

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-18564

(P2012-18564A)

(43) 公開日 平成24年1月26日(2012.1.26)

(51) Int.Cl.

G06F 17/30 (2006.01)

F I

G06F 17/30 419A

G06F 17/30 170E

テーマコード (参考)

5B075

審査請求 未請求 請求項の数 8 O L (全 30 頁)

(21) 出願番号 特願2010-155719 (P2010-155719)

(22) 出願日 平成22年7月8日(2010.7.8)

(71) 出願人 000002185

ソニー株式会社

東京都港区港南1丁目7番1号

(74) 代理人 100082131

弁理士 稲本 義雄

(74) 代理人 100121131

弁理士 西川 孝

(72) 発明者 木村 貴宣

東京都港区港南1丁目7番1号 ソニー株式会社内

Fターム(参考) 5B075 KK07 ND12 ND14 ND35 UU40

(54) 【発明の名称】 通信システム、情報処理装置、情報処理方法、およびプログラム

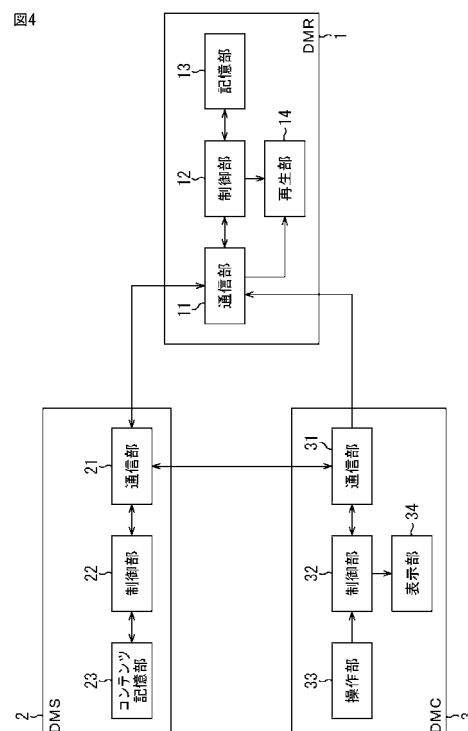
(57) 【要約】

【課題】 アイテムの探索結果を効率的に管理することができるようにする。

【解決手段】 通信システムは、DLNA機器であるDMR 1、DMS 2、およびDMC 3がネットワークを介して接続されることによって構成される。DMC 3からDMR 1に対して、DMS 2が管理するルートコンテナの再生が指示された場合、DMR 1においては、DMS 2が管理するオブジェクトの探索が開始される。探索結果は、ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかる先頭アイテムのトラック番号と、最後に見つかる最終アイテムのトラック番号と、先頭アイテムのコンテナ内における位置を表す位置情報とを、コンテナのObject IDと対応付けるようにして記憶され、DMR 1により管理される。本発明は、DLNA機器に適用することができる。

【選択図】 図4

図4



**【特許請求の範囲】****【請求項 1】**

DLNA機器である第1の他の情報処理装置が管理するルートコンテナに格納されるオブジェクトの探索を行う探索手段と、

前記ルートコンテナに格納されるオブジェクトの探索結果を表す情報として、前記ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかるアイテムである先頭アイテムのトラック番号と、最後に見つかるアイテムである最終アイテムのトラック番号と、前記先頭アイテムのコンテナ内における位置を表す位置情報とを、コンテナのObjectIDと対応付けて記憶する記憶手段と

を備える情報処理装置。

10

**【請求項 2】**

前記記憶手段は、BrowseDirectChildrenによって前記先頭アイテムが見つかったときのstartingIndexの値を、前記位置情報として記憶する

請求項1に記載の情報処理装置。

**【請求項 3】**

DLNA機器である第2の他の情報処理装置から送信されてきたPlayContainerUriを受信する受信手段をさらに備え、

前記探索手段は、前記PlayContainerUriのcontainer-id-argによってObjectIDが指定される前記ルートコンテナに格納される、first-item-id-argによってObjectIDが指定されるアイテムの親のコンテナからBrowseDirectChildrenを開始する

20

請求項2に記載の情報処理装置。

**【請求項 4】**

前記ルートコンテナを起点としてBrowseDirectChildrenを繰り返し行った場合に見つかるオブジェクトの順番は決っており、

前記探索手段は、前記first-item-id-argによってObjectIDが指定されるアイテムを格納する前記親のコンテナ以降のそれぞれのコンテナを対象としてBrowseDirectChildrenを行い、最後のオブジェクトが見つかったとき、前記ルートコンテナを起点として、前記ルートコンテナ以降のそれぞれのコンテナを対象としてBrowseDirectChildrenを行い、前記first-item-id-argによってObjectIDが指定されるアイテムが見つかったとき、探索を終了する

30

請求項3に記載の情報処理装置。

**【請求項 5】**

前記探索手段は、所定のトラック番号のアイテムの再生を行う場合、前記記憶手段に記憶されている情報に基づいて、前記所定のトラック番号のアイテムを格納するコンテナのObjectIDと、前記所定のトラック番号のアイテムのコンテナ内における位置を特定し、特定したObjectIDにより識別されるコンテナ内の、特定した位置にあるオブジェクトの探索を行い、

前記探索手段による探索結果に基づいて、前記所定のトラック番号のアイテムの送信を前記第1の他の情報処理装置に対して要求し、要求に応じて送信されてきたアイテムを再生する再生手段をさらに備える

40

請求項1に記載の情報処理装置。

**【請求項 6】**

DLNA機器である他の情報処理装置が管理するルートコンテナに格納されるオブジェクトの探索を行い、

前記ルートコンテナに格納されるオブジェクトの探索結果を表す情報として、前記ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかるアイテムである先頭アイテムのトラック番号と、最後に見つかるアイテムである最終アイテムのトラック番号と、前記先頭アイテムのコンテナ内における位置を表す位置情報とを、コンテナのObjectIDと対応付けて記憶する

ステップを含む情報処理方法。

50

**【請求項 7】**

DLNA機器である他の情報処理装置が管理するルートコンテナに格納されるオブジェクトの探索を行い、

前記ルートコンテナに格納されるオブジェクトの探索結果を表す情報として、前記ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかるアイテムである先頭アイテムのトラック番号と、最後に見つかるアイテムである最終アイテムのトラック番号と、前記先頭アイテムのコンテナ内における位置を表す位置情報とを、コンテナのObjectIDと対応付けて記憶する

ステップを含む処理をコンピュータに実行させるプログラム。

**【請求項 8】**

DLNA機器である、第1の情報処理装置と、第2の情報処理装置と、第3の情報処理装置とからなる通信システムにおいて、

前記第1の情報処理装置は、

前記第3の情報処理装置において再生可能なアイテムを記憶する記憶手段と、

ルートコンテナを設定し、前記アイテムを前記ルートコンテナに格納して管理するとともに、前記第3の情報処理装置からの探索要求に対して応答する制御手段と

を備え、

前記第2の情報処理装置は、

前記ルートコンテナのObjectIDをcontainer-id-argによって指定するとともに、前記ルートコンテナに格納されるアイテムのうちの最初に再生するアイテムのObjectIDをfirst-item-id-argによって指定するPlayContainerUriを作成する作成手段と、

前記作成手段により作成された前記PlayContainerUriを前記第3の情報処理装置に送信する送信手段と

を備え、

前記第3の情報処理装置は、

前記第2の他の情報処理装置から送信されてきた前記PlayContainerUriを受信する受信手段と、

前記PlayContainerUriのcontainer-id-argによってObjectIDが指定される前記ルートコンテナに格納される、first-item-id-argによってObjectIDが指定されるアイテムの親のコンテナからBrowseDirectChildrenを開始するように、前記第1の情報処理装置が管理する前記ルートコンテナに格納されるオブジェクトの探索を行う探索手段と、

前記ルートコンテナに格納されるオブジェクトの探索結果を表す情報として、前記ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかるアイテムである先頭アイテムのトラック番号と、最後に見つかるアイテムである最終アイテムのトラック番号と、前記先頭アイテムのコンテナ内における位置を表す位置情報とを、コンテナのObjectIDと対応付けて記憶する記憶手段と

を備える

通信システム。

**【発明の詳細な説明】****【技術分野】****【0001】**

本発明は、通信システム、情報処理装置、情報処理方法、およびプログラムに関し、特に、コンテンツの探索結果を効率的に管理することができるようにした通信システム、情報処理装置、情報処理方法、およびプログラムに関する。

**【背景技術】****【0002】**

近年、DLNA(Digital Living Network Alliance)のガイドラインに対応した機器(以下、適宜、DLNA機器という)が普及してきている。複数のDLNA機器を家庭内のネットワークを介して接続することによって、あるDLNA機器に記憶されているコンテンツを、他のDLNA機器で再生したりすることが可能になる。

10

20

30

40

50

## 【 0 0 0 3 】

図 1 は、3 つの DLNA 機器を用いて実現される通信システムの構成例を示す図である。

## 【 0 0 0 4 】

図 1 の通信システムは、DMSとして動作するDLNA機器、DMRとして動作するDLNA機器、およびDMCとして動作するDLNA機器が、LAN(Local Area Network)などのネットワークを介して接続されることによって構成される。図 1 の通信システムはいわゆる3Box System Usageの通信システムである。以下、適宜、それぞれのDLNA機器を、単に、DMS、DMR、DMCという。

## 【 0 0 0 5 】

DMS(Digital Media Server)は、音楽コンテンツやビデオコンテンツなどのコンテンツを記憶し、他のDLNA機器に配信する機器である。図 1 の例においては、HDD(Hard Disk Drive)やSSD(Solid State Drive)などの内蔵の記録媒体に音楽コンテンツを記録するオーディオ機器がDMSとされている。 10

## 【 0 0 0 6 】

以下、音楽コンテンツの再生について説明する。音楽コンテンツの再生は、例えばプレイリストを用いて行われる。適宜、コンテンツのことをアイテムともいう。

## 【 0 0 0 7 】

DMR(Digital Media Renderer)は、アイテムを再生し、音楽を出力する機器である。図 1 の例においては、ディスプレイとスピーカを有するテレビジョン受像機がDMRとされている。 20

## 【 0 0 0 8 】

DMC(Digital Media Controller)は、UI(User Interface)機能を有し、ユーザによる操作に応じて他のDLNA機器を制御する機器である。図 1 の例においては、ディスプレイや各種の操作ボタンを有する携帯機器がDMCとされている。

## 【 0 0 0 9 】

DLNAでは、プレイリストの表記方法としてPlayContainerUriが定義されている。PlayContainerUriには、プレイリストに登録されているアイテムを記憶するDMSの情報、プレイリストとして再生するコンテナと、そのコンテナ内における、最初に再生するアイテムの位置の情報、および、DMSにおいて階層的にコンテナが管理されている場合において、アイテムの探索時に階層を辿る深さの情報などが記述される。プレイリストに登録されているそれぞれのアイテムが、そのプレイリストに対応するコンテナに格納されている。 30

## 【 0 0 1 0 】

DMCから送信されたPlayContainerUriを受け取ったDMRは、その記述に基づいてアイテムの探索を行い、見つかったアイテムを再生していくことでプレイリスト再生を実現する。

## 【 0 0 1 1 】

## [ プレイリスト再生の流れ ]

ここで、図 2 のフローチャートを参照して、図 1 の通信システムにおいて行われるプレイリスト再生の流れについて説明する。

## 【 0 0 1 2 】

ステップ S 1 において、DMCは、DMSが管理するコンテナのBrowseを実行する。DMCからDMSに対してはBrowse要求 (Browse要求を表す情報) が送信される。 40

## 【 0 0 1 3 】

ステップ S 1 1 において、DMSは、DMCによるBrowse要求を受信し、ステップ S 1 2 において応答する。DMSからの応答には、プレイリストとして選択することが可能な、アイテムが格納されているコンテナの情報が含まれる。

## 【 0 0 1 4 】

ステップ S 2 において、DMCはDMSから送信されてきた情報を受信する。DMCは、受信した情報に基づいてプレイリストの一覧をディスプレイに表示し、再生するプレイリストをユーザに選択させる。

## 【 0 0 1 5 】

ステップ S 3 において、DMCは、ユーザによる操作に応じて、再生対象となるプレイリスト（コンテナ）を決定する。ここで決定されたコンテナに格納されているアイテムが順次再生される。

【 0 0 1 6 】

ステップ S 4 において、DMCは、PlayContainerUriを作成する。DMCにより作成されたPlayContainerUriには、ステップ S 3 で決定されたコンテナを指定する情報などが含まれる。

【 0 0 1 7 】

ステップ S 5 において、DMCは、PlayContainerUriをDMRに送信する。

【 0 0 1 8 】

ステップ S 2 1 において、DMRは、DMCから送信されてきたPlayContainerUriを受信し、ステップ S 2 2 において解析する。PlayContainerUriを解析することにより、アイテムの配信元となるDMSや、プレイリストに対応するRootコンテナなどが特定される。

【 0 0 1 9 】

ステップ S 2 3 において、DMRは、アイテムの配信元として特定したDMSが管理する、再生対象のプレイリストに対応するRootコンテナを対象としてBrowseを実行する。DMRからDMSに対してはBrowse要求が送信される。

【 0 0 2 0 】

ステップ S 1 3 において、DMSは、DMRによるBrowse要求を受信し、ステップ S 1 4 において応答する。DMSからの応答には、Browse対象のRootコンテナに格納されているオブジェクト（コンテナまたはアイテム）の情報が含まれる。

【 0 0 2 1 】

ステップ S 2 4 において、DMRは、DMSから送信されてきた情報を受信し、ステップ S 2 5 において、受信した情報に基づいてアイテムの探索を行う。アイテムの探索については後述する。

【 0 0 2 2 】

探索によってアイテムが見つかった場合、ステップ S 2 6 において、DMRは、アイテム取得要求（アイテム取得要求を表す情報）をDMSに送信する。ここで、探索によりオブジェクトが見つかるとは、Browseの実行結果としてDMSから送信されてきた情報の中にObjectIDなどのオブジェクトの情報が含まれており、そのオブジェクトの情報をDMRが取得したことを表す。

【 0 0 2 3 】

ステップ S 1 5 において、DMSは、DMRによるアイテム取得要求を受信し、要求されたアイテムを記録媒体から読み出す。ステップ S 1 6 において、DMSは、記録媒体から読み出したアイテムをDMRに送信する。

【 0 0 2 4 】

ステップ S 2 7 において、DMRは、DMSから送信されてきたアイテムを受信し、ステップ S 2 8 において再生する。

【 0 0 2 5 】

プレイリストに登録されているアイテム毎に、ステップ S 2 3 乃至 S 2 8 の処理がDMRにより行われ、また、DMRの処理に対応して、ステップ S 1 3 乃至 S 1 6 の処理がDMSにより行われる。図 1 の通信システムにおいては、プレイリストの再生が以上のようにして行われる。

【 0 0 2 6 】

プレイリストの再生中、DMCからDMRに対しては、適宜、再生方法の指定などが行われる。再生方法には、トラック番号順に再生するノーマル再生、プレイリストに登録されているアイテムをランダムに再生するランダム再生がある。また、トラック番号により指定されたアイテム、または、1 曲（1 トラック）前、1 曲後といったようにして指定されたアイテムを再生するトリック再生がある。

【 0 0 2 7 】

10

20

30

40

50

### [ アイテムの探索 ]

次に、DMRにより行われるアイテムの探索について説明する。

#### 【 0 0 2 8 】

図 3 は、DMSが管理するアイテムのツリー構造の例を示す図である。

#### 【 0 0 2 9 】

図 3 の例においては、あるプレイリストに対応するRootコンテナであるContainer\_Aには、第 1 階層のオブジェクトとして、左から順にItem\_8, Container\_B, Container\_C, Container\_D, Container\_E, Item\_7が格納されている。それぞれのItemは 1 曲の音楽アイテムに相当する。

#### 【 0 0 3 0 】

また、Container\_Aには、第 2 階層のオブジェクトとして、Container\_BにItem\_9が格納され、Container\_Cに、Container\_F, Item\_1, Item\_2, Container\_Gが格納されている。また、Container\_DにItem\_5が格納され、Container\_EにItem\_6が格納されている。

#### 【 0 0 3 1 】

さらに、Container\_Aには、第 3 階層のオブジェクトとして、Container\_FにItem\_10が格納され、Container\_GにItem\_3とItem\_4が格納されている。

#### 【 0 0 3 2 】

図 3 に示すツリー構造によって、DMSにおいてアイテムが管理されている場合について説明する。ここでは、PlayContainerUriにより指定される、プレイリストとして再生するコンテナ (Rootコンテナ) がContainer\_Aであり、最初に再生するアイテムがItem\_1であるものとする。DMRにおいてはItem\_1の探索が最初に行われる。

#### 【 0 0 3 3 】

はじめに、Container\_Aを対象としてDMRによりBrowseが実行される。複数のオブジェクトの情報を送信することが要求されている場合、Browse要求に対する応答として、Container\_Aに格納されている第 1 階層のオブジェクトであるItem\_8, Container\_B, Container\_C, Container\_D, Container\_E, Item\_7の情報がDMSからDMRに送信される。ObjectIDや位置情報 (URL (Uniform Resource Locator)) を含む各オブジェクトの情報が、例えば図 3 の左側に示されるオブジェクトの情報の順にソートされ、DMRに送信される。

#### 【 0 0 3 4 】

DMRにおいては、ソートされた先頭のオブジェクトから順に注目され、注目されているオブジェクトがItemである場合には、そのItemがいま探しているItemであるか否かが判断される。また、注目されているオブジェクトがコンテナである場合には、そのコンテナを対象としてBrowseが実行される。

#### 【 0 0 3 5 】

この例においては、Item\_8が注目され、注目されているItem\_8が、いま探しているItemではないとしてDMRにより判断される。

#### 【 0 0 3 6 】

次に、Item\_8の次のオブジェクト (ソート順で次のオブジェクト) であるContainer\_Bが注目され、Container\_Bを対象としてBrowseが実行される。DMSからDMRに対しては、Browse要求に対する応答として、Container\_Bに格納されているItem\_9の情報が送信される。このとき、Item\_9が注目され、Item\_9が、いま探しているItemではないとしてDMRにより判断される。

#### 【 0 0 3 7 】

次に、いま注目したItem\_9を格納するContainer\_Bの次のオブジェクトであるContainer\_Cが注目され、Container\_Cを対象としてBrowseが実行される。DMSからDMRに対しては、Browseコマンドに対する応答として、Container\_Cに格納されているContainer\_F, Item\_1, Item\_2, Container\_Gの情報が送信される。

#### 【 0 0 3 8 】

Container\_F, Item\_1, Item\_2, Container\_Gのうちの先頭のオブジェクトであるContainer\_Fが注目され、Container\_Fを対象としてBrowseが実行される。DMSからDMRに対しては

10

20

30

40

50

、Browse要求に対する応答として、Container\_Fに格納されているItem\_10の情報が送信される。Item\_10が注目され、Item\_10が、いま探しているItemではないとしてDMRにより判断される。

【0039】

次に、いま注目したItem\_10を格納するContainer\_Fの次のオブジェクトであるItem\_1が注目され、Item\_1が、いま探しているItemであるとしてDMRにより判断される。DMRからDMSに対して、Item\_1を指定してアイテム取得要求が送信され、アイテム取得要求に対する応答として送信されてきたデータが受信されたとき、DMRによりItem\_1が再生される。

【0040】

Item\_1の再生が終了したとき、Item\_1の次のオブジェクトであるItem\_2が注目される。Item\_2が、いま探しているItemであるとしてDMRにより判断された場合、DMRからDMSに対して、Item\_2を指定してアイテム取得要求が送信される。アイテム取得要求に対する応答として送信されてきたデータが受信されたとき、DMRによりItem\_2が再生される。

10

【0041】

この例においては、最初に再生するアイテムとしてPlayContainerUriにより指定されたアイテム (Item\_1) の再生が終了した後は、探索により見つかった順にアイテムを再生するものとされている。

【0042】

Item\_2の再生が終了したとき、Item\_2の次のオブジェクトであるContainer\_Gが注目され、Container\_Gを対象としてBrowseが実行される。DMSからDMRに対しては、Browse要求に対する応答として、Container\_Gに格納されているItem\_3, Item\_4の情報が送信される。

20

【0043】

DMRからDMSに対して、Item\_3を指定してアイテム取得要求が送信される。アイテム取得要求に対する応答として送信されてきたデータが受信されたとき、DMRによりItem\_3が再生される。

【0044】

Item\_3の再生が終了したとき、DMRからDMSに対して、Item\_4を指定してアイテム取得要求が送信される。アイテム取得要求に対する応答として送信されてきたデータが受信されたとき、DMRによりItem\_4が再生される。

30

【0045】

Item\_4の再生が終了したとき、次に、Item\_4を格納するContainer\_Gを格納するContainer\_Cの次のオブジェクトであるContainer\_Dが注目され、Container\_Dを対象としてBrowseが実行される。DMSからDMRに対しては、Browse要求に対する応答として、Container\_Dに格納されているItem\_5の情報が送信される。

【0046】

DMRからDMSに対して、Item\_5を指定してアイテム取得要求が送信される。アイテム取得要求に対する応答として送信されてきたデータが受信されたとき、DMRによりItem\_5が再生される。

【0047】

Item\_5の再生が終了したとき、次に、Item\_5を格納するContainer\_Dの次のオブジェクトであるContainer\_Eが注目され、Container\_Eを対象としてBrowseが実行される。DMSからDMRに対しては、Browse要求に対する応答として、Container\_Eに格納されているItem\_6の情報が送信される。

40

【0048】

DMRからDMSに対して、Item\_6を指定してアイテム取得要求が送信される。アイテム取得要求に対する応答として送信されてきたデータを受信したとき、DMRによりItem\_6が再生される。

【0049】

Item\_6の再生が終了したとき、次に、Item\_6を格納するContainer\_Eの次のオブジェク

50

トであるItem\_7が注目され、DMRからDMSに対して、Item\_7を指定してアイテム取得要求が送信される。アイテム取得要求に対する応答として送信されてきたデータを受信したとき、DMRによりItem\_7が再生される。

【0050】

再生モードとしてリピート再生などが指定されている場合、探索により既に見つかったアイテムであるItem\_8, Item\_9, Item\_10が順に再生される。Item\_8, Item\_9, Item\_10のアイテム取得要求は、既を取得され、図示せぬメモリなどに保存されていたそれらのアイテムのObjectIDや位置情報に基づいてアイテムを指定することによって行われる。

【0051】

ユーザにより選択されたプレイリストに対応するContainer\_Aの再生は以上のようにして行われる。

【0052】

なお、DMSにおいては、RootコンテナであるContainer\_Aを起点としてBrowseを実行したときに見つけることができる順に、それぞれのアイテムにTrack\_1~Track\_10のトラック番号が割り当てられている。Container\_Aに対応するプレイリストのノーマル再生が指示された場合、DMRにより実行されるBrowseにより見つかるItem\_8, Item\_9, Item\_10, Item\_1, Item\_2, Item\_3, Item\_4, Item\_5, Item\_6, Item\_7の順に再生される。

【先行技術文献】

【特許文献】

【0053】

【特許文献1】特開2008-140410号公報

【特許文献2】特表2008-515048号公報

【発明の概要】

【発明が解決しようとする課題】

【0054】

実際にアイテムの再生を行うDMRはプレイリスト内のツリー構造を知らないために以上のようにしてアイテムの探索を行う必要がある。また、アイテムの再生を行うためには、ObjectIDや位置情報によってアイテムを指定して、そのデータの送信をDMSに要求する必要がある。

【0055】

従って、例えば、DMCからトラック番号を指定してアイテムの再生が指示された場合、探索が終了していないと、DMRは、指定されたトラック番号のアイテムを特定できない。

【0056】

同様に、ランダム再生を行うことも、探索が終了していない段階では、どこを指定してデータの送信を要求して良いのか分からないことから難しい。

【0057】

ここで、探索により見つかる毎にオブジェクトの情報をDMRに記憶させていくことで、探索が完了した範囲に関して、トラック番号を指定した再生やランダム再生などのノーマル再生以外の再生を可能にすることが考えられる。

【0058】

しかしながら、オブジェクトの情報を単純に記憶させていくとすると、探索で見つかるアイテムの数が多いときには、オブジェクトの情報を記憶するためのメモリの容量が無視できない量になってしまう。例えば、1つのオブジェクトの情報量が256bytesであるとした場合、1000個のアイテムが存在するときには、それだけで約256kbytesのメモリを消費してしまうことになる。アイテムの種類によっては1000個を遙かに上回る可能性もあり、DMRのメモリの容量が乏しい場合にはこの手法では対応できない。

【0059】

本発明はこのような状況に鑑みてなされたものであり、アイテムの探索結果を効率的に管理することができるようにするものである。

【課題を解決するための手段】



## 【 0 0 6 0 】

本発明の一側面の情報処理装置は、DLNA機器である第1の他の情報処理装置が管理するルートコンテナに格納されるオブジェクトの探索を行う探索手段と、前記ルートコンテナに格納されるオブジェクトの探索結果を表す情報として、前記ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかるアイテムである先頭アイテムのトラック番号と、最後に見つかるアイテムである最終アイテムのトラック番号と、前記先頭アイテムのコンテナ内における位置を表す位置情報とを、コンテナのObjectIDと対応付けて記憶する記憶手段とを備える。

## 【 0 0 6 1 】

前記記憶手段には、BrowseDirectChildrenによって前記先頭アイテムが見つかったときのstartingIndexの値を、前記位置情報として記憶させることができる。

10

## 【 0 0 6 2 】

DLNA機器である第2の他の情報処理装置から送信されてきたPlayContainerUriを受信する受信手段をさらに設けることができる。この場合、前記探索手段には、前記PlayContainerUriのcontainer-id-argによってObjectIDが指定される前記ルートコンテナに格納される、first-item-id-argによってObjectIDが指定されるアイテムの親のコンテナからBrowseDirectChildrenを開始させる。

## 【 0 0 6 3 】

前記ルートコンテナを起点としてBrowseDirectChildrenを繰り返し行った場合に見つかるオブジェクトの順番は決まっており、前記探索手段には、前記first-item-id-argによってObjectIDが指定されるアイテムを格納する前記親のコンテナ以降のそれぞれのコンテナを対象としてBrowseDirectChildrenを行わせ、最後のオブジェクトが見つかったとき、前記ルートコンテナを起点として、前記ルートコンテナ以降のそれぞれのコンテナを対象としてBrowseDirectChildrenを行わせ、前記first-item-id-argによってObjectIDが指定されるアイテムが見つかったとき、探索を終了させることができる。

20

## 【 0 0 6 4 】

前記探索手段には、所定のトラック番号のアイテムの再生を行う場合、前記記憶手段に記憶されている情報に基づいて、前記所定のトラック番号のアイテムを格納するコンテナのObjectIDと、前記所定のトラック番号のアイテムのコンテナ内における位置を特定し、特定したObjectIDにより識別されるコンテナ内の、特定した位置にあるオブジェクトの探索を行わせることができる。前記探索手段による探索結果に基づいて、前記所定のトラック番号のアイテムの送信を前記第1の他の情報処理装置に対して要求し、要求に応じて送信されてきたアイテムを再生する再生手段をさらに設けることができる。

30

## 【 0 0 6 5 】

本発明の一側面の情報処理方法は、DLNA機器である他の情報処理装置が管理するルートコンテナに格納されるオブジェクトの探索を行い、前記ルートコンテナに格納されるオブジェクトの探索結果を表す情報として、前記ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかるアイテムである先頭アイテムのトラック番号と、最後に見つかるアイテムである最終アイテムのトラック番号と、前記先頭アイテムのコンテナ内における位置を表す位置情報とを、コンテナのObjectIDと対応付けて記憶するステップを含む。

40

## 【 0 0 6 6 】

本発明の一側面のプログラムは、DLNA機器である他の情報処理装置が管理するルートコンテナに格納されるオブジェクトの探索を行い、前記ルートコンテナに格納されるオブジェクトの探索結果を表す情報として、前記ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかるアイテムである先頭アイテムのトラック番号と、最後に見つかるアイテムである最終アイテムのトラック番号と、前記先頭アイテムのコンテナ内における位置を表す位置情報とを、コンテナのObjectIDと対応付けて記憶するステップを含む処理をコンピュータに実行させる。

## 【 0 0 6 7 】

50

本発明の他の側面の通信システムは、DLNA機器である、第1の情報処理装置と、第2の情報処理装置と、第3の情報処理装置とからなる通信システムである。前記第1の情報処理装置は、前記第3の情報処理装置において再生可能なアイテムを記憶する記憶手段と、ルートコンテナを設定し、前記アイテムを前記ルートコンテナに格納して管理するとともに、前記第3の情報処理装置からの探索要求に対して応答する制御手段とを備え、前記第2の情報処理装置は、前記ルートコンテナのObjectIDをcontainer-id-argによって指定するとともに、前記ルートコンテナに格納されるアイテムのうちの最初に再生するアイテムのObjectIDをfirst-item-id-argによって指定するPlayContainerUriを作成する作成手段と、前記作成手段により作成された前記PlayContainerUriを前記第3の情報処理装置に送信する送信手段とを備え、前記第3の情報処理装置は、前記第2の他の情報処理装置から送信されてきた前記PlayContainerUriを受信する受信手段と、前記PlayContainerUriのcontainer-id-argによってObjectIDが指定される前記ルートコンテナに格納される、first-item-id-argによってObjectIDが指定されるアイテムの親のコンテナからBrowseDirectChildrenを開始するように、前記第1の情報処理装置が管理する前記ルートコンテナに格納されるオブジェクトの探索を行う探索手段と、前記ルートコンテナに格納されるオブジェクトの探索結果を表す情報として、前記ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかるアイテムである先頭アイテムのトラック番号と、最後に見つかるアイテムである最終アイテムのトラック番号と、前記先頭アイテムのコンテナ内における位置を表す位置情報とを、コンテナのObjectIDと対応付けて記憶する記憶手段とを備える。

10

20

**【0068】**

本発明の一側面においては、DLNA機器である他の情報処理装置が管理するルートコンテナに格納されるオブジェクトの探索が行われ、前記ルートコンテナに格納されるオブジェクトの探索結果を表す情報として、前記ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかるアイテムである先頭アイテムのトラック番号と、最後に見つかるアイテムである最終アイテムのトラック番号と、前記先頭アイテムのコンテナ内における位置を表す位置情報とが、コンテナのObjectIDと対応付けて記憶される。

**【0069】**

本発明の他の側面においては、第1の情報処理装置により、第3の情報処理装置において再生可能なアイテムが記憶され、ルートコンテナが設定され、前記アイテムが前記ルートコンテナに格納して管理され、第3の情報処理装置からの探索要求に対して応答することが行われる。また、第2の情報処理装置により、前記ルートコンテナのObjectIDをcontainer-id-argによって指定するとともに、前記ルートコンテナに格納されるアイテムのうちの最初に再生するアイテムのObjectIDをfirst-item-id-argによって指定するPlayContainerUriが作成され、作成された前記PlayContainerUriが前記第3の情報処理装置に送信される。前記第3の情報処理装置により、前記第2の他の情報処理装置から送信されてきた前記PlayContainerUriが受信され、前記PlayContainerUriのcontainer-id-argによってObjectIDが指定される前記ルートコンテナに格納される、first-item-id-argによってObjectIDが指定されるアイテムの親のコンテナからBrowseDirectChildrenを開始するように、前記第1の情報処理装置が管理する前記ルートコンテナに格納されるオブジェクトの探索が行われ、前記ルートコンテナに格納されるオブジェクトの探索結果を表す情報として、前記ルートコンテナに格納されるコンテナ毎に、そのコンテナを対象とする探索により最初に見つかるアイテムである先頭アイテムのトラック番号と、最後に見つかるアイテムである最終アイテムのトラック番号と、前記先頭アイテムのコンテナ内における位置を表す位置情報とが、コンテナのObjectIDと対応付けて記憶される。

30

40

**【発明の効果】****【0070】**

本発明によれば、アイテムの探索結果を効率的に管理することができる。

**【図面の簡単な説明】**

50

## 【 0 0 7 1 】

【図 1】通信システムの構成例を示す図である。

【図 2】プレイリスト再生の流れについて説明するフローチャートである。

【図 3】アイテムのツリー構造の例を示す図である。

【図 4】本発明の一実施形態に係るDLNA機器の構成例を示すブロック図である。

【図 5】PlayContainerUriを示す図である。

【図 6】各DLNA機器の処理について説明するフローチャートである。

【図 7】DMRの探索処理について説明するフローチャートである。

【図 8】図 7 のステップ S 1 0 1 において行われる1st Item Track起点探索処理について説明するフローチャートである。

10

【図 9】コンテナ探索 0 について説明するフローチャートである。

【図 1 0】図 9 のステップ S 1 2 3 において行われるコンテナ情報作成処理について説明するフローチャートである。

【図 1 1】コンテナ探索 1 について説明するフローチャートである。

【図 1 2】コンテナ探索 2 について説明するフローチャートである。

【図 1 3】コンテナ探索 3 について説明するフローチャートである。

【図 1 4】コンテナ情報の例を示す図である。

【図 1 5】コンテナ情報の他の例を示す図である。

【図 1 6】コンテナ情報のさらに他の例を示す図である。

【図 1 7】図 7 のステップ S 1 0 2 において行われるRootコンテナ起点探索処理について説明するフローチャートである。

20

【図 1 8】コンテナ情報の例を示す図である。

【図 1 9】統合したコンテナ情報の例を示す図である。

【図 2 0】DMRの再生処理について説明するフローチャートである。

【図 2 1】コンピュータのハードウェアの構成例を示すブロック図である。

【発明を実施するための形態】

## 【 0 0 7 2 】

[ 通信システムの構成例 ]

図 4 は、本発明の一実施形態に係るDLNA機器の構成例を示すブロック図である。

## 【 0 0 7 3 】

30

図 4 の通信システムは、図 1 を参照して説明したように、DMSとして動作するDLNA機器、DMRとして動作するDLNA機器、およびDMCとして動作するDLNA機器がネットワークを介して接続されることによって構成される。

## 【 0 0 7 4 】

図 4 のDMR 1 は図 1 のテレビジョン受像機などよりなるDMRに対応し、DMS 2 は図 1 のオーディオ機器などよりなるDMSに対応する。また、図 4 のDMC 3 は図 1 の携帯機器などよりなるDMCに対応する。重複する説明については適宜省略する。

## 【 0 0 7 5 】

図 4 に示すように、DMR 1 は、通信部 1 1、制御部 1 2、記憶部 1 3、および再生部 1 4 から構成され、DMS 2 は、通信部 2 1、制御部 2 2、およびコンテンツ記憶部 2 3 から構成される。DMC 3 は、通信部 3 1、制御部 3 2、操作部 3 3、および表示部 3 4 から構成される。

40

## 【 0 0 7 6 】

DMR 1 の通信部 1 1 は、ネットワークインタフェースなどよりなり、DMS 2 の通信部 2 1 およびDMC 3 の通信部 3 1 とネットワークを介して通信を行う。通信部 1 1 により受信された情報は制御部 1 2 に供給される。例えば、通信部 1 1 は、DMC 3 の通信部 3 1 から送信されてきたPlayContainerUriを受信する。

## 【 0 0 7 7 】

また、通信部 1 1 は、制御部 1 2 による制御に従ってBrowse要求をDMS 2 の通信部 2 1 に送信し、Browse要求に対する応答として送信されてきた情報を受信する。通信部 1 1 は

50

、アイテム取得要求を送信することに応じてDMS 2 の通信部 2 1 から送信されてきたアイテムを受信し、受信したアイテムを再生部 1 4 に出力することなども行う。

【 0 0 7 8 】

制御部 1 2 は、CPU(Central Processing Unit)などよりなり、所定のプログラムを実行することによってDMR 1 の全体の動作を制御する。例えば、制御部 1 2 は、アイテムの探索時、Browse要求に対する応答としてDMS 2 から送信されてきた情報に基づいて、DMS 2 により管理されているコンテナに関する情報であるコンテナ情報を生成し、記憶部 1 3 に記憶させる。

【 0 0 7 9 】

後述するように、コンテナ情報には、アイテムのタイトルやアーティスト名、アイテムの位置情報などは含まれず、コンテナのObjectID、コンテナに格納されているアイテムのトラック番号などが含まれる。

【 0 0 8 0 】

制御部 1 2 は、アイテムのランダム再生がDMC 3 により指示された場合、または、DMC 3 により指定されたトラック番号のアイテムを再生する場合、記憶部 1 3 に記憶させているコンテナ情報に基づいて、再生対象のアイテムが格納されているコンテナを特定する。また、制御部 1 2 は、通信部 1 1 を制御し、特定したコンテナを探索することによってそのコンテナに格納されているアイテムの情報を取得し、アイテム取得要求をDMS 2 に送信する。

【 0 0 8 1 】

記憶部 1 3 は、フラッシュメモリなどよりなり、制御部 1 2 により生成されたコンテナ情報を記憶する。

【 0 0 8 2 】

再生部 1 4 は、通信部 1 1 から供給されたアイテムを再生し、再生音をスピーカから出力する。

【 0 0 8 3 】

DMS 2 の通信部 2 1 は、DMR 1 の通信部 1 1 およびDMC 3 の通信部 3 1 とネットワークを介して通信を行う。例えば、通信部 2 1 は、DMR 1 の通信部 1 1 からBrowse要求が送信されてきた場合、Browse要求を受信して制御部 2 2 に出力する。通信部 2 1 は、制御部 2 2 から供給されたBrowseの実行結果を表す情報をDMR 1 の通信部 1 1 に出力する。

【 0 0 8 4 】

また、通信部 2 1 は、DMR 1 の通信部 1 1 から送信されてきたアイテム取得要求を受信して制御部 2 2 に出力する。通信部 2 1 は、コンテンツ記憶部 2 3 から読み出されたアイテムをDMR 1 の通信部 1 1 に送信する。

【 0 0 8 5 】

制御部 2 2 は、所定のプログラムを実行することによってDMR 1 の全体の動作を制御する。例えば、制御部 2 2 は、プレイリスト毎にRootコンテナを設定し、プレイリストに登録されているアイテムを、図 3 に示すようなツリー構造によって管理する。制御部 2 2 は、アイテムの探索時にDMR 1 から送信されてきたBrowse要求に応じて、Browse要求に含まれるパラメータによって指定されたコンテナに格納されているアイテムに関する情報を、通信部 2 1 を制御してDMR 1 に送信する。

【 0 0 8 6 】

コンテンツ記憶部 2 3 はアイテムを記憶する。コンテンツ記憶部 2 3 に記憶されているアイテムは制御部 2 2 により管理され、適宜、読み出される。

【 0 0 8 7 】

DMC 3 の通信部 3 1 は、DMR 1 の通信部 1 1 およびDMS 2 の通信部 2 1 とネットワークを介して通信を行う。例えば、通信部 3 1 は、制御部 3 2 から供給されたPlayContainerUriをDMR 1 の通信部 1 1 に送信する。

【 0 0 8 8 】

また、通信部 3 1 は、DMS 2 の通信部 2 1 に対してBrowse要求を送信し、Browse要求に

10

20

30

40

50

対する応答としてDMS 2の通信部 2 1から送信されてきた情報を受信する。通信部 3 1は、受信した情報を制御部 3 2に出力する。

【 0 0 8 9 】

制御部 3 2は、所定のプログラムを実行することによってDMC 3の全体の動作を制御する。例えば、制御部 3 2は、Browse要求に対する応答としてDMS 2から送信されてきた情報に基づいてプレイリストの一覧を表示部 3 4に表示し、再生するプレイリストをユーザに選択させる。プレイリストが選択されたとき、制御部 3 2は、PlayContainerUriを生成し、通信部 3 1を制御してDMR 1に送信させる。

【 0 0 9 0 】

操作部 3 3は、DMC 3の筐体に設けられる十字キーや決定キーに対するユーザの操作を検出し、ユーザの操作の内容を表す情報を制御部 3 2に出力する。

10

【 0 0 9 1 】

表示部 3 4は、LCD(Liquid Crystal Display)などのディスプレイよりなり、制御部 3 2による制御に従って、プレイリストの一覧を表示する。

【 0 0 9 2 】

図 5 は、PlayContainerUriを示す図である。

【 0 0 9 3 】

図 5 の “ cds-udn ” は、DMSを特定するUDNを表す。“ service-id-arg ” は、DMSが持つCD SのService IDを表す。CDS(Content Directory Service)は、DMSが管理するアイテムのリストを階層化して配信する機能である。“ container-id-arg ” は、プレイリストのRootコンテナのObjectIDを表す。DMSが管理するコンテナと、コンテナに格納されているアイテムには、それぞれObjectIDが割り当てられている。

20

【 0 0 9 4 】

“ first-item-id-arg ” は、1st Item Trackとして指定するアイテムのObjectIDを表し、“ first-item-index-arg ” は、1st Item Trackのアイテムのコンテナ内の位置を表す。“ sort-arg ” はソート条件を表し、“ max-depth-arg ” は、アイテムの探索時にRootコンテナから辿る階層の深さの上限を表す。

【 0 0 9 5 】

主な記述について説明すると、図 5 のPlayContainerUriは、“ container-id-arg ” で指定されるRootコンテナ以下に存在するアイテムのうち、“ first-item-id-arg ” で指定されたアイテムを起点とし、深さ優先で探索して見つかったアイテムを順に再生することを表す。

30

【 0 0 9 6 】

[ 各機器の動作 ]

図 6 のフローチャートを参照して、DMR 1、DMS 2、DMC 3のそれぞれの処理について説明する。

【 0 0 9 7 】

図 6 には、PlayContainerUriをDMR 1が受信し、解析するまでの各機器の処理を示している。図 6 のステップ S 5 1 乃至 S 5 5 のDMC 3の処理は、図 2 のステップ S 1 乃至 S 5 の処理に対応し、図 6 のステップ S 6 1、S 6 2のDMS 2の処理は、図 2 のステップ S 1、S 1 2の処理に対応する。図 6 のステップ S 7 1、S 7 2のDMR 1の処理は、図 2 のステップ S 2 1、S 2 2の処理に対応する。

40

【 0 0 9 8 】

ステップ S 5 1において、DMC 3の制御部 3 2はBrowseを実行し、通信部 3 1を制御してBrowse要求をDMS 2に送信する。

【 0 0 9 9 】

ステップ S 6 1において、DMS 2の通信部 2 1は、DMC 3によるBrowse要求を受信する。

【 0 1 0 0 】

ステップ S 6 2において、DMS 2の制御部 2 2は、通信部 2 1を制御し、Browse要求に対する応答として、プレイリストとして選択することが可能な、アイテムが格納されてい

50

るコンテナの情報をDMC 3 に送信する。

【 0 1 0 1 】

ステップ S 5 2 において、DMC 3 の通信部 3 1 は、DMS 2 から送信されてきた情報を受信する。制御部 3 2 は、通信部 3 1 において受信された情報に基づいてプレイリストの一覧を表示部 3 4 に表示し、再生するプレイリストをユーザに選択させる。

【 0 1 0 2 】

ステップ S 5 3 において、DMC 3 の制御部 3 2 は、操作部 3 3 から供給された情報に基づいて再生対象のプレイリスト (Root コンテナ) を決定する。

【 0 1 0 3 】

ステップ S 5 4 において、DMC 3 の制御部 3 2 は、ユーザにより選択されたプレイリストに対応するRoot コンテナを指定する情報などを含む、図 5 に示すPlayContainerUriを作成する。

【 0 1 0 4 】

ステップ S 5 5 において、DMC 3 の制御部 3 2 は、通信部 3 1 を制御し、PlayContainerUriをDMR 1 に送信する。

【 0 1 0 5 】

ステップ S 7 1 において、DMR 1 の通信部 1 1 は、DMC 3 の通信部 3 1 から送信されてきたPlayContainerUriを受信する。制御部 1 2 は、通信部 1 1 により受信されたPlayContainerUriを記憶部 1 3 などに記憶させる。

【 0 1 0 6 】

ステップ S 7 2 において、DMR 1 の制御部 1 2 は、PlayContainerUriを解析することによって、アイテムの配信元がDMS 2 であることや、プレイリストに対応するRoot コンテナなどを特定する。

【 0 1 0 7 】

[ DMR の探索処理 ]

次に、図 7 のフローチャートを参照して、アイテムを探索するDMR 1 の処理について説明する。

【 0 1 0 8 】

図 7 の処理は、DMC 3 から送信されてきたPlayContainerUriの解析が終了した後に行われる。DMS 2 においては、図 3 に示すツリー構造によってアイテムが管理されているものとする。また、PlayContainerUriのcontainer-id-argによって図 3 のContainer\_Aが指定され、first-item-id-argによって、1st Item TrackとしてItem\_1が指定されているものとする。PlayContainerUriには、Container\_AのObjectIDとItem\_1のObjectIDが含まれている。

【 0 1 0 9 】

ステップ S 1 0 1 において、DMR 1 の制御部 1 2 は、通信部 1 1 を制御するなどしてDMS 2 と通信を行い、1st Item Track起点探索処理を行う。1st Item Track起点探索処理においては、1st Item Trackとして指定されたItem\_1を起点として、アイテムの探索が行われる。

【 0 1 1 0 】

ステップ S 1 0 2 において、制御部 1 2 は、Root コンテナ起点探索処理を行う。Root コンテナ起点探索処理においては、Root コンテナであるContainer\_Aを起点として、1st Item Trackとして指定されたItem\_1が見つかるまで探索が行われる。

【 0 1 1 1 】

ステップ S 1 0 3 において、制御部 1 2 は、1st Item Track起点探索処理により得られたコンテナ情報と、Root コンテナ起点探索処理により得られたコンテナ情報を統合する。コンテナ情報の統合については後述する。

【 0 1 1 2 】

ここで、図 8 乃至図 1 3 のフローチャートを参照して、図 7 のステップ S 1 0 1 において行われる1st Item Track起点探索処理について説明する。

## 【 0 1 1 3 】

図 8 のステップ S 1 1 1 において、制御部 1 2 は、PlayContainerUri の first-item-id-arg で 1st Item Track として指定されたアイテムを対象として BrowseMetaData を実行する。

## 【 0 1 1 4 】

ここで、DMR 1 が行う Browse には、BrowseMetaData と BrowseDirectChildren がある。

## 【 0 1 1 5 】

BrowseMetaData は、オブジェクトを指定して、そのオブジェクトの情報を DMR 1 が取得するための Browse である。DMR 1 から DMS 2 に送信される BrowseMetaData のコマンドには ObjectID が含まれており、BrowseMetaData のコマンドを受信した DMS 2 は、ObjectID により指定されたオブジェクトの情報を DMR 1 に対して送信する。DMS 2 から送信されるオブジェクトの情報には、ObjectID により指定されたオブジェクトを格納するコンテナ（親のコンテナ）の ObjectID である parentID などが含まれる。

10

## 【 0 1 1 6 】

BrowseDirectChildren は、コンテナを指定して、そのコンテナに格納されているオブジェクト（子のオブジェクト）の情報を DMR 1 が取得するための Browse である。DMR 1 から DMS 2 に送信される BrowseDirectChildren のコマンドには ObjectID が含まれており、BrowseDirectChildren のコマンドを受信した DMS 2 は、ObjectID により指定されたコンテナに格納されているコンテナやアイテムの ObjectID、アイテムの URL などの情報を DMR 1 に送信する。

20

## 【 0 1 1 7 】

この例においては、BrowseMetaData が実行されることにより、1st Item Track として指定された Item\_1 の親のコンテナである Container\_C の ObjectID が parentID として取得される。

## 【 0 1 1 8 】

ステップ S 1 1 2 において、制御部 1 2 は、1st Item Track として指定されたアイテムの親のコンテナを対象とする BrowseDirectChildren の準備を行う。BrowseDirectChildren の準備として、BrowseDirectChildren のパラメータである RequestCount の値として 1、startingIndex の値として、PlayContainerUri で指定された first-item-index-arg の値を設定することが行われる。

30

## 【 0 1 1 9 】

RequestCount は、BrowseDirectChildren に対する応答として情報を返すオブジェクトの数を指定するパラメータである。例えば、RequestCount の値が 1 であることは、startingIndex で指定された位置にある 1 つのオブジェクトの情報の送信を要求することを表す。

## 【 0 1 2 0 】

startingIndex は、情報の送信を要求するオブジェクトの位置を表すパラメータである。例えば、startingIndex の値が 0 であることは、BrowseDirectChildren の対象となるコンテナに格納されている最初（ソート順で最初）のアイテムの位置を表す。

## 【 0 1 2 1 】

DMC 3 から送信されてきた PlayContainerUri には、Container\_C における Item\_1 の位置を表す情報として first-item-index-arg=1 が設定されている。first-item-index-arg の値が 1 であることは、Container\_C を対象として BrowseDirectChildren を実行したときに最初に見つかるオブジェクトである Container\_F の位置を startingIndex=0 で表される位置として、1st Item Track である Item\_1 が、その次の位置にあることを表す。

40

## 【 0 1 2 2 】

すなわち、制御部 1 2 は、RequestCount の値として 1、startingIndex の値として 1 を、Container\_C を対象とする BrowseDirectChildren のパラメータとして設定する。Container\_C を対象とすることは、Container\_C の ObjectID により指定される。

## 【 0 1 2 3 】

ステップ S 1 1 3 において、制御部 1 2 はカウンタの値を 1 に初期化する。制御部 1 2

50

は、記憶部 13 に記憶させておくなどして、探索により見つかったアイテムの数を表すカウンタを管理している。その後、処理はコンテナ探索 0 に進む。

【0124】

次に、図 9 のフローチャートを参照して、コンテナ探索 0 について説明する。

【0125】

ステップ S 121 において、制御部 12 は、BrowseDirectChildrenを実行する。

【0126】

例えば、図 8 のステップ S 112 において準備が行われた後に、Container\_Cを対象とするBrowseDirectChildrenがステップ S 121 において行われた場合、それに対する応答として、DMS 2 からは、Container\_Cに格納され、startingIndex=1により指定される 1 つのオブジェクトであるItem\_1の情報が送信されてくる。Item\_1の情報には、Item\_1のObjectIDが含まれる。

10

【0127】

なお、1st Item Track起点探索処理、Rootコンテナ起点探索処理で実行されるBrowseDirectChildrenにおいては、RequestCountの値として常に1が設定される。すなわち、DMS 2 により管理されているオブジェクトの情報が、1 つずつ、BrowseDirectChildrenが行われる毎にDMR 1 により取得されることになる。

【0128】

また、startingIndexの値として、1st Item Trackで指定されたアイテムの親のコンテナに対してBrowseDirectChildrenを実行する場合を除いて、初期値として0が設定され、その後、同じコンテナを対象としてBrowseDirectChildrenを実行する毎に1を加算した値が設定される。すなわち、BrowseDirectChildrenの対象となるコンテナに格納されているオブジェクトの情報が、先頭のものから順に、1 つずつ、BrowseDirectChildrenが行われる毎にDMR 1 により取得されることになる。

20

【0129】

ステップ S 122 において、制御部 12 は、アイテムが見つかったか否かを判定する。

【0130】

アイテムが見つかったとステップ S 122 において判定した場合、ステップ S 123 において、制御部 12 は、コンテナ情報作成処理を行う。

【0131】

ステップ S 124 において、制御部 12 は、現在のコンテナ（現在探索対象としているコンテナ）を対象とするBrowseDirectChildrenのstartingIndexの値として、それまでの値に1を加算した値を設定する。

30

【0132】

startingIndexの値が更新された後、ステップ S 121 に戻り、同じコンテナを対象としてBrowseDirectChildrenが実行される。同じコンテナを対象としてBrowseDirectChildrenが繰り返し行われることにより、現在のコンテナに直接格納されている全てのアイテムが見つかることになる。

【0133】

一方、アイテムが見つからないとステップ S 122 において判定した場合、ステップ S 125 において、制御部 12 は、ステップ S 121 で実行したBrowseDirectChildrenによってコンテナが見つかったか否かを判定する。

40

【0134】

コンテナが見つかったとステップ S 125 において判定した場合、ステップ S 126 において、制御部 12 は、現在のコンテナのObjectIDと、現在のstartingIndexの値をスタックに格納する。スタック用のメモリ領域も例えば記憶部 13 に確保されている。

【0135】

ステップ S 127 において、制御部 12 は、ステップ S 121 のBrowseDirectChildrenによって見つかったコンテナを、次の探索の対象として選択する。その後、処理はコンテナ探索 1 に進む。

50



## 【 0 1 3 6 】

一方、ステップ S 1 2 5 においてコンテナが見つからないと判定した場合、ステップ S 1 2 8 において、制御部 1 2 は、現在のコンテナがRootコンテナであるか否かを判定する。現在のコンテナがRootコンテナであるか否かは、例えば、BrowseDirectChildrenによって得られたコンテナのObjectIDが、DMC 3 から送信されてきたPlayContainerUriのcontainer-id-argで指定されるObjectIDと同じであるか否かに基づいて判定される。

## 【 0 1 3 7 】

現在のコンテナがRootコンテナではないとステップ S 1 2 8 において判定した場合、制御部 1 2 はコンテナ探索 2 を行う。

## 【 0 1 3 8 】

一方、現在のコンテナがRootコンテナであるとステップ S 1 2 8 において判定した場合、制御部 1 2 は、1st Item Track起点探索処理を終了し、図 7 のステップ S 1 0 1 以降の処理を行う。

## 【 0 1 3 9 】

次に、図 1 0 のフローチャートを参照して、図 9 のステップ S 1 2 3 において行われるコンテナ情報作成処理について説明する。

## 【 0 1 4 0 】

図 1 0 の処理は、ステップ S 1 2 1 において行われたBrowseDirectChildrenによってアイテムが見つかる毎に行われる。

## 【 0 1 4 1 】

ステップ S 1 4 1 において、制御部 1 2 は、現在のコンテナからアイテムが見つかったのは初めてか否かを判定する。例えば、現在のコンテナがContainer\_Cであり、BrowseDirectChildrenによって見つかったアイテムが、先頭のアイテムであるItem\_1である場合、現在のコンテナからアイテムが見つかったのが初めてであるとして判定される。

## 【 0 1 4 2 】

現在のコンテナからアイテムが見つかったのが初めてであるとステップ S 1 4 1 において判定した場合、ステップ S 1 4 2 において、制御部 1 2 は、StartingIndex, Start Track No, End Track No, 1st Item Track No, ContainerIDを格納するメモリ領域を記憶部 1 3 に確保する。これらの情報は、コンテナ情報に含まれる情報である。

## 【 0 1 4 3 】

StartingIndexは、ContainerIDにより指定されるコンテナを対象としてBrowseDirectChildrenを行うときに、startingIndexの値としてどの値を指定すれば、そのコンテナに格納されている最初のアイテム (Start Track Noで指定されるアイテム) が見つかるのかを表す。あるコンテナを対象とするBrowseDirectChildrenによって最初にアイテムが見つかったときのstartingIndexの値が、StartingIndexとして制御部 1 2 により登録される。

## 【 0 1 4 4 】

Start Track Noは、ContainerIDにより指定されるコンテナに格納されるアイテムのうちの最初のアイテムのトラック番号を表す。1st Item Track起点探索処理においては、見つかったアイテムの順に、トラック番号として仮の番号が割り当てられる。あるコンテナを対象とするBrowseDirectChildrenによって最初に見つかったアイテムのトラック番号がStart Track Noとして制御部 1 2 により登録される。

## 【 0 1 4 5 】

End Track Noは、ContainerIDにより指定されるコンテナに格納されるアイテムのうちの最後のアイテムのトラック番号を表す。あるコンテナを対象とするBrowseDirectChildrenによって最後に見つかったアイテムのトラック番号がEnd Track Noとして制御部 1 2 により登録される。

## 【 0 1 4 6 】

1st Item Track Noは、PlayContainerUriのfirst-item-id-argによって1st Item Trackとして指定されたアイテムが、ContainerIDにより指定されるコンテナに格納されるものある場合に、その1st Item Trackのアイテムのトラック番号を表す。1st Item Trackとし

10

20

30

40

50

て指定されたアイテムが格納されていないコンテナの情報には、1st Item Track Noの値として0が登録される。

【0147】

ContainerIDは、それに対応付けられて記憶されるStartingIndex, Start Track No, End Track No, 1st Item Track Noの各情報が、どのコンテナに関する情報であるのかを表す。ContainerIDには、コンテナのObjectIDが登録される。

【0148】

ステップS143において、制御部12は、StartingIndexの値として、現在のstartingIndexの値を登録し、Start Track Noの値として、現在のカウンタ値を登録する。また、制御部12は、ContainerIDとして、BrowseDirectChildrenで利用したObjectID（現在のコンテナのObjectID）を登録する。

10

【0149】

現在のコンテナからアイテムが見つかったのが初めてではないとステップS141において判定された場合、ステップS142, S143の処理はスキップされる。

【0150】

ステップS144において、制御部12は、End Track Noの値として、現在のカウンタ値を登録する。現在のコンテナの情報として、End Track Noの値が既に登録されている場合、そのEnd Track Noの値は現在のカウンタ値によって更新される。

【0151】

ステップS145において、制御部12は、見つかったアイテムがPlayContainerUriのfirst-item-id-argで指定されたアイテムと同じであるか否かを判定する。

20

【0152】

見つかったアイテムがPlayContainerUriのfirst-item-id-argで指定されたアイテムと同じであるとステップS145において判定した場合、制御部12は、ステップS146において、1st Item Track Noの値として現在のカウンタ値を登録する。

【0153】

1st Item Track Noの値を登録した後、または、見つかったアイテムがPlayContainerUriのfirst-item-id-argで指定されたアイテムではないとステップS145において判定された場合、ステップS147において、制御部12は、カウンタの値に1を加算する。

【0154】

カウンタの値が更新された後、図9のステップS123に戻り、それ以降の処理が行われる。

30

【0155】

次に、図11のフローチャートを参照して、図9のステップS127の処理に続けて行われるコンテナ探索1について説明する。

【0156】

コンテナ探索1は、直前に実行したBrowseDirectChildrenによってコンテナが見つかった場合、その見つかったコンテナを対象として行われる。

【0157】

ステップS161において、制御部12は、startingIndexの値を0に初期化し、現在のコンテナ（直前に実行したBrowseDirectChildrenによって見つかったコンテナ）を対象とするBrowseDirectChildrenの準備を行う。

40

【0158】

ステップS162において、制御部12は、現在のコンテナを対象としてBrowseDirectChildrenを実行する。その後、図9のコンテナ探索0のステップS122以降の処理が行われる。

【0159】

次に、図12のフローチャートを参照して、現在のコンテナがRootコンテナではないと図9のステップS128において判定された場合に、コンテナ探索0に続けて行われるコンテナ探索2について説明する。

50

## 【0160】

ステップS 171において、制御部12は、スタックにデータが格納されているか否かを判定する。ここでは、コンテナ探索2が開始される前に図9のステップS 126の処理、または、後述する図13のステップS 188の処理が行われている場合、スタックにデータが格納されているとして判定されることになる。

## 【0161】

スタックにデータが格納されていないとステップS 171において判定された場合、処理はコンテナ探索3に進む。

## 【0162】

一方、スタックにデータが格納されているとステップS 171において判定した場合、ステップS 172において、制御部12は、スタックに格納しておいたObjectIDとstartingIndexの値を取り出す。取り出されたObjectIDとstartingIndexの値はスタックから消去される。

10

## 【0163】

ステップS 173において、スタックから取り出したObjectIDにより指定されるコンテナを対象としてBrowseDirectChildrenを実行する。その際、制御部12は、startingIndexの値として、スタックから取り出したstartingIndexの値に1を加算した値を用いる。その後、図9のコンテナ探索0のステップS 122以降の処理が行われる。

## 【0164】

次に、図13のフローチャートを参照して、図12のステップS 171においてスタックにデータが格納されていないと判定された場合に行われるコンテナ探索3について説明する。

20

## 【0165】

ステップS 181において、制御部12は、現在のコンテナのObjectIDを調べる。ここでは、現在のコンテナのObjectIDを とする。

## 【0166】

ステップS 182において、制御部12は、ステップS 181で調べたObjectIDを用いて現在のコンテナを対象としてBrowseMetaDataを実行し、現在のコンテナの親のコンテナのObjectIDであるparentIDを取得する。

## 【0167】

ステップS 183において、制御部12は、startingIndexの値として0を設定し、parentIDで指定されるコンテナを対象とするBrowseDirectChildrenの準備を行う。

30

## 【0168】

ステップS 184において、制御部12は、BrowseDirectChildrenを実行する。

## 【0169】

ステップS 185において、制御部12は、BrowseDirectChildrenの実行結果に基づいて、 で指定される、ステップS 182でBrowseMetaDataを実行したコンテナが見つかったか否かを判定する。

## 【0170】

ステップS 182でBrowseMetaDataを実行したコンテナが見つからないとステップS 185において判定した場合、ステップS 186において、制御部12は、現在のコンテナ（ステップS 182で取得したparentIDで指定されるコンテナ）を対象とするBrowseDirectChildrenのstartingIndexの値として、それまでの値に1を加算した値を設定する。

40

## 【0171】

startingIndexの値が更新された後、ステップS 184に戻り、同じコンテナを対象としてBrowseDirectChildrenが実行される。

## 【0172】

ステップS 182でBrowseMetaDataを実行したコンテナが見つかったとステップS 185において判定した場合、ステップS 187において、制御部12は、現在のstartingIn

50

dexの値に 1 を加算する。

【 0 1 7 3 】

ステップ S 1 8 8 において、制御部 1 2 は、現在探索していたコンテナのObjectIDとstartingIndexの値をスタックに格納する。その後、処理はコンテナ探索 0 に進む。

【 0 1 7 4 】

ここで、図 3 のツリー構造を対象として以上の探索を行った場合の処理の流れを具体的に説明する。

【 0 1 7 5 】

はじめに、PlayContainerUriのfirst-item-id-argによって1st Item Trackとして指定されたItem\_1を対象としてBrowseMetaDataが実行される（図 8 のステップ S 1 1 1）。Item\_1を対象としたBrowseMetaDataにより、親のコンテナであるContainer\_CのObjectIDが取得される。 10

【 0 1 7 6 】

次に、取得されたObjectIDを用いて、Container\_Cを対象としてBrowseDirectChildrenが実行される（図 9 のステップ S 1 2 1）。Container\_Cを対象とする 1 回目のBrowseDirectChildrenのstartingIndexの値は、PlayContainerUriで指定されたfirst-item-index-argの値と同じ1である。

【 0 1 7 7 】

Container\_Cを対象とする 1 回目のBrowseDirectChildrenによりItem\_1が見つかり、それに応じてコンテナ情報作成処理が行われる（図 9 のステップ S 1 2 3）。コンテナ情報作成処理により、Container\_Cの情報として、StartingIndex=1、Start Track No=1、ContainerID=Container\_CのObjectIDが、それぞれ登録される（図 1 0 のステップ S 1 4 3）。また、1st Item Track No=1が登録される（図 1 0 のステップ S 1 4 6）。 20

【 0 1 7 8 】

startingIndexの値は図 8 のステップ S 1 2 4 において 2 となる。Container\_Cを対象とする 2 回目のBrowseDirectChildrenによりItem\_2が見つかり、それに応じて、コンテナ情報作成処理が行われる（図 9 のステップ S 1 2 3）。コンテナ情報作成処理により、Container\_Cの情報として、End Track No=2が登録される（図 1 0 のステップ S 1 4 4）。 30

【 0 1 7 9 】

この時点で記憶部 1 3 に記憶されるコンテナ情報を図 1 4 に示す。図 1 4 に示すように、Container\_Cの情報として、StartingIndex=1、Start Track No=1、End Track No=2、1st Item Track No=1が、Container\_CのObjectIDと対応付けて記憶される。 30

【 0 1 8 0 】

このように、DMR 1 においては、あるコンテナを対象とした探索結果を表す情報として、アイテムの再生時に、そのコンテナを対象としてどのような値をパラメータとして設定してBrowseDirectChildrenを行えばアイテムの情報を取得することができるのかを表す情報が記憶される。

【 0 1 8 1 】

すなわち、Container\_Cを再生する場合、制御部 1 2 は、コンテナ情報を参照することによって、Container\_CのObjectIDを特定することが可能になる。 40

【 0 1 8 2 】

また、制御部 1 2 は、特定したObjectIDに基づいてContainer\_Cを指定するとともに、startingIndex=1を指定するBrowseDirectChildrenを実行することによって、仮のトラック番号というTrack\_1のアイテムの情報を取得することが可能になる。制御部 1 2 は、取得した情報（ObjectID、URL等）に基づいてアイテム取得要求を行うことによって、Track\_1のアイテムを取得し、再生することができる。

【 0 1 8 3 】

さらに、制御部 1 2 は、特定したObjectIDに基づいてContainer\_Cを指定するとともに、startingIndex=2を指定するBrowseDirectChildrenを実行することによって、仮のトラック番号というTrack\_2のアイテムの情報を取得することが可能になる。制御部 1 2 は、 50

取得した情報に基づいてアイテム取得要求を行うことによって、Track\_2のアイテムを取得し、再生することができる。

【0184】

また、制御部12は、Container\_Cに格納されているアイテムが、Track\_1のアイテムからTrack\_2のアイテムまでであることをコンテナ情報から特定することができる。また、制御部12は、“End Track No” - “Start Track No” + 1の計算結果から、Container\_Cには再生対象のアイテムが2個あることも特定することができる。制御部12は、1st Item Track No に1が設定されていることから、PlayContainerUriで指定された、最初に再生すべきアイテムがContainer\_Cにあることを特定することができる。

【0185】

これにより、BrowseDirectChildrenによって取得された情報を全て記憶しておく場合に較べて、DMS2から取得可能なアイテムの情報を記憶しておくために必要なメモリの容量を減らすことが可能になる。

【0186】

図3のツリー構造の探索の説明に戻り、startingIndexの値は図8のステップS124において3となり、Container\_Cを対象とする3回目のBrowseDirectChildrenによりContainer\_Gが見つかる。

【0187】

Container\_Cの情報がスタックに格納された後（図9のステップS126）、Container\_Gが、次の探索対象のコンテナとして選択される（図9のステップS127）。Container\_Gの探索に移る前にContainer\_Cの情報をスタックに格納しておく理由は、Container\_Gの探索が終わった後、Container\_Cに戻り、Container\_Cを対象として既に探索を行った位置の続きから探索を再開できるようにするためである。

【0188】

Container\_Gを対象としてコンテナ探索1が行われ（図11）、続けて、コンテナ探索0のステップS122以降の処理が行われることによって、Item\_3とItem\_4が見つかる。それぞれのアイテムが見つかる毎にコンテナ情報作成処理が行われることによって、記憶部13には、Container\_Gの情報としてItem\_3とItem\_4の情報が記憶される。

【0189】

この時点で記憶部13に記憶されるコンテナ情報を図15に示す。図15に示すように、Container\_Cの情報に追加して、StartingIndex=0、Start Track No=3、End Track No=4、1st Item Track No=0がContainer\_GのObjectIDと対応付けて記憶される。図15のコンテナ情報は、Container\_Cに格納されているアイテムを再生した後、再生対象がContainer\_Gに格納されているアイテムに移ることを表す。

【0190】

Container\_Gを対象とする2回のBrowseDirectChildrenによってItem\_3とItem\_4が見つかった後に行われる、Container\_Gを対象とする3回目のBrowseDirectChildren（図9のステップS121）によっては、アイテムもコンテナも見つからない。この場合、現在のコンテナであるContainer\_GはRootコンテナではないから、図9のステップS128の判定を経て、処理はコンテナ探索0から、図12のコンテナ探索2に進む。

【0191】

探索対象がContainer\_CからContainer\_Gに移る前にContainer\_Cの情報がスタックに格納されているから、コンテナ探索2においては、スタックに格納されている情報に基づいて、Container\_Cを対象としてBrowseDirectChildrenが実行される（図12のステップS173）。ここで行われるBrowseDirectChildrenは、既に探索が終了しているContainer\_Gの次の位置にあるオブジェクトを探索するものである。その後、処理はコンテナ探索0のステップS122以降に進む。

【0192】

Container\_Cを対象とするBrowseDirectChildrenによってはオブジェクトが見つからないから、処理はコンテナ探索0からコンテナ探索2に進む。2回目のコンテナ探索2にお

10

20

30

40

50

いては、スタックにはContainer\_Cの情報が格納されていないから、ステップS 1 7 1の判定を経て、処理はコンテナ探索2からコンテナ探索3に進む。

【0193】

コンテナ探索3においては、Container\_Cに対してBrowseMetaDataが実行され、親のコンテナであるContainer\_AのObjectIDが取得される(図13のステップS 1 8 2)。また、Container\_Aを対象として、Container\_Cが見つかるまで、BrowseDirectChildrenが繰り返される(図13のステップS 1 8 4)。

【0194】

Container\_Aを対象とする3回目のBrowseDirectChildrenによってContainer\_Cが見つかり、このとき、startIndexの値に1が加算され(図13のステップS 1 8 7)、Container\_Aの情報がスタックに格納された後(図13のステップS 1 8 8)、処理はコンテナ探索3からコンテナ探索0に進む。

【0195】

その後、Container\_Dを対象としてBrowseDirectChildrenが実行され、Item\_5が見つかることに応じてコンテナ情報作成処理が行われる。また、Container\_Eを対象としてBrowseDirectChildrenが実行され、Item\_6が見つかることに応じてコンテナ情報作成処理が行われる。

【0196】

Container\_Aを対象としてBrowseDirectChildrenが実行され、Item\_7が見つかることに応じてコンテナ情報作成処理が行われた後、図9のステップS 1 2 8の処理を経て、1st Item Track起点探索処理が終了する。

【0197】

以上のように、1st Item Track起点探索処理においては、Item\_1を起点として、Item\_1とともに同じContainer\_Cに格納されるItem\_2、Container\_Gに格納されるItem\_3、Item\_3とともに同じContainer\_Gに格納されるItem\_4、Container\_Dに格納されるItem\_5、Container\_Eに格納されるItem\_6、Container\_Aに格納されるItem\_7の順に各アイテムが見つかる。

【0198】

1st Item Track起点探索処理が終了したとき、記憶部13には、図16に示すコンテナ情報が記憶される。

【0199】

図16に示すように、コンテナ情報には、見つかったアイテムを格納するコンテナの順に、Container\_C、Container\_G、Container\_D、Container\_E、Container\_Aの各コンテナの情報が登録される。

【0200】

ただし、このままでは、PlayContainerUriで指定されたDMS2が管理するアイテムの全ての探索結果を取得したことになる。図7を参照して説明したように、1st Item Track起点探索処理が終了したとき、Rootコンテナ起点探索処理が行われる。

【0201】

Rootコンテナ起点探索処理においては、RootコンテナであるContainer\_Aを起点として、1st Item Trackとして指定されたItem\_1が見つかるまで探索が行われる。Rootコンテナ起点探索処理も、探索の起点が異なる点と、Item\_1が見つかったときに処理が終了される点を除いて、基本的に、1st Item Track起点探索処理と同様の処理である。

【0202】

図17のフローチャートを参照して、図7のステップS 1 0 2において行われるRootコンテナ起点探索処理について説明する。

【0203】

ステップS 2 0 1において、制御部12は、startIndex=0を設定し、RootコンテナであるContainer\_Aに対するBrowseDirectChildrenの準備を行う。

【0204】

ステップ S 2 0 2 において、制御部 1 2 は、カウンタの値を 1 に初期化する。

【 0 2 0 5 】

その後、処理はコンテナ探索 0 に進み、上述した処理と同様の処理が行われる。すなわち、Rootコンテナ起点探索処理においては、Container\_Aに格納されるItem\_8、Container\_Bに格納されるItem\_9、Container\_Fに格納されるItem\_10が見つかる。Item\_10が見つかった後、Container\_Cに格納されるItem\_1が見つかるから、そのときにRootコンテナ起点探索処理は終了する。

【 0 2 0 6 】

Rootコンテナ起点探索処理のBrowseDirectChildrenによって見つかったアイテムが1st Item Trackとして指定されたItem\_1と同じアイテムであるか否かは、BrowseDirectChildrenによって得られたアイテムのObjectIDが、DMC 3 から送信されてきたPlayContainerUriのfirst-item-id-argで指定されるObjectIDと同じであるか否かに基づいて判定される。Rootコンテナ起点探索処理において、Container\_Cを対象とし、startIndexの値を1としてBrowseDirectChildrenを行ったときに得られるObjectIDは、PlayContainerUriのfirst-item-id-argで指定されるObjectIDと同じものとなる。このとき、Rootコンテナ起点探索処理は終了する。

【 0 2 0 7 】

Rootコンテナ起点探索処理が終了したとき、記憶部 1 3 には、図 1 8 に示すコンテナ情報が記憶される。

【 0 2 0 8 】

図 1 8 に示すように、コンテナ情報には、Rootコンテナ起点探索処理によって見つかったアイテムを格納するコンテナの順に、Container\_A、Container\_B、Container\_F、Container\_Cの各コンテナの情報が登録される。

【 0 2 0 9 】

上述したように、DMS 2 においては、RootコンテナであるContainer\_Aを起点としてBrowseを実行したときに見つけることができる順に、それぞれのアイテムにトラック番号が割り当てられている。

【 0 2 1 0 】

Rootコンテナ起点探索処理においても、見つかった順に各アイテムにトラック番号が制御部 1 2 により割り当てられるが、各アイテムに割り当てられたトラック番号は、仮の番号ではなく、DMS 2 により割り当てられているトラック番号と同じ、正式な番号である。図 1 8 のStart Track No、End Track No、1st Item Track Noは、正式なトラック番号である。

【 0 2 1 1 】

図 7 を参照して説明したように、Rootコンテナ起点探索処理が終了したとき、1st Item Track起点探索処理によって得られたコンテナ情報と、Rootコンテナ起点探索処理によって得られたコンテナ情報の統合が行われる（ステップ S 1 0 3 ）。

【 0 2 1 2 】

図 1 9 は、1st Item Track起点探索処理によって得られた図 1 6 のコンテナ情報と、Rootコンテナ起点探索処理によって得られた図 1 8 のコンテナ情報を統合したコンテナ情報の例を示す図である。

【 0 2 1 3 】

Rootコンテナ起点探索処理において各アイテムに割り当てられたトラック番号は正式な番号であるから、図 1 9 に示すように、Rootコンテナ起点探索処理によって得られた図 1 8 の情報に続けて、1st Item Track起点探索処理によって得られた図 1 6 の情報を加えるようにしてコンテナ情報の統合が行われる。

【 0 2 1 4 】

1st Item Track起点探索処理によって得られたコンテナ情報に含まれる各アイテムのトラック番号は、PlayContainerUriによって1st Itemとして指定されたItem\_1の仮のトラック番号 1 を基準とする相対値である。Item\_1の正式なトラック番号は4であるから、コン

テナ情報の統合時、1st Item Track起点探索処理によって得られたテナ情報に含まれる各アイテムのトラック番号は、Item\_1の正式なトラック番号である4を基準として、相対的に同じ数だけ差があるように書き換えられる。すなわち、図16のテナ情報に含まれるトラック番号は、図19のContainer\_Cの行以降に示すように、3(4-1)ずつ増えるように書き換えられる。

【0215】

これにより、PlayContainerUriで指定された、DMS2が管理するContainer\_Aに格納されるアイテムの全ての探索結果を取得したことになる。

【0216】

以上のように、DMS2が管理するツリー構造全体を把握するための情報として、各テナの情報のみを記憶するようにしたため、メモリの使用量を抑えることができる。例えば、1つのテナに10個のアイテムが格納されており、そのようなテナが1000個ある場合、アイテムの情報を個々に全て記憶しておくとした場合には10000個のアイテムの情報を格納しておく分のメモリ領域が必要になるが、以上の方法によれば10分の1の領域で済む。

【0217】

また、ツリー構造全体の情報を記憶しておくことになるため、トラック番号を指定しての再生やランダム再生がDMC3から指示された場合であっても、その指示に応じた処理を行うことが可能になる。

【0218】

[ DMRの再生処理 ]

次に、図20のフローチャートを参照して、DMR1の再生処理について説明する。

【0219】

図20の処理は、DMS2が管理するアイテムの全ての探索結果が以上の処理によって取得された状態で、例えば、DMC3から、所定のアイテムの再生が指示されたときに開始される。DMC3からは、トラック番号を指定してアイテムの再生が指示される。

【0220】

ステップS211において、制御部12は、DMC3により再生が指示されたトラック番号が割り当てられたアイテムを格納するテナの情報をテナ情報から読み出す。例えば、図19のテナ情報が記憶部13に記憶されている場合において、トラック番号9のアイテムを再生することが指示されたとき、トラック番号として9が割り当てられたアイテムを格納するContainer\_EのObjectIDと、StartingIndex=0が読み出される。

【0221】

ステップS212において、制御部12は、再生が指示されたトラック番号が割り当てられたアイテムを格納するテナを対象としてBrowseDirectChildrenを実行する。StartingIndexの値にはステップS211で読み出された値が用いられる。BrowseDirectChildrenに対する応答として、DMS2からは、再生が指示されたアイテムのObjectID、URLなどの詳細情報が送信されてくる。

【0222】

ステップS213において、制御部12は、ObjectIDやURLを用いるなどして再生対象のアイテムを指定し、DMS2に対してアイテムの送信を要求する。ここで行われる要求に対する応答として、DMS2からは、コンテンツ記憶部23に記憶されているアイテムのデータが送信されてくる。DMS2から送信されてきたアイテムのデータは通信部11により受信され、再生部14に供給される。

【0223】

ステップS214において、再生部14は、通信部11から供給されたアイテムを再生する。

【0224】

トラック番号を指定して再生が指示された場合、その再生処理は以上のようにして行われる。図20の再生処理は、トラック番号を指定して再生が指示された場合だけでなく、

10

20

30

40

50



ランダム再生がDMC 3 により指示された場合にも適用可能である。

【 0 2 2 5 】

ランダム再生を行う場合、制御部 1 2 は、まず、コンテナ情報に登録されているContainer\_IDの中から、1つのContainer\_IDをランダムに選択する。次に、制御部 1 2 は、選択したContainer\_IDに対応付けて記憶されている、同じ行のStart Track NoとEnd Track Noで表される範囲のトラック番号の中から、1つのトラック番号をランダムに選択する。

【 0 2 2 6 】

トラック番号を選択し終えたとき、制御部 1 2 は、startingIndexをコンテナ情報のStartingIndexの値に基づいて決定し、Container\_IDで特定されるコンテナに対してBrowseDirectChildrenを行う。これにより、制御部 1 2 は再生すべきアイテムの情報を得ることができる。

10

【 0 2 2 7 】

なお、アイテムをランダムに選択する手法として、図 1 9 のコンテナ情報から最大Track番号が分かるので、その範囲で乱数を生成し、得られた番号とコンテナ情報を照らし合わせることで、どこのコンテナの何番目の位置に、得られたトラック番号のアイテムがあるのかを選択する方法もある。

【 0 2 2 8 】

[ コンピュータの構成例 ]

上述した一連の処理は、ハードウェアにより実行することもできるし、ソフトウェアにより実行することもできる。一連の処理をソフトウェアにより実行する場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または汎用のパーソナルコンピュータなどに、プログラム記録媒体からインストールされる。

20

【 0 2 2 9 】

図 2 1 は、上述した一連の処理をプログラムにより実行するコンピュータのハードウェアの構成例を示すブロック図である。

【 0 2 3 0 】

CPU(Central Processing Unit) 1 0 1、ROM(Read Only Memory) 1 0 2、RAM(Random Access Memory) 1 0 3 は、バス 1 0 4 により相互に接続されている。

【 0 2 3 1 】

30

バス 1 0 4 には、さらに、入出力インタフェース 1 0 5 が接続されている。入出力インタフェース 1 0 5 には、キーボード、マウスなどよりなる入力部 1 0 6、ディスプレイ、スピーカなどよりなる出力部 1 0 7 が接続される。また、入出力インタフェース 1 0 5 には、ハードディスクや不揮発性のメモリなどよりなる記憶部 1 0 8、ネットワークインタフェースなどよりなる通信部 1 0 9、リムーバブルメディア 1 1 1 を駆動するドライブ 1 1 0 が接続される。

【 0 2 3 2 】

以上のように構成されるコンピュータでは、CPU 1 0 1 が、例えば、記憶部 1 0 8 に記憶されているプログラムを入出力インタフェース 1 0 5 及びバス 1 0 4 を介してRAM 1 0 3 にロードして実行することにより、上述した一連の処理が行われる。

40

【 0 2 3 3 】

CPU 1 0 1 が実行するプログラムは、例えばリムーバブルメディア 1 1 1 に記録して、あるいは、ローカルエリアネットワーク、インターネット、デジタル放送といった、有線または無線の伝送媒体を介して提供され、記憶部 1 0 8 にインストールされる。

【 0 2 3 4 】

なお、コンピュータが実行するプログラムは、本明細書で説明する順序に沿って時系列に処理が行われるプログラムであっても良いし、並列に、あるいは呼び出しが行われたとき等の必要なタイミングで処理が行われるプログラムであっても良い。

【 0 2 3 5 】

本発明の実施の形態は、上述した実施の形態に限定されるものではなく、本発明の要旨

50

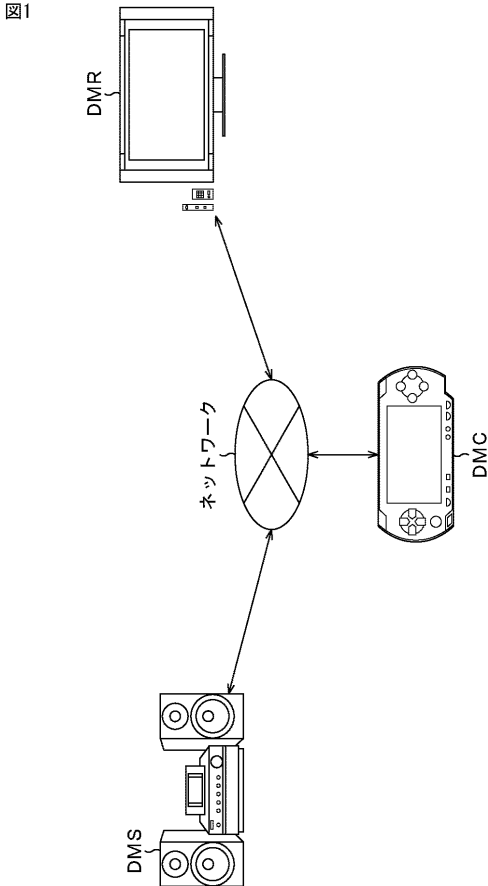
を逸脱しない範囲において種々の変更が可能である。

【符号の説明】

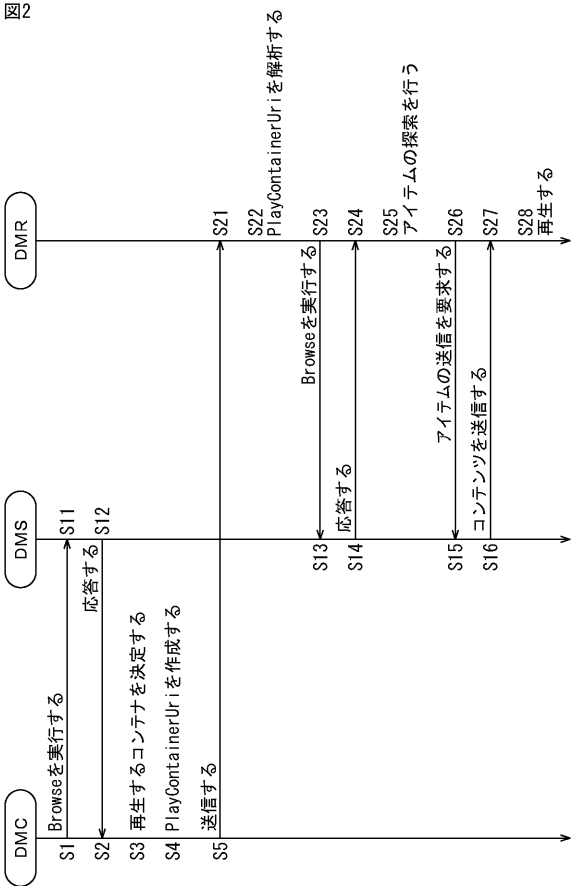
【0236】

1 DMR， 2 DMS， 3 DMC， 11 通信部， 12 制御部， 13 記憶部，  
14 再生部， 21 通信部， 22 制御部， 23 コンテンツ記憶部， 31  
通信部， 32 制御部， 33 操作部， 34 表示部

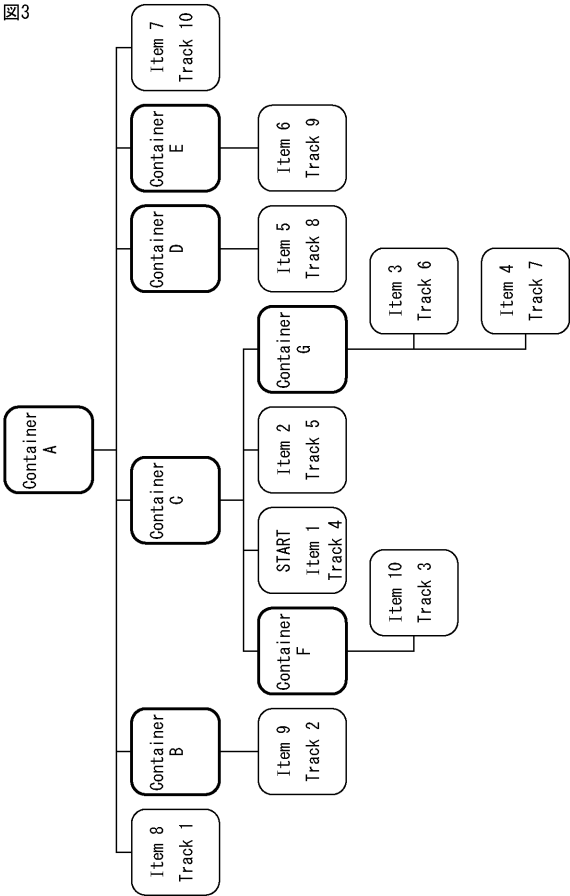
【図1】



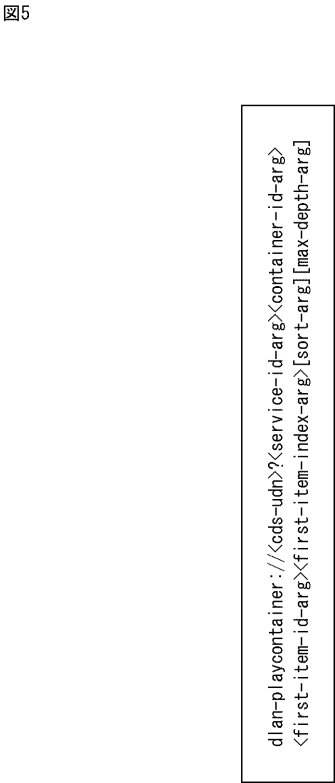
【図2】



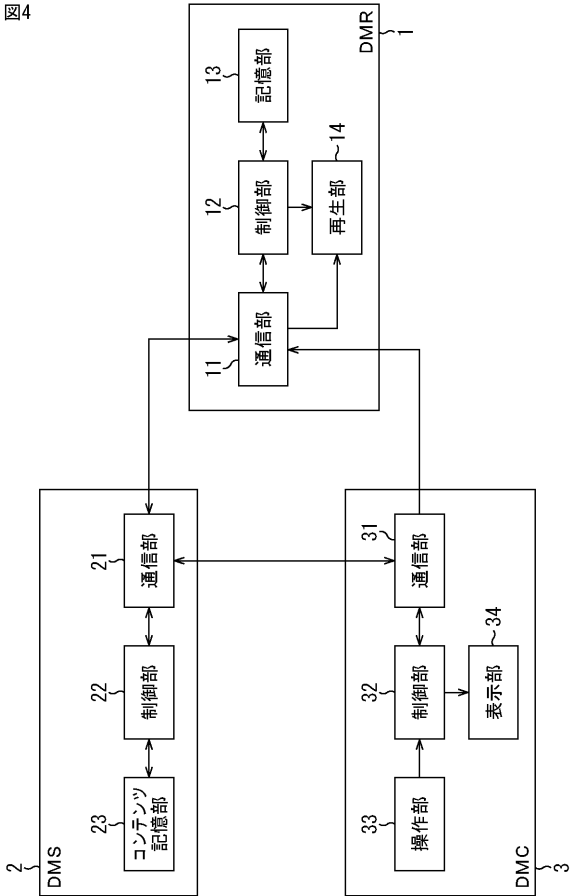
【図3】



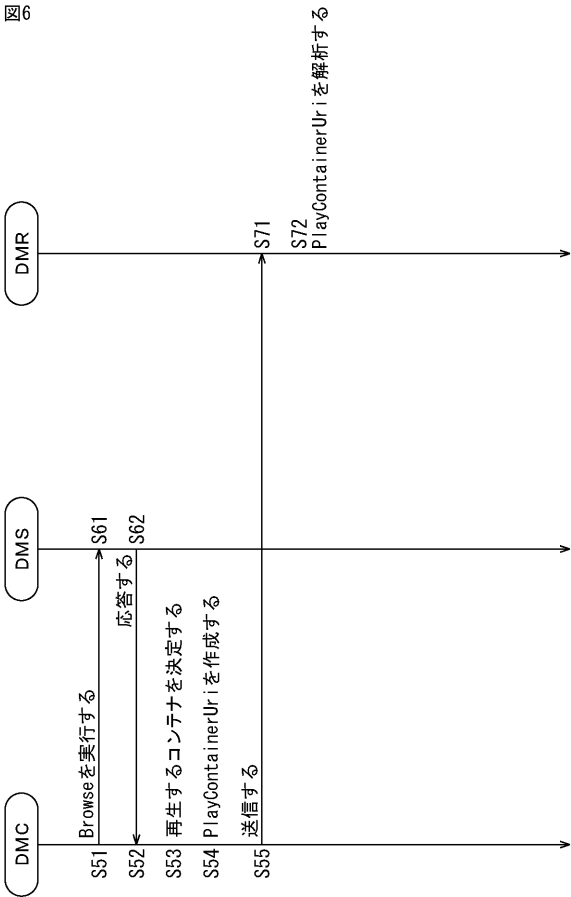
【図5】



【図4】

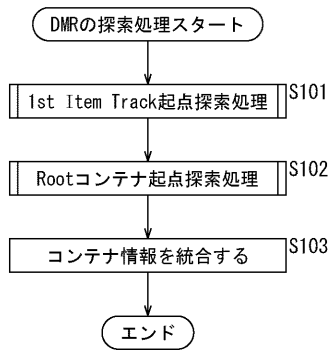


【図6】



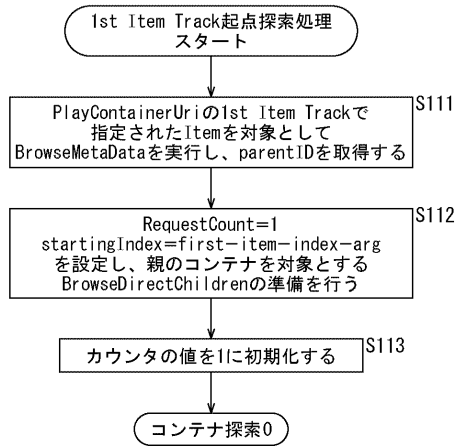
【図 7】

図7



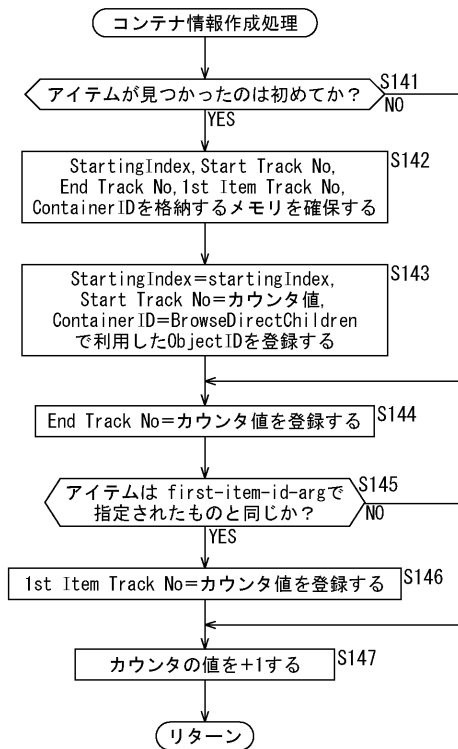
【図 8】

図8



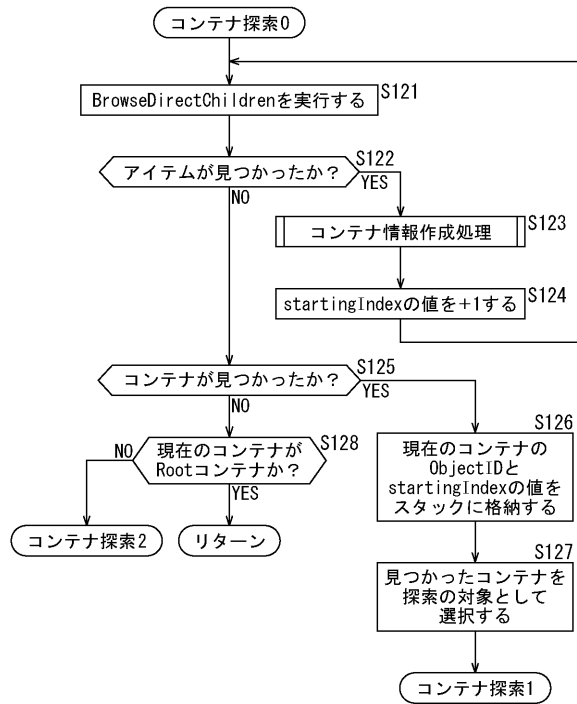
【図 10】

図10



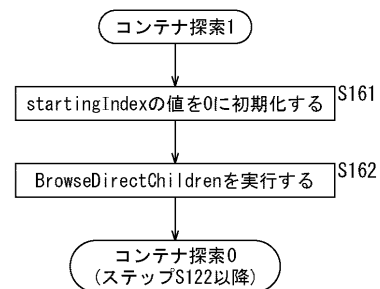
【図 9】

図9



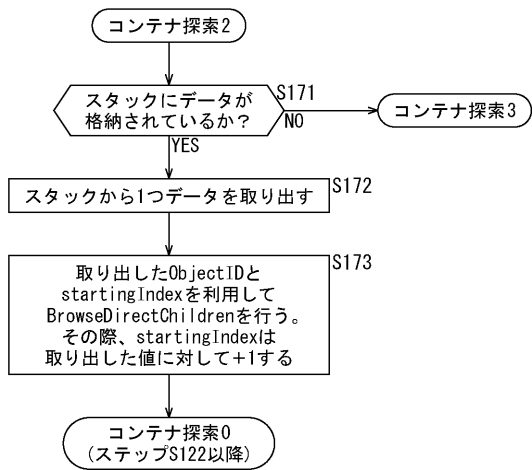
【図 11】

図11



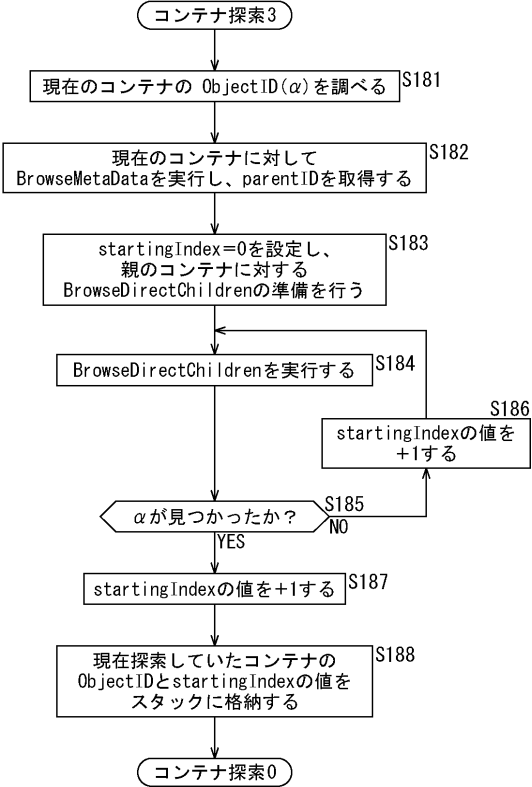
【 図 1 2 】

図12



【 図 1 3 】

図13



【 図 1 4 】

図14

Starting Index	Start Track No	End Track No	1 <sup>st</sup> Item Track No	Container ID
1	1	2	1	Container_C

【 図 1 5 】

図15

Starting Index	Start Track No	End Track No	1 <sup>st</sup> Item Track No	Container ID
1	1	2	1	Container_C
0	3	4	0	Container_G

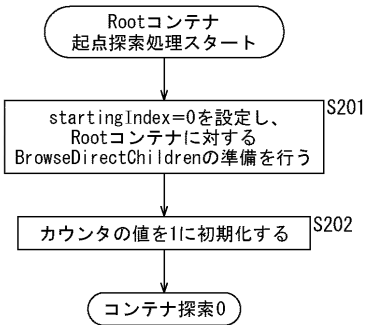
【 図 1 6 】

図16

Starting Index	Start Track No	End Track No	1 <sup>st</sup> Item Track No	Container ID
1	1	2	1	Container_C
0	3	4	0	Container_G
0	5	5	0	Container_D
0	6	6	0	Container_E
5	7	7	0	Container_A

【 図 1 7 】

図17



【 図 1 8 】

図18

Starting Index	Start Track No	End Track No	1 <sup>st</sup> Item Track No	Container ID
0	1	1	0	Container_A
0	2	2	0	Container_B
0	3	3	0	Container_F
1	4	4	4	Container_C

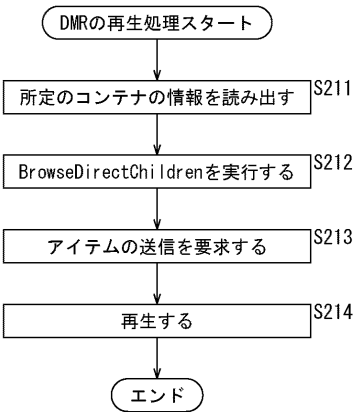
【 図 1 9 】

図19

Starting Index	Start Track No	End Track No	1 <sup>st</sup> Item Track No	Container ID
0	1	1	0	Container_A
0	2	2	0	Container_B
0	3	3	0	Container_F
1	4	5	4	Container_C
0	6	7	0	Container_G
0	8	8	0	Container_D
0	9	9	0	Container_E
5	10	10	0	Container_A

【 図 2 0 】

図20



【 図 2 1 】

図21

