオブジェクトとすることができる。ナビゲータ212は、リーダ211からトークン202を受信する。リーダ211は、プッシュ形式によってトークン202をナビゲータ212に渡すことができる。しかしながら、バッファリング要件を減らすまたは無くすために、以下で詳細に説明するさまざまな例において、ナビゲータ212は、各トークンの明示的なナビゲータ要求に応じて、プル形式によってリーダ211からトークンを一つずつ受信する。

40

【0023】

ナビゲータ212は、ナビゲータにトークンストリーム203をブックマークさせることを可能にするブックマーキング機構(BM)205を含む。ナビゲータ212は、トークンストリーム203を、実質的にメモリ214内でオブジェクトをインスタンス化するインスタンス化マネージャ213に渡す。インスタンス化マネージャ213に渡されたト

50

ークンストリーム 203 内にあるトークンは、ナビゲータ 212 に渡されたトークンストリーム 202 内にあるトークンと同じである。しかしながら、以下で説明するように、ブックマーキング機構 205 によって、これらのトークンを、トークンストリーム 202 と比べて異なる順序でトークンストリーム 203 内に渡すことができる。インスタンス化マネージャ 213 は、ブックマークが、トークンストリーム 202 においてナビゲータ 212 によって生成されるようにすることができる。各ブックマークは、後で戻る特定のトークンを表し、ブックマークされたトークンから開始されるトークンストリームの再生をシミュレートする。ナビゲータ 212 は、バッファ 206 内にブックマークから開始されるトークンストリームをバッファリングすることによって、そのトークンストリームを再生できる。バッファされたトークンストリームは、後で、ブックマークが解放 (release) された時に解放されて、そのトークンストリームはバッファから再生される。

10

【0024】

インスタンス化マネージャ 213 は、トークンストリーム 202 によって表され、ナビゲータ 212 から受信するインスタンス化命令に応じて、メモリ 214 内でオブジェクトを実質的にインスタンス化する。インスタンス化マネージャ 213 が戻ってトークンストリームの先行する部分を再生できるので、用語「トークンストリーム」は、ナビゲータ 212 からインスタンス化マネージャ 213 に渡されるトークン列に関して広い意味に用いられる。インスタンス化マネージャ 213 も、ハードウェア、ソフトウェア、またはそれらを組み合わせたもので実装でき、メモリ内、恐らくはオブジェクトをインスタンス化する同じメモリ 214 内でインスタンス化されるオブジェクトとすることもできる。

20

【0025】

インスタンス化マネージャ 213 は、一群のクラス定義 204 にアクセスできる。例示において、インスタンス化マネージャ 213 は、3 つの特定のクラス定義 204 A、204 B、204 C にアクセスできる。しかしながら、インスタンス化マネージャ 213 は、省略記号 204 D を用いて象徴的に表されているように、任意の数 (1 または複数) のクラス定義にアクセスし得る。インスタンス化マネージャ 213 は、メモリ 214 内で任意の数のオブジェクトをインスタンス化できる。例えば、インスタンス化マネージャは、メモリ内で 3 つのオブジェクト 205 A、205 B、205 C をインスタンス化しているように示されている。しかしながら、インスタンス化マネージャ 213 は、省略記号 205 D によって表されているように、メモリ内で任意の数のオブジェクトをインスタンス化できる。オブジェクトインスタンスは、そのオブジェクトに対応するクラス定義に従ってメモリ内で割り当てることができる。

30

【0026】

図 3 に、ナビゲータが、ストリーミングトークンに応じてオブジェクトのインスタンス化を行うための方法 300 のフローチャートを示す。方法 300 は、ナビゲータが、インスタンス化マネージャがトークンストリーム内の次のトークンに対して準備ができていると判断する度に行われる (動作 301)。トークンストリーム内の各トークンは、インスタンス化マネージャの或るインスタンス化のコンテキストを表すことを思い起こしていただきたい。

【0027】

ナビゲータからインスタンス化マネージャに渡されるトークンストリームは、生の (live) トークンストリームとすることができ、この場合、ナビゲータは、トークンをリーダから受信するとそのままトークンを渡す。一方、トークンストリームは、再生モードにすることもでき、その場合、ナビゲータは、トークンを、そのバッファからインスタンス化マネージャに渡す。特定のトークンストリーム内の第 1 トークンに対して、トークンストリームは生 (判定ブロック 302 の “Live”) になる。

40

【0028】

生のトークンストリーム (判定ブロック 302 の “Live”) の場合、ナビゲータは、リーダからストリームの次のトークンを得る (動作 303)。プル機構において、ナビゲータは、リーダからの次のトークンの明示的な要求を行うことができ、その要求への応

50

答としてトークンを受信する。トークンストリーム内に先行するブックマークセットがある場合（判定ブロック304の“はい”）、ナビゲータは、リーダからトークンを得るとすぐにバッファリングし（動作305）、さらにトークンをインスタンス化マネージャに渡す（動作306）。バッファリングに先行してトークンをインスタンス化マネージャに渡すことも可能である。従って、動作305と306とに対して相対的なタイミング要件はない。トークンストリーム内に先行するブックマークセットがない場合（判定ブロック304の“いいえ”）、ナビゲータは、トークンをバッファリングせずにインスタンス化マネージャに渡す（動作306）。

【0029】

一実施形態において、バッファ206は、個々に連結されたリスト（`singly linked list`）として実装され、バッファ内の各トークンが、個々に連結されたリスト内のノードに関連して開始される。個々に連結されたリスト内の最後のノードを除いて、各ノードは、リスト内の次のノードへのリンクも含む。トークンストリームを使用するすべてのブックマークは、連結されたリストを共有する。生のトークンストリーム（判定ブロック302の“Live”）の場合において、先行するブックマークセットがない場合（判定ブロック304の“いいえ”）、バッファ内にトークンはなく、連結されたリストはゼロ長（`length of zero`）を有する。一方、生のトークンストリーム（判定ブロック302の“Live”）の場合において、先行するブックマークセットがある（判定ブロック304の“はい”）場合、トークンは、リスト内に付加的なノードを生成することによって個々に連結されたリストに付加されて、リスト内の以前の最後のノードを、新しく生成された付加的なノードに連結する。

【0030】

一方、再生トークンストリーム（判定ブロック302の“Replay”）の場合、ナビゲータは、リーダから次のトークンを得るのではなく、バッファから次のトークンを渡す（動作307）。トークンストリーム内に先行するブックマークがある場合（判定ブロック308の“はい”）、ナビゲータは、バッファからのトークンをインスタンス化マネージャに渡す（動作306）。トークンストリーム内に先行するブックマークがない場合（判定ブロック308の“いいえ”）、ナビゲータは、トークンをインスタンス化マネージャに渡す（動作306）前、同時および／または後に、そのトークンのバッファに関連するメモリを自由に空にすることができる（しかし、必ずしも必要ではない）（動作309）。連結リストの実施形態において、ナビゲータは、バッファからのトークンを再生すると、連結リスト内の次のトークンに対する再生を生じさせたブックマークを進める（`advance`）。そのブックマークが、連結リスト内で一番初めのブックマークである場合、そのブックマークは、二度とアクセスされない過去のトークンを進める。ガーベジコレクションモジュールは、連結リスト内のそのような未使用のモジュールを周期的に検出でき、それらのトークンを連結リストから除去できる。

【0031】

図4に、ナビゲータにトークンをブックマークさせる方法400のフローチャートを示す。具体的には、特定のトークンをブックマークする要求をインスタンス化マネージャから受信した（動作401）後にブックマークが開始される。ナビゲータは、次に、トークンをブックマークして（動作402）、さらにそのトークンのバッファリングを開始する（動作403）。もちろん、このブックマーキングは、ブックマークが実施される間に受信された後続のトークンについての、図3の方法300のフローに作用する。

【0032】

図5に、ナビゲータを、先行するブックマークからのトークンストリームを再生する再生モードに入らせることができる方法500のフローチャートを示す。具体的には、ナビゲータは、ブックマークされた特定のトークンから開始されるストリーミングトークンを再生する要求をインスタンス化マネージャから受信する（動作501）。ナビゲータは、その後、インスタンス化マネージャに渡されるべき次のトークンとして、ブックマークされたトークンに戻ることによって再生モードに入る（動作502）。

【0033】

図6に、ナビゲータからトークンのセットを受信するインスタンス化マネージャに関する方法600のフローチャートを示す。インスタンス化マネージャがナビゲータからトークンを受信する（動作601）度に、方法600が開始される。最初に、インスタンス化マネージャは、受信した命令によって表された任意のインスタンス化命令を現在実行できるか否かを判断する（判定ブロック602）。

【0034】

インスタンス化マネージャが現在インスタンス化命令を実行できる場合（判定ブロック602の“はい”）、インスタンス化マネージャは、そのインスタンス化命令を実行する（動作603）。さらに、先行するブックマークされたトークンがあり、インスタンス化命令が、インスタンス化マネージャに先行するブックマークされたインスタンス化トークンを実行させることができる場合（判定ブロック605の“はい”）、インスタンス化マネージャは、その先行するブックマークされたトークンから開始されるトークンを再生する（動作606）。例えば、インスタンス化マネージャは、再生命令をナビゲータに渡すことができ、ナビゲータに図5の方法500を実行させることができる。そうでない場合（判定ブロック605の“いいえ”）、インスタンス化マネージャは、ナビゲータに、次のトークンに対して準備ができていることを任意に通知しながら次のトークンを単に待つ（動作607）。

【0035】

インスタンス化マネージャが現在インスタンス化命令を実行できない場合（判定ブロック602の“いいえ”）、インスタンス化マネージャは、現在のトークンをブックマークする（動作608）。例えば、インスタンス化マネージャは、ナビゲータがトークンをブックマークすることを要求することができ、ナビゲータに図4の方法400を実行させることができる。インスタンス化マネージャは、その後、次のトークンを待つ（動作607）。

【0036】

このように、本明細書で説明する実施形態は、インスタンス化マネージャが特定のトークンにすぐに応答できない場合に、トークンストリームが再生（*replay*）されることを可能にする。インスタンス化マネージャは、そのトークンをブックマークし、そのブックマークされたトークンによって渡されたインスタンス化のコンテキストを受信する準備ができた時に、そのトークンから開始されるストリームの再生に戻ることができる。

【0037】

特に具体的な2つの例を説明する。具体的な各例において、クラス定義はC#で行う。また、特定のトークンストリームと、関連する階層的なマークアップ言語文書とを示し、具体的な実施形態に従ってどのようにトークンが処理されるかを説明する。これらの例は、単に例示目的として示す。第1例は、1つのレベルのブックマーキングのみを説明する単純なものである。この第1例を、「単一ブックマーク（*single bookmark*）」例と呼ぶ。第2例はより複雑であり、ブックマーキングのいくつかの階層を示す。本明細書では、「複数ブックマーク（*multiple bookmark*）」例と呼ぶ。

【0038】

単一ブックマーク例では、その目的は、以下の定義に従ったクラスのオブジェクトインスタンスを生成することである。

```
public class Monster<TWeapon>
{
    public int SizeInMeters{get;set;}
    public string Name{get;set;}
}
```

【0039】

Monster オブジェクトの構造は、*TWeapon* 型引数に依存するので、*Monster* クラスは、*Monster* クラスを均等に割り当てるために型引数 *TWeapon*

10

20

30

40

50

nを要求する標準クラスである。以下は、一群のインスタンス化命令の非直列化形式（serialized form）の例である。

```
<Monster>
  <Monster.SizeInMeters>100</Monster.SizeInMeters>
  <x:TypeArguments>Fire</x:TypeArguments>
  <Monster.Name>Godzilla</Monster.Name>
</Monster>
```

【0040】

リーダによってビットストリームから抽出された対応するトークン列を以下のように表す。

```
1 StartElement: Monster
2 StartElement: Monster.SizeInMeters
3 Text: 100
4 EndElement: Monster.SizeInMeters
5 StartElement: x:TypeArguments
6 Text: Fire
7 EndElement: x:TypeArguments
8 StartElement: Monster.Name
9 Text: Godzilla
10 EndElement: Monster.Name
11 EndElement: Monster
```

【0041】

トークン1は、ナビゲータからインスタンス化マネージャに渡される。しかしながら、インスタンス化マネージャは、型引数 `Weapon` を認識していないので、まだ `Monster` オブジェクトを生成することができない。本例において `Monster` クラスは標準クラスなので、オブジェクトの構造を認識していても型引数が必要であることを思い起こしていただきたい。オブジェクトの構造をまだ認識していないので、インスタンス化マネージャは、ナビゲータにトークン1をブックマークさせ、その後すぐにナビゲータがトークン1をバッファリングする。その後、ナビゲータは、トークン2、3および4をインスタンス化マネージャに渡して、同様にナビゲータはこれらのトークンもバッファリングする。インスタンス化マネージャは、モンスターオブジェクトがまだ生成されてなくて、これらのトークンに従うにはモンスターオブジェクトの存在が必要であるので、トークン2-4にまだ対処することができない。

【0042】

ナビゲータは、次に、インスタンス化マネージャにトークン5 “`StartElement: x:TypeArguments`” を渡して、モンスターオブジェクトを生成するのに必要な型引数を受信しようとしていることをインスタンス化マネージャに合図する。

【0043】

ナビゲータは、次に、トークン6 “`Text: Fire`” をインスタンス化マネージャに渡して、モンスタークラスの型引数が “`Fire`” であることを知らせる。この時点で、あるいはインスタンス化マネージャがトークン7 “`EndElement: x:TypeArguments`” を受信した後で、インスタンス化マネージャは、型引数が “`Fire`” であることを認識する。インスタンス化マネージャは、ナビゲータに、先行するブックマークであるトークン1から再生するように命令する。

【0044】

インスタンス化マネージャが再度トークン1を受信すると、モンスターオブジェクトが生成される。しかしながら、インスタンス化マネージャは、型引数は “`Fire`” になることを認識している。従って、インスタンス化マネージャは、`Monster<Fire>` オブジェクトを生成する。この段階で、バッファを用いるブックマークは1つだけであ

り、そのブックマークがバッファ内のトークン2に進んでいるので、トークン1をバッファから解放できる。

【0045】

その後、トークン2 “StartElement: Monster. SizeInMeters” がナビゲータのバッファから再生されると、インスタンス化マネージャは、Monster<Fire>オブジェクトのSizeInMetersプロパティに書き込む準備をする。この段階で、トークン2はバッファから解放されるが、必ずしも必要ではない。例えば、ブックマークを、個々に連結されたバッファのリスト表示内のトークン3に進めることができる。その後で、ガーベジコレクションモジュールは、連結リスト内のトークン1および2を表すノードがもはや使われていないことを検出する。

10

【0046】

トークン3 “Text: 100” が再生されると、インスタンス化マネージャは、値“100”をMonster<Fire>オブジェクトのSizeInMetersプロパティに書き込む。トークン3は、次に、バッファから廃棄される。

【0047】

その後、トークン4 “EndElement: Monster. SizeInMeters” がナビゲータのバッファから再生されると、インスタンス化マネージャは、SizeInMetersプロパティへの書き込みが完了したことを認識する。トークン4は、次に、バッファから廃棄される。

【0048】

20

ナビゲータのバッファからトークン5から7までが再生されると、インスタンス化マネージャは、型引数がすでに設定されているので何も動作を起こす必要はないが、実際は、Monster<Fire>オブジェクトが生成される前に動作が要だった。トークン5から7までがバッファから廃棄される。

【0049】

トークン8 “StartElement: Monster. Name” がインスタンス化マネージャによって受信されると、ナビゲータが今はそのバッファからではなく（今空である）、ビットストリームにアクセスするリーダーからトークンを渡しているので、我々は、再生モード（replay mode）を終了させて生のモード（live mode）に入る。インスタンス化マネージャは、Monster<Fire>オブジェクトのNameプロパティに書き込む準備をする。

30

【0050】

トークン9 “Text: Godzilla” が、インスタンス化マネージャによって受信されると、値“Godzilla”は、Monster<Fire>オブジェクトのNameプロパティに書き込まれる。

【0051】

トークン10 “EndElement: Monster. Name” によって、インスタンス化マネージャは、Nameプロパティへの書き込みが完了したことを認識する。

【0052】

トークン11 “EndElement: Monster” によって、インスタンス化マネージャは、Monster<Fire>オブジェクトの生成およびポピュレーション（populating）が完了したことを認識する。

40

【0053】

このように、単一ブックマークの例では、インスタンス化マネージャは、トークンをブックマークすることが可能であったので、そのブックマークされたトークンに応じる適切なコンテキストがあった場合、インスタンス化マネージャを後でそのトークンに戻らせることができる。一部のバッファリングが行われたが、すべてのインスタンス化命令のセットはバッファリングされなかった。

【0054】

次の例は、複数ブックマークの例である。この具体例において、3つの相対的なクラス

50

定義があり、1つはCity、1つはCountry、そして新しい1つはMonsterであり、以下のとおりC#で書き込まれる。

```
public class City
{
    public City(Country country);
    public string Name{get;set;}
    public string Population{get;set;}
}
public class Country
{
    public string Name{get;set;}
}
public class Monster
{
    public Monster(City home);
    public int SizeInMeters{get;set;}
    public Color SkinColor{get;set;}
}
```

10

【0055】

ここに、我々が非直列化したいマークアップ言語がある。本例において、x:ConstructorArgumentsは、オブジェクトを生成するコンストラクタに渡されるオブジェクトの直列化(serialization)を含むことに留意されたい。

20

```
<Monster>
  <Monster.SizeInMeters>100</Monster.SizeInMeters>
  <x:ConstructorArguments>
    <City>
      <City.Name>Tokyo</City.Name>
      <City.Population>12800000</City.Population>
      <x:ConstructorArguments>
        <Country>
          <Country.Name>Japan</Country.Name>
        </Country>
      </x:ConstructorArguments>
    </City>
  </x:ConstructorArguments>
  <Monster.SkinColor>Green</Monster.SkinColor>
</Monster>
```

30

【0056】

以下は、トークン列の表示である。

```
1 StartElement: Monster
2 StartElement: Monster.SizeInMeters
3 Text: 100
4 EndElement: Monster.SizeInMeters
5 StartElement: x:ConstructorArguments
6 StartElement: City
7 StartElement: City.Name
8 Text: Tokyo
9 EndElement: City.Name
10 StartElement: City.Population
11 Text: 12800000
```

40

50

```

12 EndElement: City.Population
13 StartElement: x:ConstructorArguments
14 StartElement: Country
15 StartElement: Country.Name
16 Text: Japan
17 EndElement: Country.Name
18 EndElement: Country
19 EndElement: x:ConstructorArguments
20 EndElement: City
21 EndElement: x:ConstructorArguments
22 StartElement: Monster.SkinColor
23 Text: Green
24 EndElement: Monster.SkinColor
25 EndElement: Monster

```

【0057】

ナビゲータは最初に、トークンストリームの第1トークン1に対して準備ができているかを判断する。ナビゲータを動作させるインスタンス化マネージャの実施形態において、ナビゲータは、インスタンス化マネージャから可能標識（`ready indicator`）を受信できる。別の実施形態では、恐らくナビゲータが、インスタンス化マネージャから単に“`Not Ready`”標識を受信しないことによって、インスタンス化マネージャがトークンストリームの第1トークン1に対して準備ができているとナビゲータに思い込ませることができる。あるいは、ナビゲータは、インスタンス化マネージャとの通信を全く行わなくても、インスタンス化マネージャは常に準備ができていると単に思い込ませる場合もある。

【0058】

ナビゲータは、インスタンス化マネージャが準備できていると判断すると、リーダにトークン列から第1トークンを得てくるように命令する。リーダは、トークン1 `StartElement: Monster`をナビゲータに渡す。ナビゲータが今、トークンストリームを生でインスタンス化マネージャに渡しているので、ナビゲータは、トークン1 `StartElement: Monster`をインスタンス化マネージャに渡す。

【0059】

インスタンス化マネージャは、`City`コンストラクタ引数を要求する`Monster`クラスを参照する。しかしながら、インスタンス化マネージャは、その`City`コンストラクタ引数をまだ有していない。従って、インスタンス化マネージャは、トークン1によって表されたインスタンス化命令をまだ実行することができない。インスタンス化マネージャは、ナビゲータにトークン1をブックマークするように命令する。本例では複数のブックマークが使用されるので、これを「第1」ブックマークと呼ぶ。ナビゲータは、第1ブックマークをトークン1 `StartElement: Monster`において保持し、トークン1をバッファリングする。ナビゲータは、そのブックマークが解放されるまで後続のトークンをバッファリングし続ける。インスタンス化マネージャは、インスタンス化マネージャにブックマークに戻ることができるようにするイベント、すなわち`Monster`クラス（すなわち、`City`オブジェクト）のすべてのコンストラクタ引数が生成されたというイベントに気付くこともできる。

【0060】

ナビゲータは、次に、リーダにビットストリームから次のトークンを得てくるように命令する。リーダは、トークン2 `StartElement: Monster.SizeInMeters`をナビゲータに渡す。ナビゲータは、トークン1とともにこのトークンをバッファリングして、トークン2 `StartElement: Monster.SizeInMeters`をインスタンス化マネージャに渡す。インスタンス化マネージャは、コンストラクタ引数がまだ必要であるので、トークン1をまだ処理することができない

10

20

30

40

50

。

【0061】

ナビゲータは、リーダーにビットストリームから次のトークンを得てくるように命令する。リーダーは、トークン3Text: 100をナビゲータに渡す。ナビゲータは、トークン1-3をバッファリングして、トークン3Text: 100をインスタンス化マネージャに渡す。この時点ではMonsterオブジェクトすら生成されていないので、インスタンス化マネージャは、まだ動作を起こすことができない。このプロセスは、トークン4EndElement: Monster. SizeInMetersでも繰り返す。

【0062】

ナビゲータは、リーダーにビットストリームから次のトークンを得てくるように命令する。リーダーは、トークン5StartElement: x:ConstructorArgumentsをナビゲータに渡す。ナビゲータは、トークン1-5をバッファリングして、トークン5StartElement: x:ConstructorArgumentsをインスタンス化マネージャに渡す。この時点ではMonsterオブジェクトすら生成されていないので、インスタンス化マネージャは、まだ動作を起こすことができない。インスタンス化マネージャは、コンストラクタオブジェクトを生成する準備を任意にする。

【0063】

ナビゲータは、リーダーにビットストリームから次のトークンを得てくるように命令する。リーダーは、トークン6StartElement Cityをナビゲータに渡す。ナビゲータは、トークン1-6をバッファリングして、トークン6StartElement Cityをインスタンス化マネージャに渡す。この時点ではMonsterオブジェクトすら生成されていないので、インスタンス化マネージャは、トークン1の第1ブックマークに対してまだ動作を起こすことができない。さらにトークン6は、インスタンス化マネージャにCityオブジェクトをインスタンス化するように命令する。インスタンス化マネージャは、Cityクラスを参照するが、Countryコンストラクタが必要であるのでCityオブジェクトをまだインスタンス化することができない。従って、インスタンス化マネージャは、ナビゲータに、トークン6StartElement: Cityのさらに別のブックマーク（「第2」ブックマーク）を生成するように命令する。従って、今トークンストリーム内に、トークン1StartElement: Monsterの第1ブックマークと、トークン6StartElement Cityの第2ブックマークとの2つのブックマークがある。ナビゲータは、次に、第2ブックマークを生成する。インスタンス化マネージャは、インスタンス化マネージャに第2ブックマークに戻ることができるようにするイベント、すなわちCountryオブジェクトが生成されてポピュレートされるイベントにも気付くことができる。

【0064】

ナビゲータは、リーダーにビットストリームから次のトークンを得てくるように命令する。リーダーは、トークン7StartElement: City. Nameをナビゲータに渡す。ナビゲータは、トークン1-7をバッファリングして、トークン7StartElement: City. Nameをインスタンス化マネージャに渡す。この時点ではMonsterオブジェクトすら生成されていないので、インスタンス化マネージャは、トークン1StartElement: Monsterの第1ブックマークに対してまだ動作を起こすことができない。さらに、Countryコンストラクタ引数が必要であるのでCityオブジェクトがまだ生成されていないことから、インスタンス化マネージャは、トークン6StartElement Cityの第2ブックマークに対して動作を起こすことができない。このプロセスは、トークン8Text Tokyo、トークン9EndElement City. Name、トークン10StartElement: City. Population、トークン11Text: 12800000、およびトークン12EndElement City. Populationでも繰り返す。各トークンは、ナビゲータから生のトークンストリームとしてインスタンス化マネージャ

10

20

30

40

50

ャに渡されて、各々がナビゲータによってバッファリングされるが、MonsterオブジェクトとCityオブジェクトとがまだ生成されていないので、インスタンス化マネージャは、トークンに対して何もすることができない。

【0065】

ナビゲータがトークン13StartElement: xConstructorArgumentsをインスタンス化マネージャに渡すと、インスタンス化マネージャは、Cityクラスのコンストラクタ引数が渡されようとしていることを合図される。この段階では、ナビゲータは、トークン1-13をバッファリングする。

【0066】

ナビゲータは、リーダーにビットストリームから次のトークンを得てくるように命令する。リーダーは、トークン14StartElement: Countryをナビゲータに渡す。ナビゲータは、トークン1-14をバッファリングして、トークン14StartElement: Countryをインスタンス化マネージャに渡す。インスタンス化マネージャは、このトークンによってCountryオブジェクトをインスタンス化するように命令される。Countryクラスが参照される。Countryオブジェクトを割り当てることができる前に必要なコンストラクタ、型引数、または他の情報はない。従って、インスタンス化マネージャは、Countryオブジェクトを、Countryクラスによって定義されたようにメモリ内に割り当てる。

10

【0067】

ナビゲータは、リーダーにビットストリームから次のトークンを得てくるように命令する。リーダーは、トークン15StartElement: Country.Nameをナビゲータに渡す。ナビゲータは、トークン1-15をバッファリングして、トークン15StartElement: Country.Nameをインスタンス化マネージャに渡す。インスタンス化マネージャは、新しくインスタンス化されたCountryオブジェクトのNameプロパティに値を書き込む準備をする。

20

【0068】

ナビゲータは、リーダーにビットストリームから次のトークンを得てくるように命令する。リーダーは、トークン16Text: Japanをナビゲータに渡す。ナビゲータは、トークン1-16をバッファリングして、トークン16Text: Japanをインスタンス化マネージャに渡す。インスタンス化マネージャは、値“Japan”をCountryオブジェクトのNameプロパティに書き込む。

30

【0069】

リーダーは、トークン17EndElement: Country.Nameをナビゲータに渡す。ナビゲータは、トークン1-17をバッファリングして、トークン17をインスタンス化マネージャに渡す。インスタンス化マネージャは、CountryオブジェクトのNameプロパティへの書き込みが今完了したことを通知する。

【0070】

リーダーは、トークン18EndElement: Countryをナビゲータに渡す。ナビゲータは、トークン1-18をバッファリングして、トークン18をインスタンス化マネージャに渡す。インスタンス化マネージャは、Countryオブジェクトのインスタンス化およびポピュレーションが今完了したことに気付く。インスタンス化マネージャは、今Cityオブジェクトのコンストラクタとして用いるCountryオブジェクトを有し、今からCityオブジェクトを生成し得る。これは、インスタンス化マネージャに第2ブックマークに戻ることができるようにするイベントである。インスタンス化マネージャは、ナビゲータに、トークン6の第2ブックマークである最後のブックマークから開始されるトークンストリームを再生するように命令する。第2ブックマークは、次に、除去されて、トークン1StartElement: Monsterの第1ブックマークのみを残す。

40

【0071】

リーダーからの情報を要求せずに、ナビゲータは、そのバッファからトークン6Star

50

`tElement: City`をインスタンス化マネージャに渡す。インスタンス化マネージャは、`City`クラスのすべてのコンストラクタ引数を有していることに気付くので、`City`オブジェクトをインスタンス化する。インスタンス化マネージャは、たとえ`City`オブジェクトがインスタンス化されていても、`Monster`オブジェクトのコンストラクタ引数として用いるのには準備がまだ十分でないので、トークン1をまだ処理することができない。

【0072】

ナビゲータは、次に、そのバッファからトークン7 `StartElement: City.Name`をインスタンス化マネージャに渡す。インスタンス化マネージャは、`City`オブジェクトの`Name`フィールドに書き込む準備をする。

10

【0073】

ナビゲータは、次に、そのバッファからトークン8 `Text: Tokyo`をインスタンス化マネージャに渡す。インスタンス化マネージャは、`City`オブジェクトの`Name`フィールドの値として“`Tokyo`”を書き込む。

【0074】

ナビゲータは、次に、そのバッファからトークン9 `EndElement: City.Name`をインスタンス化マネージャに渡す。インスタンス化マネージャは、`Name`プロパティの値の書き込み動作が完了したことに気付く。

【0075】

ナビゲータは、次に、そのバッファからトークン10 `StartElement: City.Population`をインスタンス化マネージャに渡す。インスタンス化マネージャは、`City`オブジェクトの`Population`フィールドに値を書き込む準備をする。

20

【0076】

ナビゲータは、次に、そのバッファからトークン11 `Text: 12800000`をインスタンス化マネージャに渡す。インスタンス化マネージャは、`City`オブジェクトの`Population`プロパティに値“`12800000`”を書き込む。

【0077】

ナビゲータは、次に、そのバッファからトークン12 `EndElement: City.Population`をインスタンス化マネージャに渡す。インスタンス化マネージャは、`Population`プロパティの書き込み動作が完了したことに気付く。

30

【0078】

ナビゲータは、次に、そのバッファからトークン13 `StartElement: x:ConstructorArguments`をインスタンス化マネージャに渡す。インスタンス化マネージャは、コンストラクタ引数（複数可）を用いて`City`オブジェクトをポピュレートする準備をする。

【0079】

ナビゲータは、次に、そのバッファからトークン14 `StartElement: Country`をインスタンス化マネージャに渡す。インスタンス化マネージャは、すでに`Country`オブジェクトを生成させているので、その`Country`オブジェクトを用いて`City`オブジェクトをポピュレートする。

40

【0080】

ナビゲータは、そのバッファからトークン列14-18をインスタンス化マネージャに渡す。しかしながら、インスタンス化マネージャは、`Country`オブジェクトがすでに生成されていたので、これらのトークンに要求された動作をすでに行っている。従って、トークン14-18に要求される特定の動作はない。一実施形態において、特定のトークンに対して動作が行われると、そのトークンは完了したことをフラグで合図して、そのトークンストリームがインスタンス化マネージャに対して再生された時、ナビゲータがそのトークンを渡さないようにできる。この場合では、例えば、トークン14-18は、フラグで合図することが可能である。あるいは、インスタンス化マネージャは、フラグを見

50

ても何も動作を起こさない場合もある。

【0081】

ナビゲータは、次のトークン（トークン19）をインスタンス化マネージャに送信する準備をする。しかしながら、この段階では、ナビゲータは、トークン1-18までしかバッファリングしていない。従って、ナビゲータは、再生モードを終了して、再度生のモードに入る。従って、ナビゲータは、リーダーからの次のトークンを要求する。リーダーは、トークン19 EndElement: ConstructorArguments をナビゲータに渡す。ナビゲータは、（先行するブックマークがまだあるので）トークン1-19をバッファリングして、トークン19をインスタンス化マネージャに渡す。インスタンス化マネージャは、Cityオブジェクトのコンストラクタ引数が完了したことに気付く。

10

【0082】

リーダーは、トークン20 EndElement: City をナビゲータに渡す。ナビゲータは、トークン1-20をバッファリングして、トークン20をインスタンス化マネージャに渡す。インスタンス化マネージャは、Cityオブジェクトが今完了したことに気付く。Cityオブジェクトが完了したので、インスタンス化マネージャは、Cityオブジェクトをコンストラクタ引数として要求したトークン1に対処する準備が今できている。インスタンス化マネージャは、第1ブックマークに戻る対応もできていることに気付くことができる。インスタンス化マネージャは、ナビゲータに最後のブックマークから再生するように命令する。トークン6の第2ブックマークがすでに削除されているので、最後のブックマークは、トークン1 StartElement: Monster の第1ブックマークである。

20

【0083】

ナビゲータは、トークン1から再生して、第1トークンを削除する。ナビゲータは、そのバッファからトークン1 StartElement: Monster をインスタンス化マネージャに渡す。インスタンス化マネージャは、Monsterクラスを参照し、今Monsterクラスのすべてのコンストラクタ引数（すなわち、Cityオブジェクト）を含むことに付き、よって、Monsterオブジェクトを割り当てる。任意には、トークンストリーム内にこれ以上ブックマークがないので、ナビゲータは、トークンをインスタンス化マネージャに渡すと、各トークンをそのバッファから解放できる。あるいは、ナビゲータは、バッファの再生が完了した時点ですべてのバッファを解放することもでき、ナビゲータは、トークンストリーム内にブックマークが残っていない生のモードに再度入る。

30

【0084】

ナビゲータは、そのバッファからトークン2 StartElement: Monster.SizeInMeters をインスタンス化マネージャに渡す。インスタンス化マネージャは、MonsterオブジェクトのSizeInMetersプロパティに値を書き込む準備をする。任意には、次いで、トークン2はバッファから解放されるが、上述したように、バッファを後で開放することもできる。

【0085】

ナビゲータは、そのバッファからトークン3 Text: 100 をインスタンス化マネージャに渡す。インスタンス化マネージャは、MonsterオブジェクトのSizeInMetersプロパティに値“100”を書き込む。

40

【0086】

ナビゲータは、そのバッファからトークン4 EndElement: Monster.SizeInMeters をインスタンス化マネージャに渡す。インスタンス化マネージャは、MonsterオブジェクトのSizeInMetersプロパティへの書き込みが完了したことに気付く。

【0087】

ナビゲータは、そのバッファからトークン5 StartElement: x:Con

50

`structorArguments`をインスタンス化マネージャに渡す。インスタンス化マネージャは、コンストラクタ引数（複数可）を用いて`Monster`オブジェクトをポピュレートする準備をする。

【0088】

ナビゲータは、そのバッファからトークン`6StartElementCity`をインスタンス化マネージャに渡す。インスタンス化マネージャは、`City`オブジェクトを用いて`Monster`オブジェクトをポピュレートする。

【0089】

ナビゲータは、今度はそのバッファからトークン`7-20`を一つずつインスタンス化マネージャに渡す。インスタンス化マネージャは、`City`オブジェクトがすでに構成されているので、実質的にトークン`7-20`に応じる動作をまったく起こさない。ナビゲータは、そのバッファ内のすべてのトークン（すなわち、トークン`1-20`）を再生したので、これ以上トークンをバッファリングせずに、リーダから新しいトークンを得る生のモードに入る。

【0090】

ナビゲータは、リーダがビットストリーム内の次のトークンを渡すことを要求する。リーダは、トークン`21EndElement: x:ConstructorArguments`をナビゲータに渡す。ナビゲータは、トークンストリーム内に現在ブックマークがない生のモードである。従って、トークン`21`をバッファリングする必要はない。その代わりに、ナビゲータは、トークン`21`をバッファリングせずにインスタンス化マネージャに渡す。インスタンス化マネージャは、`Monster`オブジェクトのコンストラクタ引数が完了したことに気付く。

【0091】

リーダは、トークン`22StartElement: Monster.SkinColor`をナビゲータに渡し、ナビゲータは、バッファリングせずにインスタンス化マネージャに転送する。インスタンス化マネージャは、`Monster`オブジェクトの`SkinColor`プロパティに値を書き込む準備をする。

【0092】

リーダは、トークン`23Text: Green`をナビゲータに渡し、ナビゲータは、バッファリングせずにインスタンス化マネージャに転送する。インスタンス化マネージャは、`Monster`オブジェクトの`SkinColor`プロパティに値“`Green`”を書き込む。

【0093】

リーダは、トークン`24EndElement: Monster.SkinColor`をナビゲータに渡し、ナビゲータは、バッファリングせずにインスタンス化マネージャに転送する。インスタンス化マネージャは、`SkinColor`プロパティへの値書き込む動作が完了したことに気付く。

【0094】

リーダは、トークン`25EndElement: Monster`をナビゲータに渡し、ナビゲータは、バッファリングせずにインスタンス化マネージャに転送する。インスタンス化マネージャは、`Monster`オブジェクトの生成が完了したことに気付く。インスタンス化マネージャも、トークンストリームのすべてのインスタンス化命令に対応して完了する。

【0095】

一部のバッファリングが行われたが、バッファリングは、トークンストリーム内のすべてのトークンの数よりも少なかった。本明細書で説明した実施形態を用いて、トークンストリームが、インスタンス化マネージャが通常認識している特定のインスタンス化トークンを処理する方法で書かれていなくても、すべてのトークンストリームのほんの一部をバッファリングしながら一部のトークンストリームを評価することができる。従って、トークンストリームの構造的要件を特に要求することなく、トークンストリームに基づくイン

10

20

30

40

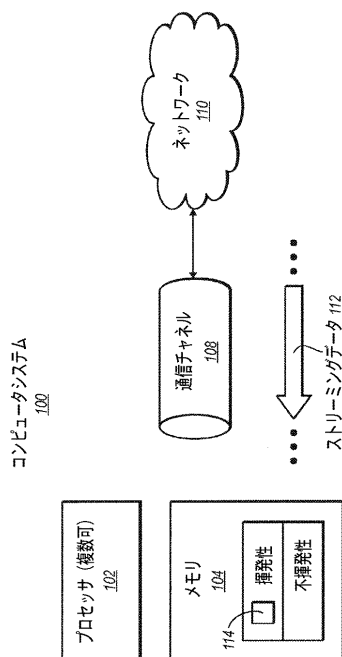
50

スタンス化要求のバッファリングが低減される。

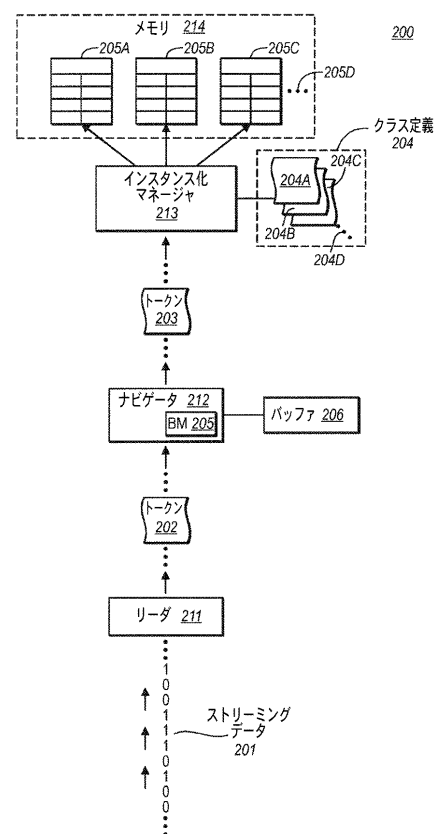
【0096】

本発明を、その趣旨または不可欠な特性から逸脱せずに他の特定の様式で実装できる。説明した実施形態は、すべての点において、限定するものではなく例として見なされるべきである。従って本発明の範囲は、前述の説明よりはむしろ特許請求の範囲の記載によって示される。特許請求の範囲の意味および同等の範囲に当てはまるすべての変更は、それらの範囲内に含まれる。

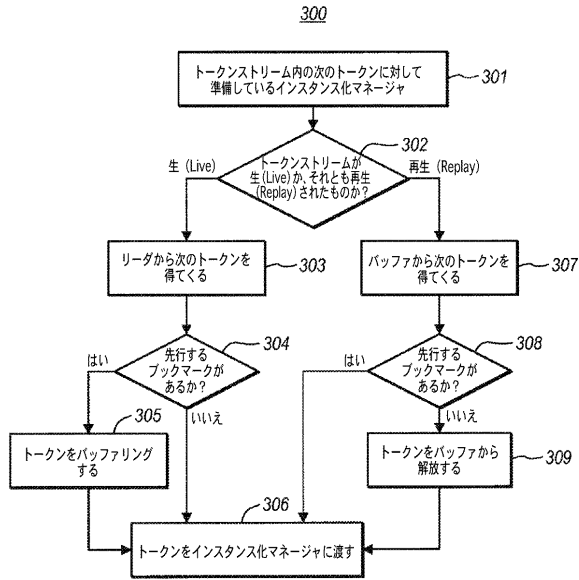
【図 1】



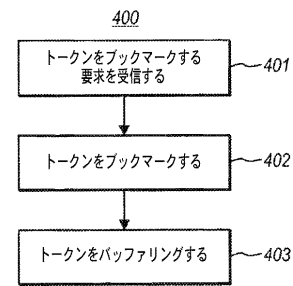
【図 2】



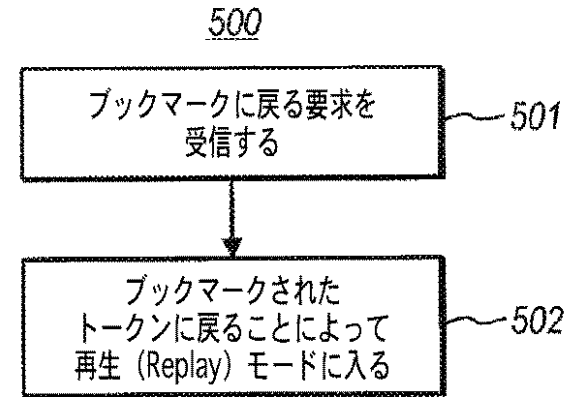
【図 3】



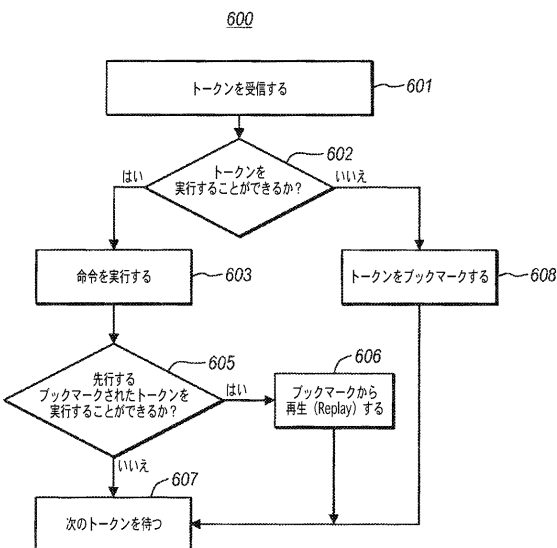
【図 4】



【図 5】



【図 6】



フロントページの続き

- (72)発明者 ジェフリー シー. シュリマー
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション インターナショナル パテント内
- (72)発明者 クリストファー リー アンダーソン
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション インターナショナル パテント内
- (72)発明者 マーティン ジェイムズ ガドジン
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション インターナショナル パテント内

審査官 鈴木 和樹

- (56)参考文献 特開2006-209748 (JP, A)
米国特許出願公開第2007/169011 (US, A1)
米国特許出願公開第2007/136435 (US, A1)

- (58)調査した分野(Int.Cl., DB名)
G06F 17/30
G06F 9/44