

(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2005-293316

(P2005-293316A)

(43) 公開日 平成17年10月20日(2005.10.20)

(51) Int. Cl. <sup>7</sup>	F I	テーマコード (参考)
G06F 9/445	G06F 9/06 640A	5B017
G06F 9/46	G06F 9/46 340B	5B042
G06F 11/00	G06F 11/00 350C	5B076
G06F 11/30	G06F 11/30 305H	5B098
G06F 12/14	G06F 12/14 510D	5K067

審査請求 未請求 請求項の数 7 O L (全 30 頁) 最終頁に続く

(21) 出願番号	特願2004-108396 (P2004-108396)	(71) 出願人	392026693 株式会社エヌ・ティ・ティ・ドコモ 東京都千代田区永田町二丁目11番1号
(22) 出願日	平成16年3月31日 (2004.3.31)	(74) 代理人	100083806 弁理士 三好 秀和
		(74) 代理人	100100712 弁理士 岩▲崎▼ 幸邦
		(74) 代理人	100095500 弁理士 伊藤 正和
		(74) 代理人	100101247 弁理士 高橋 俊一
		(72) 発明者	菊地 悠 東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内

最終頁に続く

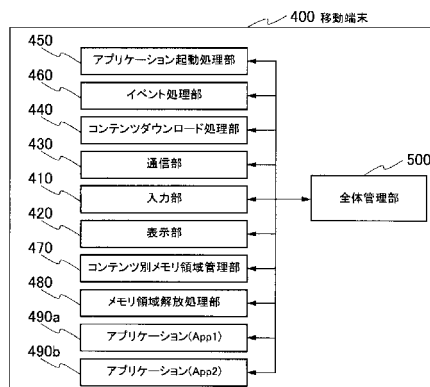
(54) 【発明の名称】 端末装置、プログラム提供装置及びプログラム取得方法

(57) 【要約】

【課題】 計算リソースを効率的に用いてアプリケーションの実行に必要なモジュールを取得すること、及び、計算リソースを効率的に用いて複数のアプリケーションを実行すること。

【解決手段】 コンテンツダウンロード処理部440が、アプリケーション情報に基づいて、実行可能と判断されたアプリケーション及び取得されていないと判断した共有モジュールダウンロードする。アプリケーション起動処理部450が、所望の共有モジュールがロードされていないと判断した場合に、当該共有モジュールをメモリ上にロードし、所望のインスタンスが生成されていないと判断した場合に、当該インスタンスをメモリ上に生成する。イベント処理部460が、検知されたイベントを、当該イベントの配送を必要とするアプリケーションに配送する。メモリ領域解放処理部480がアプリケーションによるメモリ領域への不正なアクセスを検出した場合に、アプリケーションを終了させる。

【選択図】 図4



**【特許請求の範囲】****【請求項 1】**

プログラム提供装置からアプリケーションを取得する端末装置であって、  
前記アプリケーションに係る情報を示すアプリケーション情報を取得するアプリケーション情報取得部と、

前記アプリケーション情報に基づいて、前記アプリケーションの実行に必要なモジュールが、既に取得されているか否かについて判断する取得判断部と、

前記モジュールが取得されていないと判断された場合に、前記モジュールを取得するモジュール取得部とを備えることを特徴とする端末装置。

**【請求項 2】**

特定の状態が生じたことを示すイベントを検知するイベント検知部と、

前記アプリケーションと前記イベントの種別とを関連づけて管理するイベント管理部と

、  
該イベント管理部を参照して、前記イベント検知部によって検知された前記イベントの種別に関連づけられている前記アプリケーションに対して、該イベントを通知するイベント通知部とを備えることを特徴とする請求項 1 に記載の端末装置。

**【請求項 3】**

アクティブなアプリケーションを管理するアクティブアプリケーション管理部と、

前記イベント検知部によって切り替えイベントが検知された場合に、アクティブなアプリケーションを変更するアクティブアプリケーション変更部とを備え、

前記イベント通知部は、前記端末装置に対する入力操作があったことを示す入力イベントを、アクティブなアプリケーションに通知することを特徴とする請求項 2 に記載の端末装置。

**【請求項 4】**

前記アプリケーションに割り当てられるメモリの領域であるメモリ領域を管理するメモリ領域管理部と、

該メモリ領域管理部を参照して、前記アプリケーションによるメモリ領域への不正なアクセスを検出する不正アクセス検出部と、

前記不正アクセス検出部によって、前記アプリケーションによるメモリ領域への不正なアクセスが検出された場合に、該アプリケーションを終了させる終了処理部とを備えることを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の端末装置。

**【請求項 5】**

アプリケーションを実行する端末装置であって、

前記アプリケーション、又は、該アプリケーションの実行に必要なモジュールが、メモリ上にロードされているか否かを管理するロード済みプログラム管理部と、

該ロード済みプログラム管理部を参照して、所望のモジュールがロードされていないと判断した場合に、該モジュールを前記メモリ上にロードするロード部と、

前記ロード部によってロードされた前記モジュールに含まれるクラスに基づいてインスタンスが前記メモリ上に生成されているか否かを管理する生成済みインスタンス管理部と

、  
前記生成済みインスタンス管理部を参照して、所望のインスタンスが生成されていないと判断した場合に、該インスタンスを前記メモリ上に生成するインスタンス生成部とを備えることを特徴とする端末装置。

**【請求項 6】**

端末装置に対してアプリケーションを提供するプログラム提供装置であって、

前記端末装置からのアプリケーション情報要求に応じて、前記アプリケーションに係る情報を示すアプリケーション情報を提供するアプリケーション情報提供部と、

前記アプリケーション情報に基づいて、前記アプリケーションの実行に必要なモジュールが取得されていないと判断した前記端末装置からのモジュール要求に応じて、前記モジュールを提供するモジュール提供部とを備えることを特徴とするプログラム提供装置。

10

20

30

40

50

**【請求項 7】**

端末装置が、プログラム提供装置からアプリケーションを取得するプログラム取得方法であって、

前記プログラム提供装置が、前記端末装置からのアプリケーション情報要求に応じて、前記アプリケーションに係る情報を示すアプリケーション情報を提供する工程と、

前記端末装置が、前記プログラム提供装置によって提供された前記アプリケーション情報を取得する工程と、

前記端末装置が、前記アプリケーション情報に基づいて、前記アプリケーションの実行に必要なモジュールが、既に取得されているか否かについて判断する工程と、

前記端末装置が、前記モジュールが取得されていないと判断した場合に、前記プログラム提供装置にモジュール要求を送信する工程と、

前記プログラム提供装置が、前記モジュール要求に応じて前記モジュールを提供する工程とを備えることを特徴とするプログラム取得方法。

10

**【発明の詳細な説明】****【技術分野】****【0001】**

本発明は、端末装置がプログラム提供装置からアプリケーションを取得するプログラム取得方法、及び、プログラム取得方法に用いられる端末装置及びプログラム提供装置に関する。

20

**【背景技術】****【0002】**

従来、移動端末が、インターネット等のネットワークを介して、コンテンツとしてのアプリケーションをプログラム提供装置から取得する技術が提供されている。最近では、移動端末が、上述のネットワークを介して、Java（登録商標）アプリケーションをプログラム提供装置から取得し、当該アプリケーションを実行するサービスが実現されている。

30

**【0003】**

ただし、アプリケーションを取得した移動端末が、当該アプリケーションを実行できることは保証されない。特に移動端末は、CPUやメモリ等の計算リソースが限られているという特徴を有するため、実行できるアプリケーションも大きく制限される。

**【0004】**

このため、移動端末が、ネットワークを介して、プログラム提供装置から取得したアプリケーションを当該移動端末で実行できると判断した場合にのみ、取得した当該アプリケーションを外部記憶装置に保存し、当該アプリケーションを実行する技術が提供されている（例えば、特許文献1）。

**【特許文献1】特開2004-5419号公報**

40

**【発明の開示】****【発明が解決しようとする課題】****【0005】**

しかしながら、従来の技術では、移動端末は、計算リソースが限られているにも関わらず、所望のアプリケーションの実行に必要なモジュールが既に取得されているか否かについて判断しなかったため、既に取得した複数のアプリケーションの実行に必要なモジュールを重複して取得することがあり、計算リソースを効率的に用いることができなかった。

**【0006】**

又、移動端末は、計算リソースが限られているため、複数のアプリケーションを同時に実行することが困難であった。

50

## 【0007】

そこで、本発明は、計算リソースを効率的に用いてアプリケーションの実行に必要なモジュールを取得すること、及び、計算リソースを効率的に用いて複数のアプリケーションを実行することを目的とする。

## 【課題を解決するための手段】

## 【0008】

本発明の第1の特徴は、プログラム提供装置からアプリケーションを取得する端末装置であって、アプリケーションに係る情報を示すアプリケーション情報を取得するアプリケーション情報取得部と、アプリケーション情報に基づいて、アプリケーションの実行に必要なモジュールが、既に取得されているか否かについて判断する取得判断部と、モジュールが取得されていないと判断された場合に、モジュールを取得するモジュール取得部とを備えることを要旨とする。

10

## 【0009】

かかる発明によれば、モジュール取得部が、アプリケーションの実行に必要なモジュールの内、取得されていないと判断されたモジュールを選択的に取得するため、端末装置は、モジュールを重複して取得することによる無駄な計算リソースの消費を防ぐことができる。よって、端末装置は、計算リソースを効率的に用いてアプリケーションの実行に必要なモジュールを取得することができる。

## 【0010】

又、本発明の第1の特徴において、特定の状態が生じたことを示すイベントを検知するイベント検知部と、アプリケーションとイベントの種別とを関連づけて管理するイベント管理部と、イベント管理部を参照して、イベント検知部によって検知されたイベントの種別に関連づけられているアプリケーションに対して、そのイベントを通知するイベント通知部とを備えるように構成されていてもよい。

20

## 【0011】

かかる発明によれば、イベント通知部が、検知されたイベントを当該イベントの種別に関連づけられているアプリケーションに通知するため、端末装置は、複数のアプリケーションを同時に実行する場合でも、検知したイベントを当該イベントの通知を必要としているアプリケーションに通知することができる。

## 【0012】

更に、本発明の第1の特徴において、アクティブなアプリケーションを管理するアクティブアプリケーション管理部と、イベント検知部によって、切り替えイベントが検知された場合に、アクティブなアプリケーションを変更するアクティブアプリケーション変更部とを備え、イベント通知部は、端末装置に対する入力操作があったことを示す入力イベントを、アクティブなアプリケーションに通知するように構成されていてもよい。

30

## 【0013】

かかる発明によれば、アクティブアプリケーション管理部が、アクティブなアプリケーションを管理し、切り替えイベントが検知された場合に、アクティブアプリケーション変更部が、アクティブなアプリケーションを変更する。そして、イベント通知部が、入力イベントをアクティブなアプリケーションに通知するため、端末装置は、複数のアプリケーションを実行する場合に、入力イベントの通知を受けるアクティブなアプリケーションを適切に管理することができる。

40

## 【0014】

本発明の第1の特徴において、アプリケーションに割り当てられるメモリの領域であるメモリ領域を管理するメモリ領域管理部と、メモリ領域管理部を参照して、アプリケーションによるメモリ領域への不正なアクセスを検出する不正アクセス検出部と、不正アクセス検出部によって、アプリケーションによる不正なアクセスが検出された場合に、アプリケーションを終了させる終了処理部とを備えるように構成されていてもよい。

## 【0015】

かかる発明によれば、終了処理部が、アプリケーションによるメモリ領域への不正なア

50

クセスが検出された場合に、そのアプリケーションを終了させるため、端末装置は、あるアプリケーションによるメモリ領域への不正なアクセスが他のアプリケーションに影響を与えることを防ぐことができる。

**【0016】**

本発明の第2の特徴は、アプリケーションを実行する端末装置であって、アプリケーション、又は、アプリケーションの実行に必要なモジュールが、メモリ上にロードされているか否かを管理するロード済みプログラム管理部と、ロード済みプログラム管理部を参照して、所望のモジュールがロードされていないと判断した場合に、当該モジュールをメモリ上にロードするロード部と、ロード部によってロードされたモジュールに含まれるクラスに基づいてインスタンスがメモリ上に生成されているか否かを管理する生成済みインスタンス管理部と、生成済みインスタンス管理部を参照して、所望のインスタンスが生成されていないと判断した場合に、インスタンスをメモリ上に生成するインスタンス生成部とを備えることを要旨とする。

10

**【0017】**

かかる発明によれば、ロード部が、所望のモジュールがメモリ上にロードされていないと判断された場合にそのモジュールをメモリ上にロードし、インスタンス生成部が、所望のインスタンスが生成されていないと判断した場合にそのインスタンスをメモリ上に生成するため、端末装置は、不要なモジュールをメモリにロードすることや不要なインスタンスをメモリ上に生成することを防ぐことができる。よって、端末装置は、計算リソースを効率的に用いて複数のアプリケーションを実行することができる。

20

**【0018】**

本発明の第3の特徴は、端末装置に対してアプリケーションを提供するプログラム提供装置であって、端末装置からのアプリケーション情報要求に応じて、アプリケーションに係る情報を示すアプリケーション情報を提供するアプリケーション情報提供部と、アプリケーション情報に基づいて、アプリケーションの実行に必要なモジュールが取得されていないと判断した端末装置からのモジュール要求に応じて、モジュールを提供するモジュール提供部とを備えることを要旨とする。

**【0019】**

本発明の第4の特徴は、端末装置が、プログラム提供装置からアプリケーションを取得するプログラム取得方法であって、プログラム提供装置が、端末装置からのアプリケーション情報要求に応じて、アプリケーションに係る情報を示すアプリケーション情報を提供する工程と、端末装置が、プログラム提供装置によって提供されたアプリケーション情報を取得する工程と、端末装置が、アプリケーション情報に基づいて、アプリケーションの実行に必要なモジュールが既に取得されているか否かについて判断する工程と、端末装置が、モジュールが取得されていないと判断した場合に、プログラム提供装置にモジュール要求を送信する工程と、プログラム提供装置が、モジュール要求に応じてモジュールを提供する工程とを備えることを要旨とする。

30

**【発明の効果】****【0020】**

以上説明したように、本発明によれば、計算リソースを効率的に用いてアプリケーションの実行に必要なモジュールを取得すること、及び、計算リソースを効率的に用いて複数のアプリケーションを実行することが可能となる。

40

**【発明を実施するための最良の形態】****【0021】**

(本発明の一実施形態に係るコンテンツダウンロードシステムの構成)

図1～図13を参照して、本実施形態に係るコンテンツダウンロードシステム1の構成について説明する。図1に示すように、コンテンツダウンロードシステム1は、ネットワーク100と、ゲートウェイ装置200と、コンテンツダウンロード装置300と、移動端末400とを備える。

**【0022】**

50

移動端末 400 は、プログラム提供装置であるコンテンツダウンロード装置 300 からアプリケーションを取得する端末装置である。又、コンテンツダウンロード装置 300 は、端末装置である移動端末 400 に対してアプリケーションを提供するプログラム提供装置である。ゲートウェイ装置 200 は、移動端末 400 と、インターネット等のネットワーク 100 との間の通信トラフィックを中継するものである。

【0023】

次に、コンテンツダウンロード装置 300 について詳細に説明する。図 2 に示すように、コンテンツダウンロード装置 300 は、通信部 310 と、アプリケーション情報データベース 320 と、コンテンツデータベース 330 と、アプリケーション情報提供部 340 と、アプリケーション提供部 350 と、共有モジュール提供部 360 と、全体管理部 370 とを備える。

10

【0024】

通信部 310 は、ゲートウェイ装置 200 とネットワーク 100 とを介して、移動端末 400 との間で通信するものである。

【0025】

アプリケーション情報データベース 320 は、アプリケーション毎に、アプリケーションに係る情報を示すアプリケーション情報を保持するものである。具体的には、アプリケーション情報データベース 320 は、図 3 に示すように、アプリケーション情報 600 を保持する。

【0026】

アプリケーション情報 600 は、アプリケーションに係る情報として、アプリケーション実行情報 A と、共有モジュール情報 B, C とを含む。アプリケーション実行情報 A は、アプリケーション名と、URL (Uniform Resource Locator) と、ファイルサイズと、RAM (Random Access Memory) サイズとを含む。尚、図 3 は、アプリケーション 490 a に係るアプリケーション情報を示すものとする。

20

【0027】

ここで、URL は、コンテンツダウンロード装置 300 上で、アプリケーションが格納されている場所を示す情報である。ファイルサイズは、コンテンツダウンロード装置 300 によってファイルとして格納されているアプリケーションのバイト数を示す情報である。

30

【0028】

尚、アプリケーション実行情報 A は、ファイルサイズと RAM サイズの他にも、CPU の処理速度等の計算リソースに関する情報を含んでもよい。

【0029】

又、共有モジュール情報は、当該アプリケーションが共有モジュールを必要としない場合は存在せず、当該アプリケーションが複数の共有モジュールを必要とする場合は複数存在する。図 3 の例では、アプリケーション 490 a が 2 つの共有モジュールを必要とすることが分かる。

40

【0030】

各共有モジュール情報は、共有モジュール名と、URL と、ファイルサイズと、RAM サイズと、元 URL とを含む。ここで、元 URL は、共有モジュールの提供元であるコンテンツダウンロード装置上で、当該共有モジュールが格納されている場所を示す情報である。このため、共有モジュールが、提供元のコンテンツダウンロード装置からコピーされてコンテンツダウンロード装置 300 に格納されている場合には、元 URL と URL とは異なる。元 URL は、比較する共有モジュール情報を識別するための URN (Uniform Resource Name) として用いられる。

【0031】

尚、アプリケーション情報を取得した移動端末 400 が、共有モジュールの提供元のコ

50

コンテンツダウンロード処理部からではなく、URLに基づいてコンテンツダウンロード装置300から共有モジュールを取得することにより、当該共有モジュールの提供元であるコンテンツダウンロード装置の負荷が軽減される。

【0032】

コンテンツデータベース330は、アプリケーション及びアプリケーションの実行に必要な共有モジュールをコンテンツとして格納するものである。

【0033】

アプリケーションは、特定の機能を提供するソフトウェアであり、例えば、メモ帳ソフト、スケジュール管理ソフト、ゲームソフト等を含む。又、アプリケーションは、移動端末400における基本的な機能を提供するネイティブアプリケーションであってもよいし、オブジェクト指向言語であるJava（登録商標）によって記述されたアプリケーションであってもよい。ネイティブアプリケーションには、例えば、電話の発着信機能やメールの送受信機能を提供するアプリケーションや、OSに相当するアプリケーション等が含まれる。

10

【0034】

ただし、本実施形態において、コンテンツダウンロード装置300から提供されるコンテンツは、オブジェクト指向言語によって記述されているものとする。尚、アプリケーションは、例えば、当該アプリケーションを圧縮したJARファイルとしてコンテンツデータベース330に格納されている。

【0035】

アプリケーション情報提供部340は、移動端末400から、通信部310を介して取得したアプリケーション情報要求に応じて、アプリケーション情報を提供するものである。

20

【0036】

具体的には、アプリケーション情報提供部340は、アプリケーション情報要求に応じて、アプリケーション情報データベース320から所望のアプリケーション情報を取得し、取得したアプリケーション情報を移動端末400に提供するように構成されている。ここで、アプリケーション情報要求は、URLやアプリケーション名やアプリケーションID等のアプリケーションを特定することができる情報を用いて、所望のアプリケーションを指定している。

30

【0037】

アプリケーション提供部350は、アプリケーション情報に基づいて、アプリケーションを実行可能と判断した移動端末400からのアプリケーション要求に応じてアプリケーションを提供するものである。

【0038】

具体的には、アプリケーション提供部350は、移動端末400から取得したアプリケーション要求に応じて、コンテンツデータベース330から所望のアプリケーションを取得し、取得したアプリケーションを移動端末400に提供するように構成されている。ここで、アプリケーション要求は、URLやアプリケーション名やアプリケーションID等のアプリケーションを特定することができる情報を用いて、所望のアプリケーションを指定している。

40

【0039】

共有モジュール提供部360は、アプリケーション情報に基づいて、対応するアプリケーションの実行に必要な共有モジュールが取得されていないと判断した移動端末400からの共有モジュール要求に応じて共有モジュールを提供するモジュール提供部である。

【0040】

具体的には、共有モジュール提供部360は、移動端末400から取得した共有モジュール要求に応じて、コンテンツデータベース330から所望の共有モジュールを取得し、取得した共有モジュールを移動端末400に提供するように構成されている。ここで、共有モジュール要求は、元URLや共有モジュール名や共有モジュールID等の共有モジュ

50

ールを特定することができる情報を用いて、所望の共有モジュールを指定している。

【0041】

全体管理部370は、各構成要素310～360を制御するものである。

【0042】

次に、移動端末400について詳細に説明する。図4に示すように、移動端末400は、入力部410と、表示部420と、通信部430と、コンテンツダウンロード処理部440と、アプリケーション起動処理部450と、イベント処理部460と、コンテンツ別メモリ領域管理部470と、メモリ領域解放処理部480とを備える。又、図4の例では、アプリケーション490a, 490bが、移動端末400上で実行されているものとする。

10

【0043】

入力部410は、移動端末400へのユーザの入力操作を受け付けた場合に、入力イベントを生成し、生成した入力イベントをイベント処理部460に提供するものである。ここで、イベントとは特定の状態が生じたことを示すものである。入力イベントとは、移動端末400から入力操作があったことを示すイベントであり、移動端末400上のボタン等が押下されたことを示す「Key」イベントや、指定された時刻になったことや、指定された時間が経過したことを示す「Timer」や、アプリケーションによって表示されている画面のサイズが変更されたこと等を示す「Component」等が含まれる。

【0044】

又、イベントには入力イベントの他、ネイティブアプリケーションに配送され、他のアプリケーションの実行より優先して処理されるネイティブイベントや、アクティブなアプリケーションを変更する必要があることを示す切り替えイベントや、特定のアプリケーションをアクティブにする、即ちアクティブ化する必要があることを示すアクティブ化イベントや、特定のアプリケーションを非アクティブにする、即ち非アクティブ化する必要があることを示す非アクティブ化イベントや、アプリケーションを終了させる必要があることを示す終了イベント等が含まれる。

20

【0045】

尚、アクティブなアプリケーションとは移動端末400からの入力操作が反映されるアプリケーションをいう。例えば、Windows（登録商標）上で複数のウィンドウが表示されている場合に、最前面に表示されているウィンドウを表示するアプリケーションが、アクティブなアプリケーションである。本実施形態では、最大で1のアプリケーションがアクティブなアプリケーションとなる。

30

【0046】

表示部420は、アプリケーション490a, 490bの動作の状態をユーザに対して表示するものである。通信部430は、ネットワーク100とゲートウェイ装置200とを介して、コンテンツダウンロード装置300と通信するものである。

【0047】

コンテンツダウンロード処理部440は、コンテンツダウンロード装置300からコンテンツを取得するものである。本実施形態では、コンテンツダウンロード装置300からコンテンツを取得することをダウンロードという。アプリケーション起動処理部450は、ダウンロードしたアプリケーションを実行して、起動するものである。イベント処理部460は、イベントをアプリケーションに配送するものである。

40

【0048】

コンテンツ別メモリ領域管理部470は、アプリケーションに割り当てられるメモリの領域であるメモリ領域を管理するメモリ領域管理部である。又、コンテンツ別メモリ領域管理部470は、アプリケーションが必要とする共有モジュールに割り当てられるメモリ領域も管理するものである。

【0049】

具体的には、コンテンツ別メモリ領域管理部470は、図5に示すように、アプリケーション起動処理部450から取得したコンテンツのコンテンツ名と、当該コンテンツに割

50

り当てられるメモリ領域の情報とを保持する。図5の例では、使用メモリ領域が、関連づけられるコンテンツ名のコンテンツに割り当てられるメモリ領域の情報である。

【0050】

又、コンテンツ別メモリ領域管理部470は、メモリ領域解放処理部480から取得した情報削除要求に応じて、情報削除要求に含まれるアプリケーション名又は共有モジュール名に係る情報を削除する。ここで、情報削除要求は、メモリ領域解放処理部480が、情報削除要求に含まれるアプリケーション名又は共有モジュール名に係る情報の削除を求めたものである。

【0051】

メモリ領域解放処理部480は、コンテンツ別メモリ領域管理部470に基づいて、アプリケーションによるメモリ領域への不正なアクセスを検出する不正アクセス検出部である。

10

【0052】

具体的には、メモリ領域解放処理部480は、アプリケーションによるメモリアccessを監視することによって、コンテンツ別メモリ領域管理部470によって管理される当該アプリケーションの使用メモリ領域を超えたメモリ領域へのアクセスを、不正アクセスとして検出する。

【0053】

又、メモリ領域解放処理部480は、アプリケーションによるメモリ領域への不正なアクセスが検出された場合に、当該アプリケーションを終了させる終了処理部としても機能するものである。更に、メモリ領域解放処理部480は、イベント配送処理部465から、終了させるアプリケーションのアプリケーション名を含むアプリケーション終了イベントを取得した場合にも、当該アプリケーションを終了させるものである。

20

【0054】

具体的には、メモリ領域解放処理部480は、終了させるアプリケーションに割り当てられたメモリ領域を解放するものである。メモリ領域解放処理部480は、コンテンツ別メモリ領域管理部470を参照し、終了させるアプリケーションの使用メモリ領域によって示されるメモリ領域を解放する。これにより、アプリケーションに含まれるクラスに基づいて生成されたインスタンスは消滅し、アプリケーションが終了する。

【0055】

又、メモリ領域解放処理部480は、終了させたアプリケーションによって使用されていた共有モジュールであって、他のアプリケーションによって使用されていない共有モジュールに割り当てられたメモリ領域も解放するものである。

30

【0056】

具体的には、メモリ領域解放処理部480は、コンテンツダウンロード処理部440から、共有モジュール検索要求に対する応答である共有モジュール名を取得する。ここで、共有モジュール検索要求は、終了させたアプリケーションによって使用されていた共有モジュールであって、他のアプリケーションによって使用されていない共有モジュールの共有モジュール名を要求するものである。

【0057】

メモリ領域解放処理部480は、コンテンツ別メモリ領域管理部470を参照し、取得した共有モジュール名に対応する使用メモリ領域によって示されるメモリ領域を解放するように構成されている。

40

【0058】

全体管理部500は、各構成要素410～480と、実行中のアプリケーション490a, 490bとを制御するものである。

【0059】

以下、コンテンツダウンロード処理部440と、アプリケーション起動処理部450と、イベント処理部460の各構成について詳細に説明する。

【0060】

50

まず、コンテンツダウンロード処理部 440 について詳細に説明する。図 6 に示すように、コンテンツダウンロード処理部 440 は、コンテンツダウンロード処理決定部 441 と、アプリケーション情報ダウンロード部 442 と、アプリケーション情報解析部 443 と、アプリケーション情報記憶部 444 と、記憶コンテンツ管理部 445 と、コンテンツダウンロード部 446 と、コンテンツ記憶部 447 とを備える。

【0061】

コンテンツダウンロード処理決定部 441 は、所望のアプリケーションをダウンロードするか否かを決定し、当該アプリケーションをダウンロードすると決定した場合に、当該アプリケーションによって必要とされる共有モジュールをダウンロードするか否かを決定するものである。

10

【0062】

具体的には、コンテンツダウンロード処理決定部 441 は、ウェブブラウザ等のアプリケーション 490b 等から取得した取得要求に応じて、記憶コンテンツ管理部 445 を参照し、所望のアプリケーションがダウンロードされているか否かについて判断する。

【0063】

コンテンツダウンロード処理決定部 441 は、所望のアプリケーションがダウンロードされていると判断した場合に、所望のアプリケーションをダウンロードしないと決定するように構成されている。ここで、取得要求は、特定のアプリケーションを取得することを要求するものである。

【0064】

又、コンテンツダウンロード処理決定部 441 は、アプリケーション情報解析部 443 から取得したアプリケーション情報 600 に基づいて、該当するアプリケーションを実行可能か否かについて判断する。そして、コンテンツダウンロード処理決定部 441 は、所望のアプリケーションを実行可能でないと判断した場合にも、当該アプリケーションをダウンロードしないと決定するものである。

20

【0065】

具体的には、コンテンツダウンロード処理決定部 441 は、取得要求によって指定されるアプリケーションに対応するアプリケーション情報をアプリケーション情報ダウンロード部 442 から取得する。コンテンツダウンロード処理決定部 441 は、取得したアプリケーション情報をアプリケーション情報解析部 443 に入力することによってアプリケーション情報の解析結果であるアプリケーション情報 600 を取得する。コンテンツダウンロード処理決定部 441 は、アプリケーション情報解析部 443 から取得したアプリケーション情報 600 に基づいて、当該アプリケーションを実行可能か否かについて判断する。

30

【0066】

例えば、コンテンツダウンロード処理決定部 441 は、図 3 に示すアプリケーション実行情報 A に含まれるファイルサイズと、移動端末 400 のフラッシュメモリの空き容量とを比較し、ファイルサイズの方が大きい場合に、アプリケーション 490a を実行可能でないと判断する。或いは、コンテンツダウンロード処理決定部 441 は、アプリケーション実行情報 A に含まれる RAM サイズと、移動端末 400 において利用可能な RAM の容量を比較し、RAM サイズの方が大きい場合に、アプリケーション 490a を実行可能でないと判断する。

40

【0067】

コンテンツダウンロード処理決定部 441 は、所望のアプリケーションを実行可能でないと判断した場合に、所望のアプリケーションをダウンロードしないと決定するように構成されている。

【0068】

又、コンテンツダウンロード処理決定部 441 は、アプリケーション情報に基づいて、アプリケーションの実行に必要な共有モジュールが、既にダウンロードされているか否かについて判断する取得判断部としても機能する。

50

## 【 0 0 6 9 】

具体的には、コンテンツダウンロード処理決定部 4 4 1 は、アプリケーション 4 9 0 a を実行可能と判断した場合に、取得したアプリケーション情報 6 0 0 に含まれる共有モジュール情報 B , C を取得する。コンテンツダウンロード処理決定部 4 4 1 は、共有モジュール情報 B , C に含まれる元 URL に基づいて、記憶コンテンツ管理部 4 4 5 を参照し、アプリケーションの実行に必要な全ての共有モジュールが既にダウンロードされているか否かについて判断する。

## 【 0 0 7 0 】

コンテンツダウンロード処理決定部 4 4 1 は、実行可能と判断したアプリケーションによって必要とされる共有モジュールがダウンロードされていないと判断した場合に、当該共有モジュールをダウンロードする必要があると判断するように構成されている。

10

## 【 0 0 7 1 】

又、コンテンツダウンロード処理決定部 4 4 1 は、メモリ領域解放処理部 4 8 0 から取得した共有モジュール検索要求に応じて、対応するアプリケーションによって使用されていた共有モジュールであって、他のアプリケーションによって使用されていない共有モジュールの共有モジュール名を提供するものである。

## 【 0 0 7 2 】

具体的には、コンテンツダウンロード処理決定部 4 4 1 は、共有モジュール検索要求に含まれるアプリケーション名に基づいて、アプリケーション情報記憶部 4 4 4 から対応するアプリケーション情報を取得する。コンテンツダウンロード処理決定部 4 4 1 は、取得したアプリケーション情報に含まれる共有モジュール情報を取得する。

20

## 【 0 0 7 3 】

コンテンツダウンロード処理決定部 4 4 1 は、イベント処理部 4 6 0 から、実行中アプリケーション名要求に対する応答である実行中のアプリケーションのアプリケーション名を取得する。ここで、実行中アプリケーション名要求は、アプリケーション実行状態管理部 4 6 3 によって管理されている、実行中のアプリケーションのアプリケーション名を要求するものである。

## 【 0 0 7 4 】

コンテンツダウンロード処理決定部 4 4 1 は、取得した各共有モジュール情報に含まれる元 URL と、取得したアプリケーション名とに基づいて、アプリケーション情報記憶部 4 4 4 を参照し、アプリケーション名に係るアプリケーション情報に、取得した共有モジュール情報と同一の共有モジュール情報が含まれているか否かについて判断する。

30

## 【 0 0 7 5 】

コンテンツダウンロード処理決定部 4 4 1 は、取得したアプリケーション名に係るアプリケーション情報に、取得した共有モジュール情報と同一の共有モジュール情報が含まれていない場合に、当該共有モジュール情報に係る共有モジュールが、他のアプリケーションによって使用されていないと判断する。コンテンツダウンロード処理決定部 4 4 1 は、当該共有モジュール情報に含まれる共有モジュール名を、共有モジュール検索要求に対する応答としてメモリ領域解放処理部 4 8 0 に提供するように構成されている。

## 【 0 0 7 6 】

アプリケーション情報ダウンロード部 4 4 2 は、アプリケーション情報を取得するアプリケーション情報取得部である。具体的には、アプリケーション情報ダウンロード部 4 4 2 は、コンテンツダウンロード処理決定部 4 4 1 からダウンロードしようとするアプリケーションに係る URL を取得する。アプリケーション情報ダウンロード部 4 4 2 は、取得した URL を含むアプリケーション情報要求を通信部 4 3 0 に提供し、アプリケーション情報要求に対する応答であるアプリケーション情報を取得するように構成されている。

40

## 【 0 0 7 7 】

アプリケーション情報解析部 4 4 3 は、コンテンツダウンロード処理決定部 4 4 1 を介してアプリケーション情報ダウンロード部 4 4 2 から取得したアプリケーション情報を解析し、図 3 に示される形式にフォーマットするものである。又、アプリケーション情報解

50

析部 443 は、フォーマットしたアプリケーション情報 600 を、解析結果としてコンテンツダウンロード処理決定部 441 に入力するものである。更に、アプリケーション情報解析部 443 は、フォーマットしたアプリケーション情報 600 を、アプリケーション情報記憶部 444 に格納するものである。

【0078】

アプリケーション情報記憶部 444 は、アプリケーション情報解析部 443 によって図 3 に示される形式にフォーマットされたアプリケーション情報 600 を記憶するものである。

【0079】

記憶コンテンツ管理部 445 は、移動端末 400 にダウンロードされたコンテンツの情報を管理するものである。具体的には、記憶コンテンツ管理部 445 は、図 7 に示すように、コンテンツダウンロード処理決定部 441 から取得した情報登録要求に応じて、情報登録要求に含まれるコンテンツ名と、コンテンツ種類と、URL 情報とを保持するように構成されている。

【0080】

ここで、コンテンツのコンテンツ種類は、当該コンテンツがアプリケーションであるか、又は、共有モジュールであるかを示す情報である。図 7 の例では、「Application」は、該当するコンテンツがアプリケーションであることを示し、「Shared Library」は、該当するコンテンツが共有モジュールであることを示す。又、アプリケーションに係る URL 情報は URL であり、共有モジュールに係る URL 情報は元 URL である。

【0081】

コンテンツダウンロード部 446 は、実行可能と判断された場合に、アプリケーションを取得する。又、コンテンツダウンロード部 446 は、共有モジュールがダウンロードされていないと判断された場合に、共有モジュールを取得するモジュール取得部としても機能する。即ち、コンテンツダウンロード部 446 は、ダウンロードする必要があると判断されたアプリケーション及び共有モジュールを選択的にダウンロードするものである。

【0082】

具体的には、コンテンツダウンロード部 446 は、コンテンツダウンロード処理決定部 441 から取得したコンテンツ種類が「Application」である場合に、併せて取得した URL を含むアプリケーション要求を通信部 430 に提供し、アプリケーション要求に対する応答であるアプリケーションを取得するように構成されている。

【0083】

一方、コンテンツダウンロード部 446 は、コンテンツダウンロード処理決定部 441 から取得したコンテンツ種類が「Shared Library」である場合に、併せて取得した URL を含む共有モジュール要求を通信部 430 に提供し、共有モジュール要求に対する応答である共有モジュールを取得するように構成されている。

【0084】

又、コンテンツダウンロード部 446 は、取得したアプリケーション又は共有モジュールをコンテンツ記憶部 447 に格納するものである。

【0085】

コンテンツ記憶部 447 は、コンテンツダウンロード部 446 から取得したコンテンツを記憶するものである。

【0086】

次に、アプリケーション起動処理部 450 について詳細に説明する。図 8 に示すように、アプリケーション起動部 450 は、コンテンツロード処理決定部 451 と、コンテンツロード処理部 452 と、ロード済みコンテンツ管理部 453 と、生成済みインスタンス管理部 454 と、インスタンス生成処理決定部 455 と、インスタンス生成処理部 456 とを備える。

【0087】

10

20

30

40

50

ここで、本実施形態において、コンテンツをロードするとは、コンテンツに含まれるクラスをメモリ上にロードすることである。そして、ロードされたクラスに基づいてインスタンスを生成することにより、移動端末400は当該コンテンツを実行できる。

【0088】

コンテンツロード処理決定部451は、所望のアプリケーションを起動するために、アプリケーションに必要な共有モジュールをロードするか否かについて決定するものである。

【0089】

コンテンツロード処理決定部451は、端末装置400の各構成要素410～480又はアプリケーション490a, 490bから起動しようとするアプリケーションのアプリケーション名を含む起動要求を取得する。

10

【0090】

コンテンツロード処理決定部451は、イベント処理部460からアクティブアプリケーション要求に対する応答として、アクティブなアプリケーションのアプリケーション名を取得する。

【0091】

コンテンツロード処理決定部451は、取得したアプリケーション名を含む非アクティブ化イベントを生成し、生成した非アクティブ化イベントをイベント処理部460に提供することにより、対応するアプリケーションを非アクティブ化する。

【0092】

コンテンツロード処理決定部451は、コンテンツダウンロード処理部440から、アプリケーション情報要求に対する応答であるアプリケーション情報を取得する。コンテンツロード処理決定部451は、取得したアプリケーション情報から、共有モジュール情報を取得する。コンテンツロード処理決定部451は、取得した共有モジュール情報に含まれる元URLに基づいて、ロード済みコンテンツ管理部453を参照し、対応する共有モジュールがロードされているか否かについて判断する。

20

【0093】

コンテンツロード処理決定部451は、ロードされていないと判断した共有モジュール及びアプリケーションのロードをコンテンツロード処理部452に依頼し、コンテンツロード処理部452から、ロードした共有モジュール又はアプリケーションに含まれるクラスと当該クラスのクラス名とを取得する。

30

【0094】

コンテンツロード処理決定部451は、起動するアプリケーションのアプリケーション名を含むアクティブ化イベントを生成し、生成したアクティブ化イベントをイベント処理部460に提供することにより、起動するアプリケーションをアクティブ化する。コンテンツロード処理決定部451は、所望のアプリケーションを起動するように構成されている。

【0095】

コンテンツロード処理部452は、コンテンツロード処理決定部451が、ロード済みコンテンツ管理部453を参照して、コンテンツがロードされていないと判断した場合に、当該コンテンツをメモリ上にロードするものである。

40

【0096】

即ち、コンテンツロード処理決定部451及びコンテンツロード処理部452は、ロード済みコンテンツ管理部453を参照して所望のコンテンツがロードされていないと判断した場合に、そのコンテンツをメモリ上にロードするロード部として機能する。

【0097】

具体的には、コンテンツロード処理部452は、コンテンツロード処理決定部451から取得したコンテンツ名に対応するコンテンツをコンテンツダウンロード処理部440から取得し、取得したコンテンツをロードするように構成されている。

【0098】

50

ロード済みコンテンツ管理部 453 は、アプリケーション又はアプリケーションの実行に必要な共有モジュールが、メモリ上にロードされているか否かを管理するロード済みプログラム管理部である。

【0099】

具体的には、ロード済みコンテンツ管理部 453 は、図 9 に示すようにコンテンツロード処理決定部 451 から取得したコンテンツ名と URL 情報とを保持するように構成されている。尚、図 9 はアプリケーション 490a がロードされた後のロード済みコンテンツ管理部 453 を示すものである。

【0100】

生成済みインスタンス管理部 454 は、コンテンツロード処理部 452 によってロードされたコンテンツに含まれるクラスに基づいてインスタンスがメモリ上に生成されているか否かを管理するものである。

【0101】

具体的には、生成済みインスタンス管理部 454 は、図 10 に示すように、インスタンス生成処理決定部 455 から取得したインスタンス名とクラス名とを保持するように構成されている。図 10 の例では、例えば、クラス「Class1」に基づいて「Class Instance1」が生成されたことが分かる。

【0102】

又、生成済みインスタンス管理部 454 は、コンテンツロード処理決定部 451 を介して、メモリ領域解放処理部 480 から使用メモリ領域が開放されたコンテンツのコンテンツ名を含む削除要求を取得し、当該コンテンツに含まれるクラスに係る情報を削除するものである。ここで、生成済みインスタンス管理部 454 は、コンテンツロード処理決定部 451 を介してコンテンツダウンロード処理部 440 から対応するコンテンツを取得することにより、コンテンツに含まれるクラスのクラス名を判断するものである。

【0103】

インスタンス生成処理決定部 455 は、コンテンツロード処理決定部 451 から取得したクラス名に基づいて生成済みインスタンス管理部 454 を参照して、所望のインスタンスが生成されているか否かについて判断するものである。

【0104】

例えば、インスタンス生成処理決定部 455 は、クラス名として「Class1」を取得した場合に、図 10 に示される生成済みインスタンス管理部 454 を参照し、所望のインスタンスが生成されていると判断する。或いは、インスタンス生成処理決定部 455 は、クラス名として「Class5」を取得した場合に、図 10 に示される生成済みインスタンス管理部 454 を参照し、所望のインスタンスが生成されていないと判断する。

【0105】

又、インスタンス生成処理決定部 455 は、所望のインスタンスが生成されていないと判断した場合に、当該インスタンスを生成することを決定するものである。インスタンス生成処理決定部 455 は、生成済みインスタンス管理部 454 を参照して、既に生成されたインスタンスのインスタンス名と異なるインスタンス名のインスタンスを生成することを決定する。

【0106】

インスタンス生成処理部 456 は、インスタンス生成処理決定部 455 が、生成済みインスタンス管理部 454 を参照して、所望のインスタンスが生成されていないと判断した場合に、インスタンス生成処理決定部 455 から取得したクラスに基づいて、併せて取得したインスタンス名のインスタンスをメモリ上に生成するものである。

【0107】

即ち、インスタンス生成処理決定部 455 及びインスタンス生成処理部 456 は、生成済みインスタンス管理部 454 を参照して、所望のインスタンスが生成されていないと判断した場合に、そのインスタンスをメモリ上に生成するインスタンス生成部として機能する。

10

20

30

40

50

## 【0108】

又、インスタンス生成処理部456は、インスタンスを生成する際に割り当てられたメモリ領域の情報を、インスタンス生成処理決定部455を介してコンテンツロード処理決定部451に入力する。

## 【0109】

次に、イベント処理部460について詳細に説明する。図11に示すように、イベント処理部460は、イベント配送処理決定部461と、イベント発生検知部462と、アプリケーション実行状態管理部463と、アプリケーション別イベント管理部464と、イベント配送処理部465とを備える。

## 【0110】

イベント配送処理決定部461は、アプリケーション別イベント管理部464を参照して、イベント発生検知部462によって検知されたイベントの種別に関連づけられているアプリケーションを当該イベントの配送先に決定するものである。そして、イベント配送処理決定部461は、イベントと配送先の指定とをイベント配送処理部465に入力するものである。

## 【0111】

例えば、イベント処理決定部461は、タイマーイベントを取得した場合に、アプリケーション別イベント管理部464を参照して、タイマーイベントの配送を必要とするアプリケーションのアプリケーション名を取得し、当該アプリケーションを配送先に決定する。

## 【0112】

図13の例では、アプリケーション名が「App1」のアプリケーション490aと、アプリケーション名が「App2」のアプリケーション490bとがタイマーイベントの配送を必要とする。イベント配送処理決定部461は、タイマーイベントと、その配送先の指定である「App1」と「App2」とをイベント配送処理部465に入力するように構成されている。

## 【0113】

或いは、イベント配送処理決定部461は、入力イベントを取得した場合には、アクティブなアプリケーションが、当該入力イベントの配送を必要としているか否かについて判断し、必要としている場合には、当該入力イベントと、アクティブなアプリケーションのアプリケーション名とをイベント配送処理部465に入力する。

## 【0114】

具体的には、イベント配送処理決定部461は、アプリケーション実行状態管理部463を参照し、アクティブなアプリケーションのアプリケーション名を取得する。図12の例では、イベント配送処理決定部461は、「App2」を取得する。

## 【0115】

イベント配送処理決定部461は、取得した入力イベントの種別と、アクティブなアプリケーションのアプリケーション名に基づいて、アプリケーション別イベント管理部464を参照し、当該アプリケーション名に当該入力イベントの種別が関連づけられている場合に、アクティブなアプリケーションが当該入力イベントの配送を必要としていると判断する。

## 【0116】

例えば、図12及び図13の例では、イベント配送処理決定部461は、アプリケーション別イベント管理部464を参照し、「App2」に取得した入力イベントの種別が関連づけられているか否かについて判断する。「App2」に「Key」は関連づけられているが、「Component」は関連づけられていない。このため、イベント配送処理決定部461は、取得した入力イベントが「Key」であれば、アプリケーション490bが当該入力イベントの配送を必要としていると判断し、取得した入力イベントが「Component」であればアプリケーション490bが当該入力イベントの配送を必要としないと判断する。

10

20

30

40

50

## 【 0 1 1 7 】

そして、イベント配送処理決定部 4 6 1 は、配送を必要としていると判断した場合に、当該入力イベントと、その配送先の指定である「 A p p 2 」とをイベント配送処理部 4 6 5 に入力する。

## 【 0 1 1 8 】

又、イベント配送処理決定部 4 6 1 は、イベント発生検知部 4 6 2 からネイティブイベントや、切り替えイベントや、アプリケーション終了イベント等の特別なイベントを取得した場合には、当該イベントの種別に応じて、当該イベントを処理する。

## 【 0 1 1 9 】

イベント配送処理決定部 4 6 1 は、ネイティブイベントを取得した場合に、アクティブなアプリケーションを非アクティブ化する。具体的には、イベント配送処理決定部 4 6 1 は、アプリケーション実行状態管理部 4 6 3 を参照し、アクティブなアプリケーションが存在する場合に、当該アプリケーションのアプリケーション名を含む非アクティブ化イベントを生成する。イベント配送処理決定部 4 6 1 は、生成した非アクティブ化イベントと、当該イベントの配送先であるアプリケーション実行状態管理部 4 6 3 の指定とをイベント配送処理部 4 6 5 に入力する。

## 【 0 1 2 0 】

そして、イベント配送処理決定部 4 6 1 は、ネイティブアプリケーションからネイティブイベントの処理が終了したことの通知を取得することによってネイティブイベントの処理が終了したことを検知した場合に、非アクティブ化したアプリケーションをアクティブ化する。具体的には、イベント配送処理決定部 4 6 1 は、非アクティブ化したアプリケーションのアプリケーション名を含むアクティブ化イベントを生成する。イベント配送処理決定部 4 6 1 は、生成したアクティブ化イベントと、当該イベントの配送先であるアプリケーション実行状態管理部 4 6 3 の指定とをイベント配送処理部 4 6 5 に入力する。

## 【 0 1 2 1 】

又、イベント配送処理決定部 4 6 1 は、アプリケーション切り替えイベントを取得した場合に、アクティブなアプリケーションを変更する。即ち、イベント配送処理決定部 4 6 1 は、イベント検知部によってアプリケーション切り替えイベントが検知された場合に、アクティブなアプリケーションを変更するアクティブアプリケーション変更部である。

## 【 0 1 2 2 】

具体的には、イベント配送処理決定部 4 6 1 は、アプリケーション実行状態管理部 4 6 3 を参照し、アクティブなアプリケーションが存在する場合に、当該アプリケーションのアプリケーション名を含む非アクティブ化イベントを生成する。イベント配送処理決定部 4 6 1 は、生成した非アクティブ化イベントと、当該イベントの配送先であるアプリケーション実行状態管理部 4 6 3 の指定とをイベント配送処理部 4 6 5 に入力する。

## 【 0 1 2 3 】

そして、イベント配送処理決定部 4 6 1 は、切り替えイベントに含まれるアプリケーション名を含むアクティブ化イベントを生成し、当該アクティブ化イベントと、その配送先であるアプリケーション実行状態管理部 4 6 3 の指定とをイベント配送処理部 4 6 5 に入力する。

## 【 0 1 2 4 】

或いは、イベント配送処理決定部 4 6 1 は、アプリケーション終了イベントを取得した場合に、当該アプリケーション終了イベントと、その配送先であるメモリ領域解放処理部 4 8 0 の指定とを、イベント配送処理部 4 6 5 に入力する。

## 【 0 1 2 5 】

更に、イベント配送処理決定部 4 6 1 は、取得した入力イベントを、アクティブなアプリケーションに配送する必要がない場合等、取得したイベントを処理する必要がない場合に、取得したイベントを破棄する。

## 【 0 1 2 6 】

イベント発生検知部 4 6 2 は、特定の状態が生じたことを示すイベントを移動端末 4 0

0内の各構成要素410～450, 470～480と、アプリケーション490a, 490bとから、イベントを取得することによって検知するイベント検知部である。

【0127】

アプリケーション実行状態管理部463は、アクティブなアプリケーションを管理するアクティブアプリケーション管理部である。具体的には、アプリケーション実行状態管理部463は、図12に示すように、イベント配送処理決定部461から取得したアプリケーション名とアプリケーション実行状態とを保持する。ここで、アプリケーション実行状態は、当該アプリケーションがアクティブであるか否かを示す情報である。

【0128】

図12の例では、「Active」は、該当するアプリケーションがアクティブであることを示し、「Non-active」は、該当するアプリケーションがアクティブでない、即ち、非アクティブであることを示す。 10

【0129】

アプリケーション実行状態管理部463は、イベント配送処理部465からアクティブ化イベントを取得した場合に、アクティブ化イベントに含まれるアプリケーション名に係る実行状態を「Active」に変更し、非アクティブ化イベントを取得した場合に、非アクティブ化イベントに含まれるアプリケーション名に係る実行状態を「Non-active」に変更する。

【0130】

更に、アプリケーション実行状態管理部463は、イベント配送処理決定部461を介して、メモリ領域解放処理部480から情報削除要求を取得した場合に、当該情報削除要求に含まれるアプリケーション名に係る情報を削除する。 20

【0131】

アプリケーション別イベント管理部464は、アプリケーションとイベントの種別とを関連づけて管理するイベント管理部である。具体的には、アプリケーション別イベント管理部464は、イベント配送処理決定部461を介してアプリケーション起動処理部450から取得したアプリケーション名と、イベントの種別であるイベント種別とを図13に示すように保持する。

【0132】

図13の例では、アプリケーション名が「App1」のアプリケーション490aは、「Key」、「Timer」、「Component」という種別のイベントの配送を必要とすることが分かる。 30

【0133】

又、アプリケーション別イベント管理部464は、イベント配送処理決定部461を介して、メモリ領域解放処理部480から情報削除要求を取得した場合に、当該情報削除要求に含まれるアプリケーション名に係る情報を削除する。

【0134】

イベント配送処理部465は、イベント配送処理決定部461から、イベントと当該イベントの配送先の指定を取得し、取得したイベントを指定される配送先に配送するものである。 40

【0135】

又、イベント配送処理決定部461によって取得されたイベントがタイマーイベントや入力イベント等である場合に、イベント配送処理部465は、イベント配送処理決定部461から、イベント発生検知部462によって検知されたイベントと、アプリケーション別イベント管理部464において、そのイベントの種別に関連づけられているアプリケーションのアプリケーション名とを取得する。そして、イベント配送処理部465は、取得したイベントを、取得したアプリケーション名のアプリケーションに配送する。

【0136】

即ち、イベント配送処理決定部461及びイベント配送処理部465は、アプリケーション別イベント管理部464を参照して、イベント発生検知部462によって検知された 50

イベントの種別に関連づけられているアプリケーションに対して当該イベントを通知するイベント通知部として機能する。このように、本実施形態では、イベントを通知することを、イベントを配送するという。

【0137】

更に、イベント配送処理決定部461によって取得されたイベントが入力イベントである場合に、イベント配送処理部465は、イベント配送処理決定部461からアクティブなアプリケーションのアプリケーション名を取得する。そして、イベント配送処理部465は、取得したイベントを、取得したアプリケーション名のアプリケーションに対して配送する。

【0138】

即ち、イベント配送処理部465は、入力イベントをアクティブなアプリケーションに配送するものである。

【0139】

(本発明の一実施形態に係るコンテンツダウンロード方法)

図14は、本実施形態に係るコンテンツダウンロードシステム1において、移動端末400が、所望のアプリケーションをコンテンツダウンロード装置300からダウンロードするシーケンスの一例を示す図である。

【0140】

ステップ101において、移動端末400上の通信部430が、所望のアプリケーションに係るアプリケーション情報を要求するアプリケーション情報要求をコンテンツダウンロード装置300上の通信部310に送信し、コンテンツダウンロード装置300の通信部310が、当該アプリケーション情報要求を受信する。

【0141】

ステップ102において、コンテンツダウンロード装置300上の通信部310が、アプリケーション情報要求に応じて、所望のアプリケーション情報を移動端末400上の通信部430に送信し、移動端末400上の通信部430が、当該アプリケーション情報を受信する。

【0142】

ステップ103において、アプリケーション情報解析部443が、取得したアプリケーション情報を解析し、図3に示すアプリケーション情報600をコンテンツダウンロード処理決定部441に入力する。

【0143】

ステップ104において、コンテンツダウンロード処理決定部441が、解析されたアプリケーション情報600に基づいて所望のアプリケーションがダウンロードされているか否かについて判断する。コンテンツダウンロード処理決定部441は、所望のアプリケーションがダウンロードされていると判断した場合に、当該アプリケーションをダウンロードしないと決定し、ステップ105～ステップ114を行わない。

【0144】

ステップ105において、コンテンツダウンロード処理決定部441が、アプリケーション情報に基づいて所望のアプリケーションを実行可能か否かについて判断する。コンテンツダウンロード処理決定部441は、所望のアプリケーションを実行可能でないと判断した場合に、当該アプリケーションをダウンロードしないと決定し、ステップ106～ステップ114を行わない。

【0145】

ステップ106において、アプリケーション情報解析部443が、コンテンツダウンロード処理決定部441に入力したアプリケーション情報600をアプリケーション情報記憶部444に格納する。

【0146】

ステップ107において、コンテンツダウンロード処理決定部441が、アプリケーション情報に含まれる共有モジュール情報を取得する。

10

20

30

40

50

## 【0147】

ステップ108において、コンテンツダウンロード処理決定部441が、共有モジュール情報に基づいて、アプリケーションの実行に必要な全ての共有モジュールがダウンロードされているか否かについて判断する。

## 【0148】

ステップ108において必要な共有モジュールがダウンロードされていないと判断された場合に、ステップ109において、移動端末400上の通信部430が、コンテンツダウンロード装置300上の通信部310にアプリケーション要求と共有モジュール要求とを送信し、コンテンツダウンロード装置300上の通信部310が、当該アプリケーション要求と当該共有モジュール要求とを受信する。

10

## 【0149】

ステップ110において、コンテンツダウンロード装置300上の通信部310が、アプリケーション要求と共有モジュール要求とに応じて、アプリケーションと共有モジュールとを移動端末400上の通信部430に送信し、移動端末400上の通信部430が、当該アプリケーションと当該共有モジュールとを受信する。

## 【0150】

ステップ108において全ての共有モジュールがダウンロードされていると判断された場合に、ステップ111において、移動端末400上の通信部430が、アプリケーション要求をコンテンツダウンロード装置300上の通信部310に送信し、コンテンツダウンロード装置300上の通信部310が、アプリケーション要求を受信する。

20

## 【0151】

ステップ112において、コンテンツダウンロード装置300上の通信部310が、アプリケーション要求に応じて、アプリケーションを移動端末400上の通信部430に送信し、移動端末400上の通信部430が、当該アプリケーションを受信する。

## 【0152】

ステップ113において、コンテンツダウンロード部446が、ダウンロードしたアプリケーションと共有モジュールとをコンテンツとしてコンテンツ記憶部447に格納する。

## 【0153】

ステップ114において、コンテンツダウンロード処理決定部441が、ダウンロードされたコンテンツに係る情報を記憶コンテンツ管理部445に入力し、登録する。

30

## 【0154】

図15は、本実施形態に係るコンテンツダウンロードシステム1において、移動端末400が、所望のアプリケーションを起動するシーケンスの一例を示す図である。

## 【0155】

ステップ201において、コンテンツロード処理決定部451が、イベント処理部460からアクティブアプリケーション要求に対する応答を取得することにより、アクティブなアプリケーションがあるか否かについて判断する。

## 【0156】

コンテンツロード処理決定部451は、ステップ201においてアクティブなアプリケーションがあると判断した場合に、ステップ202において、アクティブなアプリケーションを非アクティブ化する。

40

## 【0157】

ステップ203において、コンテンツロード処理決定部451が、コンテンツダウンロード処理部440から取得したアプリケーション情報に含まれる共有モジュール情報を取得する。

## 【0158】

ステップ204において、コンテンツロード処理決定部451が、ロード済みコンテンツ管理部453を参照し、各共有モジュール情報に係る共有モジュールがロードされているか否かについて判断する。

50

## 【0159】

ステップ204において共有モジュールがロードされていないと判断された場合に、ステップ205において、コンテンツロード処理部452が、当該共有モジュールをロードする。

## 【0160】

ステップ206において、コンテンツロード処理決定部451が、ロードされた共有モジュールに係る情報をロード済みコンテンツ管理部453に、ロード済みコンテンツとして登録する。

## 【0161】

ステップ207において、インスタンス生成処理決定部455が、生成済みインスタンス管理部454を参照して、共有モジュールに含まれるクラスに基づいて、インスタンスが生成されているか否かについて判断する。

## 【0162】

ステップ207において所望のインスタンスが生成されていないと判断された場合に、ステップ208において、インスタンス生成処理部456が、共有モジュールに含まれるクラスに基づいて、当該インスタンスをメモリ上に生成する。

## 【0163】

ステップ209において、インスタンス生成処理決定部455が、生成したインスタンスに係る情報を生成済みインスタンス管理部454に入力し、生成済みインスタンスとして登録する。

## 【0164】

ステップ210において、コンテンツロード処理決定部451が、ロードした共有モジュールの共有モジュール名と、当該共有モジュールに割り当てられたメモリ領域の情報をコンテンツ別メモリ領域管理部470に提供し、使用メモリ領域として登録する。

## 【0165】

ステップ211において、コンテンツロード処理決定部451が、全ての共有モジュールについて処理を行ったか否かについて判断し、全ての共有モジュールについて処理を行っていない場合と判断した場合に、ステップ204に戻って処理していない共有モジュールに対してステップ204～ステップ210の処理を行う。

## 【0166】

ステップ212において、コンテンツロード処理部452が、所望のアプリケーションをロードする。

## 【0167】

ステップ213において、コンテンツロード処理決定部451が、ロードしたアプリケーションに係る情報をロード済みコンテンツ管理部453に、ロード済みコンテンツとして登録する。

## 【0168】

ステップ214において、インスタンス生成処理部456が、アプリケーションに含まれるクラスに基づいて、インスタンスをメモリ上に生成する。

## 【0169】

ステップ215において、インスタンス生成処理決定部455が、生成したインスタンスに係る情報を生成済みインスタンス管理部454に入力し、生成済みインスタンスとして登録する。

## 【0170】

ステップ216において、コンテンツロード処理決定部451が、起動するアプリケーションのアプリケーション名と、当該アプリケーションが配送を必要とするイベントの種類とをイベント処理部460に提供し、アプリケーション別イベント管理部464に登録する。

## 【0171】

ステップ217において、コンテンツロード処理決定部451が、起動するアプリケー

10

20

30

40

50

ションのアプリケーション名と、生成されたインスタンスに割り当てられたメモリ領域の情報とをコンテンツ別メモリ領域管理部 470 に、使用メモリ領域として登録する。

【0172】

ステップ 218 において、コンテンツロード処理決定部 451 が、起動するアプリケーションをアクティブ化する。

【0173】

ステップ 219 において、コンテンツロード処理決定部 451 が、所望のアプリケーションを起動する。

【0174】

図 16 は、本実施形態に係るコンテンツダウンロードシステム 1 において、移動端末 400 がイベントを検知し、処理するシーケンスの一例を示す図である。 10

【0175】

ステップ 301 において、イベント発生検知部 462 が、イベントを検知する。

【0176】

ステップ 302 において、イベント配送処理決定部 461 が、検知されたイベントがネイティブイベントであると判断した場合に、ステップ 303 において、アクティブなアプリケーションを非アクティブ化する。

【0177】

ステップ 304 において、イベント配送処理決定部 461 が、ネイティブイベントの処理が終了したことを検知する。ステップ 305 において、イベント配送処理決定部 461 が、非アクティブ化したアプリケーションをアクティブ化する。 20

【0178】

ステップ 306 において、イベント配送処理決定部 461 が、検知されたイベントがアプリケーション切り替えイベントであると判断した場合に、ステップ 307 においてアクティブなアプリケーションを変更する。

【0179】

ステップ 308 において、イベント配送処理決定部 461 が、検知されたイベントがタイマーイベントであると判断した場合に、ステップ 309 において、タイマーイベントの配送を必要とするアプリケーションを検索する。ステップ 310 において、イベント配送処理部 465 が、タイマーイベントの配送を必要とするアプリケーションの全てにタイマーイベントを配送する。 30

【0180】

本実施例においては、ステップ 310 までに処理されなかったイベントは主に入力イベントであるため、ステップ 311 において、イベント配送処理決定部 461 が、アクティブなアプリケーションを検索する。ステップ 312 において、イベント配送処理決定部 461 が、アクティブなアプリケーションが、検知されたイベントの配送を必要とすると判断した場合に、ステップ 313 において、イベント配送処理部 465 が、アクティブなアプリケーションに当該イベントを配送する。

【0181】

ステップ 314 において、イベント配送処理決定部 461 が、検知されたイベントがアプリケーション終了イベントであると判断した場合に、ステップ 315 において、メモリ領域解放処理部 480 が、アプリケーション終了処理を行う。 40

【0182】

ステップ 316 において、イベント配送処理決定部 461 が、ステップ 301 ~ ステップ 315 において処理されなかったイベントを破棄する。

【0183】

図 17 は、本実施形態に係るコンテンツダウンロードシステム 1 において、移動端末 400 がアプリケーションを終了させた後に行う終了処理のシーケンスの一例を示す図である。

【0184】

ステップ401において、アプリケーション実行状態管理部463が、メモリ領域解放処理部480から取得した情報削除要求に含まれるアプリケーション名に係る情報を削除する。

【0185】

ステップ402において、アプリケーション別イベント管理部464が、メモリ領域解放処理部480から取得した情報削除要求に含まれるアプリケーション名に係る情報を削除する。

【0186】

ステップ403において、メモリ領域解放処理部480が、終了させるアプリケーションの使用メモリ領域を取得し、ステップ404において取得した使用メモリ領域によって示されるメモリ領域を解放する。

10

【0187】

ステップ405において、メモリ領域解放処理部480が、終了させたアプリケーションが使用していた共有モジュールに係る共有モジュール情報を取得する。

【0188】

ステップ406において、メモリ領域解放処理部480が、取得した共有モジュール情報に基づいて、終了させたアプリケーションによって使用されていた共有モジュールを、他のアプリケーションが使用しているか否かについて判断し、他のアプリケーションが使用していると判断した場合にステップ410に進む。

【0189】

ステップ407において、メモリ領域解放処理部480が、他のアプリケーションに使用されていない共有モジュールの使用メモリ領域を取得し、ステップ408において、取得した使用メモリ領域によって示されるメモリ領域を解放する。

20

【0190】

ステップ409において、コンテンツ別メモリ領域管理部470が、メモリ領域解放処理部480から取得した情報削除要求に含まれるアプリケーション名と共有モジュール名とに係る情報を削除し、情報を更新する。

【0191】

ステップ410において、生成済みインスタンス管理部454が、解放されたメモリ領域を使用していたコンテンツに含まれるクラスに係る情報を削除する。

30

【0192】

(変更例)

なお、本発明は上記実施形態に限定されるものではなく、種々の変更が可能である。例えば、1の共有モジュールが複数のクラスを含む場合に、アプリケーション情報は当該アプリケーション情報に係るアプリケーションによって必要とされるクラスのクラス名や当該クラスに基づいて生成されるインスタンス名を含んでもよい。

【0193】

或いは、本実施形態においては、アプリケーションが必要とする共有モジュールに含まれるクラスのクラス名に基づいて、インスタンス生成処理決定部455が、所望のインスタンスが生成されているか否かについて判断することとしている。しかし、インスタンス生成処理決定部455は、生成しようとするインスタンスのインスタンス名も取得し、クラス名とインスタンス名に基づいて、所望のインスタンスが生成されているか否かについて判断してもよい。

40

【0194】

これによれば、同一のインスタンス名のインスタンスを必要とするアプリケーションのみが同一のインスタンスを共有できる。このため、他のアプリケーションによって利用されているインスタンスと同一のクラスに基づいて生成されるインスタンスであっても、異なる動作をするインスタンスを必要とするアプリケーションに係るアプリケーション情報に異なるインスタンス名を含むことにより、当該アプリケーションは、異なるインスタンスを利用することができる。

50

## 【0195】

又、移動端末400は、アプリケーションが発する例外を処理する例外処理部を備えてもよい。ここで、例外とは、整数が0で除されたことや、必要なファイルが見つからなかったこと等の問題が生じたことを示すものである。

## 【0196】

更には、全体管理部500等が、不要になったインスタンスに割り当てられていたメモリ領域を解放する機能を備えていれば、メモリ領域解放処理部480は、終了させたアプリケーションや不要になった共有モジュールに割り当てられたメモリ領域を解放する必要がない。このような機能には例えばJava(登録商標)の仮想マシンによるガーベジコレクションが含まれる。又、全体管理部500等が、アプリケーションによる不正なメモリ領域へのアクセスを検出する機能も備えていれば、コンテンツ別メモリ領域管理部470は必要ない。

10

## 【0197】

又、移動端末400がコンテンツダウンロード装置300から取得するアプリケーション及び共有モジュールはオブジェクト指向言語によって記述されていなくてもよい。これによれば、アプリケーション及び共有モジュールは、メモリ上にロードすることによって実行可能となり、インスタンスの生成処理は不要となる。

## 【0198】

(本実施形態に係るコンテンツダウンロードシステムの効果)

このようなコンテンツダウンロードシステム1によれば、コンテンツダウンロード部446が、アプリケーションの実行に必要な共有モジュールの内、ダウンロードされていないと判断された共有モジュールを選択的にダウンロードするため、移動端末400は、共有モジュールを重複して取得することによる無駄な計算リソースの消費を防ぐことができる。よって、移動端末400は、CPUやメモリ等の計算リソースを効率的に用いてアプリケーションの実行に必要な共有モジュールを取得することができる。

20

## 【0199】

又、コンテンツロード処理決定部451によって所望のコンテンツがロードされていないと判断された場合に、コンテンツロード処理部452が、当該コンテンツをメモリ上にロードし、インスタンス生成処理決定部455によって所望のインスタンスが生成されていないと判断した場合に、インスタンス生成処理部456が、当該インスタンスをメモリ上に生成するため、移動端末400は、不要なコンテンツをメモリにロードすることや不要なインスタンスをメモリ上に生成することを防ぐことができる。

30

## 【0200】

このように、移動端末400上で実行される複数のアプリケーションが、共有モジュールやインスタンスを共有することができ、移動端末400は、計算リソースを効率的に用いて複数のアプリケーションを実行することができる。

## 【0201】

又、コンテンツロード処理部452は、他のアプリケーションが実行されている際であっても、アプリケーションをロードするため、例えば、複数のアプリケーションを同一の仮想マシン上で動作させる場合であっても、コンテンツロード処理部452は、他のアプリケーションが実行されている際にダウンロードされたアプリケーションをメモリ上にロードすることができる。

40

## 【0202】

更に、イベント配送処理決定部461が、検知されたイベントがタイマーイベントや入力イベント等である場合に、当該イベントを当該イベントの種別に関連づけられているアプリケーションに配送することを決定し、イベント配送処理部465が、決定されたアプリケーションに当該イベントを配送するため、移動端末400は、複数のアプリケーションを、例えば同一の仮想マシン上で同時に実行する場合でも、検知したイベントを当該イベントの配送を必要としているアプリケーションに配送することができる。

## 【0203】

50

又、アプリケーション実行状態管理部 463 が、アクティブなアプリケーションを管理し、切り替えイベントが検知された場合に、イベント配送処理決定部 461 がアクティブなアプリケーションを変更する。そして、イベント配送処理部 465 が、入力イベントをアクティブなアプリケーションに配送するため、移動端末 400 は、複数のアプリケーションを同時に実行する場合に、入力イベントが配送されるアクティブなアプリケーションを適切に管理することができる。

【0204】

尚、本実施例においては、最大で1のアプリケーションのみがアクティブなアプリケーションであるため、唯一のアプリケーションをユーザが同時に操作できる一方、非アクティブなアプリケーションは、タイマーイベントを利用することにより、必要な処理をすることができ

10

【0205】

更に、メモリ領域解放処理部 480 が、アプリケーションによるメモリ領域への不正なアクセスが検出された場合に、当該アプリケーションを終了させるため、移動端末 400 は、あるアプリケーションによるメモリ領域への不正なアクセスが他のアプリケーションに影響を与えることを防ぐことができる。

【0206】

このため、移動端末 400 を提供する複数のベンダは、移動端末 400 上で、他のアプリケーションの実行を妨げる等の悪意のあるアプリケーションが実行されるような場合を考慮せずに、アプリケーションをダウンロードして、当該アプリケーションを実行できる環境を提供することができる。

20

【0207】

又、アプリケーションが Java (登録商標) アプリケーションである場合に、移動端末 400 は、複数の仮想マシン上で単独のアプリケーションを動作させるのではなく、同一の仮想マシン上で複数のアプリケーションを動作させることが可能になるため、移動端末 400 の少ない計算リソースを効率的に用いて複数のアプリケーションを実行することができる。

【図面の簡単な説明】

【0208】

【図1】本発明の実施の形態に係るコンテンツダウンロードシステムの構成を示す図である。

30

【図2】本発明の実施の形態に係るコンテンツダウンロード装置の構成を示す図である。

【図3】本発明の実施の形態に係るアプリケーション情報を示す図である。

【図4】本発明の実施の形態に係る移動端末の構成を示す図である。

【図5】本発明の実施の形態に係るコンテンツ別メモリ領域管理部を示す図である。

【図6】本発明の実施の形態に係るコンテンツダウンロード処理部の構成を示す図である。

【図7】本発明の実施の形態に係る記憶コンテンツ管理部を示す図である。

【図8】本発明の実施の形態に係るアプリケーション起動処理部の構成を示す図である。

【図9】本発明の実施の形態に係るロード済みコンテンツ管理部を示す図である。

40

【図10】本発明の実施の形態に係る生成済みインスタンス管理部を示す図である。

【図11】本発明の実施の形態に係るイベント処理部の構成を示す図である。

【図12】本発明の実施の形態に係るアプリケーション実行状態管理部を示す図である。

【図13】本発明の実施の形態に係るアプリケーション別イベント管理部を示す図である。

【図14】本発明の実施の形態に係るコンテンツダウンロード方法の手順を示す図である。

【図15】本発明の実施の形態に係るアプリケーション起動方法の手順を示す図である。

【図16】本発明の実施の形態に係るイベント処理方法の手順を示す図である。

【図17】本発明の実施の形態に係るアプリケーション終了処理の手順を示す図である。

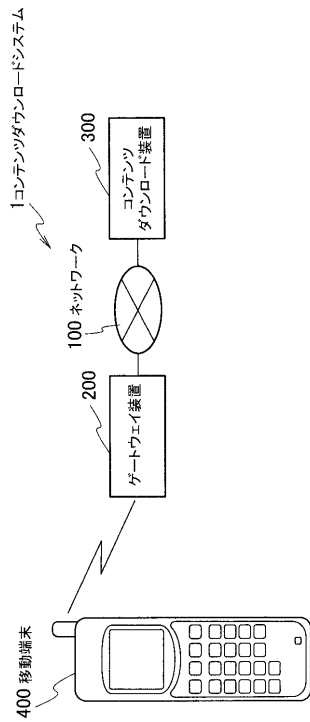
50

## 【符号の説明】

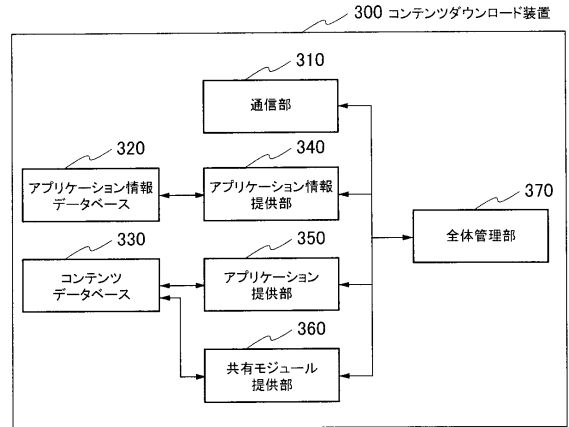
## 【0209】

1	コンテンツダウンロードシステム	
100	ネットワーク	
200	ゲートウェイ装置	
300	コンテンツダウンロード装置	
310	通信部	
320	アプリケーション情報データベース	
330	コンテンツデータベース	
340	アプリケーション情報提供部	10
350	アプリケーション提供部	
360	共有モジュール提供部	
370	全体管理部	
400	移動端末	
410	入力部	
420	表示部	
430	通信部	
440	コンテンツダウンロード処理部	
441	コンテンツダウンロード処理決定部	
442	アプリケーション情報ダウンロード部	20
443	アプリケーション情報解析部	
444	アプリケーション情報記憶部	
445	記憶コンテンツ管理部	
446	コンテンツダウンロード部	
447	コンテンツ記憶部	
450	アプリケーション起動処理部	
451	コンテンツロード処理決定部	
452	コンテンツロード処理部	
453	ロード済みコンテンツ管理部	
454	生成済みインスタンス管理部	30
455	インスタンス生成処理決定部	
456	インスタンス生成処理部	
460	イベント処理部	
461	イベント配送処理決定部	
462	イベント発生検知部	
463	アプリケーション実行状態管理部	
464	アプリケーション別イベント管理部	
465	イベント配送処理部	
470	コンテンツ別メモリ領域管理部	
480	メモリ領域解放処理部	40
490 a	アプリケーション	
490 b	アプリケーション	
500	全体管理部	
600	アプリケーション情報	

【 図 1 】



【 図 2 】

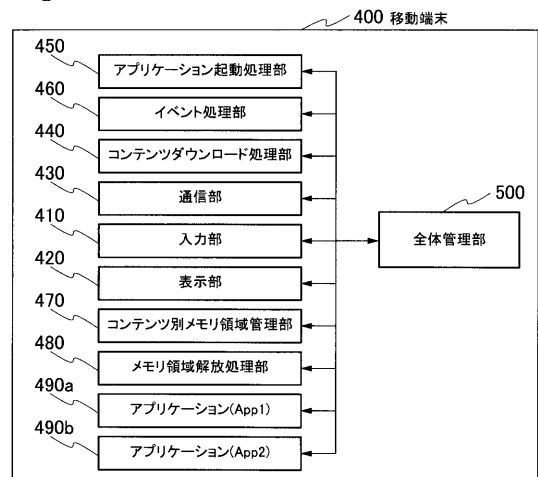


【 図 3 】

600 アプリケーション情報

A	アプリケーション名	App1
	URL	http://aa.com/App1
	ファイルサイズ	15924(Byte)
B	RAMサイズ	16384(Byte)
	共有モジュール名	SLibrary1
	URL	http://aa.com/SL1
C	ファイルサイズ	125626(Byte)
	元URL	http://bb.com/SL1
	RAMサイズ	131072(Byte)
	共有モジュール名	SLibrary2
	URL	http://aa.com/SL2
	ファイルサイズ	6264(Byte)
	元URL	http://cc.com/SL2
	RAMサイズ	6324(Byte)

【 図 4 】

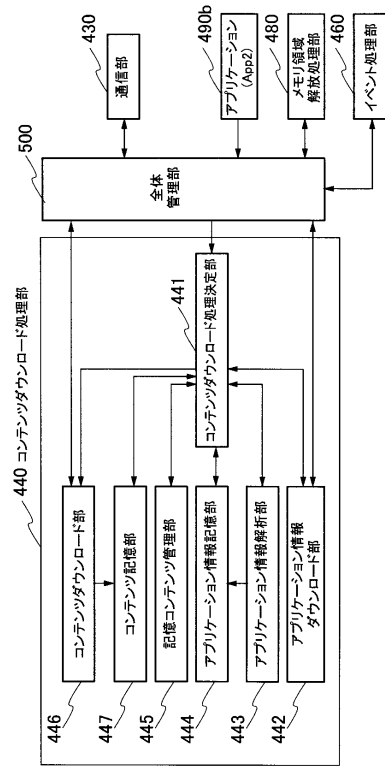


【 図 5 】

470 コンテンツ別メモリ領域管理部

コンテンツ名	使用メモリ領域
App1	0x...0x...0x...0x...
SLibrary1	0x...0x...0x...
SLibrary2	0x...0x...0x...
...	...

【 図 6 】

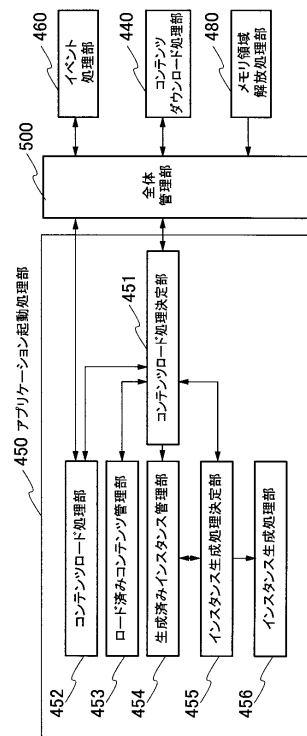


【 図 7 】

445 記憶コンテンツ管理部

コンテンツ名	コンテンツ種類	URL情報
App1	Application	http://aa.com/App1
App2	Application	http://aa.com/App2
SLibrary1	Shared Library	http://bb.com/SL1
SLibrary2	Shared Library	http://cc.com/SL2
...	...	...

【 図 8 】



【 図 9 】

453 ロード済みコンテンツ管理部

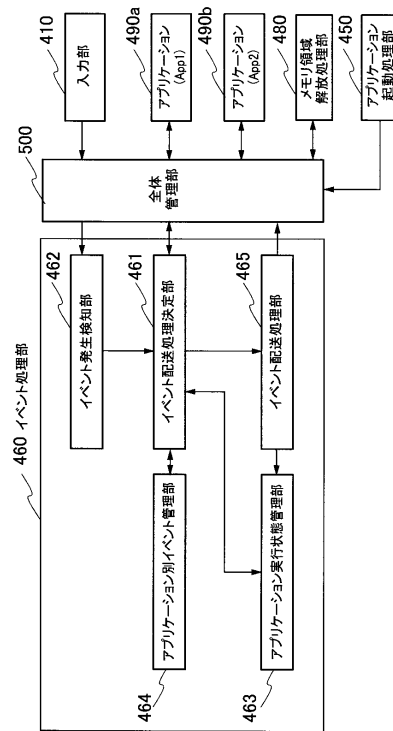
コンテンツ名	URL情報
App1	http://aa.com/App1
SLibrary1	http://bb.com/SL1
SLibrary2	http://cc.com/SL2
⋮	⋮

【 図 10 】

454 生成済みインスタンス管理部

インスタンス名	クラス名
ClassInstance1	Class1
ClassInstance2	Class2
ClassInstance3	Class3
⋮	⋮

【 図 11 】

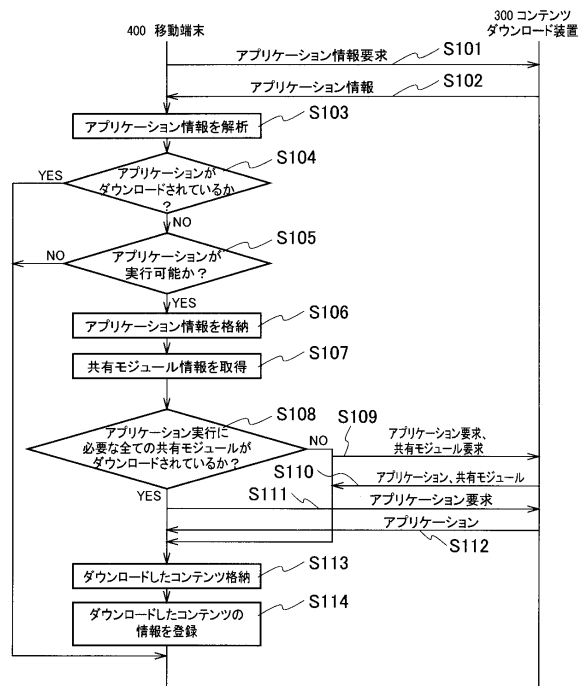


【 図 12 】

463 アプリケーション実行状態管理部

アプリケーション名	アプリケーション実行状態
App1	Non-active
App2	Active
⋮	⋮

【 図 14 】

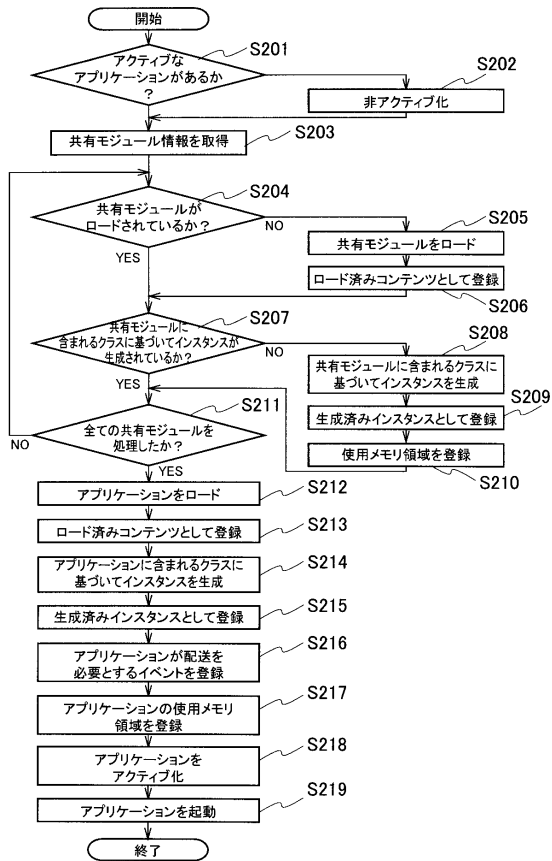


【 図 13 】

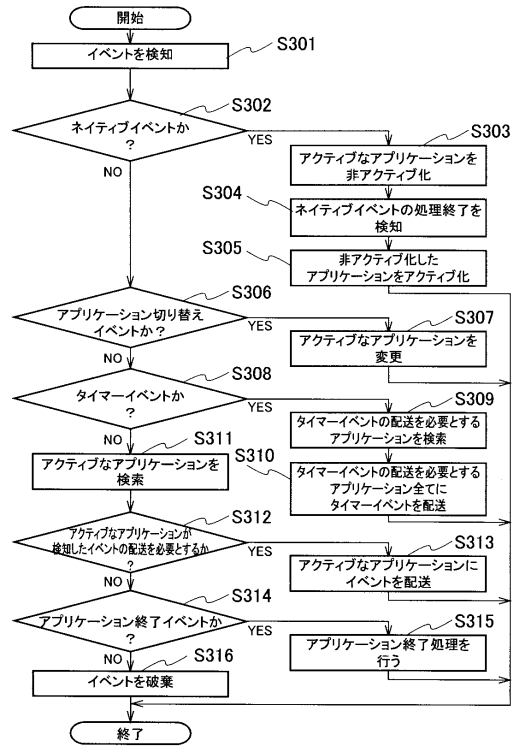
464 アプリケーション別イベント管理部

アプリケーション名	イベント種別
App1	Key,Timer,Component
App2	Key,Timer
⋮	⋮

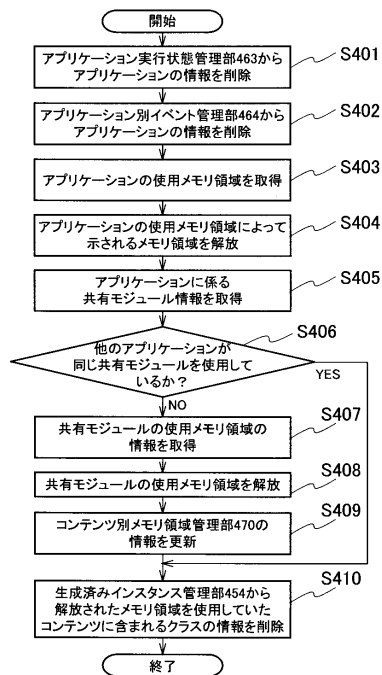
【 図 1 5 】



【 図 1 6 】



【 図 1 7 】



## フロントページの続き

(51)Int.Cl. <sup>7</sup>	F I	テーマコード(参考)
H 0 4 B 7/26	H 0 4 M 11/08	5 K 1 0 1
H 0 4 M 11/08	G 0 6 F 9/06	6 5 0 C
	H 0 4 B 7/26	M

(72)発明者 富岡 淳樹

東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内

(72)発明者 倉掛 正治

東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内

Fターム(参考) 5B017 AA01 CA01

5B042 GA36 GB09 JJ47 KK01

5B076 BB02 BB04 BB06 BB17

5B098 AA09 GA01 GC11

5K067 AA21 BB04 BB21 DD51 EE02 EE10 EE16 FF02 HH22 KK13

KK15

5K101 KK18 LL12 MM07 NN21