

(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2006-120056

(P2006-120056A)

(43) 公開日 平成18年5月11日(2006.5.11)

(51) Int.CI.	F 1	テーマコード (参考)
GO6F 12/00 (2006.01)	GO6F 12/00	512 5B075
GO6F 17/30 (2006.01)	GO6F 17/30	414A 5B082
	GO6F 17/30	419A

審査請求 未請求 請求項の数 23 O L (全 73 頁)

(21) 出願番号	特願2004-309439 (P2004-309439)	(71) 出願人	503003854 ヒューレット・パッカード デベロップメント カンパニー エル.ピー. アメリカ合衆国 テキサス州 77070 ヒューストン 20555 ステイト ハイウェイ 249
(22) 出願日	平成16年10月25日 (2004.10.25)	(74) 代理人	110000039 特許業務法人アイ・ピー・エス
		(72) 発明者	山本 昭夫 東京都杉並区高井戸東3丁目29番地21 号 日本ヒューレット・パッカード株式会社内

最終頁に続く

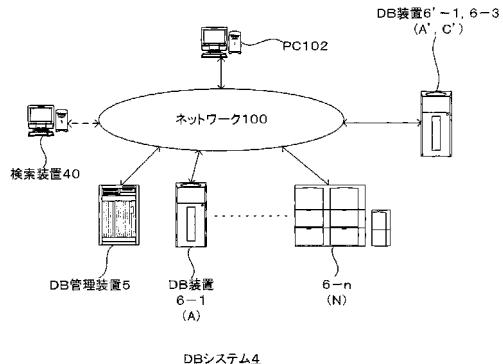
(54) 【発明の名称】データベースシステムおよびその方法

## (57) 【要約】

【課題】 DBシステムにおいて、各DB装置の負荷分散を行い、性能を保証する。

【解決手段】 DB装置6は、各エントリに対するアクセスの頻度およびアクセスに用いられた検索条件を、DB管理装置5に通知する。DB管理装置5は、各エントリに対するアクセスの頻度および検索条件を集計し、あるDB装置6のサブツリーに対するアクセスが大きい場合などに、DB装置6を制御して、サブツリーの新設を行い、この新設に合わせて、ディレクトリツリーを示す情報を変更する。DB管理装置5は、内容が変更された検索条件の変換に用いられる情報を、検索装置40に対して提供し、検索装置40は、検索者の検索操作に従つて、DB管理装置5から提供された情報を用いた変換を行い、DB装置6に対する検索を行う。

【選択図】図24



**【特許請求の範囲】****【請求項 1】**

1つ以上の関連ノードと、1つ以上の見出ノードとを関連付けてディレクトリツリーの形式のデータベースとするデータベースシステムであって、前記関連ノードそれぞれは、1つ以上の前記見出ノードと関連付けられ、前記見出ノードそれぞれには1つ以上の情報が属し、前記関連付けられた前記関連ノードと前記見出ノードとの間それぞれに対しては、これらの間の関連を示す関連属性が定義され、

前記関連ノードそれぞれの関連ノードエントリと、前記関連属性それぞれの関連属性エントリとを作成し、これらのエントリを、前記関連ノードと前記関連属性との関連に従って対応付けて、ディレクトリツリーを作成するディレクトリツリー作成手段と、10

前記見出ノードそれぞれに対応する情報を、前記作成された関連属性エントリに、前記見出ノードと前記関連属性との関連に従って対応付ける情報対応付手段と  
を有するデータベースシステム。

**【請求項 2】**

前記関連属性は、前記関連ノードと前記見出ノードとの間に定義される役割を示す  
請求項 1 に記載のデータベースシステム。

**【請求項 3】**

前記ディレクトリツリーは、1つ以上のサブツリーを含み、前記ディレクトリツリーおよび前記サブツリーそれぞれには、最上位のエントリが定義され、前記サブツリーそれぞれの最上位のエントリは、前記ディレクトリツリーおよび他の前記サブツリーのエントリのいずれかを参照し、20

前記ディレクトリツリーおよびサブツリーにおいて、前記関連ノードおよび前記見出ノードは、階層的に対応付けられる

請求項 1 または 2 に記載のデータベースシステム。

**【請求項 4】**

前記サブツリーそれぞれの最上位のエントリそれぞれと、これら最上位のエントリが参照する前記ディレクトリツリーまたは他の前記サブツリーに含まれるエントリとを対応付ける最上位エントリ対応付手段

をさらに有する請求項 3 に記載のデータベースシステム。

**【請求項 5】**

前記ディレクトリツリーおよび前記サブツリーのいずれかに従って、これらのディレクトリツリーのいずれかに含まれる見出ノードに対応する情報と、前記関連ノードに対応する情報と、前記見出ノードと前記関連ノードとの間に定義される役割とをそれぞれ記憶する1つ以上の第1のデータベース装置

を有する請求項 3 または 4 に記載のデータベースシステム。

**【請求項 6】**

ある第1のデータベース装置に記憶された情報を、他の第1のデータベース装置に転送して記憶させる情報転送手段

をさらに有する請求項 5 に記載のデータベースシステム。

**【請求項 7】**

前記情報転送手段は、前記第1のデータベース装置の動作状態に基づいて、ある第1のデータベース装置から、他の第1のデータベース装置に転送する情報を決める

請求項 6 に記載のデータベースシステム。

**【請求項 8】**

前記情報転送手段は、前記ディレクトリツリーまたは前記サブツリーに対するアクセス状態に基づいて、ある第1のデータベース装置から、他の第1のデータベース装置に転送する情報を決める

請求項 6 に記載のデータベースシステム。

**【請求項 9】**

前記第1のデータベース装置の1つ以上に記憶された情報の複製を記憶する第2のデータベース装置

10

20

30

40

50

データベース装置

をさらに有し、

前記最上位エントリ対応付手段は、前記第1のデータベース装置の最上位エントリと、前記第1のデータベース装置に記憶された情報の複製を記憶する前記第2のデータベース装置とをさらに対応付ける

請求項5～8いずれかに記載のデータベースシステム。

【請求項10】

前記ディレクトリツリーおよび前記サブツリーそれぞれには、さらに1つ以上の前記サブツリーが含まれることがあり、

前記ディレクトリツリーおよび前記サブツリーそれぞれから、さらに前記サブツリーを作成するサブツリー作成手段

をさらに有する請求項9に記載のデータベースシステム。

【請求項11】

前記ディレクトリツリーおよび前記サブツリーそれぞれには、前記関連ノードおよび見出しノードの分類に用いられる分類情報が定義され、

前記前記ディレクトリツリーおよび前記サブツリーそれぞれと、これらのディレクトリツリーそれぞれに定義された分類情報とを対応付ける分類情報対応付手段

をさらに有する請求項9または10に記載のデータベースシステム。

【請求項12】

前記記憶された見出ノードに対応する情報を検索する検索装置

20

をさらに有する請求項11に記載のデータベースシステム。

【請求項13】

前記検索装置は、

検索条件を受け入れる検索条件受入手段と、

前記受け入れられた検索条件に対応する前記分類情報と対応付けられた前記ディレクトリツリーまたは前記サブツリーを検索するディレクトリツリー検索手段と、

前記検索の結果として得られたディレクトリツリーに含まれる見出ノードに対応する情報を検索する情報検索手段と

を有する請求項12に記載のデータベースシステム。

【請求項14】

1つ以上の関連ノードと、1つ以上の見出ノードとを関連付けてディレクトリツリーのデータベースにおける情報管理方法であって、前記関連ノードそれぞれは、1つ以上の前記見出ノードと関連付けられ、前記見出ノードそれぞれには1つ以上の情報が属し、前記関連付けられた前記関連ノードと前記見出ノードとの間それぞれに対しては、これらの間の関連を示す関連属性が定義され、

前記関連ノードそれぞれの関連ノードエントリと、前記関連属性それぞれの関連属性エントリとを作成し、これらのエントリを、前記関連ノードと前記関連属性との関連に従って対応付けて、ディレクトリツリーを作成し、

前記見出ノードそれぞれに対応する情報を、前記作成された関連属性エントリに、前記見出ノードと前記関連属性との関連に従って対応付ける

情報管理方法。

【請求項15】

前記ディレクトリツリーは、1つ以上のサブツリーを含み、前記ディレクトリツリーおよび前記サブツリーそれぞれには、最上位のエントリが定義され、前記サブツリーそれぞれの最上位のエントリは、前記ディレクトリツリーおよび他の前記サブツリーのエントリのいずれかを参照し、

前記ディレクトリツリーおよび前記サブツリーそれぞれには、さらに1つ以上の前記サブツリーが含まれることがあり、

前記ディレクトリツリーおよび前記サブツリーそれぞれから、さらに前記サブツリーを作成する

30

40

50

請求項 1 4 に記載の情報管理方法。

**【請求項 1 6】**

前記ディレクトリツリーは、1つ以上のサブツリーを含み、前記ディレクトリツリーおよび前記サブツリーそれぞれには、最上位のエントリが定義され、前記サブツリーそれぞれの最上位のエントリは、前記ディレクトリツリーおよび他の前記サブツリーのエントリのいずれかを参照し、

前記ディレクトリツリーおよび前記サブツリーのいずれかに従って、これらのディレクトリツリーのいずれかに含まれる見出ノードに対応する情報と、前記関連ノードに対応する情報と、前記見出ノードと前記関連ノードとの間に定義される役割とをそれぞれ記憶し、

前記記憶された見出ノードに対応する情報を検索する

請求項 1 4 に記載の情報管理方法。

**【請求項 1 7】**

検索条件を受け入れ、

前記受け入れられた検索条件に対応する前記分類情報と対応付けられた前記ディレクトリツリーまたは前記サブツリーを検索し、

前記検索の結果として得られたディレクトリツリーに含まれる見出ノードに対応する情報を検索する

請求項 1 6 に記載の情報管理方法。

**【請求項 1 8】**

前記ディレクトリツリーおよび前記サブツリーのいずれかに従って、これらのディレクトリツリーのいずれかに含まれる見出ノードに対応する情報と、前記関連ノードに対応する情報と、前記見出ノードと前記関連ノードとの間に定義される役割とを、複数の第1のデータベース装置に記憶し、

ある第1のデータベース装置に記憶された情報を、他の第1のデータベース装置に転送して記憶させる

請求項 1 4 に記載の情報管理方法。

**【請求項 1 9】**

コンピュータを含み、1つ以上の関連ノードと、1つ以上の見出ノードとを関連付けてディレクトリツリーのデータベースとするデータベースシステムのプログラムであって、前記関連ノードそれぞれは、1つ以上の前記見出ノードと関連付けられ、前記見出ノードそれぞれには1つ以上の情報が属し、前記関連付けられた前記関連ノードと前記見出ノードとの間それぞれに対しては、これらの間の関連を示す関連属性が定義され、

前記関連ノードそれぞれの関連ノードエントリと、前記関連属性それぞれの関連属性エントリとを作成し、これらのエントリを、前記関連ノードと前記関連属性との関連に従って対応付けて、ディレクトリツリーを作成するステップと、

前記見出ノードそれぞれに対応する情報を、前記作成された関連属性エントリに、前記見出ノードと前記関連属性との関連に従って対応付けるステップと

をコンピュータに実行させるプログラム。

**【請求項 2 0】**

前記ディレクトリツリーは、1つ以上のサブツリーを含み、前記ディレクトリツリーおよび前記サブツリーそれぞれには、最上位のエントリが定義され、前記サブツリーそれぞれの最上位のエントリは、前記ディレクトリツリーおよび他の前記サブツリーのエントリのいずれかを参照し、

前記ディレクトリツリーおよび前記サブツリーそれぞれには、さらに1つ以上の前記サブツリーが含まれることがあり、

前記ディレクトリツリーおよび前記サブツリーそれぞれから、さらに前記サブツリーを作成するステップ

をさらにコンピュータに実行させる請求項 1 9 に記載のプログラム。

**【請求項 2 1】**

10

20

30

40

50

前記ディレクトリツリーは、1つ以上のサブツリーを含み、前記ディレクトリツリーおよび前記サブツリーそれぞれには、最上位のエントリが定義され、前記サブツリーそれぞれの最上位のエントリは、前記ディレクトリツリーおよび他の前記サブツリーのエントリのいずれかを参照し、

前記ディレクトリツリーおよび前記サブツリーのいずれかに従って、これらのディレクトリツリーのいずれかに含まれる見出ノードに対応する情報と、前記関連ノードに対応する情報と、前記見出ノードと前記関連ノードとの間に定義される役割とをそれぞれ記憶するステップと、

前記記憶された見出ノードに対応する情報を検索するステップと  
をさらにコンピュータに実行させる請求項19に記載のプログラム。

10

#### 【請求項22】

検索条件を受け入れるステップと、

前記受け入れられた検索条件に対応する前記分類情報と対応付けられた前記ディレクトリツリーまたは前記サブツリーを検索するステップと、

前記検索の結果として得られたディレクトリツリーに含まれる見出ノードに対応する情報を検索するステップと

をさらにコンピュータに実行させる請求項21に記載のプログラム。

#### 【請求項23】

前記ディレクトリツリーおよび前記サブツリーのいずれかに従って、これらのディレクトリツリーのいずれかに含まれる見出ノードに対応する情報と、前記関連ノードに対応する情報と、前記見出ノードと前記関連ノードとの間に定義される役割とを、複数の第1のデータベース装置に記憶するステップと、

ある第1のデータベース装置に記憶された情報を、他の第1のデータベース装置に転送して記憶させるステップと

をさらにコンピュータに実行させる請求項19に記載のプログラム。

20

#### 【発明の詳細な説明】

##### 【技術分野】

##### 【0001】

本発明は、データを記憶する方法及び記憶するデータ構造に関する。

より詳細には、相互に関連性を持った一連のデータを記憶する方法及び記憶するデータ構造に関する。

##### 【背景技術】

##### 【0002】

関連するデータを記憶し、記憶されたこのようなデータを検索するために、リレーションナル・データベースと呼ばれる装置が用いられている。

また、例えば、非特許文献1および特許文献1,2は、関連するデータを記憶する方法、および、記憶されたこのようなデータを検索する方法を開示する。

しかしながら、リレーションナル・データベースにおいて、完成されたデータベースの構造(スキーマ)を変更することは、容易ではない。

また、これらの文献に開示されている方法では、データの記述内容が複雑になり、また、データの表記・格納方法が一意的ではない。

また、これらの文献に開示されている方法では、データベースが複数のサーバを用いて実現されているときに、これら複数のサーバ間ににおける負荷分散が容易ではない。

【非特許文献1】The Associative Model of Data White Paper (Lazy Software, September 2000)

【特許文献1】特開2001-209647号公報

【特許文献2】国際公開公報WO 00/29980

#### 【発明の開示】

#### 【発明が解決しようとする課題】

##### 【0003】

40

50

本発明は、上述のような背景からなされたものであり、データベースの構造を変更することなく、様々な種類の情報の追加が容易なデータベースシステムおよびその方法を提供することを目的とする。

また、本発明は、情報の記述が簡単で、情報の表記および格納方法が一意的なデータベースシステムおよびその方法を提供することを目的とする。

また、本発明は、データベースが複数の装置に分散されているときに、これら複数の装置間における負荷分散が容易なデータベースシステムおよびその方法を提供することを目的とする。

また、本発明は、このように登録された情報を検索するために適したデータベースシステムおよびその方法を提供することを目的とする。 10

#### 【課題を解決するための手段】

##### 【0004】

上記目的を達成するために、本発明にかかるデータベースシステムは、1つ以上の関連ノードと、1つ以上の見出ノードとを関連付けてディレクトリツリーの形式のデータベースとするデータベースシステムであって、前記関連ノードそれぞれは、1つ以上の前記見出ノードと関連付けられ、前記見出ノードそれぞれには1つ以上の情報が属し、前記関連付けられた前記関連ノードと前記見出ノードとの間それぞれに対しては、これらの間の関連を示す関連属性が定義され、前記関連ノードそれぞれの関連ノードエントリと、前記関連属性それぞれの関連属性エントリとを作成し、これらのエントリを、前記関連ノードと前記関連属性との関連に従って対応付けて、ディレクトリツリーを作成するディレクトリツリー作成手段と、前記見出ノードそれぞれに対応する情報を、前記作成された関連属性エントリに、前記見出ノードと前記関連属性との関連に従って対応付ける情報対応付手段とを有する。 20

##### 【0005】

好適には、前記関連属性は、前記関連ノードと前記見出ノードとの間に定義される役割を示す。

##### 【0006】

好適には、前記ディレクトリツリーは、1つ以上のサブツリーを含み、前記ディレクトリツリーおよび前記サブツリーそれぞれには、最上位のエントリが定義され、前記サブツリーそれぞれの最上位のエントリは、前記ディレクトリツリーおよび他の前記サブツリーのエントリのいずれかを参照し、前記ディレクトリツリーおよびサブツリーにおいて、前記関連ノードおよび前記見出ノードは、階層的に対応付けられる。 30

##### 【0007】

好適には、前記サブツリーそれぞれの最上位のエントリそれぞれと、これら最上位のエントリが参照する前記ディレクトリツリーまたは他の前記サブツリーに含まれるエントリとを対応付ける最上位エントリ対応付手段をさらに有する。

##### 【0008】

好適には、前記ディレクトリツリーおよび前記サブツリーのいずれかに従って、これらのディレクトリツリーのいずれかに含まれる見出ノードに対応する情報と、前記関連ノードに対応する情報と、前記見出ノードと前記関連ノードとの間に定義される役割とをそれぞれ記憶する1つ以上の第1のデータベース装置を有する。 40

##### 【0009】

好適には、ある第1のデータベース装置に記憶された情報を、他の第1のデータベース装置に転送して記憶させる情報転送手段をさらに有する。

##### 【0010】

好適には、前記情報転送手段は、前記第1のデータベース装置の動作状態に基づいて、ある第1のデータベース装置から、他の第1のデータベース装置に転送する情報を決める。 50

##### 【0011】

好適には、前記情報転送手段は、前記ディレクトリツリーまたは前記サブツリーに対す

るアクセス状態に基づいて、ある第1のデータベース装置から、他の第1のデータベース装置に転送する情報を決める。

【0012】

好適には、前記第1のデータベース装置の1つ以上に記憶された情報の複製を記憶する第2のデータベース装置をさらに有し、前記最上位エントリ対応付手段は、前記第1のデータベース装置の最上位エントリと、前記第1のデータベース装置に記憶された情報の複製を記憶する前記第2のデータベース装置とをさらに対応付ける。

【0013】

好適には、前記ディレクトリツリーおよび前記サブツリーそれぞれには、さらに1つ以上の前記サブツリーが含まれることがあり、前記ディレクトリツリーおよび前記サブツリーそれだから、さらに前記サブツリーを作成するサブツリー作成手段をさらに有する。

【0014】

好適には、前記ディレクトリツリーおよび前記サブツリーそれぞれには、前記関連ノードおよび見出しノードの分類に用いられる分類情報が定義され、前記前記ディレクトリツリーおよび前記サブツリーそれと、これらのディレクトリツリーそれぞれに定義された分類情報とを対応付ける分類情報対応付手段をさらに有する。

【0015】

好適には、前記記憶された見出ノードに対応する情報を検索する検索装置をさらに有する。

【0016】

好適には、前記検索装置は、検索条件を受け入れる検索条件受入手段と、前記受け入れられた検索条件に対応する前記分類情報と対応付けられた前記ディレクトリツリーまたは前記サブツリーを検索するディレクトリツリー検索手段と、前記検索の結果として得られたディレクトリツリーに含まれる見出ノードに対応する情報を検索する情報検索手段とを有する。

【0017】

また、本発明にかかる情報管理方法は、1つ以上の関連ノードと、1つ以上の見出ノードとを関連付けてディレクトリツリーのデータベースにおける情報管理方法であって、前記関連ノードそれは、1つ以上の前記見出ノードと関連付けられ、前記見出ノードそれには1つ以上の情報が属し、前記関連付けられた前記関連ノードと前記見出ノードとの間それに対しては、これらの間の関連を示す関連属性が定義され、前記関連ノードそれとの関連ノードエントリと、前記関連属性それとの関連属性エントリとを作成し、これらのエントリを、前記関連ノードと前記関連属性との関連に従って対応付けて、ディレクトリツリーを作成し、前記見出ノードそれに対応する情報を、前記作成された関連属性エントリに、前記見出ノードと前記関連属性との関連に従って対応付ける。

【0018】

好適には、前記ディレクトリツリーは、1つ以上のサブツリーを含み、前記ディレクトリツリーおよび前記サブツリーそれぞれには、最上位のエントリが定義され、前記サブツリーそれぞれの最上位のエントリは、前記ディレクトリツリーおよび他の前記サブツリーのエントリのいずれかを参照し、前記ディレクトリツリーおよび前記サブツリーそれぞれには、さらに1つ以上の前記サブツリーが含まれることがあり、前記ディレクトリツリーおよび前記サブツリーそれだから、さらに前記サブツリーを作成する。

【0019】

好適には、前記ディレクトリツリーは、1つ以上のサブツリーを含み、前記ディレクトリツリーおよび前記サブツリーそれぞれには、最上位のエントリが定義され、前記サブツリーそれぞれの最上位のエントリは、前記ディレクトリツリーおよび他の前記サブツリーのエントリのいずれかを参照し、前記ディレクトリツリーおよび前記サブツリーのいずれかに従って、これらのディレクトリツリーのいずれかに含まれる見出ノードに対応する情報を、前記関連ノードに対応する情報を、前記見出ノードと前記関連ノードとの間に定義される役割とをそれぞれ記憶し、前記記憶された見出ノードに対応する情報を検索する。

## 【0020】

好適には、検索条件を受け入れ、前記受け入れられた検索条件に対応する前記分類情報と対応付けられた前記ディレクトリツリーまたは前記サブツリーを検索し、前記検索の結果として得られたディレクトリツリーに含まれる見出ノードに対応する情報を検索する。

## 【0021】

好適には、前記ディレクトリツリーおよび前記サブツリーのいずれかに従って、これらのディレクトリツリーのいずれかに含まれる見出ノードに対応する情報と、前記関連ノードに対応する情報と、前記見出ノードと前記関連ノードとの間に定義される役割とを、複数の第1のデータベース装置に記憶し、ある第1のデータベース装置に記憶された情報を、他の第1のデータベース装置に転送して記憶させる。

10

## 【0022】

また、本発明にかかるプログラムは、コンピュータを含み、1つ以上の関連ノードと、1つ以上の見出ノードとを関連付けてディレクトリツリーのデータベースとするデータベースシステムのプログラムであって、前記関連ノードそれぞれは、1つ以上の前記見出ノードと関連付けられ、前記見出ノードそれぞれには1つ以上の情報が属し、前記関連付けられた前記関連ノードと前記見出ノードとの間それぞれに対しては、これらの間の関連を示す関連属性が定義され、前記関連ノードそれぞれの関連ノードエントリと、前記関連属性それぞれの関連属性エントリとを作成し、これらのエントリを、前記関連ノードと前記関連属性との関連に従って対応付けて、ディレクトリツリーを作成するステップと、前記見出ノードそれぞれに対応する情報を、前記作成された関連属性エントリに、前記見出ノードと前記関連属性との関連に従って対応付けるステップとをコンピュータに実行せらる。

20

## 【発明の効果】

## 【0023】

本発明によれば、データベースの構造を変更することなく、様々な種類の情報の追加が容易なデータベースシステムおよびその方法を提供することができる。

また、本発明によれば、情報の記述が簡単で、情報の表記および格納方法が一意的なデータベースシステムおよびその方法を提供することができる。

また、本発明によれば、データベースが複数の装置に分散されているときに、これら複数の装置間における負荷分散が容易なデータベースシステムおよびその方法を提供することができる。

30

また、本発明によれば、このように登録された情報を検索するために適したデータベースシステムおよびその方法を提供することができる。

## 【発明を実施するための最良の形態】

## 【0024】

## [本発明がなされるに至った経緯]

本発明の理解を助けるために、まず、本発明がなされるに至った経緯を説明する。様々な要素を含むデータを、効率的に収集し、それらのデータを記憶することが行われている。

記憶されたデータから、必要に応じてデータを読み出しこれらのデータを正確に再現することが望まれている。

40

一般に、直積集合  $A \times B$  の部分集合  $R \subseteq A \times B$  において、順序対  $(a, b) \in R$  に対して、 $a R b$  と表記すると、これは「 $a$  は  $b$  と関係  $R$  を有する」ことを意味する。

簡単なデータの一例として、「作家シェークスピアは戯曲ハムレットを書いた。」を例にとる。

## 【0025】

これらのデータは2項関係にあるので、「作家シェークスピア」 =  $a$ 、 $R$  = 作者 - 作品、 $b$  = 「戯曲ハムレット」と置くことにより、「作家シェークスピアは戯曲ハムレットを書いた。」なるデータが、「 $a R b$ 」と表記される。

これらのデータは、データベースに記憶する際に、「 $a$ 」、「 $R$ 」及び「 $b$ 」として記

50

憶され、データを読み出す場合にも一意に再現できる。

しかし、データの構成要素数が増えた場合は、すなわち2項関係ではなくn項関係になる場合には、これら関連性をもつ一連のデータはハイパーグラフ構造として表現され、処理は簡単ではない。

このため、一般にはn項関係を2項関係に分割して、それらの組合せとして表現し、データベースに記憶する手法がとられる。

#### 【0026】

第1の例として「作家シェークスピアは戯曲ハムレットを英国で1600年頃書いた。」を例にとって説明する。

この例は4項関係を表したものであるが、2項関係に展開すると、 $4C2 = 6$  であるので表1に示すように6つの2項関係の組合せとして表現される。

#### 【0027】

【表1】

a	R	b
作家シェークスピア	作者－作品	戯曲ハムレット
作家シェークスピア	作者－創作年代	1600年頃
作家シェークスピア	作者－創作国	英国
戯曲ハムレット	作品－創作年代	1600年頃
戯曲ハムレット	作品－創作国	英国
1600年頃	創作年代－創作国	英国

#### 【0028】

すなわち、nが大きくなると、n項関係を表現するために $nC2$ なる数の2項関係の表現が必要となってしまう。

#### 【0029】

さらに、第2の例として「作家シェークスピアは戯曲十二夜を1600年頃書いた。」を例として取り2項関係の組合せで表現すると、3項関係すなわち $n = 3$  であるので、 $3C2 = 3$ となり、表2に示すようになる。

#### 【0030】

【表2】

a	R	b
作家シェークスピア	作者－作品	戯曲十二夜
作家シェークスピア	作者－創作年代	1600年頃
戯曲十二夜	作品－創作年代	1600年頃

#### 【0031】

ここで、第1の例の情報と第2の例の情報とが同じデータベースに記憶されている場合、「作家シェークスピア」 - 「作者 - 創作年代」 - 「1600年頃」からなる同一の2項関係が記憶されるという問題が起こる。

また、上記第1及び第2の例として示された2項関係のうち、どれを組み合わせて元の

10

20

30

40

50

情報を再現すればよいか判断できないという問題も生じる。

これらは2項関係毎に識別子を付ければ解消される問題であるが、データ構造や処理が煩雑になるという欠点がある。

#### 【0032】

2項以上のデータ構成要素を有するデータを記憶する方法としては、リレーションナル・データベースによる記憶方法が知られている。

これは、データ項目名（データ属性）が列に割り当てられたテーブルを定義・作成し、具体的なデータをテーブルの行に順次入力していくものである。

#### 【0033】

「作家シェークスピアは戯曲ハムレットを英国で1600年頃書いた。」を例にとって説明する。10

データ項目名としては、(1)「誰が」、(2)「何を」、(3)「何時」、(4)「どこで」、(5)「何故」、(6)「どのように」を指定することができる。

そしてこれらのデータ項目に対応して、(1)「作家シェークスピア」、(2)「戯曲ハムレット」、(3)「1600年頃」、(4)「英国」、(5)「(ブランク)」、(6)「書いた」のデータを記憶することができる。

#### 【0034】

しかしながら、この方法では、次のような問題が生じる。

(1) データ項目名を、取得したデータに合わせて、後から追加することが簡単にはできない。定型の構成要素からなるデータの場合には、問題にはならないが、様々な構成要素を含むデータを入力する場合には、構成要素が増加するたびに、データ項目名を追加するためのスキーマの変更が必要となる。スキーマの変更はオンラインでデータの入力時には一般に困難である。20

(2) 後から追加することが簡単にはできないために、最初のデータベース構築時に、最大限のデータ項目数を挙げて、スキーマを構築することが考えられる。しかし入力する可能性の低いデータに対しても、データ項目としてスキーマを構築することは、メモリの使用効率が低くならざるを得ない。

#### 【0035】

このような問題を解決するために、従来の関係データモデルに代わるモデルとして、英國「Lazy Software」社による「Associative Model of Data（データ関連モデル）」法によるデータ記憶方法が提案されている。30

このデータモデルでは情報を事物と事物間の関連性と見なし、この関連性を「Source-Verb-Target」構文で表現している。

これによれば、上記のリレーションナル・データベースによる記憶方法で生じる問題の幾つかは解消される。

しかし、複雑なデータ関係を取扱う場合には、データ間の関係の表現方法が、複雑となり直感的でないこと、及び2分木的な形式を用いてn項関係を表現するため表現が一意に定まらず、データベースにデータを記憶する際に、作業者の任意にデータの構造を変えることができるために、データをデータベースから読み取り再現する場合に、正確に入力情報を再現できない場合が発生するという問題がある。40

#### 【0036】

上記のように、従来、相互に関係のある複数のデータを記憶する方法としては、リレーションナル・データベースが知られていた。

これは、記憶しようとするデータ項目を予め定めておき、データ項目に該当するデータを記憶するものである。

データの関係が見易いという利点はあるものの、一旦構成されたデータベースのデータ項目を追加すること、すなわち、一旦構成されたデータベースの構造（スキーマ）を変更することは簡単ではない。

また後から追加したデータ項目に対応するデータは、従前から記憶されているデータに対して、空白エリアが生じメモリの使用効率が悪くなるという問題が生じている。50

これに対して、Lazy Software社が提唱しているデータ関連モデルでは、データ項目数を追加する場合の困難さ、及びメモリの使用効率の低下といった問題は解決されるものの、データの記述内容が複雑になりデータ構造を見ても直感的ではなく、データの表記・記憶方法が一意的でないといった別の問題が生じる。

#### 【0037】

本発明では、見出ノード群と各ノード間の関連性を示すエッジ群から構成される関連性ネットワーク型データに対して、関連性（エッジ）を新たに関連ノード（以下、本願明細書では「Aノード」と呼ぶ。）とし、当該Aノードと関連性のある見出ノード（以下、本願明細書では「Tノード」と呼ぶ。）がその関連において果たす役割（以下、本願明細書では「関連性役割」と呼ぶ。）を属性とするエッジから成るデータモデルを定義する（図1）。

このモデルに従って変換されたデータ構造から、1つのAノードとTノード、及びその間の関連性役割を基本構成要素として抽出し、これらを表3に示すようにリレーションナル・データベースにおいて定義される関連性役割テーブル（以下、本願明細書では「ARテーブル」と呼ぶ。）の行（レコード）に対応付け、リレーションナル・データベース管理システムにより記憶・管理する方法を実現する。

#### 【0038】

【表3】

Aノード	Tノード	関連性役割
A1	T1	関連性役割1
A1	T2	関連性役割2

#### 【0039】

これにより、あるデータ（見出ノード）に関する新たな属性情報を別の関連性データとして定義し、それに対応するAノード、Tノード及びノード間のエッジの組合せ（すなわち、本発明のデータモデルにおける基本構成要素）を用いてARテーブルの行に対応付けられるデータとして表現すれば、既存のテーブル構造（データベース・スキーマ）を変更することなく新たな属性情報の追加が可能となる。

さらに、Aノード及びTノードにそれぞれ一意に同定できる識別子を付与し、各識別子に対して、ノードの属性型を表わすノードタイプと、属性値すなわちノードの具体的な内容を表わすノード名を有する識別子テーブル（以下、本願明細書では「IDテーブル」と呼ぶ。）を定義する（表5）。

#### 【0040】

これらのデータを、上記ARテーブルと同様に、リレーションナル・データベース管理システムにより記憶・管理する。

また、あるAノードにより示される関連性のもつ意味を具体的に記述するデータとして1つのTノードを新しく設け、これら2つのノードを「具象化」という予め定義された関連性役割で関連付ける。

当該Tノードと、別のAノードに関して同様に新しく設けられたTノードとの間での関連性を定義・記述することにより、元の2つのAノードにより示される関連性の間に存在する関係を表現することができる。

#### 【0041】

Aノードのもつ意味を記述するために特に導入された関連性役割「具象化」により関連付けられるTノードは、上記のARテーブルにより記憶・管理できる。

これらIDテーブル及びARテーブルを用いて各ノードを管理することにより、AノードとTノードの関連性だけでなく、AノードとAノードの関連性についても表記可能となるデータの表現方法を実現する。

## 【0042】

一般に、一つの共通の関連性を有する  $n$  個のデータからなるデータ集合について、2項関係で表現すると、 $nC2$  個のデータの組が必要となるが、本願発明によれば、 $n$  個のデータの組でよい。

言い換えると一つの共通の関連性を有する  $n$  個の要素を持つデータ集合について（図2、(a)）、これらのデータをデータベースに記憶する場合に、当該データ集合に共通するノード（Aノード）を新たに1つ設け、次に各要素毎にその関連性役割を定義する（図2、(b)）。

これにより、「Aノード」、「Tノード」及び「関連性役割」を一組のデータとして定義すれば、本願発明のデータ構造は構築できる（表4）。

10

20

30

40

50

## 【0043】

## 【表4】

Aノード	Tノード	関連性役割
A1	T1	関連性役割1
A1	T2	関連性役割2
A1	T3	関連性役割3
A1	T4	関連性役割4
:	:	:
A1	T(n-1)	関連性役割(n-1)
A1	Tn	関連性役割n

## 【0044】

図2(b)及び表4で使用されている「A1」はAノードに付与された識別子であり、別の識別子「T1」～「Tn」が付与された一群のデータ（Tノード）が何等かの共通の関連性を有していることを示すものである。

Aノードに付与された識別子が異なれば、それらの一連のデータは別の意味で何等かの共通の関連性を有していることになる。

さらに、識別子が付与されたAノード及びTノードに対して、表5に示すようにノードタイプとノード名をデータ属性としてもつIDテーブルを作成する。

## 【0045】

## 【表5】

ノード識別子	ノードタイプ	ノード名
A1	Aノードタイプ1	Aノード名1
T1	Tノードタイプ1	Tノード名1
T2	Tノードタイプ2	Tノード名2
:	:	:
Tn	Tノードタイプn	Tノード名n

## 【0046】

また、TノードT1とT2がAノードA1により関連付けられ、TノードT1とT3がAノードA2により関連付けられる関連性ネットワーク型データに対し(図3、(a))、AノードA1が示す関連性の意味を具体的に記述するTノード(識別子T11)を新たに設け、「具象化」という予め定義された関連性役割でA1と関連付ける。

同様に、AノードA2と新たなTノードT12を関連性役割「具象化」で関連付け、これら2つのTノードT11とT12との間にある関連性を、AノードA11を用いて定義する(図3、(b))。

これにより、元の2つのAノードA1とA2の間の関係が、表6に示すようなARテーブルを使って表現できる。

10

## 【0047】

## 【表6】

Aノード	Tノード	関連性役割
A1	T11	具象化
A2	T12	具象化
A11	T11	関連性役割11
A11	T12	関連性役割12

20

## 【0048】

## [データ記憶方法・データ構造]

以下、本発明におけるデータ記憶方法およびデータ構造を説明する。

具体例として、「作家シェークスピアは戯曲ハムレットを英国で1600年頃書いた。」を第1のデータとして説明する。

これらの情報は、(1)「作家シェークスピア」、(2)「戯曲ハムレット」、(3)「1600年頃」、(4)「英国」の要素をもつ4項関係を表したものである。  
2項関係に分割してデータの関係を表現すると、 $4C2 = 6$ であるので以下の表7に示す様に6個の組合せとして表現される。

30

## 【0049】

## 【表7】

見出ノード1	エッジ(関連性)	見出ノード2
作家シェークスピア	作者一作品	戯曲ハムレット
作家シェークスピア	作者一創作年代	1600年頃
作家シェークスピア	作者一創作国	英国
戯曲ハムレット	作品一創作年代	1600年頃
戯曲ハムレット	作品一創作国	英国
1600年頃	創作年代一創作国	英国

40

## 【0050】

本願発明に従って、これらの情報を変換する。先ず第1番目のデータについて変換する

50

見出ノード1にある「作家シェークスピア」は、この情報において「作者」であることを示しているので、「作家」と「シェークスピア」とに分けて、「シェークスピア」をTノード、「作者」を関連性役割とする。また、後述するように、「作家」をTノードのノードタイプとする。

また関連性を示すエッジ「作者 - 作品」については、上記の一連の情報が同じグループであることを示すために「ハムレット著作について」としてAノードを設ける。

「ハムレット著作について」のデータは、一連の情報が同じグループであることを示すために用いるので、他のグループの情報との識別ができるものであれば、他の表現でもかまわない。

10

20

【0051】

「Aノード」 - 「Tノード」 - 「関連性役割」の構造で記述すると、表8に示すようになる。

【0052】

【表8】

Aノード	Tノード	関連性役割
ハムレット著作について	シェークスピア	作者

【0053】

同様に見出ノード2にある「戯曲ハムレット」は、この情報において「作品」であることから、変換すると表9のようになる。

また、「戯曲」をTノードのノードタイプとする。

【0054】

【表9】

Aノード	Tノード	関連性役割
ハムレット著作について	ハムレット	作品

30

【0055】

次に、第2番目のデータについて同様に変換すると、表10のようになる。

【0056】

【表10】

Aノード	Tノード	関連性役割
ハムレット著作について	シェークスピア	作者
ハムレット著作について	1600年頃	創作年代

40

【0057】

ここで、「ハムレット著作について」「シェークスピア」「作者」の情報は、冗長なデータなので省略することができる。以下同様に全てのデータについて変換し、冗長なデータを削除すると表11のようになる。

【0058】

【表11】

Aノード	Tノード	関連性役割
ハムレット著作について	シェークスピア	作者
ハムレット著作について	ハムレット	作品
ハムレット著作について	1600年頃	創作年代
ハムレット著作について	英国	創作国

10

【0059】

従って、一つの共通の関連性を有する4個のデータからなるデータ集合について、2項関係で表現すると、 $4C2 = 6$ であるので6個のデータの組が必要となるが、本願発明によれば、4個のデータ組でよいことが分かる。

言い換えると、4個の要素を持つデータ集合について、これらのデータをデータベースに記憶する場合に、当該データ集合に共通するノード（Aノード）を新たに1つ設け、次に要素毎にその関連性役割を定義し、「Aノード」、「Tノード」及び「関連性役割」を一組のデータとして定義すれば、本願発明のデータ構造は構築できる。  
20

ここで、Aノード「ハムレット著作について」に識別子「A1」を付与する。ノード識別子「A1」を共通にもつこれらのデータは、一つのグループに属することが分かる。

さらに、4つのTノード「シェークスピア」、「ハムレット」、「1600年頃」、「英国」に対して識別子「T11」～「T14」をそれぞれ付与する。

これにより、第1の例から作成されるA Rテーブルは表12のようになる。

【0060】

【表12】

Aノード	Tノード	関連性役割
A1	T11	作者
A1	T12	作品
A1	T13	創作年代
A1	T14	創作国

30

40

【0061】

また、「ハムレット著作について」を示すAノード（識別子A1）のノードタイプを「著作関連情報」とし、識別子T11～T14が付与されたTノードのノードタイプをそれぞれ「作家」、「戯曲」、「年代」、「国」とすることにより、表13に示すようなI Dテーブルが作成される。

【0062】

【表13】

A1	著作関連情報	(null)
T11	作家	シェークスピア
T12	戯曲	ハムレット
T13	年代	1600年頃
T14	国	英国

10

【0063】

さらに、第2のデータとして「戯曲ハムレットを原作とする日本語翻訳版が2003年2月に○○出版社から出版された。」の情報を例にとる。

これらの情報を本願発明に従ってデータを構成すると次のようになる。ここで一連の情報要素に共通するAノードは「ハムレット日本語訳について」とし、識別子「A2」を付与すると表14のようになる。

【0064】

【表14】

Aノード	Tノード	関連性役割
A2	ハムレット	原作
A2	ハムレット	翻訳版
A2	2003年2月	出版日
A2	○○出版社	出版

20

30

【0065】

原作であるハムレットには既に識別子「T12」が付けられているので、翻訳版ハムレットに「T22」、出版日及び出版社に対してそれぞれ識別子「T23」、「T24」を付与すると、第2のデータに対するA Rテーブルは表15のようになる。

【0066】

【表15】

Aノード	Tノード	関連性役割
A2	T12	原作
A2	T22	翻訳版
A2	T23	出版日
A2	T24	出版

40

【0067】

また、表16に示すIDテーブルが作成される。

【表16】

ノード識別子	ノードタイプ	ノード名
A2	著作関連情報	(null)
T22	戯曲	ハムレット
T23	日付	2003年2月
T24	出版社	○○出版社

10

【0068】

これら第1及び第2のデータ、さらに同様のデータ変換処理が施された他のデータも同じデータベースに記憶されるので、最終的に表17及び表18に示すようなARテーブル及びIDテーブルが得られる。

【0069】

【表17】

Aノード	Tノード	関連性役割
A1	T11	作者
A1	T12	作品
A1	T13	創作年代
A1	T14	創作国
A2	T12	原作
A2	T22	翻訳版
A2	T23	出版日
A2	T24	出版
:	:	:

20

30

【0070】

【表18】

ノード識別子	ノードタイプ	ノード名
A1	著作関連情報	(null)
T11	作家	シェークスピア
T12	戯曲	ハムレット
T13	年代	1600年頃
T14	国	英国
A2	著作関連情報	(null)
T22	戯曲	ハムレット
T23	日付	2003年2月
T24	出版社	○○出版社
:	:	:

10

20

【0071】

さらに、図4に示すように、識別子がA1及びA2であるAノードにより示される関連性のもつ意味を具体的に記述するためのTノードを設け、それぞれに識別子「T31」及び「T32」を付与し、ノードA1及びノードA2と関連性役割「具象化」で関連付ける。

これら2つの新たなTノードのノードタイプを「著作情報」とし、ノード名としてはそれぞれ「ハムレット著作について」及び「ハムレット日本語訳について」とする。

ノードT31とノードT32との関連性を示すAノードを、識別子A3としてさらに新しく設け、ノードタイプを「原作 - 翻訳情報」とする。

30

また、ここでの関連においてノードT31及びノードT32が果たす役割をそれぞれ「原作情報」、「翻訳情報」とする。以上の処理から、表19に示したARテーブル及び表20に示したIDテーブルが追加される。

【0072】

【表19】

Aノード	Tノード	関連性役割
A1	T31	具象化
A2	T32	具象化
A3	T31	原作情報
A3	T32	翻訳情報

40

【0073】

## 【表20】

ノード識別子	ノードタイプ	ノード名
T31	著作情報	ハムレット著作について
T32	著作情報	ハムレット日本語訳について
A3	原作-翻訳情報	(null)

10

## 【0074】

上記の第1及び第2のデータを、リレーションナル・データベースに直接記憶するためには、データ項目名として、新たに翻訳版としての「ハムレット」あるいは「出版社」などに対応する項目（データ属性）を追加する必要がある。これはデータベースのテーブル構造を変更することになるので簡単ではない。

本発明では、例に示した様に既存のARテーブルに対し行を追加することにより、異なるデータ属性を有する情報を複数の組として記憶することが可能である。

また、第1及び第2のデータを、Aノード、Tノード及び各ノード間の関連を示すエッジを用いて表現すると図4のようになる。

本発明では、こうした複雑な構造をもつデータを容易に表現するための方法とリレーションナル・データベースへの記憶・管理方法を提供している。 20

## 【0075】

次に本願発明のデータベースから所望のデータを検索する場合について説明する。

以下に、ユーザが「作家シェークスピアの書いた戯曲ハムレットの日本語訳の出版社を知りたい」場合を例にとって説明する。

図5にそのフローチャートを示す。以下に図5のフローチャートの説明をする。

## 【0076】

110：ユーザは検索条件を「作家」である「シェークスピア」の「戯曲」として入力する。

120：検索条件に合致する複数組のデータを収集する。

30

130：複数組検出された情報の中から、「戯曲」に対応する情報を順次表示する。

140：ユーザはこれらの中から所望の戯曲名、すなわち「ハムレット」を選択する。

150：「ハムレット」及び「翻訳版」を新たな条件として、データを検索する。

160：検索条件に合致する複数組データを収集する。

170：検出された複数組の中から「出版社」及び「出版日」に関するデータを順次表示する。

180：ユーザは所望の出版社を選択する。

## 【0077】

以下、詳細に説明する。

ユーザは検索条件として、Tノードのノード名が「シェークスピア」であり、関連性役割が「作家」であることを入力し情報を検索する。 40

データベースの中でARテーブルを参照し、関連性役割が「作家」であるTノードの識別子を検出し、続いて、IDテーブルに記憶されているノード名属性を参照し、「シェークスピア」をノード名とするTノードに対応するAノード識別子が複数組検出される。

複数組検出されたAノード識別子をもつ情報の中から、別の検索条件、つまり関連性役割が「戯曲」となっているTノード識別子がARテーブルより選出される。

この識別子に基づいて、IDテーブルから対応する戯曲に関する情報が順次表示される。

## 【0078】

すなわち「ハムレット」、「じゃじゃ馬ならし」、「ベニスの商人」、「真夏の夜の夢

50

」、「リア王」などの戯曲名が順次表示される。ユーザはこれらの中から所望の戯曲名、すなわち「ハムレット」を選択する。

引き続き、選択された「ハムレット」の識別子「T12」をキーとして、TノードIDに「ハムレット」の識別子を含み、かつ、関連性役割が「翻訳版」であるAノード識別子をA Rテーブルから検索する。

これにより、検索条件を満足するAノード識別子により関連付けられた複数組の情報が検出される。

検出された複数組の情報から、関連性役割として「出版社」及び「出版日」をもつTノードのノード名を、順次、IDテーブルを参照しながら表示する。

ユーザはその中から所望の出版社、すなわち最近「2003年2月」に戯曲ハムレットの翻訳版を出版した「出版社」を選択することができる。10

#### 【0079】

以上、図5に示した例は、検索例の一例として説明したものである。

つまり検索条件が多くなると、データベースの検索は、図5に示されているように2回に限定されるものではなく、所望の条件に応じて任意回数繰り返し行われることは言うまでもない。

また、ここで示した具体例においては、関連性役割として単一の属性が割り当てられているが、これに限定されるものではない。

すなわち、関連性役割に複数の属性をもたせることも可能である。上記の例では、関連性役割「戯曲」に対して「悲劇」、「喜劇」、「ロマンス劇」、「史劇」等のジャンルを追加して指定することにより、さらに詳細な関連性役割を定義することができる。20

#### 【0080】

##### [本発明にかかるデータ記憶方法・データ構造の特徴]

以上説明したように、本発明にかかるデータ記憶方法およびデータ構造によると、現在、広く一般に用いられているリレーションナル・データベースのテーブル形式を用いて、3項以上の関係、一般にn項の相互に関連性をもった一連のデータの関係を表すハイパーテーブルの構造を保持したまま、データを記憶及び管理することができる。

また、関連性ネットワーク型データをリレーションナル・データベースのテーブルに直接マッピングする方法では、3項以上の関係、一般にn項関係を表すデータを効率よく記憶・管理できないという問題が解決される。30

#### 【0081】

また、既にデータベースに記憶されているデータに対して、属性情報を追加するなどの変更を行うにはリレーションナル・データベースのテーブル設計の変更が必要となり、柔軟性に欠けるとともに膨大な労力がかかるという問題が解決される。

さらに、同一のデータベース・スキーマの枠組みで、データに付与された識別子と関連性役割を用いて一連の関連性のもつ意味を記述することができる。

#### 【0082】

##### [第1のデータベース装置1]

以下、本発明にかかる第1のデータベースを説明する。

図6は、本発明にかかる第1のデータベースシステム(DBシステム)1の構成を例示する図である。40

図6に示すように、本発明にかかる第1のDBシステム1は、データベース装置(DB装置)12が、必要に応じて、LAN、WANおよびインターネットなどのネットワーク100を介して、データ入力および検索のためなどに用いられるコンピュータ(PC)102と接続されて構成される。

なお、以下の説明においては、図4、図5および表7～表20を参照して上述した本発明にかかるデータ記憶方法およびデータ構造の説明においてと、一部、用語が異なることがあるが、これらの説明の間で対応する用語は、実質的に同じである。

また、DBシステム1の説明については、以下の用語が、上述した本発明にかかるデータ記憶方法およびデータ構造の説明における用語よりも優先される。50

また、以下の各図において、同様な構成部分には同じ符号が付される。

#### 【0083】

##### [ハードウェア構成]

図7は、図6に示したDB装置12およびPC102のハードウェア構成を例示する図である。

図7に示すように、DB装置12およびPC102は、CPU122、メモリ124およびこれらの周辺回路などを含む本体120、表示装置およびキーボードなどを含む入出力装置126、CD装置およびHDD装置などの記録装置128から構成される。

また、さらに、DB装置12およびPC102（以下、通信を行う構成部分を通信ノードと総称することがある）が、ネットワーク100に接続されるときには、ネットワーク100を介した他の通信ノードとの間の通信を行う通信装置132が付加されることがある。

つまり、DB装置12およびPC102は、他の通信ノードとの間で通信を行う機能を有しうるコンピュータとしての構成部分を含む。

#### 【0084】

##### [データ構造]

DB装置12においては、上述した本発明にかかるデータ記憶方法およびデータ構造（図4、図5および表7～表20）を応用したデータの記憶および記憶されたデータの検索が行われる。

まず、本発明にかかるDB装置12の詳細を説明する前に、その理解を助けるために、DB装置12におけるデータ構造およびデータ検索について説明する。

#### 【0085】

図8は、図3(a)に例示したデータの関連を書き換えて示す図である。

図8に示すように、DB装置12においては、上述の本発明にかかるデータ記憶方法およびデータ構造においてと同様に、見出ノード(Topic Node；以下、Tノードと記す)が、1つ以上の関連ノード(Association Node；以下、Aノードと記す)と関連付けられ、関連付けられたTノードとAノードとの間には、関連属性Rが定義される。

なお、関連属性Rは、TノードとAノードとの関連について定義されうるどのような属性でもよいが、以下、説明の具体化および明確化のために、上述の本発明にかかるデータ記憶方法およびデータ構造の説明においてと同様に、関連属性Rが、関連性役割Rである場合を具体例とする。

#### 【0086】

図3(a)に示したデータの関係は、書き換えにより、図8に示すように表される。

図8に示すAノードA1～An（以下、nは1以上の整数を示す。但し、全てのnが同じ数を示すとは限らない。）の内、AノードA1と、AノードA1に関連するTノードT1-1～T1-3、T2-1とは、エッジで結ばれている。

同様に、AノードA2に関連するTノードT2-1、T2-2、Tn-1と、AノードA2とはエッジで結ばれている。

また、AノードAnについても同様であって、AノードAnに関連するTノードTn-1～Tn-4とAノードAnとはエッジで結ばれている。

つまり、図8には、TノードT2-1が、AノードA1、A2の両方に関連性を有し、TノードTn-1は、AノードA2、Anの両方に関連性を有することが示されている。

#### 【0087】

図9は、図8に示したデータ構造を一般化して示す図である。

図8において、TノードT1-1から、AノードA1、TノードT2-1、AノードA2およびTノードTn-1を経て、AノードAnに至るようにエッジをたどると、図9において、上から下の方向に伸びるパスが得られる。

#### 【0088】

なお、図9には、

(1) TノードT1-1～T1-m1、T2-1が、AノードA1に関連付けられ、T

10

20

30

40

50

ノード T 1 - 1 ~ T 1 - m 1 , T 2 - 1 と A ノード A 1 との間の関連（エッジ）には、関連性役割 R 1 - 1 ~ R 1 - m 1 , R 1 - 0 が定義され、

(2) T ノード T 2 - 1 ~ T 2 - m 2 および図 9において省略された T ノードが、A ノード A 2 に関連付けられ、T ノード T 2 - 1 ~ T 2 - m 2 と A ノード A 2 との間の関連には、関連性役割 R 2 - 1 ~ R 2 - m 2 が定義され、

(3) 以下、同様に、図 9において省略されている T ノードと A ノードとが関連付けられ、これらの間に関連性役割 R が定義され、

(4) T ノード T n - 1 ~ T n - m n および図 9において省略された T ノードが、A ノード A n に関連付けられ、T ノード T n - 1 ~ T n - m n と A ノード A n との間の関連には、関連性役割 R n - 1 ~ R n - m n (m 1 ~ m n , n は整数) が定義され  
10 ている場合が具体例として示されている。

#### 【0089】

つまり、DB 装置 12においては、T ノードそれが、1つ以上の A ノードと関連付けられ、また、A ノードそれが、1つ以上の T ノードと関連付けられることにより、A ノードを介して複数の T ノードが関連付けられ、また、T ノードを介して複数の A ノードが関連付けられる。

DB 装置 12においては、図 9に示すように関連付けられた A ノードと T ノードとの組み合わせが、複数、記憶されうる。

#### 【0090】

図 10 は、図 3 (b) , 図 4 に示したデータの関係を、一般化して示す図である。  
20

図 3 (b) , 図 4 に示したデータの関係は、図 10 に示すように一般化して表されうる。  
。

図 10において、A ノード A 1 に関連する T ノード T 1 - 1 ~ T 1 - 3 ( , T 3 - 1 )  
と A ノード A 1 とはエッジで結ばれ、A ノード A 2 に関連する T ノード T 2 - 1 ~ T 2 -  
3 ( , T 3 - 2 ) と A ノード A 2 とはエッジで結ばれ、A ノード A n と T ノード T n - 1  
~ T n - 3 , T 2 - 3 ( , T 3 - n ) とはエッジで結ばれている。

T ノード T 2 - 3 は、A ノード A 2 , A n の両方とエッジで結ばれており、このことは  
、T ノード T 2 - 3 が、A ノード A 2 , A n の両方に関連することを示している。

ここで、A ノード A 1 , A 2 , A n により関連付けられる一連の情報が、共通の関連性  
30 を有するときには、新たな関連ノード A 3 が定義されうる。

例えば、A ノード A 1 が、ハムレットの原作に関する情報であり、A ノード A 2 が、ハムレットの翻訳に関する情報であり、A ノード A n が、ハムレットの公演に関する情報であるときには、これらの A ノード A 1 , A 2 , A n により示される関連情報は、ハムレットに関する情報としての共通性を有する。

#### 【0091】

そこで、A ノード A 1 , A 2 , A n により関連付けられる情報が、共通性を有すること  
を示すために、新たな関連ノード A 3 が定義され、データベースに格納される。

また、図 10 中に破線で囲って示すように、A ノード A 1 および T ノード T 1 - 1 ~ T  
1 - 3 により示される一連の情報を具体的に記述するために、新たな T ノード T 3 - 1 が  
定義され、データベースに格納される。  
40

さらに、同様に、A ノード A 2 , A n の関連を具体的に記述する新たな T ノード T 3 -  
2 , T 3 - n が定義され、データベースに格納される。

例えば、T ノード T 3 - 1 には、見出しの内容として「ハムレット著作について」、T  
ノード T 3 - 2 には、見出しの内容として「ハムレット日本語訳について」、T ノード T  
3 - n には、見出し内容として「ハムレット公演について」などのデータが定義され、デ  
ータベースに格納される。

#### 【0092】

さらに、新たな A ノード A 3 と、T ノード T 3 - 1 ~ T 3 - n それぞれとの間に、関連性  
役割 R が定義され、データベースに格納される。

例えば、新たな A ノード A 3 と T ノード T 3 - 1 との間に、関連性役割 R として「原作  
50

「情報」が定義され、新たなAノードA<sub>3</sub>とTノードT<sub>3-2</sub>との間に、関連性役割Rとして「翻訳情報」が定義され、新たなAノードA<sub>3</sub>とTノードT<sub>3-n</sub>との間に、関連性役割Rとして「公演情報」などのデータが定義され、データベースに格納される。

同様に、例えば、AノードA<sub>1</sub>, A<sub>2</sub>, A<sub>n</sub>とTノードT<sub>3-1</sub>, T<sub>3-2</sub>, T<sub>3-n</sub>との間には、「具象化」という予めシステムにより定義された関連性役割Rが定義される。

#### 【0093】

図11は、図9に示した構造を探るデータを記憶するために用いられる関連性役割(A R (Association Role))テーブルを示す図である。

図12は、図9に示した構造を探るデータを記憶するために用いられるTノード用識別子(I D (Identifier))テーブルを示す図である。

図13は、図9に示した構造を探るデータを記憶するために用いられるAノード用識別子(I D)テーブルを示す図である。

DB装置12において、図9に示した構造により関連付けられたAノードおよびTノードと、Tノードのデータは、図11に示すARテーブル、および、図12に示すIDテーブルを用いて記憶される。

#### 【0094】

図11に示すARテーブルのエントリそれぞれは、ある1つのAノードと、このAノードに関連付けられた1つのTノードと、これら関連付けられたAノードとTノードとの間に定義される関連性役割(R)を示し、あるAノードの識別子(ID)と、このAノードに関連付けられたTノードの識別子と、これら関連付けられたAノードとTノードとの間に定義された関連性役割(R)を含む。

つまり、ARテーブルのエントリそれぞれは、図9に示したいずれかのエッジの一端にあるAノードの識別子と、このエッジの他端にあるTノードの識別子と、このエッジに対して定義される関連性役割とを含む。

#### 【0095】

このようなエントリを、図9に示した全てのエッジ(T<sub>1-1</sub>~A<sub>1</sub>の間のエッジ~T<sub>n-mn</sub>~A<sub>n</sub>間のエッジ)について作成し、ARテーブルに記憶することにより、図9に示したAノードとTノードとの関連は、図11に示すARテーブルに記憶される。

また、Tノードそれぞれは、内容(Tノードの名称、Tノード自体のデータおよびTノードにより参照されるデータなど)を有し、さらに、Tノードそれぞれには、ARテーブルの各エントリに記憶される識別子(ID)の他に、このTノードの属性(ノードタイプ(N T); 見出属性)が定義される(以下、Tノードが、その内容として、その名称(ノード名(N))のみを有する場合を具体例とする。)。

#### 【0096】

図12に示すTノード用のIDテーブルのエントリそれぞれは、図9に示したいずれかのTノードの識別子(ID)と、このTノードに対して定義された属性(ノードタイプ(N T))と、このTノードの名称(ノード名(N))とを含む。

このようなエントリを、図9に示した全てのTノードT<sub>1-1</sub>~T<sub>n-mn</sub>について作成し、Tノード用のIDテーブルに記憶することにより、図9に示した全てのTノードについてのデータが記憶される。

図13に示すAノード用のIDテーブルのエントリそれぞれは、図9に示したいずれかのAノードの識別子(ID)と、このAノードに対して定義された属性(ノードタイプ(N T'))と、このAノードの名称(ノード名(N'))とを含む。

このようなエントリを、図9に示した全てのAノードA<sub>1</sub>~A<sub>n</sub>について作成し、Aノード用のIDテーブルに記憶することにより、図9に示した全てのAノードについてのデータが記憶される。

#### 【0097】

なお、図9に示したAノードとTノードとの関連、AノードのデータおよびTノードのデータを記憶するためには、テーブル形式以外に、同様な形式が採られうるが、以下の説

10

20

30

40

50

明においては、ARテーブルおよびIDテーブルを用いる場合が具体例とされる。

なお、DB装置12の用途、構成あるいは処理内容によっては、図11に示すように、ARテーブルにおいて、各エントリは、Tノードの識別子(ID)の代わりに、Tノードの内容(ノード名(N))を含んでもよい。

また、同様に、ARテーブルにおいて、各エントリは、Tノードの内容を、さらに含んでもよい。

#### 【0098】

##### [データ検索]

図14は、図6、図7に示したDB装置12におけるデータ検索方法を例示する図である。10

ここでは、図14に示すように、あるAノードに、TノードT<sub>1</sub>～T<sub>n</sub>および検索結果(出力)とされるべきTノードT<sub>ret</sub>(T return)が関連付けられ、AノードとTノードT<sub>1</sub>～T<sub>n</sub>、T<sub>ret</sub>との間に、関連性役割R<sub>1</sub>～R<sub>n</sub>、R<sub>ret</sub>が定義され、TノードT<sub>1</sub>～T<sub>n</sub>、T<sub>ret</sub>が、ノード名N<sub>1</sub>～N<sub>n</sub>、N<sub>ret</sub>を有する場合を具体例とする。

#### 【0099】

DB装置12においては、検索結果として得たいTノードに対して定義される関連性役割R<sub>ret</sub>と、検索のために用いられるAノードの属性(ノードタイプNT；図14においてANT1, ANT2)と、検索のために用いられるTノードの関連性役割Rとノード名Nの1つ以上の組み合わせとが検索条件として用いられる。20

この検索条件は、図14に示すように、例えば、(R<sub>ret</sub>, (ANT1, ANT2, . . .), Filter), Filter=((R<sub>1</sub>, N<sub>1</sub>), (R<sub>2</sub>, N<sub>2</sub>), . . ., (R<sub>n</sub>, N<sub>n</sub>))と表記される。

なお、この検索条件は、後述するように、さらにTノードの属性NT(第3の条件データ)を含みうる。

上記検索条件の内、Filterに含まれるTノードの関連性役割Rとノード名Nの1つ以上の組み合わせ(R<sub>1</sub>, N<sub>1</sub>), (R<sub>2</sub>, N<sub>2</sub>), . . ., (R<sub>n</sub>, N<sub>n</sub>)それぞれは、検索のためのフィルタとして用いられるので、以下、検索フィルタとも記載される。

#### 【0100】

また、上記検索条件の内、Aノードの属性(ANT1, ANT2, . . .)は、省略されうる。30

図15は、図6、図7に示したDB装置12における検索の全体的な処理(S20)を示す第1のフローチャートである。

図15に示すように、ステップ200(S200)において、DB装置12は、例えば、PC102(図6)あるいはDB装置12の入出力装置126に対する検索者の操作に応じて、図14に例示した検索条件を受け入れる。

#### 【0101】

ステップ22(S22)において、図16を参照して後述する検索フィルタに基づく関連ノードの選択を行う。

ステップ24(S24)において、図17を参照して後述するノードIDおよびノード名の取得を行う。40

#### 【0102】

ステップ202(S202)において、DB装置12は、S24の処理により検索結果として得られたTノードT<sub>ret</sub>の識別子(ノードID)およびノード名(N<sub>ret</sub>)から、検索者に対して返す応答を作成する。

この応答としては、ノード名N<sub>ret</sub>のみ、ノードT<sub>ret</sub>により参照される様々なデータ、あるいは、ノードT<sub>ret</sub>自体を示すデータなど、様々なデータを例示されうる。

#### 【0103】

ステップ204(S204)において、DB装置12は、検索者による問い合わせが終了したか否かを判断する。

DB装置12は、検索者による問い合わせが終了した場合には処理を終了し、これ以外50

のときには S 2 0 0 の処理に戻る。

#### 【 0 1 0 4 】

図 1 6 は、図 1 5 に示した検索フィルタに基づく関連ノードの選択処理 ( S 2 2 ) を示すフローチャートである。

図 1 6 に示すように、図 1 5 に示した S 2 0 0 の処理において、DB 装置 1 2 は、検索条件 ( R r e t , ( A N T 1 , A N T 2 , . . . ) , F i l t e r ) , F i l t e r = ( ( R 1 , N 1 ) , ( R 2 , N 2 ) , . . . , ( R n , N n ) ) を受け入れると、ステップ 2 2 0 ( S 2 2 0 ) において、処理のために用いられる関連ノードリストを初期化する。

#### 【 0 1 0 5 】

この関連ノードリストには、A R テーブル ( 図 1 1 ) から得られた A ノードのうち、検索条件中の A ノードの属性 ( ノードタイプ ; A N T 1 , A N T 2 , . . . ) のいずれかを、その属性 ( ノードタイプ N T ) として含む A ノードの識別子が記憶される。 10

なお、検索条件において、A ノードの属性が省略された ( ( A N T 1 , A N T 2 , . . . ) = n u l l ) ときには、S 2 2 0 の処理において、A R テーブル ( 図 1 1 ) から得られた A ノードの全ての識別子が記憶される。

ステップ 2 2 2 ( S 2 2 2 ) において、DB 装置 1 2 は、全ての検索フィルタ ( R i , N i ) について処理を行ったか否かを判断する。

#### 【 0 1 0 6 】

DB 装置 1 2 は、全ての処理について処理を行ったときには S 2 4 ( 図 1 5 , 図 1 7 ) の処理に進み、これ以外のときには、まだ処理の対象とされていないいずれかの検索フィルタ ( R i , N i ) を、次の処理対象として S 2 2 4 の処理に進む。 20

ステップ 2 2 4 ( S 2 2 4 ) において、DB 装置 1 2 は、T ノード用の I D テーブル ( 図 1 2 ) を検索し、検索フィルタ ( R i , N i ) のノード名 N i を含むエントリの全てを探し出し、探し出されたエントリに含まれる T ノードの識別子の集合 ( ノード I D 集合 T ) を作成する ( T = { T i | ノード名 = N i i n I D テーブル } ) 。

なお、検索条件に T ノードの属性 ( ノードタイプ ; N T ) が含まれ、検索フィルタが ( R i , N i , N T i ) と表されるときには、DB 装置 1 2 は、S 2 2 4 の処理において、I D テーブルから、検索フィルタ ( R i , N i , N T i ) のノード名 N i およびノードタイプ N T i を含むエントリを探し出し、探し出されたエントリに含まれる T ノードの識別子の集合をノード I D 集合 T とすればよい。 30

#### 【 0 1 0 7 】

ステップ 2 2 6 ( S 2 2 6 ) において、DB 装置 1 2 は、S 2 2 4 の処理により得られたノード I D 集合 T が空集合であるか否かを判断する。

DB 装置 1 2 は、ノード I D 集合 T が空集合であるときには検索処理を終了するための処理 ( 検索が失敗した旨を検索者に表示するなど ) を行い、検索処理を終了し、これ以外のときには S 2 2 8 の処理に進む。

ステップ 2 2 8 ( S 2 2 8 ) において、DB 装置 1 2 は、A R テーブル ( 図 1 1 ) を検索し、関連ノードリスト A を更新する。

つまり、DB 装置 1 2 は、A R テーブルから、検索フィルタ ( R i , N i ) の関連性役割 R i と、S 2 2 4 の処理により得られたノード I D 集合 T に含まれるいずれかの T ノードの識別子を含むエントリの全てを探し出し、探し出されたエントリに含まれる A ノードの識別子を、関連ノードリスト A に格納する ( A = { A j | 役割 R i , T ノードの識別子 T i ( a l l i ) , A ノード識別子 A i n A R テーブル } ) 。 40

#### 【 0 1 0 8 】

ステップ 2 3 0 ( S 2 3 0 ) において、DB 装置 1 2 は、S 2 2 8 の処理により得られた関連ノードリスト A が空集合であるか否かを判断する。

DB 装置 1 2 は、関連ノードリスト A が空集合であるときには検索処理を終了するための処理を行い、検索処理を終了し、これ以外のときには S 2 3 2 の処理に進む。

ステップ 2 3 2 ( S 2 3 2 ) において、DB 装置 1 2 は、検索条件に含まれる処理されていない検索フィルタを読み込み、S 2 2 2 の処理に戻る。 50

## 【0109】

図17は、図15、図16に示したノードIDおよびノード名取得処理(S24)を示すフローチャートである。

図17に示すように、検索フィルタに基づく関連ノードの選択処理(S22)が終了すると、ステップ240(S240)において、DB装置12は、ARテーブル(図11)を検索し、Tノード識別子集合Tを作成する。

つまり、DB装置12は、ARテーブルから、検索条件に含まれる関連性役割Retと、S22(S228)の処理により得られた関連ノードリストAに含まれるいずれかのAノードの識別子とを含むエントリの全てを探し出し、探し出されたエントリに含まれるTノードの識別子の集合(Tノード識別子集合T)を作成する( $T = \{ T_m | \text{役割} = R_{r e t}, A \text{ノードの識別子 } A \text{ in ARテーブル} \}$ )。

## 【0110】

ステップ242(S242)において、DB装置12は、S240の処理により得られたTノード識別子集合Tが空集合であるか否かを判断する。

DB装置12は、Tノード識別子集合Tが空集合であるときには終了処理を行って検索処理を終了し、これ以外のときにはS244の処理に進む。

ステップ244(S244)において、DB装置12は、Tノード用のIDテーブル(図12)を検索し、ノードIDとノード名の組の集合Pを作成する。

つまり、DB装置12は、IDテーブルから、S240の処理により作成されたTノード識別子集合Tに含まれるTノードの識別子Tmのいずれかを含むエントリの全てを探し出し、このエントリに含まれるノード名Nmと、Tノードの識別子Tmとの組の集合Pを作成する( $P = \{ (T_m, N_m) | T \text{ノードの識別子} = T_m (all m) \text{ in IDテーブル} \}$ )。

## 【0111】

ステップ246(S246)において、DB装置12は、S244の処理により得られたTノード識別子とTノード名との組の集合Pが空集合であるか否かを判断する。

DB装置12は、Tノード識別子とTノード名との組の集合Pが空集合であるときには終了処理を行って検索処理を終了し、これ以外のときにはS202の処理に進む。

この集合Pは、図15に示したS202の処理において、検索者への応答を作成するために用いられる。

## 【0112】

## [DBプログラム2]

図18は、図6、図7に示したDB装置12において実行されるDBプログラム2の構成を示す図である。

なお、図18においては、図示の明確化のために、データの流れを示す線は、適宜、省略されている。

図18に示すように、DBプログラム2は、DB管理部20、DB部24およびDB検索部26から構成される。

## 【0113】

DB管理部20は、管理操作受入部200、ARエントリ作成部202、IDエントリ作成部204、ARデータベース管理部(ARDB管理部)206およびIDデータベース管理部(IDDB管理部)208から構成される。

DB部24は、ARデータベース(ARDB)240、Tノード用のIDデータベース(IDDB)242およびAノード用のIDDB244から構成される。

DB検索部26は、検索操作受入部260、検索条件作成部262、検索制御部264、ARデータベース検索部(ARDB検索部)266およびIDデータベース検索部(IDDB検索部)268から構成される。

## 【0114】

DBプログラム2は、例えば、記録媒体130(図7)を介してDB装置12に供給され、メモリ124にロードされ、DB装置12で実行されるOS上で、DB装置12のハ

10

20

30

40

50

ードウエアを具体的に利用して実行される（以下の各プログラムについて同様）。

D B プログラム 2 は、これらの構成部分により、図 9 ~ 図 17 を参照して説明した A R データベース（図 11）および I D データベース（図 12, 図 13）の作成、および、これらのデータベースを用いたデータの検索（図 14 ~ 図 17）を行う。

#### 【0115】

D B 部 24において、A R D B 240は、図 11に示したA R テーブルを記憶する。

I D D B 242は、図 12に示したT ノード用のI D テーブルを記憶する。

I D D B 244は、図 13に示したA ノード用のI D テーブルを記憶する。

また、図 18には、図 12, 図 13に示したT ノード用のI D テーブルおよびA ノード用のI D テーブルが、それぞれI D D B 242, 244に記憶される場合が具体例とされているが、T ノード用のI D テーブルと、A ノード用のI D テーブルとは、同じデータベースに記憶されてもよい。10

また、T ノード用のI D テーブルと、A ノード用のI D テーブルとは、必ずしも分けて作成されなくともよく、1つのデータベース内に一体に作成されてもよい。

#### 【0116】

D B 管理部 20において、管理操作受入部 200は、A R テーブルおよびI D テーブルに記憶されたデータを管理し、あるいは、変更するための操作を、入出力装置 126（図 7）から、あるいは、ネットワーク 100を介して P C 102（図 6）から受け入れ、A R D B 管理部 206 および I D D B 管理部 208 に対して出力する。

また、管理操作受入部 200は、A ノードおよびT ノード、A ノードとT ノードとの関連付け、A ノードとT ノードとの間（エッジ）に定義される関連性役割 R、A ノードおよびT ノードに付される識別子（I D）、T ノードに付されるノード名（N）、および、T ノードに定義される属性（図 9）を指定するユーザの操作を受け入れ、A R エントリ作成部 202 および I D エントリ作成部 204 に対して出力する。20

例えば、管理操作受入部 200は、A ノード、T ノードおよびこれらの間の関係などを、図 14 に示したように表示するユーザインターフェース（U I）画像を、入出力装置 126 に表示し、この U I 画像に対するユーザの操作を受け入れて、これらの指定を受け入れる。

#### 【0117】

A R エントリ作成部 202 は、管理操作受入部 200 から入力されるユーザの指定に従って、図 11 に示した A R テーブルのエントリを作成し、A R D B 管理部 206 に対して出力する。30

A R D B 管理部 206 は、A R エントリ作成部 202 から入力される A R テーブルのエントリを、A R D B 240 に記憶されている A R テーブルに追加する。

また、A R D B 管理部 206 は、管理操作受入部 200 から入力されるユーザの操作に従って、A R D B 240 に記憶されている A R テーブルの内容を変更する。

#### 【0118】

また、A R D B 管理部 206 は、A R D B 検索部 266 の検索に応じて、A R D B 240 に記憶された A R テーブルのエントリを探しだし、A R D B 検索部 266 に対して出力する。40

I D エントリ作成部 204 は、管理操作受入部 200 から入力される検索者の指定に従って、図 12, 図 13 に示した T ノード用および A ノード用の I D テーブルのエントリを作成し、I D D B 管理部 208 に対して出力する。

#### 【0119】

I D D B 管理部 208 は、I D エントリ作成部 204 から入力される T ノード用の I D テーブルのエントリを、I D D B 242 に記憶されている T ノード用の I D テーブルに追加する。

また、I D D B 管理部 208 は、I D エントリ作成部 204 から入力される A ノード用の I D テーブルのエントリを、I D D B 244 に記憶されている A ノード用の I D テーブルに追加する。50

また、I D D B 管理部 2 0 8 は、管理操作受入部 2 0 0 から入力されるユーザの操作に従って、I D D B 2 4 2 , 2 4 4 に記憶されている I D テーブルの内容を変更する。

また、I D D B 管理部 2 0 8 は、I D D B 検索部 2 6 8 の検索に応じて、I D D B 2 4 2 , 2 4 4 に記憶された I D テーブルのエントリを探し出し、I D D B 検索部 2 6 8 に対して出力する。

#### 【 0 1 2 0 】

D B 検索部 2 6 において、検索操作受入部 2 6 0 は、図 1 4 ~ 図 1 7 に示した検索処理に用いられる検索条件（図 1 4 、さらに任意の T ノードの属性（ノードタイプ（N T ））が含まれることがある）を指定するための検索者の操作を、入出力装置 1 2 6 （図 7 ）から、あるいは、ネットワーク 1 0 0 を介して P C 1 0 2 （図 6 ）から受け入れる。

検索操作受入部 2 6 0 は、受け入れた操作を、検索条件作成部 2 6 2 に対して出力する。

#### 【 0 1 2 1 】

検索条件作成部 2 6 2 は、例えば、検索操作受入部 2 6 0 が、自然言語による質問文の形式で検索条件を受け入れるときには、この質問文を構文解析して単語を取り出す。

次に、検索条件作成部 2 6 2 は、A R D B 検索部 2 6 6 、I D D B 検索部 2 6 8 、A R D B 管理部 2 0 6 およびI D D B 管理部 2 0 8 を介してA R D B 2 4 0 およびI D D B 2 4 2 , 2 4 4 に記憶されたA R テーブルおよびI D テーブルを検索して、検索条件として用いられ得る単語を抽出する。

さらに、検索条件作成部 2 6 2 は、抽出された単語を、質問文の構造に沿って組み合わせ、図 1 4 に示した形式の検索条件（R r e t , ( A N T 1 , A N T 2 , . . . ) , ( ( R 1 , N 1 ) , ( R 2 , N 2 ) , . . . , ( R n , N n ) ) ）を導出し、検索制御部 2 6 4 に対して出力する。

#### 【 0 1 2 2 】

なお、検索者が、図 1 4 に示した形式（R r e t , ( A N T 1 , A N T 2 , . . . ) , ( ( R 1 , N 1 ) , ( R 2 , N 2 ) , . . . , ( R n , N n ) ) ）で検索条件を、直接、指定するときには、検索条件作成部 2 6 2 は省略され得る。

また、検索条件作成部 2 6 2 は、検索者による検索条件（R r e t , ( A N T 1 , A N T 2 , . . . ) , ( ( R 1 , N 1 ) , ( R 2 , N 2 ) , . . . , ( R n , N n ) ) ）の導出を補助するためのツールであってもよい。

#### 【 0 1 2 3 】

検索制御部 2 6 4 は、検索条件作成部 2 6 2 （検索操作受入部 2 6 0 ）から入力される検索条件（R r e t , ( A N T 1 , A N T 2 , . . . ) , ( ( R 1 , N 1 ) , ( R 2 , N 2 ) , . . . , ( R n , N n ) ) ）に従って、A R D B 検索部 2 6 6 およびI D D B 検索部 2 6 8 を制御し、図 1 5 ~ 図 1 7 に示したように、A R D B 管理部 2 0 6 およびI D D B 管理部 2 0 8 を介したA R D B 2 4 0 （A R テーブル；図 1 1 ）およびI D D B 2 4 2 , 2 4 4 （I D テーブル；図 1 2 , 図 1 3 ）の検索を行わせる。

また、検索制御部 2 6 4 は、検索条件に従った検索により検索結果（集合 P ；図 1 7 ）が得られたときには、この検索結果に基づいて応答を作成し、入出力装置 1 2 6 （図 7 ）に表示し、あるいは、ネットワーク 1 0 0 （図 6 ）を介して P C 1 0 2 の入出力装置 1 2 6 に表示し、検索者に示す。

#### 【 0 1 2 4 】

A R D B 検索部 2 6 6 は、検索制御部 2 6 4 の制御に従って、A R D B 管理部 2 0 6 を介してA R D B 2 4 0 （A R テーブル；図 1 1 ）を検索し、検索結果を検索制御部 2 6 4 に返す。

I D D B 検索部 2 6 8 は、検索制御部 2 6 4 の制御に従って、I D D B 管理部 2 0 8 を介してI D D B 2 4 2 , 2 4 4 （I D テーブル；図 1 2 , 図 1 3 ）を検索し、検索結果を検索制御部 2 6 4 に返す。

#### 【 0 1 2 5 】

[ 全体動作 ]

10

20

30

40

50

以下、図6、図7に示したDB装置12(DBプログラム2;図18)の全体的な動作を、具体例を挙げて説明する。

#### 【0126】

##### [ARテーブルおよびIDテーブルの作成]

まず、DBプログラム2のDB管理部20によるARテーブルおよびIDテーブルの作成処理を説明する。

図19は、図6、図7に示したDB装置12(DBプログラム2;図18)に対して入力されるデータと、これに含まれるデータの検索に含まれる検索条件とを例示する図である。

例えば、DBプログラム2の管理操作受入部200に対して、図19に示すように関連付けられたデータが入力される。 10

図19に示すデータには、以下のように関連付けられたAノードとTノードとが含まれ、Tノードにはそれぞれノード名が付されている(ただし、Aノードの属性(ノードタイプ(NT))およびノード名、Tノードの属性(ノードタイプ(NT))は、以下の(1)~(8)および図19において省略)。

#### 【0127】

(1)「AノードA1」と「TノードT11」とが関連付けられ、これらの間に「作者」という関連性役割Rが定義され、「TノードT11」には、「シェークスピア(同名の別人)」というノード名が付されている。

(2)「AノードA9」と「TノードT92,T41」とが関連付けられ、これらの間に「作品」および「作者」という関連性役割Rが定義され、「TノードT92,T42」には、「ベニスの商人」および「シェークスピア」というノード名が付されている。 20

(3)「AノードA4」と「Tノード41,T42」とが関連付けられ、これらの間に「作者」および「作品」という関連性役割Rが定義され、「TノードT42」には、「ハムレット」というノード名が付されている。

(4)「AノードA13」と「TノードT42」とが関連付けられ、これらの間に「脚本」という関連性役割Rが定義されている。

(5)「AノードA19」と「TノードT42」とが関連付けられ、これらの間に「原作」という関連性役割Rが定義されている。

(6)「AノードA10」と「TノードT42」とが関連付けられ、これらの間に「原作」という関連性役割Rが定義されている。 30

(7)「AノードA10」と「TノードT103,T101」とが関連付けられ、これらの間に「出版」および「翻訳版」という関連性役割Rが定義され、「TノードT103」には、「出版」というノード名が付されている。

(8)「AノードA19」と「TノードT191」とが関連付けられ、これらの間に「翻訳版」という関連性役割Rが定義されている。

#### 【0128】

管理操作受入部200は、入力されたデータを受け入れ、ARエントリ作成部202およびIDエントリ作成部204に対して出力する。

ARエントリ作成部202は、図19に示したデータから、ARテーブルの各エントリを作成し、ARDDB管理部206に対して出力する。 40

#### 【0129】

図20は、ARエントリ作成部202(図18)およびARDDB管理部206により作成され、ARDDB240に記憶されるARテーブルを例示する図である。

なお、以下の図において、nul1は、属性(ノードタイプ)・名称(ノード名)がないことを示す。

ARDDB管理部206は、ARエントリ作成部202から入力されたARテーブルのエントリを、ARDDB240に記憶されているARテーブルに、順次、追加する。

ARエントリ作成部202およびARDDB管理部206による処理の結果、図19に示したデータから、図20に示すようなARテーブルが作成され、ARDDB240に記憶さ 50

れる。

### 【0130】

図21は、IDエントリ作成部204(図18)およびIDDB管理部208により作成され、IDDB242に記憶されるTノード用のIDテーブルを例示する図である。

IDエントリ作成部204は、図19に示したデータから、Tノード用のIDテーブルの各エントリを作成し、IDDB管理部208に対して出力する。

IDDB管理部208は、IDエントリ作成部204から入力されたTノード用のIDテーブルのエントリを、IDDB242に記憶されているIDテーブルに、順次、追加する。

IDエントリ作成部204およびIDDB管理部208による処理の結果、図19に示したデータから、図21に示すようなTノード用のIDテーブルが作成され、IDDB242に記憶される。 10

### 【0131】

図22は、IDエントリ作成部204(図18)およびIDDB管理部208により作成されるAノードについてのIDテーブルを例示する図である。

また、IDエントリ作成部204は、図19に示したデータから、Aノード用のIDテーブルの各エントリを作成し、IDDB管理部208に対して出力する。

IDDB管理部208は、IDエントリ作成部204から入力されたIDテーブルのエントリを、IDDB244に記憶されているAノード用のIDテーブルに、順次、追加する。 20

IDエントリ作成部204およびIDDB管理部208による処理の結果、図19に示したデータから、図22に示すようなAノード用のIDテーブルが作成され、IDDB244に記憶される。 20

### 【0132】

#### [データの検索]

例えば、検索者が、DB装置12の入出力装置126(図7)に対して、「作家シェークスピアを作者とする作品のうち、戯曲ハムレットを原作とする翻訳版を出版している出版社名は?」と、検索条件を質問文の形式で入力すると、検索条件作成部262は、この質問文を分析し、図19に示すように、この質問文を、前半部分と後半部分の2つに分割する。 30

検索条件作成部262は、質問文の前半部分「作家シェークスピアを作者とする作品のうち、」について、「作品」に関するデータを探すということから、検索結果とされるべきTノードT<sub>ret</sub>の関連性役割R<sub>ret</sub>を「作品」とする。

また、検索条件作成部262は、前半部分の「シェークスピアを作者とする」の部分から、検索フィルタ(R1='作者', N1='シェークスピア')を作成する。

さらに、検索条件作成部262は、TノードT<sub>ret</sub>の関連性役割(R<sub>ret</sub>='作品')と、検索フィルタ(R1='作者', N1='シェークスピア')とから、質問文の前半に対応する検索条件(R<sub>ret</sub>, (ANT1), ((R1, N1)))=(作品, (null), ((作者, シェークスピア)))を作成する。

### 【0133】

また、検索条件作成部262は、後半部分「戯曲ハムレットを原作とする翻訳版を出版している出版社名は?」について、「出版している出版社」に関する情報を検索することから、検索結果とされるべきTノードT<sub>ret</sub>の関連性役割R<sub>ret</sub>を「出版」とする。 40

また、検索条件作成部262は、後半部分に含まれる「ハムレット」を「原作」とするという条件から、第1の検索フィルタ(R1='原作', N1='ハムレット')を作成し、さらに、後半部分に含まれる「翻訳版」から、第2の検索フィルタ(R2='翻訳版', N2='null(指定なし)')を作成する。

さらに、検索条件作成部262は、関連性役割(R<sub>ret</sub>='出版')と、第1の検索フィルタ(R1='原作', N1='ハムレット')および第2の検索フィルタ(R2='翻訳版', N2='null')とから、質問文の後半に対応する検索条件(R<sub>ret</sub> 50

, (A N T 1) , ((R 1, N 1) , (R 2, N 2))) = (出版, (n u l l) , ((原作, ハムレット), (翻訳版, n u l l)))を作成する。

### 【0134】

検索制御部264は、検索条件作成部262が作成した検索条件に基づいて、以下に示すように、ARDDB検索部266およびIDDB検索部268を介してARDDB240(ARテーブル；図20)およびIDDB242(Tノード用IDテーブル；図21)を検索し、検索結果を得る。

まず、質問文の前半部分から得られた検索条件から、

(1) 検索制御部264は、IDDB242(図18)に記憶されたTノードのIDテーブルを参照し、ノード名が「シェークスピア」であるTノードの識別子(ID)を全て探し出し、この処理の結果として、T11, T31, T41, T51, T81(図21)を得る。10

(2) 検索制御部264は、ARDDB240に記憶されたARテーブル(図20)を参照し、役割が「作者」であり、Tノード識別子が(1)の処理により得られたTノードの識別子と一致するAノード識別子を全て探し出し、この処理の結果として、A1, A4, A9を得る。

(3) 検索制御部264は、ARテーブルを参照し、(2)の処理により得られたAノード識別子のうち、関連性役割Rが「作品」であるAノードに対応するTノードの識別子(ID；一般には複数個)を探し出し、この処理の結果として、T42, T92を得る。

(4) 検索制御部264は、IDテーブルを参照し、識別子が(3)の処理により得られたTノード識別子に対応するノード名を、そのTノードの識別子(ID)とともに、検索結果とする。20

つまり、検索制御部264は、(1)～(4)の処理により、質問文の前半部分から得られた検索条件(作品, (n u l l), ((作者, シェークスピア)))に基づく検索を行い、検索結果(T42, ハムレット), (T92, ベニスの商人)を得る。

次に、質問文の後半部分から得られた検索条件、および、前半部分に対応する検索結果から、

(5) 検索制御部264は、質問文の前半部分に対応する検索結果(T42, ハムレット), (T92, ベニスの商人)の内、ノード名が、第1の検索フィルタ(原作, ハムレット)と対応する(T42, ハムレット)を選択し、そのノード識別子(ID)T42を得る。30

(6) 検索制御部264は、ARテーブルを参照し、役割が「原作」であり、Tノード識別子が、(5)の処理により得られたノード識別子(ID)と一致するAノードの識別子を全て探し出し、この結果としてA10, A19を得る。

(7) 検索制御部264は、ARテーブルを参照し、(6)の処理により得られたAノード識別子のうち、役割が「翻訳版」であるものを全て探し出し、この結果として、A10, A19を得る。

(8) 検索制御部264は、ARテーブルを参照し、(7)の処理により得られたAノード識別子のうち、役割が「出版」であるAノードに対応するTノード識別子を探し出し、この結果としてT103を得る(複数の出版社から翻訳版が出されている場合には、複数のTノード識別子が得られる)。40

(9) 検索制御部264は、IDテーブルを参照し、ノードIDが(8)で得られたTノード識別子に対応するノード名を、そのノード識別子とともに、検索結果(T103, 出版)とする。

### 【0135】

つまり、検索制御部264は、(5)～(9)の処理により、質問文の後半部分から得られた検索条件(出版, (n u l l), ((原作, ハムレット), (翻訳版, n u l l)))に基づく検索を行い、検索結果(T013, 出版)を得る。

(10) 検索制御部264は、検索結果を、入出力装置126(図7)などに表示し、検索者に示す。

## 【0136】

図23は、第2のDBシステム3の構成を例示する図である。

図23に示すように、第2のDBシステム3は、DB装置12と、DB装置12と同様なハードウェア(図7)上で、DBプログラム2のDB管理部20およびDB部24を実行するDB装置30と、DB装置12と同様なハードウェア上で、DBプログラム2のDB検索部26を実行する検索装置32とが、ネットワーク100を介して接続されて構成される。

このように、DBプログラム2は、必ずしも1つのコンピュータ上で実行されなくともよく、ネットワークを介して接続された複数のコンピュータに分散され、実行されてもよい。

10

## 【0137】

## [第3のDBシステム4]

以下、ディレクトリ構造により関連情報を記憶し、検索の対象とするように構成された本発明にかかる第3のDBシステム4を説明する。

図24は、本発明にかかる第3のDBシステム4の構成を例示する図である。

図24に示すように、DBシステム4は、PC102、検索装置40、DB管理装置5およびn台のDB装置6-1~6-nが、ネットワーク100を介して接続されて構成される。

## 【0138】

なお、以下、DB装置6-1~6-nなど、複数ある構成部分のいずれかを特定せずに記載するときには、単にDB装置6と記載することがある。

20

また、検索装置40、DB管理装置5およびDB装置6のハードウェアは、図7に示したような構成を探る。

また、図6に示した第1のDBシステム1におけるDB装置12(図6)およびそのDBプログラム2(図18)のように、DB装置6が検索機能を含むときには、検索装置40は不要とされることがある。

また、DBシステム4においては、DB装置6それぞれに対する複製装置(ミラーサーバ；DB装置6')が設けられることがある。

また、以下、DB装置6-1~6-nを、それぞれDB装置A~Nと記すことがある。

30

## 【0139】

次に、図24に示したDBシステム4における情報の表現を説明する。

図25は、図24に示したDBシステム4に記憶される情報のグラフ形式の表現を例示する図である。

図26は、図25に示した情報のディレクトリ情報ツリー形式の表現を例示する図である。

図8を参照して既に説明したように、AノードおよびTノードそれぞれに対してノード名(Aノードのノード名などが「関連ノードに対応する情報」であり、Tノードのノード名などが「見出ノードに対応する情報」である)などが定義され、AノードとTノードとの間に関連性役割が定義されると、これらのノードにより関連付けられる関連情報は、グラフ形式で、図25に例示するように表現されうる。

40

## 【0140】

なお、図25には、TノードおよびAノードのノードタイプ(クラス)を定義するノード(T1, T3-1, T3-2)それぞれに、ノード名として「ロケーション情報」・「店」・「エリア」が定義され、Tノード(T2-1, T2-2)それぞれに、ノード名として「店1」・「エリア3」が定義され、Aノード(A1)に、ノード名として「ロケーション」が定義され、TノードとAノードとの間それぞれに、関連性役割として、「建物」・「地域」が定義された場合が示されている。

また、AノードA2-1, A2-2は、ノードタイプ(クラス)を定義するノードと、そのインスタンス(実体に対応するノード)であるノードとの関連を示すので、これらのノードA2-1, A2-2は、利用者が目的に応じて独自に定義した役割ではなく、システ

50

ムにより事前に定義された「クラス」、「インスタンス」という2つの役割によりTノードを関連付けるという意味で、特別なノードである。

#### 【0141】

図25に示したようにグラフ形式で表現された関連情報は、図26に示すように、LDAP (Lightweight Directory Access Protocol)の情報モデルであるDIT (Directory Information Tree; ディレクトリ情報ツリー)形式によって表現されうる。

つまり、図25に示した関連情報は、DIT形式に従って、図26に点線で示すように、DB装置6それぞれの最上位のエントリとして仮想的に設けられる仮想ルートエントリ、および、第1, 第2・・・の関係(関係#1, #2...)を示すエントリの下に設けられ、図26に実線で示すように、Aノードに対応するエントリ、および関連性役割に対応し、Tノードの情報を属性としてもつエントリにより表現されうる。10

つまり、図26に例示した場合においては、ノードタイプ(assocType)が「ロケーション情報」である関連情報を示すエントリ「ロケーション」の直下に、2つの役割(role)「建物」と「地域」に対応するエントリが設けられ、それぞれの役割を果たすメンバ(member)「店1」と「エリア3」は対応するエントリの属性となっている。

図25に示したようにグラフ形式で表現された関連情報は、図26に示すように、LDAP (Lightweight Directory Access Protocol)の情報モデルであるDIT (Directory Information Tree; ディレクトリ情報ツリー)形式によって表現されうる。

#### 【0142】

なお、図26に示したDIT形式において、『ディレクトリ』は、実世界の様々な事物に関する情報を、できる限り実世界での関係に即したわかりやすい形で構成し、また、情報の在り処を持っていて、それらの情報の検索や更新の手段を提供するものとして定義される。20

また、『エントリ』は、実世界のオブジェクトに関する情報を、オブジェクトクラスに従って整理・分類し、ディレクトリ情報として表現したものとして定義され、あるいは、ディレクトリに格納したデータと定義される。

また、『ディレクトリ情報ツリー』は、エントリ(ディレクトリ情報)を、階層的に管理する際、その階層関係をツリー構造に表現したものとして定義される。

つまり、図26の各四角はエントリに対応し、これらの階層関係をツリー構造に表現したもののがディレクトリ情報ツリー(DIT)である。30

#### 【0143】

なお、上述したように、エントリはオブジェクトに関する情報の集まりであって、このオブジェクトに関する情報は『属性』と呼ばれる。

この属性は『属性型』と、1つ以上の値、『属性値(群)』からなり、図26においては、各エントリの構造が、『属性型：属性値』の形式で、下表21～表23に示すように、LDAPにおける各エントリの構造表現として例示されている。

例えば、「ou=関係#1」直下のエントリでは、属性型objectclass(オブジェクトクラス)の属性値が関連ノード、属性型cn(一般名)の属性値がロケーション、・・・というように、エントリの各属性が定義される。

#### 【0144】

##### 【表21】

dn: associd=関連ノード識別子
objectclass: 関連ノード
associd: 関連ノード識別子
assocType: ロケーション情報
cn: ロケーション
role: 建物
role: 地域

#### 【0145】

10

20

30

40

【表22】

```
dn: role=建物,associd=関連ノード識別子,…
objectclass: 関連役割
cn: 建物
role: 建物
member: 店1
```

【0146】

【表23】

```
dn: role=地域,associd=関連ノード識別子,…
objectclass: 関連役割
cn: 地域
role: 地域
member: エリア3
```

10

【0147】

図27は、ディレクトリ構造で表現された関連情報の分割方法を例示する図である。

図28は、図27に示したDB装置6-1(A)のエントリdn\_A2と、DB装置6-n(N)のエントリdn\_Nとの参照関係を示す図である。

図27に示すように、DIT形式により表現された関連情報は、ディレクトリ構造に従って分割されうる。

20

【0148】

図27に示す場合においては、関連情報は、エントリdn\_Aおよびその下位のエントリdn\_A1, dn\_A2を含むディレクトリ構造により表現されており、さらに、エントリdn\_A1以下のディレクトリに対応する関連情報は、エントリdn\_B～dn\_(N-1)に対応して分割され、エントリdn\_A2以下のディレクトリに対応する関連情報は、エントリdn\_N以下のディレクトリ構造に対応付けられている（なお、図27に示した例においては、エントリdn\_B～dn\_Nを最上位エントリとするディレクトリ構造がサブツリーであり、これらのサブツリー全てを含み、エントリdn\_Aを最上位エントリとするディレクトリ構造がディレクトリツリーである）。

30

例えば、図28に示すように、図27に示したDB装置6-n(N)の最上位エントリdn\_Nは、DB装置6-1(A)のエントリdn\_A2の下位にあって、これを参照する。

このように、ディレクトリ構造に従って分割された関連情報は、図27に示すように、DB装置6-1～6-nに、分割されて記憶されうる。

【0149】

図29は、図27に示したようにDB装置6-1～6-n(DB装置A～N)に分割されて記憶された関連情報を管理するために用いられるディレクトリツリーテーブルを例示する図である。

図27, 図28に示したように分割されてDB装置6-1～6-n(A～N)に記憶された関連情報を管理するためには、図29に示すようなディレクトリツリーテーブルが用いられる。

40

図29に示すように、ディレクトリツリーテーブルには、DB装置6-1～6-nそれぞれの装置名(A～N)と、これらのDB装置6それぞれにおける最上位エントリ(dn\_A～dn\_N; 仮想ルートエントリ(図26)に同じ)と、DB装置6それぞれの最上位エントリが参照するエントリ(上位参照エントリ)と、各DB装置6に複製装置(ミラーサーバ(第2のデータベース装置))があるときのその装置名(A', C'など)とが、対応付けられて含まれる。

【0150】

図30は、図27, 図28に示したDB装置6-1～6-n(A～N)に記憶されてい

50

るディレクトリツリー（またはサブツリー）それぞれの最上位エントリ  $d_n_A \sim d_n_N$  と、ディレクトリツリー（またはサブツリー）それぞれに記憶されている関連情報の分類に用いられる分類型・分類値（分類情報）とを対応付ける情報テーブルを例示する図である。

図 25, 図 26 に示したように、DIT 形式のディレクトリ構造で表現され、このディレクトリ構造に従って分割され、DB 装置 6-1~6-n (A~N) それぞれに記憶された関連情報それは、DB 装置 6-1~6-n (A~N) それおにおいて関連性を有する情報の集合としてとらえられる。

#### 【0151】

従って、図 27, 図 28 に示した DB 装置 6-1~6-n (A~N) に記憶されているディレクトリツリー（またはサブツリー）それおに対して、そこに記憶されている関連情報の分類に用いるための属性などを示す分類型および分類値を定義することができる。

DB 装置 6-1~6-n (A~N) それぞれに記憶された関連情報の管理のためには、図 30 に示すように、図 27, 図 28 に示した DB 装置 6-1~6-n (A~N) の最上位エントリ  $d_n_A \sim d_n_N$  の下のサブツリー  $a \sim n$  それおと、最上位エントリ ( $d_n_A \sim d_n_N$ ) それおと、これらの最上位エントリ以下のサブツリーに対して定義される分類型および分類値それおとが対応付けて含まれる情報テーブルが用いられる。

#### 【0152】

なお、図 30 に示した「分類型」は、関連タイプ（A ノードのノードタイプ（図 22）；属性）に対応する。

また、「分類値」は、LDAP 情報モデルにおけるインスタンスに対応する。

図 27～図 30 に示した最上位エントリとなり得るエントリとして、  
(1) ある分類型（関連タイプ；A ノードのノードタイプ）の特定の分類値（インスタンス）を含むサブツリーのルートエントリ、および、  
(2) ある特定の分類値（インスタンス）が含まれる関連情報を含むサブツリーのルートエントリ

などを、具体例として挙げることができる。

#### 【0153】

ただし、ここには、「分類型」が関連タイプに対応し、「分類値」がそのインスタンスに対応する場合が例示されているが、これは、図 30 を図 27, 図 28 に対応付けた一例であって、「分類型」は、関連タイプに対応しないことがあり、また、「分類値」は、そのインスタンスに対応しないことがある。

例えば、図 30 に示されたサブツリー  $a$  には、「分類型：カテゴリ、分類値：提携会社情報」の関連情報が含まれ、これらは、「分類型」が関連タイプに対応せず、「分類値」がそのインスタンスに対応しない例である。

#### 【0154】

また、例えば、図 30 に示した情報テーブル中の最上位エントリ  $d_n_B$  を含むエントリの分類型として示された「位置情報」は、上記(1)「ある分類型（関連タイプ；A ノードのノードタイプ）の特定の分類値（インスタンス）を含むサブツリーのルートエントリ」の分類型の一例であり、その分類値の 1 つとして分類値「ロケーション(located-in)」を含む。

このエントリは、分類値として「ロケーション(located-in)」を有する全ての関連情報が、DB 装置 6-2 (B) に記憶されることを示している。

分類型「位置情報」の分類値のその他の例として、分類値「隣接(next-to)」などを挙げることができる。

#### 【0155】

あるいは、図 30 に示した最上位エントリ  $d_n_{(N-1)}$  を含むエントリの分類型として示された「建物」は、上記(2)「ある特定の分類値（インスタンス）が含まれる関連情報を含むサブツリーのルートエントリ」の分類型の一例であり、その分類値の 1 つとして分類値「店 1」を含む。

10

20

30

40

50

このエントリは、分類値として「店1(Store\_1)」を有する全ての関連情報が、DB装置6-(n-1)(N-1)に記憶されることを示している。

### 【0156】

#### [関連情報の作成および検索]

以下、図25～図30を参照して説明したDIT形式のディレクトリ構造により表現された関連情報の作成、および、作成された関連情報に対する検索を説明する。

### 【0157】

#### [第1のDB管理プログラム50]

図31は、図24に示したDB管理装置5において動作する第1のDB管理プログラム50の構成を示す図である。10

図31に示すように、DB管理プログラム50は、ユーザインターフェース(UI)部500、情報登録部502、情報送信部504、関連情報作成・管理部510(ディレクトリツリー作成手段および情報対応付手段)、ツリー情報提供部512、ディレクトリツリーテーブル作成部522(最上位エントリ対応付手段)、ディレクトリツリーテーブル管理部524、ディレクトリツリーテーブルDB526、情報テーブル作成部532(分類情報対応付手段)、情報テーブル管理部534および情報テーブルDB536から構成される。

### 【0158】

DB管理プログラム50は、これらの構成部分により、図25～図26に示したように、DIT形式のディレクトリ構造で表現された関連情報を作成する。20

また、DB管理プログラム50は、図27、図28に示したように、作成した関連情報を分割してDB装置6-1～6-nに記憶させ、図29、図30に示したディレクトリツリーテーブルおよび情報テーブルを用いて関連情報を管理する。

また、DB管理プログラム50は、検索装置40(図24)に対して、ディレクトリツリーテーブルおよび情報テーブルの情報に含まれ、検索のために必要とされる情報(ツリー情報)を、適宜、提供する。

### 【0159】

図32は、図31に示したDB管理プログラム50が、DB管理装置5の入出力装置126(図7)に表示するGUI画像を例示する図である。

DB管理プログラム50において、UI部500は、DB管理装置5あるいはPC102のユーザのためにGUI環境を提供し、入出力装置126(図7)などに対して、図32に示すようなGUI画像を表示する。30

さらに、UI部500は、表示されたGUI画像に対してユーザが行った関連情報の登録、管理および変更のための操作を受け入れ、情報登録部502および関連情報作成・管理部510など、DB管理プログラム50の各構成部分に対して出力し、これらの動作を制御する。

また、UI部500は、DB管理プログラム50の各構成部分が作成した情報を、入出力装置126に対して表示・出力する。

なお、UI部500は、DB管理装置5におかれてても、PC102におかれててもよく、PC102において、上記機能をユーザに対して提供してもよい。

### 【0160】

関連情報作成・管理部510は、ディレクトリツリーテーブル作成部522および情報テーブル作成部532などの動作を制御して、UI部500から入力されるTノード、Aノードおよび関連属性を示す情報から、図26に示したように、DIT形式のディレクトリ構成により表現される関連情報を作成し、作成した関連情報を、図29、図30に示したディレクトリツリーテーブルおよび情報テーブルを用いて管理する。

また、関連情報作成・管理部510は、要求に応じて、ディレクトリツリーテーブルおよび情報テーブルに含まれ、検索装置40による検索のために用いられる情報(ツリー情報)を、ツリー情報提供部512に対して出力する。

また、関連情報作成・管理部510は、ユーザの操作などに従って、図27、図28に50

示したように、D I T 形式のディレクトリ構造に表現された関連情報を分割し、情報登録部 502 に対して出力する。

#### 【0161】

情報登録部 502 は、関連情報作成・管理部 510 から入力される関連情報を、情報送信部 504 を介して、DB 装置 6-1 ~ 6-n それぞれに対して送信し、登録を要求する。

ツリー情報提供部 512 は、検索装置 40 からの要求に応じて、関連情報作成・管理部 510 から入力されるツリー情報を提供する。

#### 【0162】

ディレクトリツリーテーブル作成部 522 は、関連情報作成・管理部 510 の制御に従って、図 29 に示したディレクトリツリーテーブルを作成し、ディレクトリツリーテーブル管理部 524 に対して出力する。

ディレクトリツリーテーブル管理部 524 は、ディレクトリツリーテーブル作成部 522 から入力されたディレクトリツリーテーブルを、ディレクトリツリーテーブル DB 526 に記憶させる。

また、ディレクトリツリーテーブル DB 526 は、要求に応じて、ディレクトリツリー テーブル DB 526 に記憶させたディレクトリツリーテーブルを、関連情報作成・管理部 510 に対して出力する。

なお、図 18 に示した DB プログラム 2 においては、DB 管理部 20 が、関連情報から A R テーブルおよび I D テーブルを作成する処理を行うのに対し、図 31 に示した DB 管理プログラム 50 においては、関連情報作成・管理部 510 が、ディレクトリ構造の関連情報（図 21, 図 22）を作成し、DB 装置 6 が、関連情報作成・管理部 510 および情報送信部 504 を介して、作成されたディレクトリ構造の関連情報を受け取り、記憶・管理する。

#### 【0163】

情報テーブル作成部 532 は、関連情報作成・管理部 510 の制御に従って、図 30 に示した情報テーブルを作成し、情報テーブル管理部 534 に対して出力する。

情報テーブル管理部 534 は、情報テーブル作成部 532 から入力された情報テーブルを、情報テーブル DB 536 に記憶させる。

また、情報テーブル管理部 534 は、要求に応じて、情報テーブル DB 536 に記憶させた情報テーブルを、関連情報作成・管理部 510 に対して出力する。

#### 【0164】

以下、DB 管理プログラム 50 の全体的な処理を説明する。

図 33 は、図 31 に示した DB 管理プログラム 50 による関連情報の登録処理 (S30) を示すフローチャートである。

図 33 に示すように、ステップ 300 (S300) において、DB 管理プログラム 5 または P C 102 の U I 部 500 は、入出力装置 126 に、図 32 に例示した G U I 画像を表示する。

ユーザは、表示された G U I 画像に対して操作を行い、新たに追加する関連情報に関する情報（データ）を各フィールドに入力する。

#### 【0165】

ステップ 302 (S302) において、関連情報作成・管理部 510 は、S300 の処理において入力した関連情報に基づき、関連性（テーマ；例えば「シェークスピア作品」など）を同定する。

ステップ 304 (S304) において、関連情報作成・管理部 510 は、関連情報のノードタイプ（分類型；例えば「著作情報」）を同定し、これをエントリの属性として設定する。

ステップ 306 (S306) において、関連情報作成・管理部 510 は、情報テーブル管理部 534 を介して、情報テーブルを参照し、設定されたテーマおよびノードタイプにより、作成されるエントリが、いずれのサブツリーに分類されるかを特定する。

10

20

30

40

50

## 【0166】

ステップ310(S310)において、関連情報作成・管理部510は、ディレクトリツリー・テーブル管理部524を介して、ディレクトリツリー・テーブルを参照し、S306の処理により求められたサブツリーが、いずれのDB装置6に記憶されているかを特定する。

ステップ312(S312)において、関連情報作成・管理部510は、UI部500を介して入力された関連情報に関するデータに基づき、それに対応するエントリを作成する。

## 【0167】

ステップ314(S314)において、関連情報作成・管理部510は、情報登録部502および情報送信部504を介して、S310の処理により特定されたDB装置6に、追加された関連情報のエントリを送信する。DB装置6は、受信した関連情報のエントリが既に記憶されているか否かを判断する。10

なお、関連情報作成・管理部510は、DB装置6に記憶する際の前処理として、DB装置6内の情報を利用する。

この情報には、各エントリdn(distinguished name:識別名)が割り当てられており、この識別名により、DIT内の全てのエントリが一意に識別され得るので、同一のエントリの存在の有無が判断されうる。

DB管理プログラム50は、追加された関連情報のエントリが既に存在するとの結果をDB装置6から受信したときには処理を終了し、これ以外のときにはS316の処理に進む。20

ステップ316(S316)において、関連情報作成・管理部510は、追加された関連情報のエントリを、特定されたDB装置6に対して登録する。

## 【0168】

図34は、図31に示したDB管理プログラム50による情報テーブル(図30)の変更処理(S34)を示すフローチャートである。

ステップ340(S340)において、ユーザは、DB管理装置5(図24)またはPC102に対して、変更しようとするサブツリーを指定する操作を行い、さらに、指定したサブツリーの分類型・分類値を設定する操作を行う。

関連情報作成・管理部510は、UI部500を介してこれらの操作を受け入れる。30

ステップ342(S342)において、ユーザは、DB管理装置5(図24)またはPC102に対して、指定したサブツリーの最上位エントリを設定する操作を行う。

関連情報作成・管理部510は、UI部500を介して、この操作を受け入れる。

## 【0169】

ステップ344(S344)において、関連情報作成・管理部510は、S340,S342の処理により受け入れた操作が示す分類型、分類値および最上位エントリの全てが、指定されたサブツリーに対して設定されているか否かを判断する。

関連情報作成・管理部510は、情報テーブル管理部534を介して情報テーブルを読み出し、これらの全てが指定されたサブツリーに対して設定されているときには処理を終了し、これ以外のときにはS346の処理に進む。40

## 【0170】

ステップ346(S346)において、関連情報作成・管理部510は、受け入れた設定値(指定されたサブツリー、分類型、分類値および最上位エントリ)を、情報テーブル管理部534に対して出力する。

情報テーブル管理部534は、関連情報作成・管理部510から入力された設定値を、情報テーブルに設定し、情報テーブルDB536に記憶する。

## 【0171】

図35は、図31に示したDB管理プログラム50によるディレクトリツリーテーブルの変更処理(S36)を示すフローチャートである。

図35に示すように、ステップ360(S360)において、UI部500は、DB管50

理装置 5 から、ユーザによるディレクトリ情報ツリー分割のための操作を受ける。

ステップ 362 (S362)において、関連情報作成・管理部 510 は、ディレクトリツリーテーブル管理部 524 からディレクトリツリーテーブル（図 29）を得て、DB 装置 6 の配置およびそのディレクトリ構造を示す情報を取得し、ユーザに対して表示する。

#### 【0172】

ステップ 364 (S364)において、UI 部 500 は、表示したディレクトリ構造に対するユーザの操作に応じて、分割後のサブツリーの最上位エントリおよびその上位参照エントリの設定を受ける。

ステップ 366 (S366)において、関連情報作成・管理部 510 は、ユーザによるディレクトリ情報ツリー分割のための操作に基づき、分割されたサブツリーが記憶される DB 装置 6 に、複製装置を設定するか否かを判断する。10

DB 管理プログラム 50 は、複製装置を設定するときには S368 の処理に進み、これ以外のときには S370 の処理に進む。

#### 【0173】

ステップ 368 (S368)において、関連情報作成・管理部 510 は、S366 の処理において複製装置を設定すると判断されたとき、その複製装置を特定する。

ステップ 370 (S370)において、関連情報作成・管理部 510 は、分割後のサブツリーの最上位エントリとその上位参照エントリおよび複製装置を設定するときにはその複製装置名を、ディレクトリツリーテーブル管理部 524 に対して出力する。

ディレクトリツリーテーブル管理部 524 は、関連情報作成・管理部 510 から入力されたこれらの情報を、ディレクトリツリーテーブル DB526 の所定の部分に書き込みに反映させる。20

#### 【0174】

ステップ 372 (S372)において、関連情報作成・管理部 510 は、全ての DB 装置 6 についてのディレクトリツリーテーブルの変更処理が終了したか否かを判断する。

DB 管理プログラム 50 は、全ての DB 装置 6 についての変更処理が終了していないときには S364 の処理に戻る。

#### 【0175】

次に、DB 管理装置 5 の全体的な処理を説明する。

図 36 は、図 31 に示した DB 管理プログラム 50 の全体的な動作 (S40) を示すシーケンス図である。30

図 36 に示すように、ステップ 400 (S400) ユーザは、DB 管理プログラム 50 の UI 部 500 に対して、関連情報を入力する。

ステップ 402 (S402) において、UI 部 500 は、入力された関連情報に関する情報（データ）を、関連情報作成・管理部 510 に対して出力する。

関連情報作成・管理部 510 は、これらの情報を受け入れる。

#### 【0176】

ステップ 404 (S404) において、関連情報作成・管理部 510 は、受け入れた関連情報に関する情報のうち、テーマおよびノードタイプなど関連情報の分類に用いられる分類情報（分類型・分類値）を、情報テーブル管理部 534 に対して出力する。40

ステップ 406 (S406) において、情報テーブル管理部 534 は、関連情報作成・管理部 510 からの関連情報の分類に用いられる情報を受け入れ、これらに対応するサブツリーの最上位エントリを、関連情報作成・管理部 510 に返す。

関連情報作成・管理部 510 は、このサブツリー最上位エントリを受け入れる。

なお、これらに対応するサブツリーがない場合には、対応するサブツリー最上位エントリの代わりに、DB 装置 6 - 1 の最上位エントリ（すなわち、ルートエントリ）を関連情報作成・管理部 510 に返す。

#### 【0177】

S406 の処理においては、情報テーブル（図 30）に格納されている分類型・分類値に対応する情報が用いられ、図 32 に示した GUI 画像を介して入力される関連情報に關50

するデータは、全て、関連情報作成・管理部 510 で利用可能である。

つまり、関連情報作成・管理部 510 が、ノードタイプと特定のインスタンスとを組にして情報テーブル管理部 534 に対して与えることにより、情報テーブル管理部 534 は、関連情報の分類に用いられる情報に対応するサブツリーの最上位エントリを得ることができ、他には、例えば、「カテゴリ」が分類型として、「シェークスピア作品」が分類値として、情報テーブル管理部 534 に与えられ得る。

#### 【0178】

ステップ 408 (S408)において、関連情報作成・管理部 510 は、ディレクトリツリー・テーブル管理部 524 に対して、S406 の処理により得られたサブツリー最上位エントリを出力する。10

ディレクトリツリー・テーブル管理部 524 は、このサブツリー最上位エントリを受け入れる。

ステップ 410 (S410)において、ディレクトリツリー・テーブル管理部 524 は、受け入れたサブツリー最上位エントリを用いてディレクトリツリー・テーブル DB526 を検索し、このサブツリーを記憶する DB 装置 6 を特定する。特定された DB 装置 6 を示す情報を、関連情報作成・管理部 510 に返す。

#### 【0179】

ステップ 412 (S412)において、関連情報作成・管理部 510 は、DIT 形式とした関連情報と、特定された DB 装置 6 を示す情報を情報登録部 502 に対して出力する。20

情報登録部 502 は、入力された関連情報を、特定された DB 装置 6 に対して転送し、登録する。

ステップ 416～422 (S416～S422)において、情報登録部 502 から UUI 部 500 に対して、登録の結果（正常終了または異常終了）が通知され、ユーザに対して表示される。

#### 【0180】

##### [第 2 の DB プログラム 60]

図 37 は、図 24 に示した DB 装置 6 それぞれにおいて動作する第 2 の DB プログラム 60 を示す図である。

図 37 に示すように、DB プログラム 60 は、情報管理部 600、情報 DB602、情報通信部 604 および検索実行部 606 から構成される。30

DB プログラム 60 は、これらの構成部分により、DB 管理装置 5 (DB 管理プログラム 50; 図 31) から、DIT 形式のディレクトリ構造で表現され、適宜、分割された関連情報の登録を受け、記憶する。

また、DB プログラム 60 は、検索装置 40 からの検索に応じて、登録された関連情報を提供する。

また、DB プログラム 60 は、DB 管理装置 5 からの制御に従って、登録された関連情報の変更を行う。

#### 【0181】

DB プログラム 60 において、情報通信部 604 は、DB 管理装置 5 による関連情報およびその登録要求、および、関連情報の変更要求を受け入れ、情報管理部 600 に対して出力する。40

検索実行部 606 は、検索装置 40 からの検索条件（図 38 に示して後述する検索プログラム 42 の検索条件作成部 420 により作成される LDAp 操作、および、図 50 に示す LDAp 操作）に従って、情報管理部 600 に対して、関連情報の検索を行う。

また、検索実行部 606 は、検索の結果として情報管理部 600 から入力される関連情報を、検索を要求した検索装置 40 に対して返す。

#### 【0182】

情報管理部 600 は、DB 管理装置 5 からの登録要求および変更要求に従って、DB 管理装置 5 または他の DB 装置 6 から送られてきた関連情報（図 26～図 28）を、情報 D

10

20

40

50

B 6 0 2 に記憶させる。

また、情報管理部 6 0 0 は、検索装置 4 0 による検索に応じて、情報 D B 6 0 2 から関連情報を読み出して、検索実行部 6 0 6 に対して出力する。

#### 【 0 1 8 3 】

[ 第 1 の検索プログラム 4 2 ]

図 3 8 は、図 2 4 に示した検索装置 4 0 において動作する検索プログラム 4 2 を示す図である。

図 3 8 に示すように、検索プログラム 4 2 は、図 1 8 に示した D B 検索部 2 6 に、検索条件作成部 4 2 0 および検索結果出力部 4 2 2 を付加した構成をとる。

検索プログラム 4 2 は、これらの構成部分により、検索者（ユーザ）の検索操作に従って、D B 装置 6 に対する検索処理を行い、検索の結果として得られた関連情報を検索者に示す。

#### 【 0 1 8 4 】

検索プログラム 4 2 において、検索操作受入部 2 6 0 は、検索装置 4 0 の入出力装置 1 2 6（図 7 ）などに対して、検索者が入力した検索操作を受け入れ、検索条件作成部 4 2 0 に対して出力する。

検索条件作成部 4 2 0（ディレクトリツリー検索手段）は、検索操作受入部 2 6 0 から入力された検索操作の内容を解析し、さらに、この検索操作の内容に適合したツリー情報（ディレクトリツリーテーブルおよび情報テーブル（図 2 9 , 図 3 0 ）の内、検索に必要とされる情報）を、D B 管理装置 5 から得て、検索操作およびツリー情報に適合した検索条件を作成し、検索制御部 2 6 4 に対して出力する。

#### 【 0 1 8 5 】

具体的には、検索条件作成部 4 2 0 は、検索操作の内容を解析して得られる、関連情報の分類型および分類値に従って、D B 管理装置 5 からの情報テーブル（図 3 0 ）を受けてその内容を検索し、検索操作の内容に適合した関連情報の最上位エントリを求める（但し、このような最上位エントリがない場合には、検索条件作成部 4 2 0 は、D B 装置 6 - 1 の最上位エントリ d n \_ A を、検索操作の内容に適合した関連情報の最上位エントリとする）。

さらに、検索条件作成部 4 2 0 は、D B 管理装置 5 から受けたディレクトリツリーテーブル（図 2 9 ）の内容を検索し、情報テーブルの検索により求められた最上位エントリを記憶する D B 装置 6 を特定する。

さらに、検索条件作成部 4 2 0 は、検索操作の内容、および、情報テーブルから得られる関連情報の分類に用いる属性（分類型・分類値）を用いて、D B 装置 6 に実行させるべき L D A P 操作のコマンドおよびパラメータを作成し、検索条件として、検索制御部 2 6 4 に対して出力する。

#### 【 0 1 8 6 】

検索制御部 2 6 4（情報検索手段）は、検索条件作成部 4 2 0 から入力される検索条件を D B 装置 6 に対して出力し、関連情報の検索の制御を行う。

検索結果出力部 4 2 2 は、検索の結果として D B 装置 6 から返された関連情報を、検索装置 4 0 の入出力装置 1 2 6（図 7 ）などに対して表示・出力し、検索者に示す。

#### 【 0 1 8 7 】

以下、検索プログラム 4 2 の全体的な処理を説明する。

図 3 9 は、図 3 8 に示した検索プログラム 4 2 による検索処理（S 4 4 ）を示すフローチャートである。

図 3 9 に示すように、ステップ 4 4 0（S 4 4 0 ）において、検索操作受入部 2 6 0 は、ユーザによる検索条件を受け入れる。

#### 【 0 1 8 8 】

ステップ 4 4 2（S 4 4 2 ）において、検索操作受入部 2 6 0 は、検索要求メッセージ（図 5 0 を参照して後述）を作成し、検索条件作成部 4 2 0 に対して出力する。

ステップ 4 4 4（S 4 4 4 ）において、検索条件作成部 4 2 0 は、検索要求メッセージ

10

20

30

40

50

中に、分類型および分類値の指定があるか否かを判断する。

検索条件作成部420は、分類型および分類値の指定があるときにはS446の処理に進み、これ以外のときにはS454の処理に進む。

#### 【0189】

ステップ446(S446)において、検索条件作成部420は、DB管理装置5に対して分類型および分類値を送り、これらに対応するサブツリーの最上位エントリを特定する。

ステップ448(S448)において、検索条件作成部420は、S446の処理により、サブツリーが特定されたか否かを判断する。

検索プログラム42は、サブツリーが特定されたときにはS450の処理に進み、これ以外のときにはS454の処理に進む。 10

#### 【0190】

ステップ450(S450)において、検索条件作成部420は、DB管理装置5に対して特定された最上位エントリを送り、この最上位エントリに対応するサブツリーを記憶するDB装置6を特定する。

ステップ452(S452)において、検索条件作成部420は、特定されたサブツリーの最上位エントリを、検索を開始するエントリ(検索ベース)とする。

ステップ454(S454)において、検索条件作成部420は、DB管理装置5からディレクトリツリーテーブルを受け、受けたディレクトリツリーテーブルを参照し、全てのDB装置6に記憶されているサブツリーを通して、最も上位のエントリ(ルートエントリ)を検索ベースとし、このエントリを記憶するDB装置6を特定する。 20

#### 【0191】

ステップ456(S456)において、検索制御部264は、検索条件作成部420から、検索要求および検索ベースなどの情報を得て、特定されたDB装置6に対するLDAP操作コマンドを作成する。

ステップ458(S458)において、検索制御部264は、作成したLDAP操作コマンドを用いて、DB装置6に対する検索を実行し、その結果を受けて出力する。

#### 【0192】

以下、さらに、検索プログラム42の処理を、DB管理プログラム50の処理と関連付けて、全体的に説明する。 30

図40は、図31に示したDB管理プログラム50および図38に示した検索プログラム42の全体的な処理(S52)を示すシーケンス図である。

図40に示すように、ステップ520(S520)において、検索プログラム42の検索操作受入部260は、ユーザの問い合わせ操作を受け入れる。

ステップ522(S522)において、検索操作受入部260は、検索条件作成部420に対して、検索要求メッセージを送信する。

#### 【0193】

ステップ526(S526)において、検索条件作成部420は、DB管理プログラム50の関連情報作成・管理部510に対して、検索要求メッセージに含まれる分類型および分類値を送信する。 40

なお、S526の処理は、S446において、検索条件作成部420からDB管理装置5に対して分類型・分類値を送信する処理に対応する。

ステップ528(S528)において、関連情報作成・管理部510は、検索条件作成部420から受けた分類型・分類値を情報テーブル管理部534に対して出力する。

#### 【0194】

ステップ530(S530)において、情報テーブル管理部534は、関連情報作成・管理部510から入力された分類型・分類値を用いて情報テーブルを参照し、入力された分類型・分類値に対応する最上位エントリを得る。

情報テーブル管理部534は、得られた最上位エントリを、関連情報作成・管理部510に返す。 50

## 【0195】

ステップ532(S532)において、関連情報作成・管理部510は、S530の処理により得られた最上位エントリをディレクトリツリー・テーブル管理部524に対して出力する。

ステップ534, 536(S534, S536)において、ディレクトリツリー・テーブル管理部524は、入力された最上位エントリに対応するDB装置6を、関連情報作成・管理部510を介して検索条件作成部420に返す。

## 【0196】

ステップ538(S538)において、検索条件作成部420は、検索条件を作成し、特定されたDB装置6の名前などとともに、検索制御部264に対して出力する。

ステップ540(S540)において、検索制御部264は、LDAPコマンドを作成し、特定されたDB装置6にアクセスして検索を行う。

ステップ542～546(S542～S546)において、DB装置6から検索結果が返され、ユーザに対して表示される。

## 【0197】

[DBシステム4における関連情報およびその検索の例(1)]

以下、DBシステム4においてどのように関連情報が登録され、登録された関連情報が検索されるかについて、具体例を挙げて説明する。

図41は、図24に示したDBシステム4においてDB装置6に登録されるディレクトリ構造の関連情報を例示する第1の図であって、フラットなディレクトリ構造を示す。

図41に示されたディレクトリ構造においては、図26に点線で示されたエントリ(o = 関係 #1～#n)がなく、ルートエントリの直下に全ての関連情報がある。

このようなディレクトリ構造を、「フラットな構造のディレクトリ構造」と呼ぶ。

## 【0198】

ユーザが、DB管理装置5(図24)あるいはPC102の入出力装置126(図7)に表示されたGUI画像(図32)に対して、順次、関連情報を登録するための操作を行う。

この操作により、(関連名称、関連型、名称、役割)の形式で、DB管理装置5に対して、以下に示すような、

(ハムレット・著作について、著作情報、シェークスピア、作者),

(ハムレット・著作について、著作情報、ハムレット、作品),

(ベニスの商人・著作について、著作情報、シェークスピア、作者),

(ベニスの商人・著作について、著作情報、ベニスの商人、作品),

(ハムレット・日本語訳について、翻訳情報、ハムレット、原作),

(ハムレット・日本語訳について、翻訳情報、ハムレット日本語訳、翻訳版),

(ハムレット・日本語訳について、翻訳情報、出版、〇〇出版), . . .

などの情報が入力される。

## 【0199】

ユーザによる登録操作の結果として、図41に示すようなDIT形式のディレクトリ構造により関連情報が作成される。

なお、図41に例示するフラットな構造のディレクトリ構造においては、ディレクトリ情報ツリーをサブツリーに分割することは考慮されておらず、分類型および分類値は、ディレクトリ情報ツリー全体に対して定義されるが、必要に応じて、フラットな構造のディレクトリ構造を、サブツリーに分割することも可能である。

このディレクトリ構造においては、仮想ルートエントリの下に、5つのAノード(「ハムレット・著作について」, 「ベニスの商人・著作について」, 「ハムレット・日本語訳について」, 「ハムレット・中国語訳について」, 「ハムレット・上演について」; A4, A9, A10, A19, A13(図19などを参照))に対応するエントリが設けられる。仮想ルートエントリに対しては、分類型および分類値が定義され、定義された分類型および分類値は、DB管理装置5において、図30に示した情報テーブルにより管理され

10

20

30

40

50

る。

#### 【0200】

また、このディレクトリ構造においては、これらAノードに対応するエントリそれぞれに、Aノードそれぞれがもつ関連性役割に対応する下位エントリが設けられ、それぞれの役割を果たすTノード(T41, T42, T92, T101, T103, T191; 図19を参照)に関する情報をエントリの属性とする。

図41に示した関連情報は、全て1つのDB装置6-1(図24)に記憶され、あるいは、最上位エントリごとに分割されて、5つのDB装置6-1~6-5に記憶され、検索プログラム42による検索の対象とされる。

#### 【0201】

図42,図43は、図41に示したディレクトリ構造の関連情報を例示する第2,第3の図である。

なお、図42には、シェークスピアによる作品という一つの関連性(テーマ)に着目し、それに対応する「シェークスピア作品」エントリをルートエントリ下に新たに設け、その配下に関連情報を置くことにより、テーマに基づく階層化を実現する場合が例示されており、「シェークスピア作品」というエントリは、図26に点線で示されたエントリ(○  
u = 関係 #1 ~ #n)に対応する。

#### 【0202】

さらに、図43には、「著作」・「翻訳」・「上演」という異なる分類型(関連タイプ; Aノードのノードタイプ)を表現するエントリが定義され、「シェークスピア作品」エントリ下に新たに設けられる場合が例示されている。

このように、関連情報を、それぞれの関連タイプ(ノードタイプ)に応じたエントリの配下に登録することにより、関連タイプ(ノードタイプ)に基づく階層化が行われる。

#### 【0203】

図42に示すように、図41に示した関連情報は、さらに階層化されうる。

つまり、図41に示した関連情報に含まれる各ノードに対しては、共通のテーマとして、「シェークスピア作品」が定義されうるので、ユーザは、DB管理装置5を操作して、図42に示すように、5つのAノードに対応するエントリの上に、これらエントリ以下それに記憶される情報が、シェークスピア作品であることを示すエントリを新たに作成し、関連情報のテーマに基づいて、関連情報を階層化してもよい。

#### 【0204】

図42に示す場合においては、関連情報は、エントリ「シェークスピア作品」の下位にエントリ「ハムレット著作について(A4)」~「ハムレット上演について(A13)」を含むディレクトリ構造により表現されている。

例えば、エントリ「シェークスピア作品」およびエントリ「ハムレット著作について(A4)」~「ハムレット中国語訳について(A19)」以下のエントリを、サブツリーとしてDB装置6-1に記憶させ、「ハムレット・上演について(A13)」以下のエントリを、サブツリーとしてDB装置6-2に記憶させるなど、ディレクトリ構造に従って関連情報をサブツリーに分割することができ、分割の結果として得られたサブツリーは、DB装置6-1~6-nに、分割されて記憶されうる。

#### 【0205】

このようなときには、これら「シェークスピア作品」および「ハムレット・上演について」のディレクトリが、最上位エントリとされ、これらの最上位エントリに対しては、分類型および分類値が定義され、定義された分類型および分類値は、DB管理装置5において、図30に示した情報テーブルにより管理され、その参照関係は、図29に示したディレクトリツリーテーブルにより管理される。

#### 【0206】

別の例では、図42に示したように階層化された関連情報のディレクトリ構造内、「シェークスピア作品」およびその下の「ハムレット・著作について」および「ベニスの商人・著作について」に対応する関連情報が、DB装置6-1に記憶され、「ハムレット・

10

20

30

40

50

日本語訳について」および「ハムレット・中国語訳について」に対応する関連情報が、DB 装置 6 - 2 に記憶され、「ハムレット・上演について」に対応する関連情報が、DB 装置 6 - 3 に記憶される。

なお、最上位エントリは、サブツリー内のルート（最も上位にある）エントリなので、各 DB 装置 6 に記憶されているサブツリーごとに、ただ 1 つの最上位エントリが定義される。また、分類型・分類値は、この最上位エントリに対して定義される。

#### 【0207】

従って、図 4 2 に示した例においては、「ハムレット・日本語訳について」および「ハムレット・中国語訳について」に対応する関連情報を、DB 装置 6 - 2 に記憶するためには、これら 2 つのエントリをまとめる上位エントリ（例えば、図 4 3 に示す「翻訳情報」に対応するエントリ）が設けられ、これを最上位エントリとするサブツリーが記憶されることになる。10

このように分割されたサブツリーの参照関係は、図 2 9 に示したディレクトリテーブルを用いることにより管理され、サブツリーそれぞれに付された分類型および分類値は、図 3 0 に示した情報テーブルを用いることにより管理される。10

#### 【0208】

また、さらに、図 4 1 に示した 5 つの A ノードの内、A 4 ノード（「ハムレット・著作について」）と A 9 ノード（「ベニスの商人・著作について」）とに、「著作情報」が、関連タイプ（ノードタイプ）として定義され、A 10 ノード（「ハムレット・日本語訳について」）と A 19 ノード（「ベニスの商人・中国語訳について」）とに、「翻訳情報」が、関連タイプ（ノードタイプ）として定義され、A 13 ノード（「ハムレット・上映について」）に、「上演情報」が関連タイプ（ノードタイプ）として定義されるようなときには、図 4 3 に示すように、定義された関連タイプ（ノードタイプ）ごとの階層化が可能である。20

#### 【0209】

例えば、図 4 3 に示したように階層化された関連情報のディレクトリ構造の内、「シェークスピア作品」およびその下の「著作情報」に対応する関連情報が、DB 装置 6 - 1 に記憶され、「翻訳情報」に対応する関連情報が、DB 装置 6 - 2 に記憶され、「上演情報」に対応する関連情報が、DB 装置 6 - 3 に記憶される。30

このようなときには、「シェークスピア作品」、「翻訳情報」および「上演情報」のエントリが、最上位エントリとされ、これらの最上位エントリに対しては、分類型および分類値が定義され、定義された分類型および分類値は、DB 管理装置 5 において、図 3 0 に示した情報テーブルにより管理され、その参照関係は、図 2 9 に示したディレクトリツリーテーブルにより管理される。30

#### 【0210】

検索者（ユーザ）が、検索プログラム 4 2（図 2 4）に対して、図 4 1 ~ 図 4 3 に示したように DB 装置 6 に記憶された関連情報に対する検索のための操作を行うと、検索プログラム 4 2 は、検索操作の内容を解析し、さらに、この検索操作の内容に適合したツリー情報（ディレクトリツリーテーブル）および情報テーブル（図 2 9, 図 3 0）の内、検索に必要とされる情報を、DB 管理装置 5 から得る。40

さらに、検索プログラム 4 2 は、検索操作およびツリー情報に適合した検索条件（LDA P 操作のコマンドおよびパラメータ）を作成し、検索条件に適合するディレクトリ情報を記憶している DB 装置 6 に対する検索を行い、検索結果を得る。

このようにして得られた検索結果は、検索プログラム 4 2 のユーザに対して表示される。40

#### 【0211】

[DB システム 4 における関連情報およびその検索の例（2）]

以下、DB システム 4 においてどのように関連情報が登録され、登録された関連情報が検索されるかについて、商品、販売店および商品販売促進のためのプログラムを示す販売情報を、さらに具体例として挙げて説明する。50

図44、図45は、図24に示したDBシステム4においてDB装置6に登録されるディレクトリ構造の関連情報を例示する第4、第5の図であって、図44は、関東地域のみの販売情報を示し、図45は、関東地域および関西地域を含む販売情報を示す。

#### 【0212】

図24に示したDB管理装置5あるいはPC102に対するユーザの操作により、関東地域の販売情報が、図44に示すディレクトリ構造で表現される関連情報として登録される。

つまり、ユーザにより入力された関連情報の登録データに基づき、仮想ルートエントリの直下に、「販売情報」に関するディレクトリ情報をまとめるエントリが設けられ、このエントリの下位に、「関東地域」の販売情報に関するエントリが設けられる。

また、「関東地域」の販売情報に関して、「位置情報」、「提供情報」および「参加情報」を関連タイプ(ノードタイプ)とするエントリが設けられ、これらのエントリの下に、「店舗」、「地区」、「提供先」および「主催」などの関連役割に対応するエントリが設けられ、さらに、これらのエントリそれぞれの属性(この属性は、DB装置6それぞれに記憶されており、この属性を管理するために、独自の管理テーブルは定義、作成されない)として、「店A」・「新宿」など役割を果たすメンバに関する情報(Tノード用IDテーブル(図21)のノードID、ノード名に対応)が登録される。

#### 【0213】

なお、図44などに示す「役割」は、図20に示したARテーブルにおける関連性役割に対応し、「メンバ」は、それぞれのTノードに対応する。

図20に示した例においては、関連ノード(Aノード)A4について、メンバT41が「作者」という役割を果たし、メンバT42が「作品」という役割を果たしている。

#### 【0214】

図45に示すように、販売情報が、関東地域以外の情報(関西地域の販売情報など)をさらに含むときには、仮想ルートエントリの直下に、「販売情報」のエントリが設けられ、その下に、「関東地域」および「関西地域」のエントリが設けられ、「関西地域」のエントリの下に、関連タイプ(ノードタイプ)「位置情報」、「提供情報」および「参加情報」を表すエントリが設けられる。

さらに、これらのエントリの下には、「店舗」、「提供物」、「地区」および「主催」などの役割に対応するエントリが設けられ、これらのエントリそれぞれのエントリの属性として、「店B」、「店C」、「大阪」および「キャンペーンA」など、役割を果たすメンバに関する情報が登録される。

例えば、図44、図45に示した販売情報はサブツリーに分割され、仮想ルートエントリおよびその下位にある「販売情報」以外のディレクトリ情報は、例えば、DB装置6-1(A;図24)に記憶され、「販売情報」の下位にある「関東地域」以下のディレクトリ情報は、DB装置6-2(B)に記憶され、「関西地域」以下のディレクトリ情報は、DB装置6-3(C)に記憶され、これらの地域以外のディレクトリ(図示せず)情報は、DB装置6-4(D)などに記憶される。

#### 【0215】

図46は、図44、図45に示した販売情報(関連情報)の最上位エントリおよびその参照先などを示すディレクトリツリーテーブルを例示する図である。

図47は、図44、図45に示した販売情報(関連情報)の最上位エントリの分類型および分類値を示す情報テーブルを例示する図である。

DB装置6-1~6-3などに記憶されている販売情報の最上位エントリ、上位参照先のエントリ、および、分割されたディレクトリツリー(またはサブツリー)の最上位エントリ、それぞれのサブツリーに記憶されるディレクトリ情報の分類型・分類値は、DB管理装置5により、図46および図47に示すディレクトリツリーテーブルおよび情報テーブルを用いて管理されている。

#### 【0216】

図48は、図44、図45に示した販売情報のために用いられるGUI画像を例示する

10

20

30

40

50

図である。

図48に示すようなG U I画像がD B管理装置5あるいはP C 1 0 2の入出力装置1 2 6に表示され、このG U I画像に対するユーザの操作に従って、以上説明したD B装置6 - 1 ~ 6 - 3などに登録された販売情報に対する検索が行われる。

#### 【0217】

[関東地域のみの販売情報を対象とする検索]

まず、図44に示した関東地域のみの販売情報を対象とする検索を説明する。

但し、図44には、関東地域のみの販売情報が例示されており、実際のディレクトリ情報としては、その他の地域や販売情報以外の情報などもすべて登録・記憶されている。

検索者（ユーザ）は、検索装置40の入出力装置126に表示されたG U I画像（図48）に対して、例えば、「関東地域において、何らかの販売プログラムへの参加を通じて、店Bと提携している店舗」を検索するための操作を行う。  
10

#### 【0218】

図49は、図46、図47に示したディレクトリツリー・テーブルおよび情報テーブルから、図24に示した検索装置40が、検索の対象とするD B装置6を求めるための処理を示す図である。

検索装置40は、例えば、図49に示すように、検索者の検索要求の内容を解析し、さらに、D B管理装置5から得られたツリー情報に含まれる情報テーブル（図47）の内容を参照し、「地域(region)」および「関東地域」が、分類型および分類値として対応付けられたサブツリーの最上位エントリd n\_B（「関東地域」のディレクトリ）を得る。  
20

さらに、検索装置40は、ツリー情報に含まれるディレクトリツリー・テーブル（図46）の内容を検索し、最上位エントリd n\_Bが記憶されているD B装置6 - 2（B）を求める。

#### 【0219】

図50は、図48に示したような検索条件から得られ、図44に示した関東地域のみの販売情報を対する検索のために用いられるL D A P操作を作成する検索要求メッセージを示す図である。

なお、図50には、ユーザにより入力された検索要求が、X M Lメッセージとして例示されており、この検索要求から、実際に検索を実行するために用いられるL D A P操作コマンドが生成される。  
30

操作者により入力された検索条件と、ツリー情報から得られたサブツリー最上位エントリと、それがD B装置6 - 2（B）に記憶されているとの情報とから、「関東地域において、何らかの販売プログラムへの参加を通じて、店Bと提携している店舗」を検索するためのL D A P操作（L D A Pのコマンドおよびパラメータ）を作成し、D B装置6 - 2（B）に対する検索処理を行う。

#### 【0220】

また、図50には、関東地域のディレクトリ以下を記憶するD B装置6 - 2（B；図24）に対して、検索装置40が、以下（1）、（2）に示すような検索を実行するためのL D A P操作を作成する検索要求メッセージが示されている。

（1）「関連タイプ（図22に示すAノードのノードタイプ；関連タイプ）」が「参加情報」の全てのエントリから、直下に「役割」が「参加者」であり、「メンバ」が「店B」であるエントリをもつものを求め、その下位エントリのうち「役割」が「主催」であるエントリからメンバ属性値（メンバ名）を得る。  
40

この処理の結果として、「プログラムA」が、店Bが参加している販売プログラムとして得られる。

（2）「関連タイプ」が「参加情報」の全てのエントリから、直下に「役割」が「主催」であり、「メンバ」が「プログラムA」であるエントリをもつものを求め、その下位エントリのうち「役割」が「参加者」であるエントリからメンバ属性値（メンバ名）を得る。

この処理の結果として、「店A」および「店B」が販売プログラム「プログラムA」に参加している店舗として得られ、検索装置40は、これらの内、検索操作の内容（図48  
50

)に含まれる「店B」と異なる「店A」を、最終的な検索結果として選択する。

#### 【0221】

[関東地域および関西地域などの販売情報を対象とする検索]

次に、図45に示した関東地域およびそれ以外の地域の販売情報を対象とする検索を説明する。

検索者(ユーザ)は、検索装置40の入出力装置126に表示されたGUI画像(図48)に対して、例えば、「店Aが商品の提供により参加しているキャンペーンに、地域を問わず(全ての地域において)参加している店舗」を検索するための操作を行う。

#### 【0222】

検索装置40は、検索者の検索操作の内容を解析して、DB管理装置5から得られるツリー情報に含まれる情報テーブル(図47)を検索するが、検索操作の内容が、地域を限定しないために、検索装置40は、「関東地域」および「関西地域」などの地域のエントリが参照する「販売情報」のエントリ(図46などにおいてdn\_A)を得る。

さらに、検索装置40は、ディレクトリツリー(図46)を検索し、最上位エントリdn\_Aが記憶されているDB装置6-1(A)を求める。

#### 【0223】

操作者により入力された検索条件と、ツリー情報から得られた最上位エントリと、それがDB装置6-1(A)に記憶されているとの情報とから、「店Aが商品の提供により参加しているキャンペーンに、地域を問わず(全ての地域において)参加している店舗」を検索するためのLDAP操作(LDAPのコマンドおよびパラメータ)を作成し、DB装置6に対する検索処理を行う。

#### 【0224】

このようなLDAP操作には、以下のような検索操作が含まれる。

(1)「関連タイプ(図22に示したAノードのノードタイプ；関連タイプ)」が「提供情報」の全てのエントリから、直下に「メンバ」が「店A」であり、「役割」が「提供先」であるエントリをもつものを求め、それら下位エントリのうち「役割」が「提供物」であるエントリからメンバ属性値(メンバ名)を得る。

この処理の結果として、「商品A」が、店Aが提供している商品として得られる。  
(2)「関連タイプ」が「参加情報」の全てのエントリから、直下に「メンバ」が「商品A」であり、「役割」が「参加者」であるエントリをもつものを求め、その下位エントリのうち「役割」が「主催」であるエントリからメンバ属性値(メンバ名)を得る。

この処理の結果として、「キャンペーンA」が、商品Aが提供されているキャンペーンプログラムとして得られる。

(3)「関連タイプ」が「参加情報」の全てのエントリから、直下に「メンバ」が「キャンペーンA」であり、「役割」が「主催」であるエントリをもつものを求め、その下位エントリのうち「役割」が「参加者」であるエントリからメンバ属性値(メンバ名)を得る。

この処理の結果として、検索装置40は、「商品A」および「商品C」を、キャンペーンAに参加する店舗が提供している商品として得る。

(5)「関連タイプ」が「提供情報」の全てのエントリから、直下に「メンバ」が「商品A」であり、「役割」が「提供物」であるようなエントリ、あるいは、「メンバ」が「商品C」であり、「役割」が「提供物」であるようなエントリ、をもつものを求め、さらに、それら下位エントリのうち「役割」が「提供先」であるエントリからメンバ属性値(メンバ名)を得る。

この処理の結果として、「店A」および「店C」が、商品Aまたは商品CをキャンペーンAに提供している店舗として得られ、検索装置40は、「店A」、「店C」の内、検索操作の内容(図48)に含まれる「店A」と異なる「店C」を、最終的な検索結果として選択する。

#### 【0225】

[ディレクトリ構造の動的変更およびDB装置6間の負荷分散]

10

20

30

40

50

以上、説明したように、DBシステム4(図24)においては、関連情報がディレクトリ構造で表現されるので、DB装置6間におけるサブツリー単位での分割および複製が容易である。

従って、DBシステム4においては、例えば、DB装置6に記憶されているあるエントリに対するアクセスが、何らかの理由により過大になったようなときでも、そのエントリを記憶する複製装置(ミラーサーバ)を追加したり、そのエントリおよびその下位のエントリ(以下、あるエントリおよびその下位のエントリの集合をサブツリーとも記す)ごと、他のDB装置6に移したりするといった負荷分散のための対策が簡単である。

#### 【0226】

以下、このようなDBシステム4の特徴を活かして、DB装置6間で負荷を分散し、また、DBシステム4のシステム性能低下を予防するために適したディレクトリ構造の動的変更、および、DB装置6間におけるサブツリーの交換および移動を説明する。

DBシステム4におけるディレクトリ構成の動的変更、および、DB装置6間におけるサブツリーの交換・移動のためには、以下のような方法が採られる。

- (1) 同一DB装置6内における新たなサブツリーの作成(サブツリーの新設)、および、
- (2) 高負荷のDB装置6から、低負荷のDB装置6へのサブツリーの転送、あるいは、高負荷のDB装置6においてアクセスが多いサブツリーと、低負荷のDB装置6においてアクセスが少ないサブツリーとの交換(サブツリーの交換・移動)。

#### 【0227】

##### [サブツリーの新設]

まず、DBシステム4におけるサブツリーの新設を説明する。

図51は、図24に示したDBシステム4のDB装置6におけるサブツリーの新設を例示する図であって、(A)は、サブツリーの新設前のディレクトリ構成を示し、(B)は、サブツリーの新設後のディレクトリ構成を示す。

図51(A)に示すように、検索装置40から、query Base DNを検索ベースとし、フィルタquery Filterを用いた検索を実行したとき、この検索条件にマッチするエントリが、あるDB装置6に記憶されたディレクトリツリー(またはサブツリー)の一部に集中する場合を考える。

#### 【0228】

なお、このフィルタquery Filterは、LDAPでの検索条件であって、ここまでに示した検索操作においては、「役割」='提供物'、「メンバ」='商品C'などの条件がフィルタquery Filterになっている。

このフィルタquery Filterにおいては、いくつかの条件の論理積から、複雑な条件が表される。

#### 【0229】

エントリそれぞれに対して、検索操作ごとに、何回、検索条件にマッチしたかを、DB装置6が計数し、この計数結果を、DB管理装置5において集計することにより、特定の検索条件の下で、アクセス頻度が多いエントリの集合と、その上位エントリとを求めることができる。

例えば、サブツリー新設前のディレクトリ構成(図51(A))において、このような計測を行うことにより、ある検索条件(検索ベース:query Base DN、フィルタ:query Filter)では、図中、網掛け表示されたエントリが検索にマッチし、これらエントリへのアクセスが集中していることが検出されうる。

このようなアクセスの集中が検出されたときに、アクセスが集中するエントリの集合に上位エントリ(group DN)を設け、これ以下のエントリを新たなサブツリーとすることにより、エントリ集合へのアクセスを局所化することができ、DBシステムの性能低下を回避することができる。

さらには、アクセスが集中するサブツリーを他のDB装置6との間で移動させるための準備をすることもできる。

10

20

30

40

50

## 【0230】

図52は、図51(A),(B)に示したサブツリーの新設のために用いられる検索条件管理テーブルを例示する図である。

上位エントリ(groupDN)以下のサブツリーが新設されたときには、DB管理装置5(図24)は、図52に示すように、検索の起点エントリ(検索ベース;queryBaseDN)と、検索に用いられたフィルタ(queryFilter)と、このエントリ以下で、アクセスが集中したエントリ集合の上位に付されたエントリ(新検索ベース;groupDN)とを対応付けた検索条件管理テーブルを作成し、検索装置40に提供する。

## 【0231】

検索装置40は、検索条件管理テーブルを受けて、検索条件を作成するときに、変更があったエントリ(検索ベース;queryBaseDN)以下のサブツリーに対して、フィルタに、上記の特定のフィルタ(queryFilter)を論理積として含む検索が生じたときに、元のエントリ(検索ベース;queryBaseDN)以下のサブツリーにではなく、新たなエントリ(新検索ベース;groupDN)以下のサブツリーに対して、特定のフィルタ(queryFilter)を除いたフィルタを用いて検索を行うことにより、検索に要する時間を短縮することができる。

フィルタに、上記の特定のフィルタ(queryFilter)を含まない場合、元のエントリ(queryBaseDN)を起点エントリとし、これ以下のサブツリー全体を検索範囲とする検索を行う。

なお、サブツリー新設後も、検索条件管理テーブルを用いないときと同様に、情報テーブル(図47)によっても、検索条件管理テーブルによっても、検索のための最上位エントリが得られないときには、検索装置40は、DB装置6-1~6-n(A~N)において最も上位にあるディレクトリから、検索を行う。

## 【0232】

## [サブツリーの移動・交換]

図53は、図24に示したDBシステム4のDB装置6-1,6-2(A,B)間ににおけるサブツリーの交換・移動を例示する図であって、(A)は、サブツリー交換前のディレクトリ構造を示し、(B)は、サブツリー交換後のディレクトリ構造を示す。

図53(A)に示すように、DB装置6-2(B)において、アクセスが集中するサブツリー(エントリdn\_3以下のサブツリー)が存在し、一方、DB装置6-1(A)において、アクセスがほとんどないサブツリー(エントリdn\_2以下のサブツリー)が存在する場合を考える。

エントリそれぞれに対して、何回、アクセスがあったかを、DB装置6が計数し、この計数結果を、DB管理装置5において集計することにより、アクセス頻度が多いサブツリーを求めることができる。同時に、DB装置6の負荷状態を求めることができる。また、図52に示した検索条件管理テーブルを用いて、アクセスが集中するサブツリーを求める 것도できる。

## 【0233】

このようなアクセスの集中(偏り)が検出されたとき、図53(B)に示すように、高負荷状態のDB装置6-2(B)のエントリdn\_3以下のサブツリーを、DB装置6-1(A)のエントリdn\_A-1以下のサブツリーとし、反対に、DB装置6-1(A)のエントリdn\_2以下のサブツリーを、DB装置6-2(B)のdn\_2を最上位エントリとするサブツリーとして、サブツリーをDB装置6-1,6-2(A,B)の間で交換することにより、DB装置6-1,6-2(A,B)間で、負荷の分散を図ることができる。

ただし、マクロなレベルでのアクセス集中・偏り(例えば、そもそも検索の対象となる頻度が少ないため、アクセス(負荷)が低くなるといった状況)を想定すると、サブツリーの移動・交換については、必ずしも検索条件管理テーブルを用いる必要はない。

## 【0234】

10

20

30

40

50

サブツリーの新設（検索条件管理テーブル）では、ある時点までに発生した特定の検索条件でのアクセス集中状況に応じて、それ以降の検索処理を高速化（効率化）することを目的とする。

一方、サブツリーの移動・交換は、それぞれの検索条件に依らず、また、特定の検索条件に限らず、各サーバあるいはサブツリーに対するアクセスの絶対数（頻度）の増大や減少によって負荷の著しい偏りが発生した場合、それを解消するための処理となる。

#### 【0235】

図54は、図53（A），（B）に示したサブツリーの交換に伴うDB装置6-1，6-2（A，B）についてのディレクトリツリーテーブル（図46）の更新を示す図であって、（A）は、更新前のディレクトリツリーテーブルを示し、（B）は、更新後のディレクトリツリーテーブルを示す。10

なお、このようなサブツリーの交換は、図51（A），（B）に示したように新設されたサブツリーおよび既存のサブツリーについて行われる。

このような更新が行われたときには、DB管理装置5は、図54（A），（B）に示すように、DB装置6-1，6-2（A，B）についてのディレクトリツリーテーブルの内容を更新する（但し、この場合、DB装置6-1（A）についてのディレクトリツリーテーブルの内容に変更はない）。

また、例えば、図53（A）に示したDB装置6-1（A）のディレクトリd n\_1に対するアクセスが集中し、DB装置6-2（B）全体に対するアクセスが少なく、低負荷状態であるときなどには、単に、ディレクトリd n\_1以下のサブツリーをDB装置6-1（A）からDB装置6-2（B）に移動させるだけで、DB装置6間で負荷が分散される。20

#### 【0236】

以上説明したサブツリーの移動・交換は、以下のような手順で実行される。

（1）アクセス頻度の測定：DB装置6は、仮想ルートエントリの直下から、関連情報の基本構成単位（図26）の1階層上までのエントリについて、アクセスの回数（頻度）を計測し、DB管理装置5がこの計測値を集計する。

（2）移動・交換元のサブツリーの選択：DB管理装置5は、DB装置6において、同一階層に属する複数のエントリの下のサブツリーに対するアクセスの大きな偏り（集中）を検出したときに、これらのエントリの下のサブツリーを、移動・交換元のサブツリーとして選択する。30

（3）移動・交換先のDB装置の選択：DB管理装置5は、移動・交換元のサブツリーを選択すると、この移動・交換元のサブツリーを下位とせず、かつ、移動・交換元のサブツリーが記憶されていたDB装置6よりもアクセスの頻度（負荷）が全体として低い他のDB装置6を選択する。

#### 【0237】

##### [第3のDBプログラム62]

図55は、図24に示したDBシステム4において、サブツリーの新設および移動・交換が行われるときに、DB装置6において動作する第3のDBプログラム62を示す図である。

図55に示すように、DBプログラム62は、図37に示した第2のDBプログラム60に、情報転送部620およびアクセス監視部624を付加した構成を採る。40

DBプログラム62は、これらの構成部分により、第2のDBプログラム60と同様な機能の他、サブツリーの新設および移動・交換のために必要な機能を実現する。

#### 【0238】

DBプログラム62において、情報転送部620は、DB管理装置5からの制御に従つて、図53（A），（B）に示したような、サブツリーの移動・交換のために必要とされる関連情報の転送を行う。

アクセス監視部624は、DB装置6に記憶されたディレクトリ情報のエントリそれれに対するアクセスの回数（頻度）を計測し、計測値をDB管理装置5に対して送信する50

。

### 【0239】

#### [第2のDB管理プログラム56]

図56は、図24に示したDBシステム4において、サブツリーの新設および移動・交換が行われるときに、DB管理装置5において動作する第2のDB管理プログラム56を示す図である。

図56に示すように、第2のDB管理プログラム56は、図31に示した第1のDB管理プログラム50に、構成変更処理部560、情報転送制御部562（情報転送手段）、DB監視部570、監視用DB572、検索条件管理部580、検索条件DB582および検索条件提供部584を附加した構成を探る。  
10

DB管理プログラム56は、これらの構成部分により、第1のDB管理プログラム50と同様な機能の他、サブツリーの新設および移動・交換のために必要な機能を実現する。

### 【0240】

DB管理プログラム56において、DB監視部570は、DB装置6それぞれから送られてきた各エントリに対するアクセス回数（頻度）およびアクセスの際に用いられたLDAPによる検索操作に関する情報（検索ベース、および、フィルタ）を集計し、集計値を監視用DB572に記憶する。

DB監視部570は、記憶したアクセス頻度およびその検索操作に関する情報の集計値を、構成変更処理部560に対して出力する。

### 【0241】

構成変更処理部560は、DB監視部570から入力される各エントリに対するアクセス回数の集計値に応じて、情報転送制御部562を制御し、図51(A), (B)に例示したサブツリーの新設のための処理を行う。

また、構成変更処理部560は、各エントリに対するアクセス回数の集計値に応じて、情報転送制御部562を制御し、図53(A), (B)に例示したサブツリーの移動・交換のために必要な処理を行わせる。

また、構成変更処理部560は、情報テーブル管理部534および情報テーブル管理部534を制御して、図54(A), (B)に例示したような、サブツリーの新設および移動・変更に伴うディレクトリツリーテーブル（図46）などの内容の変更のために必要な処理を行わせる。  
30

### 【0242】

検索条件管理部580は、サブツリーの新設および移動・変更があったときに、図52に例示した検索条件管理テーブルを作成し、検索条件DB582に記憶する。

さらに、検索条件管理部580は、検索条件提供部584に対して、記憶した検索条件管理テーブルを出力する。

検索条件提供部584は、検索条件管理部580から入力された検索条件管理テーブルを、検索装置40（検索プログラム44；図59）に対して提供する。

### 【0243】

以下、アクセス回数の集計をさらに説明する。

図57は、図56に示したDB管理プログラム56の処理(S56)を示すフローチャートである。  
40

図57に示すように、ステップ560(S560)において、DB管理プログラム50は、初期化処理を行う。

ステップ562(S562)において、DB監視部570は、DB装置6のエントリそれぞれに対するアクセス回数の計測を開始する。

### 【0244】

ステップ564(S564)において、DB装置6は、検索制御部264からLDAP検索操作コマンドを受け入れる。

ステップ566(S566)において、DB監視部570は、DB装置6に入力された検索操作コマンドに含まれる検索条件を取得する。  
50

ステップ 568 (S568)において、DB 監視部 570 は、検索条件に合うエントリの情報を DB 装置 6 から取得する。

なお、S564～S568 の処理は、DB 装置 6 で実行される検索操作に関する情報を集計するために行われ、これらの集計値は DB 装置 6 から DB 監視部 570 に送られる。

#### 【0245】

ステップ 570 (S570)において、DB 監視部 570 は、検索操作に関する情報を集計するための所定の計測時間が経過したか否かを判断する。

DB 管理プログラム 56 は、計測時間を経過したときには S572 の処理に進み、これ以外のときには S564 の処理に進む。

ステップ 572 (S572)において、DB 監視部 570 は、アクセス回数の計測を終了する。 10

#### 【0246】

ステップ 574 (S574)において、DB 監視部 570 は、計測が開始されてから終了するまでの計測時間において、全てのエントリに対するアクセス回数、および、全ての DB 装置 6 の負荷状態を算出する。

ステップ 576 (S576)において、構成変更処理部 560 は、DB 監視部 570 により算出された計測結果に基づき、全てのエントリについて、アクセス集中の有無を検出する。

ステップ 578 (S578)において、DB 監視部 570 は、処理を継続するか否かを判断する。 20

DB 管理プログラム 56 は、処理を継続するときには S560 の処理に進み、これ以外のときには処理を終了する。

#### 【0247】

図 58 は、複数の DB 装置にまたがる同一階層のエントリを例示する図である。

なお、エントリおよびサブツリーに対するアクセスの集中は、例えば、以下(1)，(2)に示すような方法で検出される。

(1) 所定の計測時間内で、すべての検索操作について、アクセス集中指標（アクセス集中指標 = 検索条件に合うエントリ数 / 特定の階層に存在するエントリ数（図 51(A)））を算出する。

それぞれの検索操作の実行回数が、事前に定められた回数よりも多く、かつ、算出されたアクセス集中指標が、ある閾値以下となったときには、その検索条件に対してアクセス集中が発生したと検出する。 30

(2) 分割されたサブツリーを記憶する DB 装置 6 を計測の対象とし、サブツリーそれぞれの最上位エントリ、および、最上位エントリと同じ階層に属する全てのエントリが、操作要求に応じて検索ベースとされた回数（頻度）を、そのサブツリーのアクセス頻度として、(1)に準じた方法で、アクセスの集中を検出する。

#### 【0248】

##### [第 2 の検索プログラム 44]

図 59 は、図 24 に示した DB システム 4 において、サブツリーの新設および移動・交換が行われるときに、検索装置 40 において動作する第 2 の検索プログラム 44 を示す図である。 40

図 59 に示すように、第 2 の検索プログラム 44 は、図 38 に示した第 1 の検索装置 40 に、検索条件変換部 440 を付加した構成を探る。

検索プログラム 44 は、これらの構成部分により、第 1 の検索装置 40 と同様な機能の他、検索条件管理テーブルを用いた検索条件の変換を行う。

#### 【0249】

検索プログラム 44 において、検索条件変換部 440 は、DB 管理装置 5 から、検索条件管理テーブル（図 52）を受け取り、検索条件作成部 420 が作成した検索条件に含まれるフィルタおよび検索対象の起点エントリ（検索ベース）が、検索条件管理テーブルに示された条件に合うか否かを判断する。 50

検索条件変換部 440 は、検索条件に含まれるフィルタおよび検索ベースが、検索条件管理テーブルに示された条件に合うときには、検索対象の元の起点エントリ（検索ベース；query Base DN）を、新しい検索起点エントリ（新検索ベース；group DN）とし、検索条件のフィルタから検索条件管理テーブルに示されたフィルタ（query Filter）を除いた新しいフィルタに変換するように検索条件作成部 420 を制御する。

#### 【0250】

[サブツリーを新設するときのDBシステム4の動作]

以下、サブツリーの新設を行うときのDBシステム4の全体的な動作を説明する。

図51(A)を参照して既に述べたように、エントリそれぞれに対して、検索操作ごとに、何回、検索条件にマッチしたかを、DB装置6が計数し、この計数結果を、DB管理装置5において集計することにより、特定の検索条件の下で、アクセス頻度が多いエントリの集合と、その上位エントリとを求めることがきる。10

このようなアクセスの集中が検出されたときに、アクセスが集中するエントリの集合に上位エントリ（group DN）を設け、これ以下のエントリを新たなサブツリーとすることにより、エントリ集合へのアクセスを局所化することができ、DBシステムの性能低下を回避することができる。

DB装置6は、各エントリに対するアクセスの回数（頻度）およびアクセスに用いられたフィルタを、DB管理装置5に通知する。20

DB管理装置5は、DB装置6から得られた各エントリに対するアクセスの回数（頻度）およびアクセスに用いられたフィルタを集計する。20

#### 【0251】

DB管理装置5は、あるDB装置6のサブツリーに対するアクセスが大きい場合などに、図51(A), (B)に示したように、DB装置6を制御して、サブツリーの新設を行う。

さらに、DB管理装置5は、このサブツリーの新設に合うように、検索条件管理テーブル(図52)の内容を変更する。

DB管理装置5は、内容が変更された検索条件管理テーブルを、検索装置40に対して提供する。

検索装置40は、検索者（ユーザ）の検索操作に従って、検索操作の内容から、検索条件を作成するときに、DB管理装置5から提供された検索条件管理テーブルを用いた変換を行い、DB装置6に対する検索を行う。30

#### 【0252】

[サブツリーを移動・交換するときのDBシステム4の動作]

次に、サブツリーの移動・交換を行うときのDBシステム4の全体的な動作を説明する。40

図58を参照して既に述べたように、複数のDB装置にまたがる同一階層のエントリそれぞれに対して、検索ベースとされた回数を集計することにより、サブツリーへのアクセス集中を検出することができる。

DB装置6は、各エントリが検索ベースとして利用された回数を計測し、DB管理装置5に通知する。

DB管理装置5は、それぞれのDB装置の最上位エントリ、および、それら最上位エントリと同じ階層にある他のDB装置の全てのエントリが検索ベースとされた回数を集計し、各サブツリーのアクセス頻度として算出する。

#### 【0253】

DB管理装置5は、あるDB装置6のサブツリーに対するアクセスが大きい場合などに、図53(A), (B)に示したように、DB装置6を制御して、サブツリーの移動・交換を行う。

さらに、DB管理装置5は、このサブツリーの移動・交換に合うように、ディレクトリツリーテーブル(図46)および情報テーブル(図47)などの内容を変更する。50

D B 管理装置 5 は、内容が変更されたディレクトリツリーテーブルおよび情報テーブルを、検索装置 4 0 に対して提供する。

検索装置 4 0 は、検索者（ユーザ）の検索操作に従って、提供されたディレクトリツリーテーブルおよび情報テーブルを用いて、D B 装置 6 に対する検索を行う。

【産業上の利用可能性】

【0254】

本発明は、データベースおよびデータ検索のために利用可能である。

【図面の簡単な説明】

【0255】

【図1】図1は、本発明において関連性ネットワーク型データを変換する際のデータモデルを示す図である。 10

【図2】図2は、記憶の対象となる一連の関連性を有するn個のデータを、図1に示したデータモデルに従って変換したデータ構造を示す図である。

【図3】図3は、関連ノード間の関連性を表現する方法の一例を示す図である。

【図4】図4は、第1及び第2の例のデータをノード群及び各ノード間の関連を示すエッジ群を用いて表現した関連性ネットワーク型データの構造を示す図である。

【図5】図5は、本発明の好適実施例に関するブロック図である。

【図6】図6は、本発明にかかる第1のデータベースシステム（D Bシステム）の構成を例示する図である。 20

【図7】図7は、図6に示したD B装置およびP Cのハードウェア構成を例示する図である。

【図8】図8は、図3（a）に例示したデータの関連を書き換えて示す図である。

【図9】図9は、図8に示したデータ構造を一般化して示す図である。

【図10】図10は、図3（b），図4に示したデータの関係を、一般化して示す図である。

【図11】図11は、図9に示した構造を探るデータを記憶するために用いられる関連性役割（A R (Association Role)）テーブルを示す図である。 30

【図12】図12は、図9に示した構造を探るデータを記憶するために用いられるTノード用識別子（I D (Identifier)）テーブルを示す図である。

【図13】図13は、図9に示した構造を探るデータを記憶するために用いられるAノード用識別子（I D）テーブルを示す図である。 30

【図14】図14は、図6，図7に示したD B装置におけるデータ検索方法を例示する図である。

【図15】図15は、図6，図7に示したD B装置における検索の全体的な処理（S 2 0）を示す第1のフローチャートである。

【図16】図16は、図15に示した検索フィルタに基づく関連ノードの選択処理（S 2 2）を示すフローチャートである。

【図17】図17は、図15，図16に示したノードIDおよびノード名取得処理（S 2 4）を示すフローチャートである。

【図18】図18は、図6，図7に示したD B装置において実行されるD Bプログラム2の構成を示す図である。 40

【図19】図19は、図6，図7に示したD B装置（D Bプログラム；図18）に対して入力されるデータと、これに含まれるデータの検索に含まれる検索条件とを例示する図である。

【図20】図20は、A Rエントリ作成部（図18）およびA R D B管理部により作成され、A R D Bに記憶されるA Rテーブルを例示する図である。

【図21】図21は、I Dエントリ作成部（図18）およびI D D B管理部により作成され、I D D Bに記憶されるTノード用のI Dテーブルを例示する図である。

【図22】図22は、I Dエントリ作成部（図18）およびI D D B管理部により作成されるAノードについてのI Dテーブルを例示する図である。 50

【図23】図23は、第2のDBシステムの構成を例示する図である。

【図24】本発明にかかる第3のDBシステムの構成を例示する図である。

【図25】図24に示したDBシステムにおける情報のグラフ形式の表現を例示する図である。

【図26】図25に示した情報のディレクトリ情報ツリー形式の表現を例示する図である。

【図27】ディレクトリ構造で表現された関連情報の分割方法を例示する図である。

【図28】図27に示したDB装置(A)のエントリdn\_A2と、DB装置(N)のエントリdn\_Nとの参照関係を示す図である。

【図29】図27に示したようにDB装置(A~N)に分割されて記憶された関連情報を管理するために用いられるディレクトリツリーテーブルを例示する図である。 10

【図30】図27、図28に示したDB装置(A~N)に記憶されているディレクトリツリー(またはサブツリー)それぞれの最上位エントリdn\_A~dn\_N、および、サブツリーそれに記憶されている関連情報の分類に用いられる情報テーブルを例示する図である。

【図31】図24に示したDB管理装置において動作する第1のDB管理プログラムの構成を示す図である。

【図32】図31に示したDB管理プログラムが、データ入力のために用いられるコンピュータ(PC)の入出力装置(図6、図7)に表示するGUI画像を例示する図である。

【図33】図32に示したDB管理プログラムによる関連情報の登録処理(S30)を示すフローチャートである。 20

【図34】図32に示したDB管理プログラムによる情報テーブル(図30)の変更処理(S34)を示すフローチャートである。

【図35】図31に示したDB管理プログラムによるディレクトリツリーテーブルの変更処理(S36)を示すフローチャートである。

【図36】図31に示したDB管理プログラムの全体的な動作(S40)を示すシーケンス図である。

【図37】図24に示したDB装置それぞれにおいて動作する第2のDBプログラムを示す図である。

【図38】図24に示した検索装置において動作する検索プログラムを示す図である。 30

【図39】図38に示した検索プログラムによる検索処理(S44)を示すフローチャートである。

【図40】図31に示したDB管理プログラムおよび図38に示した検索プログラムの全体的な処理(S52)を示すシーケンス図である。

【図41】図24に示したDBシステムにおいてDB装置に登録されるディレクトリ構造の関連情報を例示する第1の図であって、フラットなディレクトリ構造を示す。

【図42】図41に示したディレクトリ構造の関連情報を例示する第2の図である。

【図43】図41に示したディレクトリ構造の関連情報を例示する第3の図である。

【図44】図24に示したDBシステムにおいてDB装置に登録されるディレクトリ構造の関連情報を例示する第4の図であって、関東地域のみの販売情報を示す。 40

【図45】図24に示したDBシステムにおいてDB装置に登録されるディレクトリ構造の関連情報を例示する第4の図であって、関東地域および関西地域を含む販売情報を示す。

【図46】図44、図45に示した販売情報(関連情報)の最上位エントリおよびその参照先などを示すディレクトリツリーテーブルを例示する図である。

【図47】図44、図45に示した販売情報(関連情報)をサブツリー分割したときの最上位エントリとその分類型および分類値を示す情報テーブルを例示する図である。

【図48】図44、図45に示した販売情報に対する検索のために用いられるGUI画像を例示する図である。

【図49】図46、図47に示したディレクトリツリーテーブルおよび情報テーブルから 50

、図24に示した検索装置が、検索の対象とするDB装置を求めるための処理を示す図である。

【図50】図48に示したような検索条件から得られ、図44に示した関東地域のみの販売情報に対する検索のために用いられるLDAP操作を作成する検索要求メッセージを例示する図である。

【図51】図24に示したDBシステムのDB装置におけるサブツリーの新設を例示する図であって、(A)は、サブツリーの新設前のディレクトリ構成を示し、(B)は、サブツリーの新設後のディレクトリ構成を示す。

【図52】図51(A), (B)に示したサブツリーの新設のために用いられる検索条件管理テーブルを例示する図である。

【図53】図24に示したDBシステムのDB装置(A, B)間におけるサブツリーの交換・移動を例示する図であって、(A)は、サブツリー交換前のディレクトリ構造を示し、(B)は、サブツリー交換後のディレクトリ構造を示す。

【図54】図53(A), (B)に示したサブツリーの交換に伴うDB装置(A, B)についてのディレクトリツリーテーブル(図46)の更新を示す図であって、(A)は、更新前のディレクトリツリーテーブルを示し、(B)は、更新後のディレクトリツリーテーブルを示す。

【図55】図24に示したDBシステムにおいて、サブツリーの新設および移動・交換が行われるときに、DB装置において動作する第3のDBプログラムを示す図である。

【図56】図24に示したDBシステムにおいて、サブツリーの新設および移動・交換が行われるときに、DB管理装置において動作する第2のDB管理プログラムを示す図である。

【図57】図56に示したDB管理プログラム56の処理(S56)を示すフローチャートである。

【図58】複数のDB装置にまたがる同一階層のエントリを例示する図である。

【図59】図24に示したDBシステムにおいて、サブツリーの新設および移動・交換が行われるときに、検索装置において動作する第2の検索プログラムを示す図である。

#### 【符号の説明】

##### 【0256】

4 . . . DBシステム、

102 . . . PC、

40 . . . 検索装置、

42, 44 . . . 検索プログラム、

260 . . . 検索操作受入部、

264 . . . 検索制御部、

420 . . . 検索条件作成部、

440 . . . 検索条件変換部、

442 . . . 検索結果受入部、

5 . . . DB管理装置、

50, 56 . . . DB管理プログラム、

500 . . . UI部、

502 . . . 情報登録部、

504 . . . 情報送信部、

510 . . . 関連情報作成・管理部、

512 . . . ツリー情報提供部、

522 . . . ディレクトリツリーテーブル作成部、

524 . . . ディレクトリツリーテーブル管理部、

526 . . . ディレクトリツリーテーブルDB、

532 . . . 情報テーブル作成部、

534 . . . 情報テーブル管理部、

10

20

30

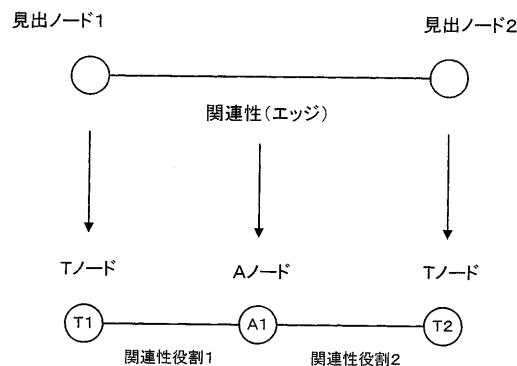
40

50

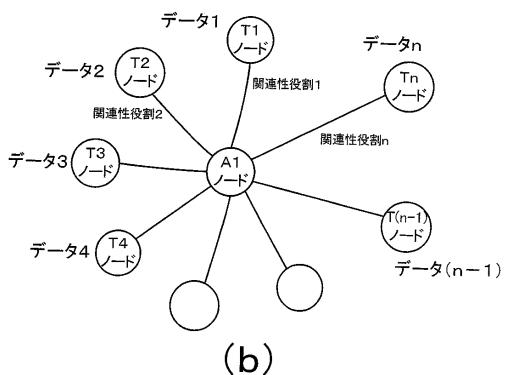
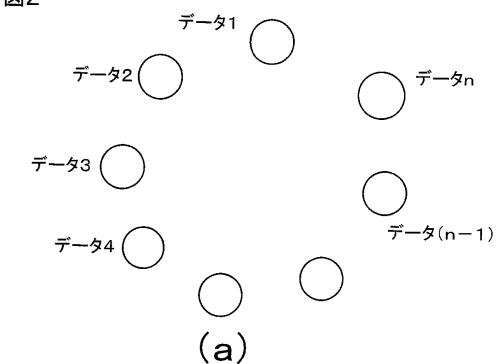
5 3 6 . . . 情報テーブル DB、  
 5 6 0 . . . 構成変更処理部、  
 5 6 2 . . . 情報転送制御部、  
 5 7 0 . . . DB監視部、  
 5 7 2 . . . 監視用DB、  
 5 8 0 . . . 検索条件管理部、  
 5 8 2 . . . 検索条件DB、  
 5 8 4 . . . 検索条件提供部、  
 6 . . . DB装置、  
 6 0 , 6 2 . . . DBプログラム、  
 6 0 0 . . . 情報管理部、  
 6 0 2 . . . 情報DB、  
 6 0 4 . . . 情報通信部、  
 6 0 6 . . . 検索実行部、  
 6 2 0 . . . 情報転送部、  
 6 2 4 . . . アクセス監視部、

10

【図1】

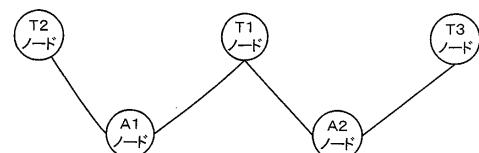


【図2】

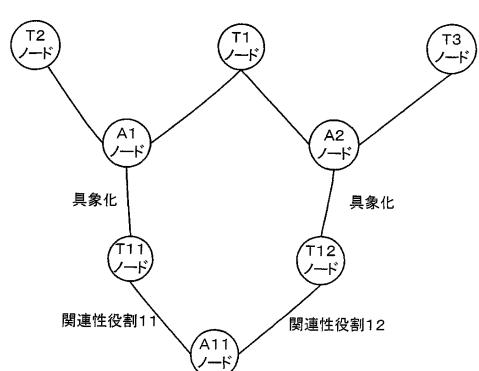


【図3】

図3

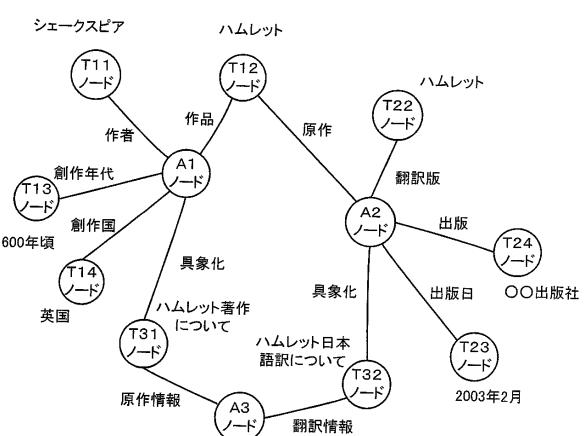


(a)

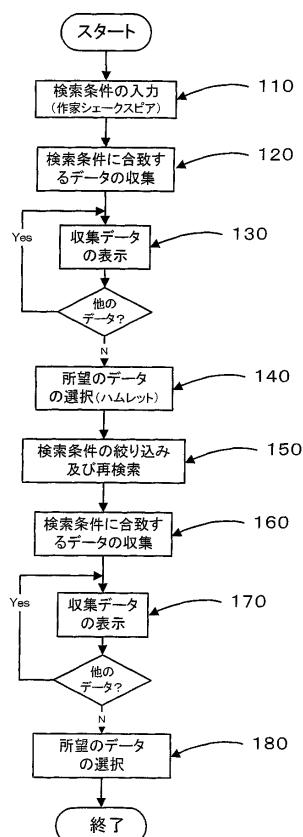


(b)

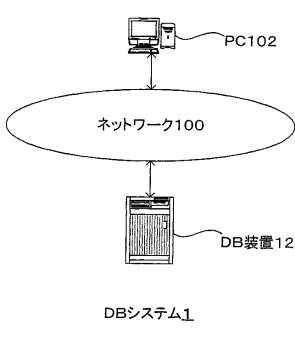
【図4】



【図5】

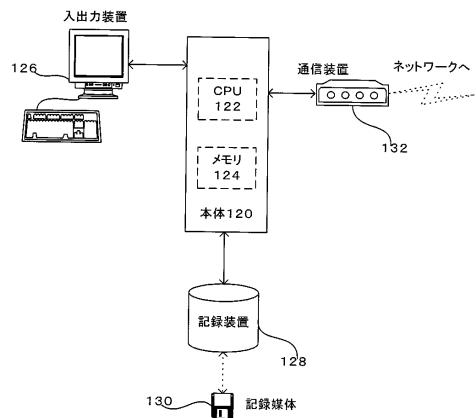


【図6】

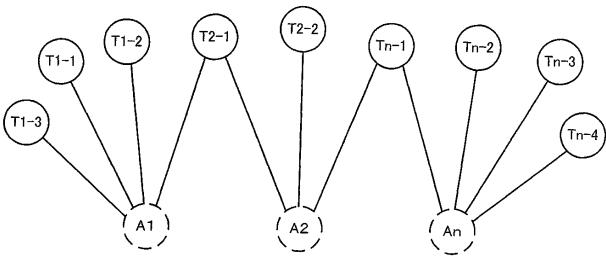


DBシステム1

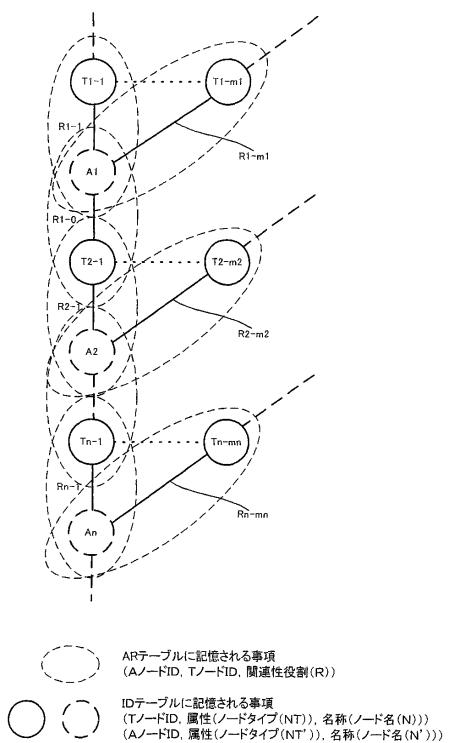
【図7】

12, 102(30, 32, 40, 5, 6)

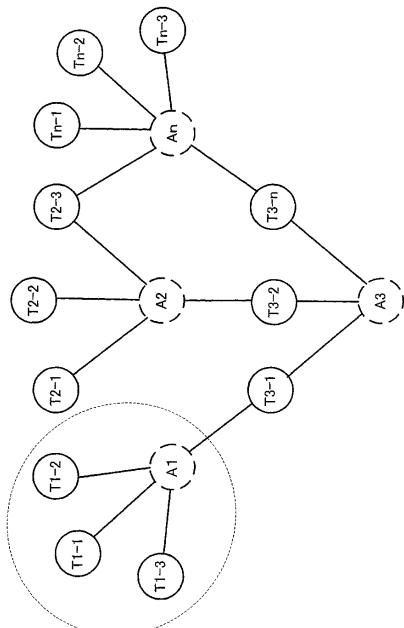
【図8】



【図9】



【図10】



【図11】

ARテーブル		
AノードID	TノードID (ノード名(N))	関連性役割R
A1	T1-1	R1-1
A1	T1-m1	R1-m1
A1	T2-1	R1-0
A2	T2-1	R2-1
A2	T2-m2	R2-m2
An	Tn-1	Rn-1
An	Tn-mn	Rn-mn

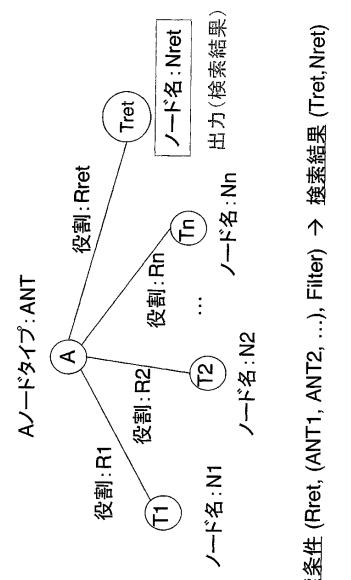
【図12】

IDテーブル(Tノード用)		
TノードID	ノードタイプ(NT)	ノード名(N)
T1-1	NT1-1	N1-1
T1-m1	NT1-m1	N1-m1
T2-1	NT2-1	N2-1
T2-m2	NT2-m2	N2-m2
Tn-1	NTn-1	Nn-1
Tn-mn	NTn-mn	Nn-mn

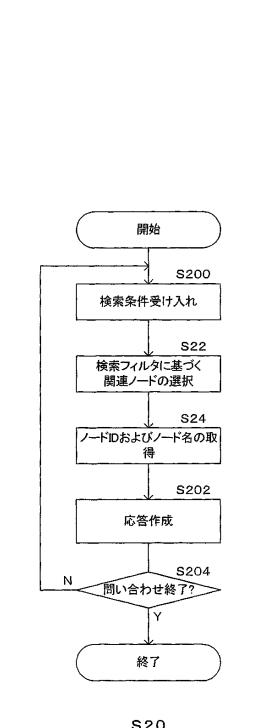
【図13】

IDテーブル(Aノード用)		
AノードID	ノードタイプ(NT)	ノード名(N)
A1	NT1	N1
A2	NT2	N2

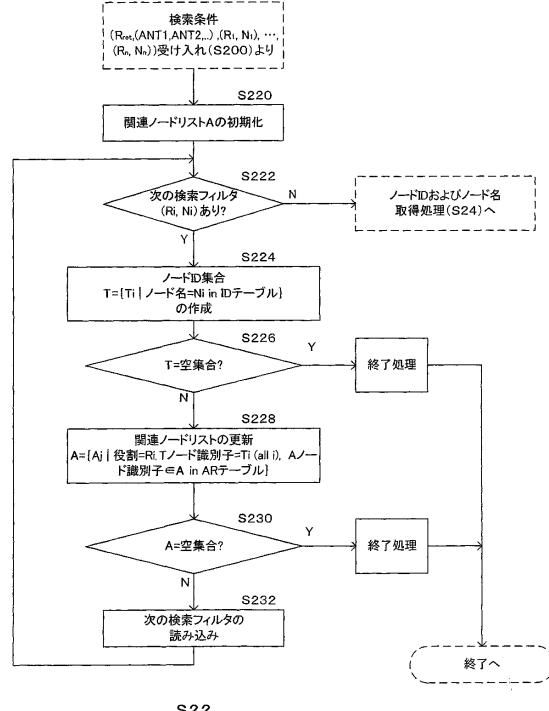
【図14】



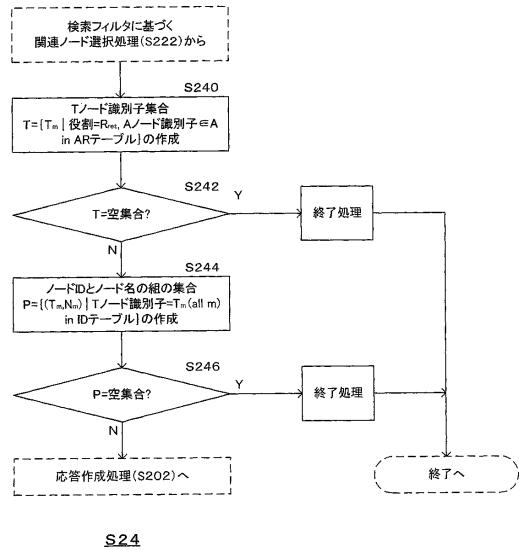
【図15】



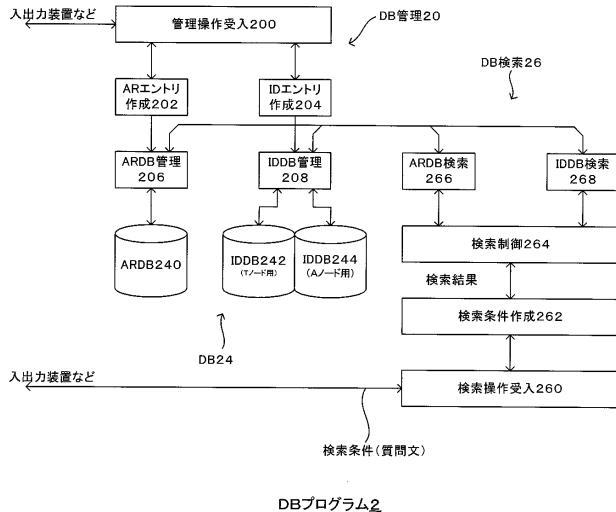
【図16】



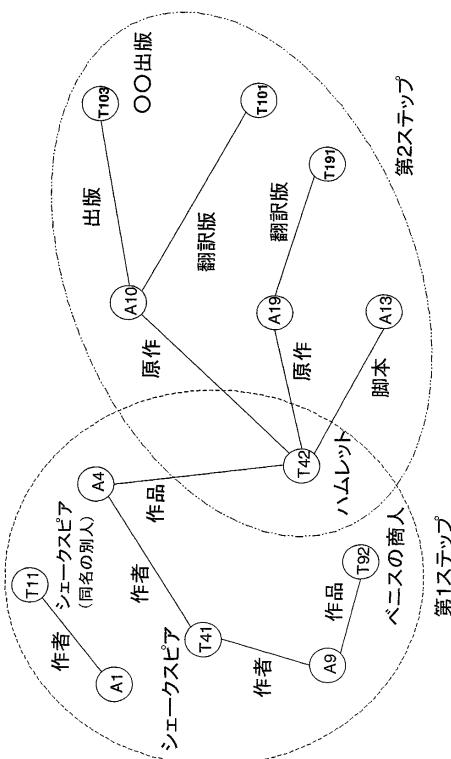
【図17】



【図18】



【図19】



検索例「作家シェークスピアを作者とする作品のうち、戯曲ハムレットを原作とする  
の翻訳版を出版している出版社名を知りたい」

【図20】

ARテーブル

Aノード	Tノード	関連性役割
A1	T11	作者
A4	T41	作者
A4	T42	作品
A9	T41	作者
A9	T92	作品
A10	T42	原作
A10	T101	翻訳版
A10	T103	出版
A13	T42	脚本
A19	T42	原作
A19	T191	翻訳版

【図21】

【図22】

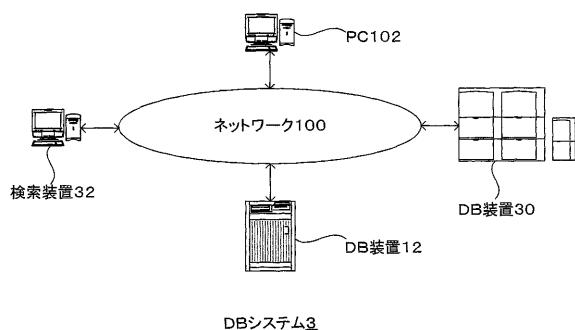
IDテーブル(Tノード用)

ノードID	ノードタイプ	ノード名
T11	作家	シェークスピア
T31	アメリカ人	シェークスピア
T41	作家	シェークスピア
T42	戯曲	ハムレット
T51	商店主	シェークスピア
T81	場所	シェークスピア
T92	戯曲	ベニスの商人
:	:	:
T101	戯曲	ハムレット日本語訳
T103	出版社	○○出版
T191	戯曲	ハムレット中国語訳

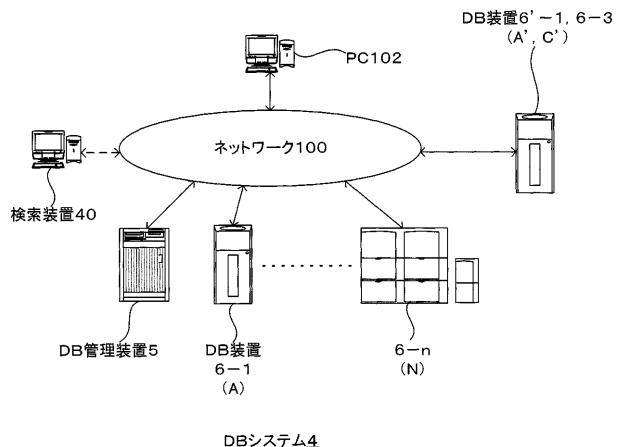
IDテーブル(Aノード用)

ノードID	ノードタイプ	ノード名
A1	出生情報	(null)
A4	著作情報	(null)
A9	著作情報	(null)
A10	翻訳情報	(null)
A13	公演(上演)情報	(null)
A19	翻訳情報	(null)
:	:	:

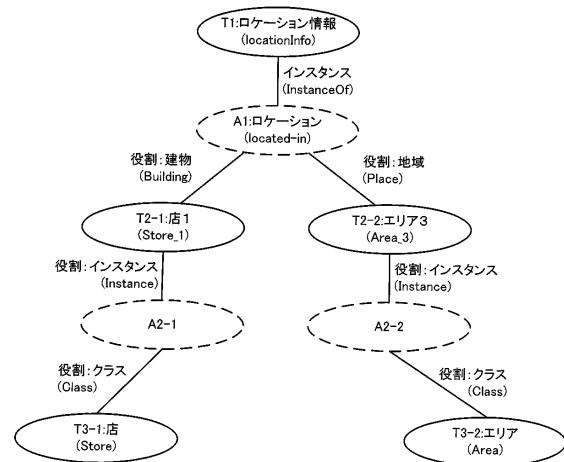
【図23】



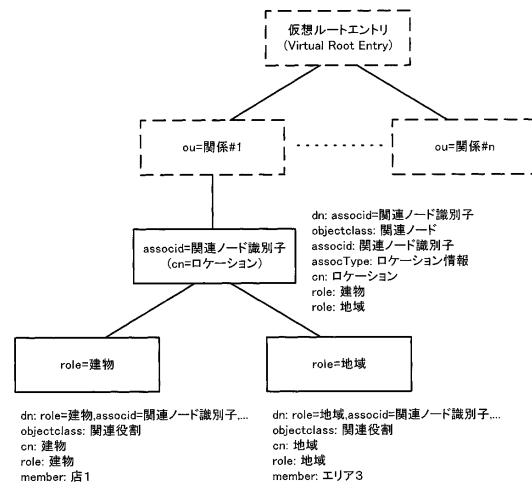
【図24】



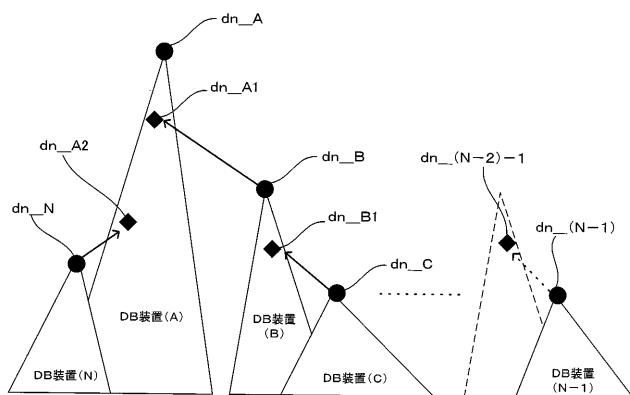
【図25】



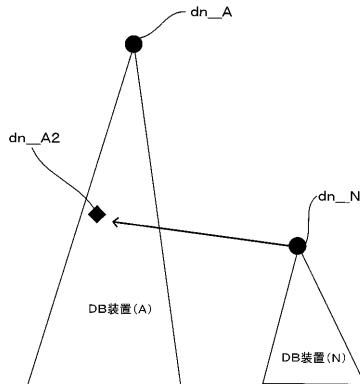
【図26】



【図27】



【図28】



【図29】

DB装置名	最上位エントリ	上位参照エントリ	複製装置
A	dn_A	なし	A'
B	dn_B	dn_A-1	なし
C	dn_C	dn_B-1	C'
D	dn_D	dn_A-2	なし
⋮	⋮	⋮	⋮
N	dn_N	dn_(N-1)-1	なし

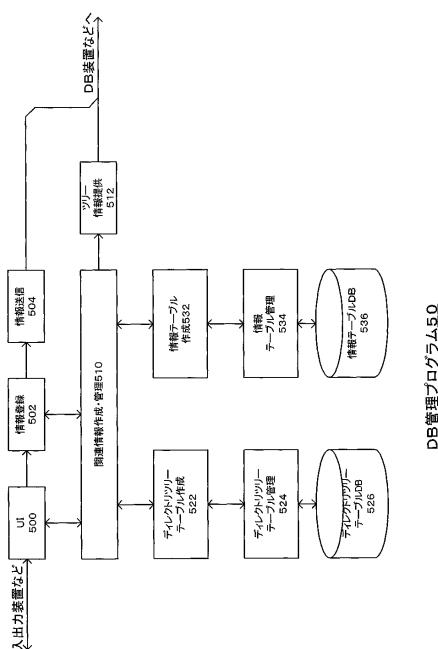
ディレクトリツリーテーブル

【図30】

サブツリー	分類型	分類値	最上位エントリ
a	カテゴリ(Category)	提携会社情報(Affiliate)	dn_A
b	位置情報(LocationInfo)	ロケーション(located-in)	dn_B
⋮	⋮	⋮	⋮
n-1	建物(Building)	店1(Store_1)	dn_(N-1)
n			

情報テーブル

【図31】



【図32】

シェークスピア作品に関するデータの入力

関連の名称: ハムレットについて

関連のタイプ: 著作情報

その他:

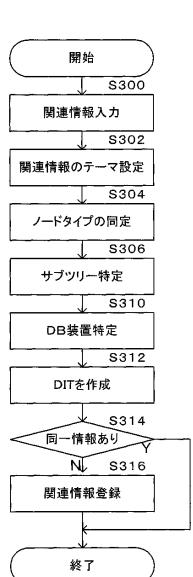
名称: シェークスピア

役割: 作者

その他:

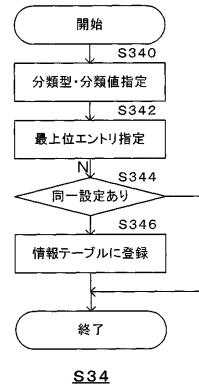
追加 クリア 終了

【図3-3】



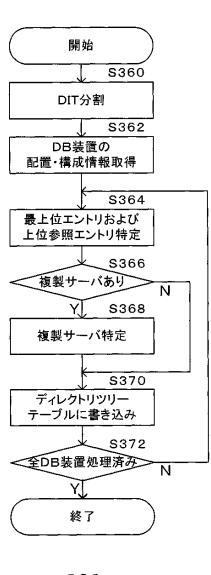
S30

【図3-4】



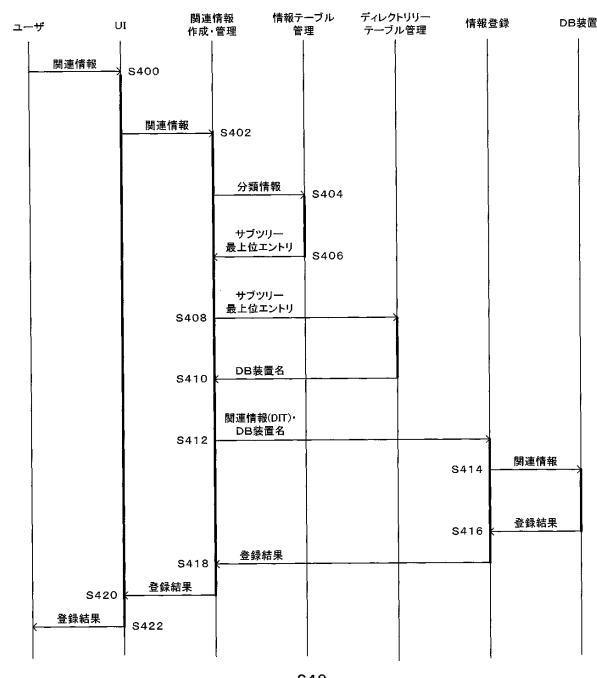
S34

【図3-5】



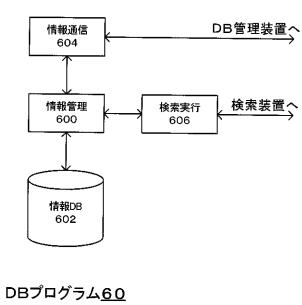
S36

【図3-6】



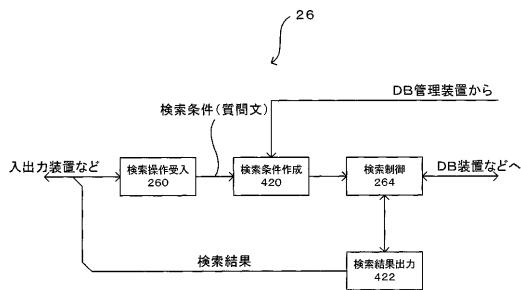
S40

【図37】



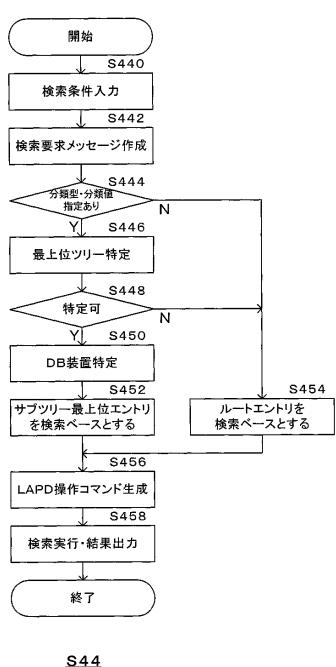
DBプログラム6.0

【図38】



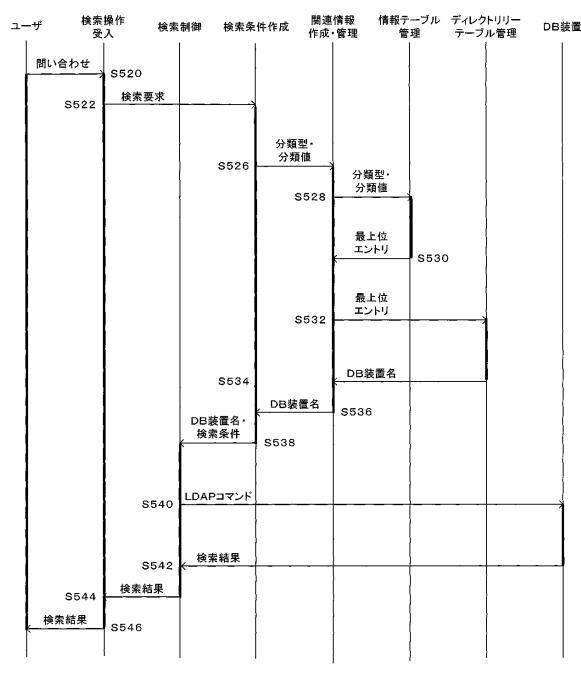
検索プログラム4.2

【図39】



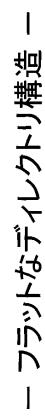
S44

【図40】

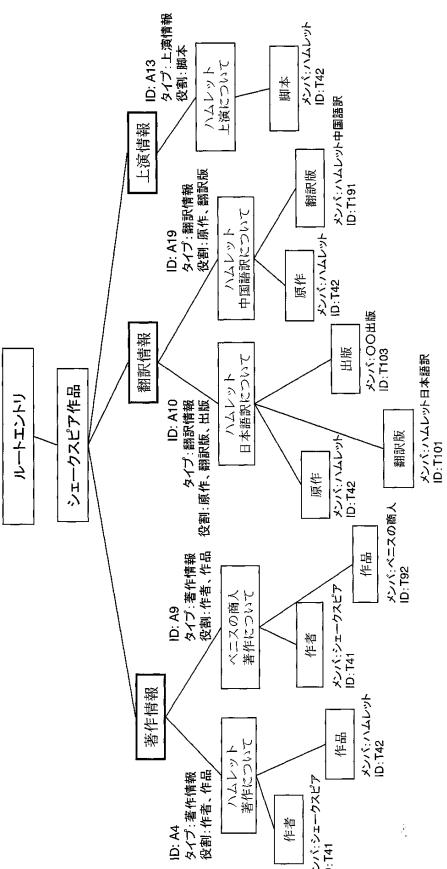


S52

【 図 4 1 】



【 図 4 3 】

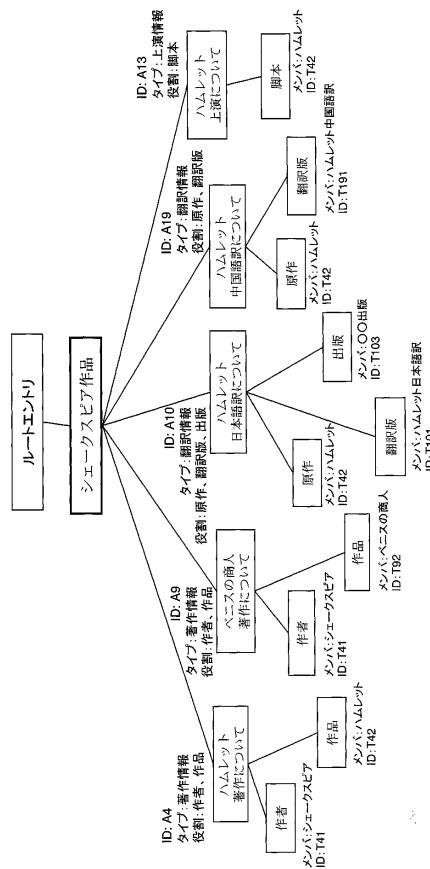


一 関連タイプに基づく階層化

— 関東地域のみ —

二、隋唐五代

【 図 4 2 】

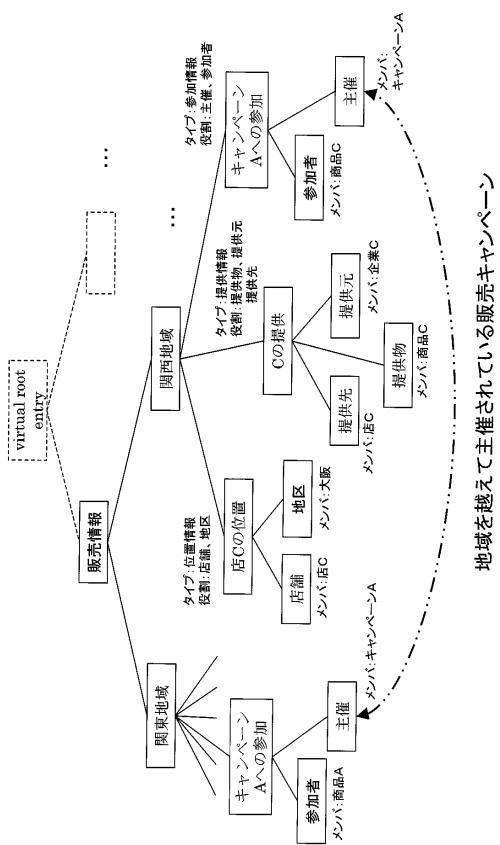


```

graph TD
    A[タイプ：店舗情報  
役割：店舗、地区] --> B[タイプ：接客情報  
役割：接客提供先]
    A --> C[タイプ：販売情報  
役割：販売提供先]
    A --> D[タイプ：在庫情報  
役割：在庫提供先]
    A --> E[タイプ：会員情報  
役割：会員提供先]
    B --> F[店舗の位置]
    B --> G[店舗名、新規]
    B --> H[店舗A]
    C --> I[店舗の参加]
    C --> J[店舗Aへの参加]
    C --> K[店舗Bへの参加]
    D --> L[店舗の提供]
    D --> M[店舗Aへの提供]
    D --> N[店舗Bへの提供]
    E --> O[店舗の主催]
    E --> P[店舗Aへの主催]
    E --> Q[店舗Bへの主催]
    F --> R[地区]
    G --> S[メンバ商品A]
    H --> T[メンバ商品B]
    I --> U[参加者]
    I --> V[メンバ商品A]
    J --> W[メンバ商品B]
    K --> X[メンバ商品C]
    L --> Y[提供元]
    M --> Z[メンバ企業A]
    N --> AA[メンバ企業B]
    O --> BB[提供物]
    P --> CC[メンバ商品A]
    Q --> DD[メンバ商品B]

```

【図45】



【図47】

サブツリー	分類型	分類値	最上位エントリ
a	category	提携会社情報	dn_A
b	region	関東地域	dn_B
:	:	:	:

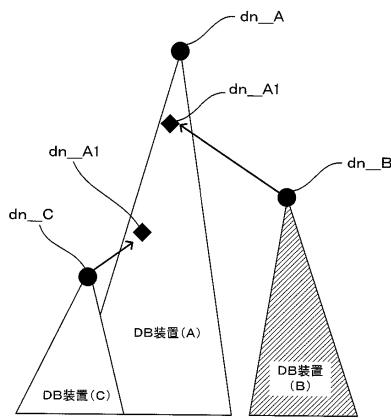
【図46】

サーバ名	最上位エントリ	上位参照エントリ	複製サーバ
サーバA	dn_A	なし	サーバA'
サーバB	dn_B	dn_A1	なし
サーバC	dn_C	dn_B1	サーバC'
:	:	:	:

【図48】

関東地域	における販売情報の検索
参加情報	▼ に關して
役割	▼ をもつメンバの検索
検索フィルタ:	
名称	店B
設定(追加)	クリア
終了	

【図49】

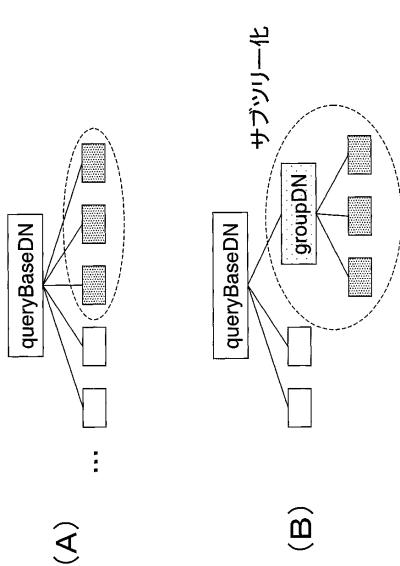


【図50】

```

<searchRequest>
  <condition region="関東地域" category="販売情報" />
  <query result="res1">
    <associationType>参加情報</associationType>
    <targetRole>主催</targetRole>
    <memberSpec>
      <memberID>店B</memberID>
      <role>参加者</role>
    </memberSpec>
  </query>
  <query result="res2">
    <associationType>参加情報</associationType>
    <targetRole>参加者</targetRole>
    <memberSpec>
      <memberID>$res1;</memberID>
      <role>主催</role>
    </memberSpec>
  </query>
</searchRequest>
  
```

【図51】

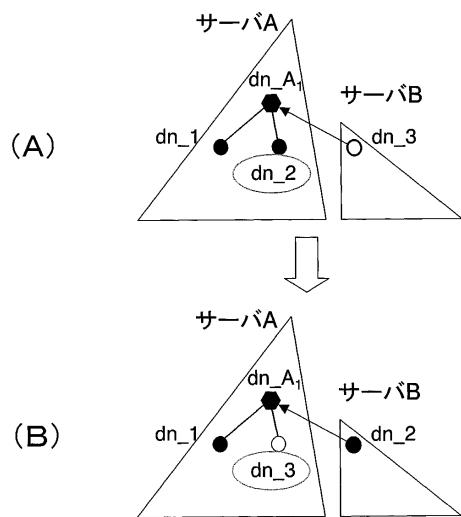


【図52】

検索条件管理テーブル

検索ベース	フィルタ	新検索ベース
queryBaseDN	queryFilter	groupDN
⋮	⋮	⋮
⋮	⋮	⋮

【図53】



【図54】

DB装置名	最上位エントリ	上位参照エントリ	複製装置
A	dn_A-1	なし	なし
B	dn_3	dn_A-1	なし

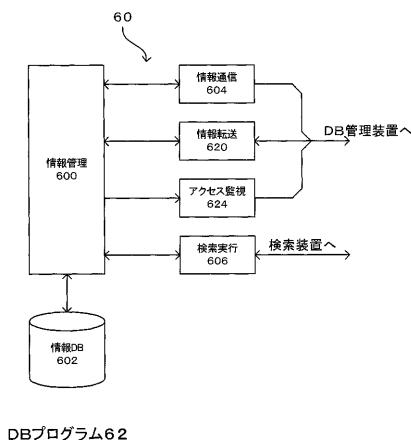
ディレクトリツリーテーブル  
(変更前)

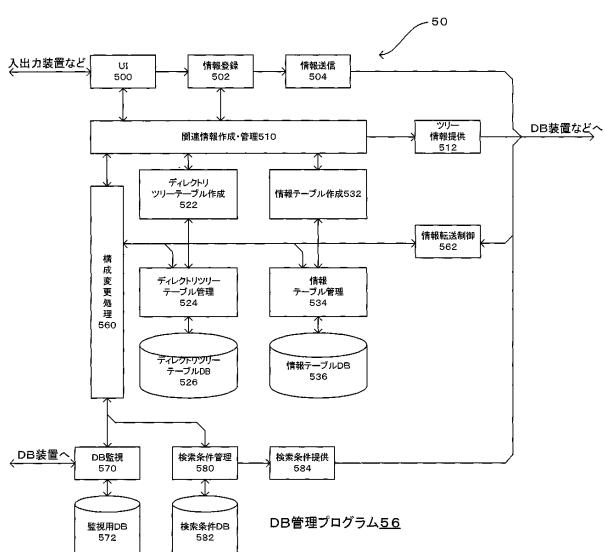
DB装置名	最上位エントリ	上位参照エントリ	複製装置
A	dn_A-1	なし	なし
B	dn_2	dn_A-1	なし

ディレクトリツリーテーブル  
(変更後)

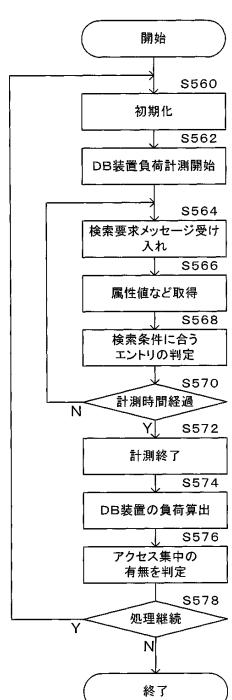
【図55】



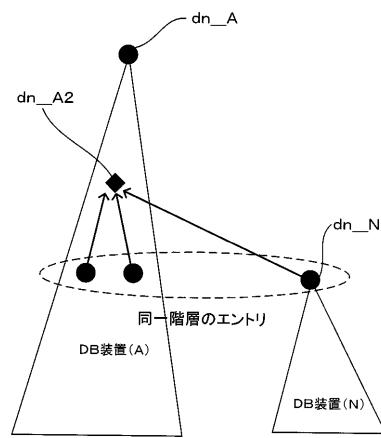
【図56】



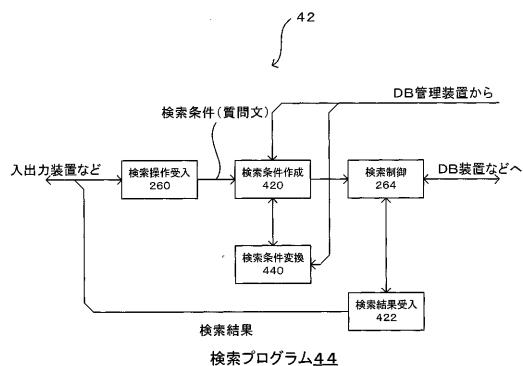
【図57】



【図58】



【図59】



フロントページの続き

(72)発明者 清水 裕之

東京都杉並区高井戸東3丁目29番地21号 日本ヒューレット・パッカード株式会社内

(72)発明者 鵜飼 文敏

東京都杉並区高井戸東3丁目29番地21号 日本ヒューレット・パッカード株式会社内

F ターム(参考) 5B075 ND35 NK02 NK43 NR02 NR12

5B082 GA07 GA16