



(22) Date de dépôt/Filing Date: 2003/04/29

(41) Mise à la disp. pub./Open to Public Insp.: 2004/10/29

(51) Cl.Int.⁷/Int.Cl.⁷ G06F 19/00, G06F 17/60, G06F 17/30

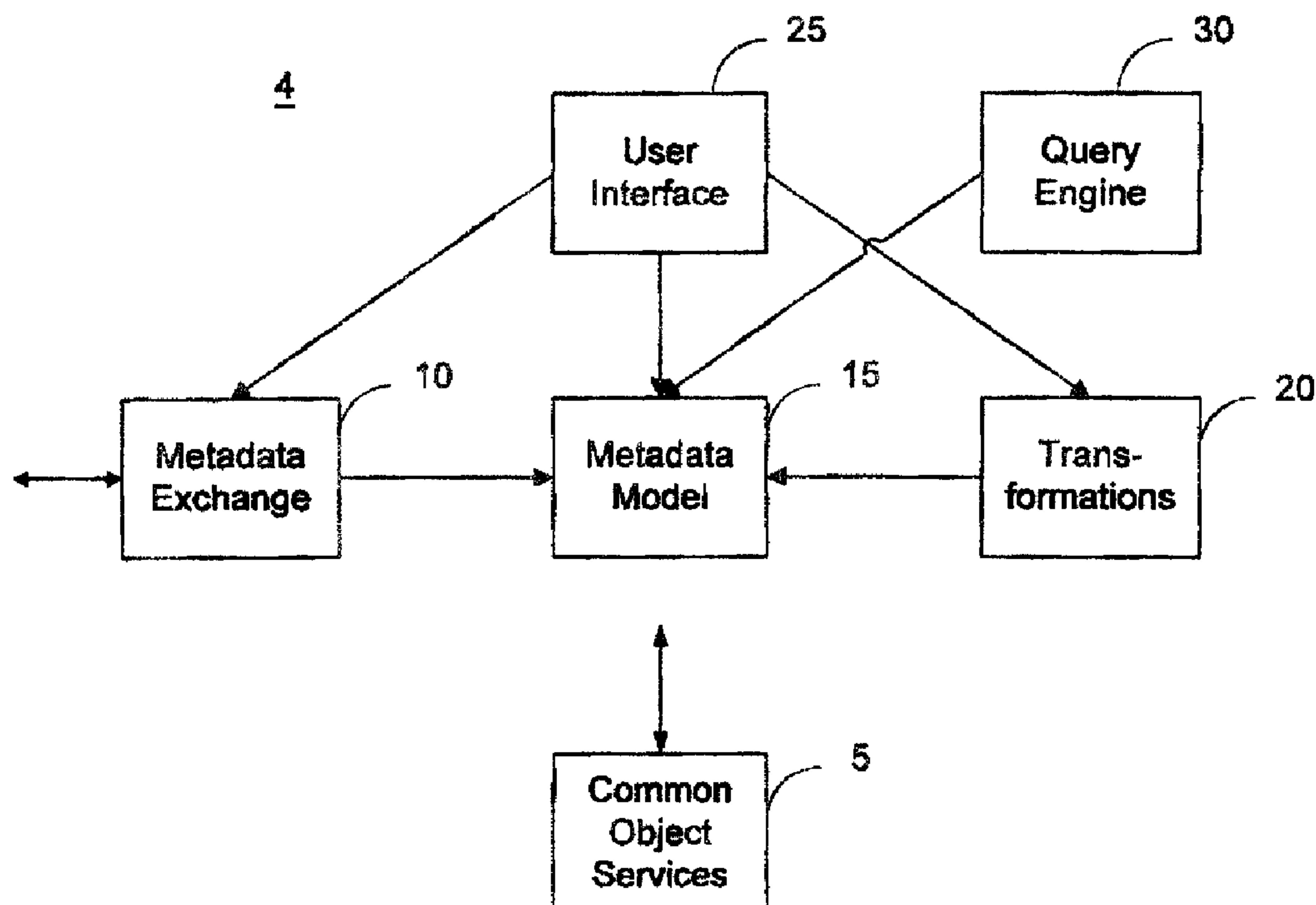
(71) Demandeur/Applicant:
COGNOS INCORPORATED, CA

(72) Inventeurs/Inventors:
POTTER, CHARLES MIKE, CA;
SEEDS, GLEN MICHAEL, CA

(74) Agent: GOWLING LAFLEUR HENDERSON LLP

(54) Titre : **MODELISATION SIMPLIFIEE DE META-DONNEES A DES FINS DE RAPPORT**

(54) Title: **SIMPLIFIED METADATA MODELLING FOR REPORTING**



(57) **Abrégé/Abstract:**

The invention introduces a methodology to automatically derive the relationships between tables and columns within tables so that problems such as those that result in double counting of information are avoided. It comprises a method for creating a report by first defining a model based on an existing database comprising two or more querySubjects, each querySubject comprising at least one fact column and one primary key column, then generating a hierarchy within the model. Next a further fact is introduced into one of the querySubjects. The expression is decomposed into sub-queries so that double-counting of data in ambiguously defined situations is prevented using the further fact to determine that such decomposition is required. Finally, the data is accessed using the sub-queries, and a report is produced.



ABSTRACT

The invention introduces a methodology to automatically derive the relationships
5 between tables and columns within tables so that problems such as those that result in
double counting of information are avoided. It comprises a method for creating a report
by first defining a model based on an existing database comprising two or more
querySubjects, each querySubject comprising at least one fact column and one primary
key column, then generating a hierarchy within the model. Next a further fact is
10 introduced into one of the querySubjects. The expression is decomposed into sub-queries
so that double-counting of data in ambiguously defined situations is prevented using the
further fact to determine that such decomposition is required. Finally, the data is accessed
using the sub-queries, and a report is produced.

Simplified Metadata Modelling for Reporting

FIELD OF THE INVENTION

The present invention relates generally to a metadata model, and more particularly to a metadata model which is suitably used in a reporting system that accesses one or
5 more data stores including relational databases..

BACKGROUND OF THE INVENTION

Currently, information reporting systems use predefined query techniques to hide the complexity of Structured Query Language (SQL) and relational databases. Typically, members of Management Information System (MIS) staff build a database solution by
10 creating user-dedicated tables, relational views or predefined SQL queries. These are then made available to users by means of menus or similar techniques. In these systems, if end-users want to change the purpose of a query they ask the MIS staff to program another query. Alternatively the user may program the SQL query or command themselves. However, the syntax of non-procedural languages (in particular SQL) is
15 complex, and typically, the data structure is not expressed in terms of the users' everyday work. Relational databases store information as well as metadata (data describing the data organisation) such as tables, columns, keys, indices, and their structure and design. Although suited to the overall needs of the customer organisation, these databases will likely contain much that is not of interest to a particular user. In addition, although a
20 query may be syntactically correct, its results may not be what are expected, due to the inherent complexity of a large scale database. Indeed, the results may be totally meaningless.

For these and other reasons modeling tools are often used that allow conceptual modeling of databases. These tools provide a layer on top of the database, and allow the
25 underlying database to be accessed in terms that are more relevant to a particular end application. Such modeling tools include "Impromptu", "Transformer", and "Architect" by Cognos Incorporated.

A previous application "Metadata Modeling", Gowlings file 0884485US, Canadian Application 2,317,166, filed 31 August 2000, describes a three level abstraction model for use in such environments. This is described with reference to Figure 1.

5 The lowest level in the database abstraction is the internal level 1. In the internal level 1, the database is viewed as a collection of files organized according to an internal data organization. The internal data organization may be anyone of several possible internal data organizations, such as B+-tree data organization and relational data organization..

10 The middle level in the database abstraction is the conceptual level 2. In the conceptual level 2, the database is viewed at an abstract level. The user of the conceptual level 2 is thus shielded from the internal storage details of the database viewed at the internal level 1.

15 The highest level in the database abstraction is the external level 3. In the external level 3, each group of users has their own perception or view of the database. Each view is derived from the conceptual level 2 and is designed to meet the needs of a particular group of users. To ensure privacy and security of data, each group of users only has access to the data specified by its particular view for the group.

20 The mapping between the three levels of database abstraction is the task of the DBMS. When the data structure or file organization of the database is changed, the internal level 1 is also changed. When changes to the internal level 1 do not affect the conceptual level 2 and external level 3, the DBMS is said to provide for physical data independence. When changes to the conceptual level 2 do not affect the external level 3, the DBMS is said to provide for logical data independence.

25 Typical DBMSs use a data model to describe the data and its structure, data relationships, and data constraints in the database. Some data models provide a set of operators that are used to update and query the database. DBMSs may be classified as either record based systems or object based systems. Both types of DBMSs use a data model to describe databases at the conceptual level 2 and external level 3.

Data models may also be called metadata models as they store metadata, i.e., data about data in databases.

Three main existing data models used in record based systems are the relational model, the network model and the hierarchical model.

5 In the relational model, data is represented as a collection of relations. To a large extent, each relation can be thought of as a table. A typical relational database contains catalogues, each catalogue contains schemas, and each schema contain tables, views, stored procedures and synonyms. Each table has columns, keys and indexes. A key is a set of columns whose composite value is distinct for all rows. No proper subset of the key
10 is allowed to have this property. A table may have several possible keys. Data at the conceptual level 2 is represented as a collection of interrelated tables. The tables are normalized so as to minimize data redundancy and update anomalies. The relational model is a logical data structure based on a set of tables having common keys that allow the relationships between data items to be defined without considering the physical
15 database organization.

A known high level conceptual data model IS the Entity-Relationship (ER) model. In an ER model, data is described as entities, attributes and relationships. An entity is anything about which data can be stored. Each entity has a set of properties, called attributes, that describe the entity. A relationship is an association between entities.
20 For example, a professor entity may be described by its name, age, and salary and can be associated with a department entity by the relationship "works for". Existing information systems use business intelligence tools or client applications that provide data warehousing and business decision making and data analysis support services using a data model. In a typical information system, a business intelligence tool is conceptually
25 provided on the top of a data model, and underneath of the data model is a database. The data model of existing information systems typically has layers corresponding to the external level 3 and the internal level 1. Some data models may use a layer corresponding to both the external level 3 and the conceptual level 2

Existing data models are used for the conceptual design of databases. When a system designer constructs an information system, the designer starts from a higher abstraction level 3 and moves down to a lower abstraction level 1, as symbolized in Figure 1 by arrows.

5 That is, the system designer first performs logical design. At the logical design stage, the designer considers entities of interest to the system users and identifies at an abstract level information to be recorded about entities. The designer then determines conceptual scheme, i.e., the external level 3 and/or conceptual level 2 of a data model. After the logical design is completed, the designer next performs physical design. At the
10 physical design stage, the designer decides how the data is to be represented in a database. The designer then creates the corresponding storage scheme, i.e., the structure of a database, and provides mapping between the internal level 1 of the data model and the database.

Existing business intelligence tools thus each provides a different paradigm for
15 retrieving and delivering information from a database. Accordingly, it is difficult to share information in the database among different business intelligence tools.

It is common that in a single organization, each group of users has its own established information system that uses its corresponding database. Thus, the single organization often has multiple databases. Those databases often contain certain types of
20 information which are useful for multiple groups of users. Such types of information may include information about business concepts, data retrieval, and user limits and privileges.

However, each information system was designed and constructed in accordance with specific needs of the group, and may use a different business intelligence tool from
25 others. These differences in the information systems and business intelligence tools used do not allow sharing the information already existing in the databases among multiple groups of users.

The previous invention describes a data model or metadata model which realizes the three abstraction levels and provides information that can be shared by multiple users who use those different business intelligence tools or client applications.

5 It does this by providing a metadata model that defines model objects to represent one or more data sources. The metadata model comprises a data access layer, a business layer and a package layer. The data access layer contains data access model objects. The data access model objects include a data access model object that describes how to retrieve data from the data sources. The business layer contains business model objects. The business model objects include a business model object that describes a business
10 view of data in the data sources. The package layer contains package model objects. The package model objects include a package model object which references a subset of business model objects.

It also provides for a metadata model to contain model objects representing one or more data sources. The data sources contain tables having columns. The metadata model
15 comprises a data access layer, a business layer and a package layer. The data access layer contains data access model objects. The data access model objects include table objects that describe definitions of the tables contained in the data sources, and column objects that describe definitions of the columns of the tables contained in the data sources. The business layer contains business model objects. The business model objects include
20 entities that are constructed based on the table objects in the data access layer, and attributes that are constructed based on the column objects in the data access layer. The package layer contains package model objects. The package model objects include a package model object that reference a subset of the business model objects.

What is needed is a way to reduce the apparent complexity and inherent learning
25 barrier to the use of the earlier invention.

SUMMARY OF THE INVENTION

The present invention introduces a methodology that reduces the apparent complexity faced by a user when confronted by this modeling environment.

In one aspect the invention provides for a metadata model that represents one or more data sources, the metadata model comprising a combined data access and business layer including query subjects that describe how to retrieve data from the data sources, the query subjects containing query items.

5 In a second aspect the invention provides for a metadata model containing model objects representing one or more data sources, the data sources containing tables having columns, the metadata model comprising a combined data access and business layer containing query subjects, the query subjects including table objects that describe definitions of the tables contained in the data sources, and query items that describe
10 definitions of the columns of the tables contained in the data sources.

BRIEF DESCRIPTION OF DRAWINGS

Embodiments of the invention will be described with reference to the following figures:

Figure 1 is a diagram showing an example of database abstractions;

15 Figure 2 is a diagram showing an example of a reporting system to which an embodiment of the present invention is applied;

Figure 2A is a diagram showing functions of the metadata exchange, metadata model and transformations shown in Figure 2; and

Figure 3 describes the use of the metadata model by the query engine.

20 DETAILED DESCRIPTION OF THE INVENTION

We now describe preferred embodiments of the present invention. All such embodiments may conveniently be implemented on any general purpose computing platform, including one incorporated in a client/server or networked environment.

Although the abstraction levels described in the earlier invention are necessary, presenting them directly to modelers creates a substantial learning barrier to those using the tool because the levels are not related to how the users perceive the data.

In order to make the model as useful as practicable, it is necessary to reduce these barriers to the point where the operation of the model is intuitive to users that understand the results expected from a reporting application, but who only have limited knowledge of the underlying database, or of database analysis techniques generally. In this context, users are the people (model designers and report authors) who develop report outlines or templates.

We therefore introduce new concepts, requiring (or rather resulting in) the definition of related new terminology. This new terminology is described in this new context. The new terms are then used with the concepts to describe one embodiment of the invention. In doing so, the various simplifications, and rationalizations made to the original concept as described in the previous patent (Metadata Modeling) are made clear.

Figure 2 illustrates a reporting system 4 to which an embodiment of the present invention is suitably applied. The reporting system 4 provides a single administration point for metadata that supports different business intelligence tools or client applications. Thus, it enables different business intelligence tools to extract and interpret data from various data sources in the same way.

The reporting system 4 includes common object services (COS) 5, a metadata exchange 10, a metadata model 15, a metadata model transformer or transformations 20, a user interface 25 and a query engine 30. The fundamental objective of the reporting system 4 is to provide a rich business-oriented metadata model 15 that allows the query engine 30 to generate the best queries of which it is capable, and allows users to build queries, reports and cubes with the aid of the query engine 30 to obtain desired reports from underlying data sources. To this end, COS 5, metadata exchange 10 and transformations 20 are provided.

COS 5 defines the framework for object persistence. Object persistence is the storage, administration and management of objects on a physical device and transfer of those objects to and from memory as well as the management of those objects on the physical device. The double head arrow from COS 5 in Figure 2 represents that COS 5 communicates with all other elements shown in Figure 2. COS 5 performs functions such as creating new objects, storing them on disk, deleting them, copying them, moving them, handling change isolation (check-in, check-out) and object modeling.

The metadata exchange 10 is used to obtain metadata from external physical sources. Metadata is obtained from one or more external sources of metadata. As shown in Figure 2A, external sources of metadata may be one or more data sources 100 and/or one or more metadata sources 101. Data sources 100 contain physical data. Examples of data sources 100 include databases and files. Metadata sources 101 contain descriptive information about data sources. Metadata sources 101 are also known as metadata repositories. Metadata repositories may be third party repositories. Metadata sources 101 generally have underlying data sources 100 containing physical data. The metadata exchange 10 facilitates importation of metadata from external sources 100 and 101 into the metadata model 15. Also, the metadata exchange 10 may facilitate exportation of metadata from the metadata model 15 to external metadata repositories.

The metadata model 15 stores metadata about its underlying one or more data sources 100. It is used to provide a common set of business-oriented abstractions of the underlying data sources 100. The metadata model 15 defines the objects that are needed to define client applications that users build. The metadata model 15 provides three layers to realize three levels of abstractions of data sources 100 as described above referring to Figure 1. The three layers are a physical layer or data access layer 102, a business layer 104 and a presentation layer or package layer 106.

Transformations 20 are used to complete the metadata model 15. For example, when a database is introduced to the reporting system 4, metadata is imported from the database into the metadata model 15. Metadata may also be imported from one or more metadata repositories or other data sources. Sufficient metadata may be imported from a

database that would build only a small number of the objects that would actually be needed to execute queries. However, if such metadata does not have good mapping to the metadata model 15, then the transformations 20 can be used to complete the metadata model 15.

5 The user interface 25 is layered on top of the metadata model 15 as a basic maintenance facility. The user interface 25 provides users with the ability to browse through the metadata model 15 and manipulate the objects defined thereby.

 The query engine 30 is responsible for taking the metadata model 15 and a user's request for information, and generating a query that can be executed against the
10 underlining data sources, e.g., a relational database. The query engine 30 is basically the reason for the existence of the rest of the blocks. The objective of the query engine 30 is to function as efficiently as possible and to preserve the semantics of the original question. A user may ask a question that is not precise. The request may be for something from "customers" and something from "products". But these may be related
15 in multiple ways. The query engine 30 needs to figure out which relationship is used to relate "customers" and "products" to provide the user with information requested.

 The use of the metadata model 15 by the query engine 30 is briefly described with reference to Figure 3. A user uses a business intelligence tool or client application (not shown) to generate a user's request for information. Upon the receipt of the user's
20 request, the client application generates an initial specification 35 based on the request. The specification 35 may be ambiguous. Also, it is not in a form that can be applied to the data sources directly. Using the information that is built in the metadata model 15, the query engine 30 makes the specification 35 unambiguous and builds a query in terms of the data access layer 102 for the specification 35. This intermediate formulation of the
25 query is also called a physical query and is subsequently translated into a data source specification language. The data source specification language may be Structured Query Language (SQL). A query in a data source specification language can be executed on the data sources. Thus, the correct data 40 may be obtained.

The metadata model 15 provides a unified and centralized modeling environment, and application program interfaces for business intelligence tools. The architecture of the metadata model 15 will now be described in further detail.

5 The metadata model 15 is composed of several layers, namely, a physical layer or data access layer 102, a business layer 104 and a presentation layer or package layer 106. These layers correspond to those abstraction levels described earlier and shown in Figure 1. However, in the present invention, the access layer 102 and business layer 104 are combined. This is possible because the entities and attributes used in defining the model have been replaced by the more powerful concepts of concepts of query subjects and
10 query items. When a database schema is imported, the system now creates a set of unified database query subjects that are directly tied to the underlying database, and are also directly usable in creating reports without the need for a separate business layer as was previously the case. This set of objects bring together all of the abstraction and mapping that the previous invention (Metadata Modeling) handles with 'connections' between
15 physical and logical segments of the model. In contrast to a 'view', which is defined by the database administrator DBA, a query subject is abstracted and separate from the underlying database, is applicable to and able to translate different kinds of databases, and provides a translation of the data and metadata into the terminology of the user.

The combined data access and business layer 102, 104 contains metadata that
20 describes how to retrieve physical data from data sources 100. It is used to formulate and refine queries against the underlying data sources 100. The underlying data sources 100 may be a single or multiple data sources, as described above. Examples of data sources 100 include relational databases, such as Oracle, Sybase, DB2, SQL Server and Informix.

The combined data access and business layer 102, 104 contains a part of the
25 model objects that directly describe actual physical data in the data sources 100 and their relationships. These model objects are called query subjects, and contain query items, which are attributes and relate the columns of the underlying databases. In addition, the query subjects may include, among other things, databases, catalogues, schemas, tables, files, columns, data access keys, indexes and data access joins, as well as SQL code that

assists in the transformation of the data. Each table has one or more columns. Data access joins exist between tables..

The query subjects in the combined data access and business layer 102, 104 may be thought of as extended metadata, which are created as a result of importing metadata from data sources 100 and metadata sources 101 provided by users. The information of some data access objects may be available from the underlying data sources 100. The user can customize some objects in the data access layer 102 in order to create data access joins, i.e., relationships between objects that were imported from various data sources.

The combined data access and business layer 102, 104 also describes the business view of the physical data in the underlying data sources 100. It is used to provide business abstractions of the physical data with which the query engine 30 can formulate queries against the underlying data sources 100.

Thus the combined data access and business layer 102, 104 contains information encapsulated in the query subjects and query items that can be used to define in abstract terms the user's business entities and their inter-relationships. The query subjects are reusable objects that represent the concepts and structure of the business to be used in business intelligence environments. They present a single unified business model, with direct relationships to the underlying databases, and can be related to physical data in a number of different data sources 100. The business model also includes business rules and display rules. As well as query subjects and query items, the business model may include keys and joins. Since the query subjects have the ability to incorporate SQL, they may be used directly in creating reports without the need for a separate business layer as previously. This set of objects bring together all of the abstraction and mapping that the previous invention (Metadata Modeling) handles with 'connections' between physical and logical segments of the model. In contrast to a 'view', which is defined by the database administrator, a query subject is abstracted and separate from the underlying database, is applicable to and able to translate different kinds of databases, and provides a translation of the data and metadata into the terminology of the user.

The various parts of the business model are closely related to the underlying data in that, for example, query subjects are related to tables in the data access layer 102 indirectly; and query items correspond to columns in the underlying data sources 100. Business joins exist between query subjects. The new terms query subjects and query items are not totally analogous to older terms, since they imply an extra level of referencing is possible thereby providing powerful new benefits as will be obvious from the following description. Each business model object has a partner in the data access layer 102, i.e., a relationship exists between a table and an entity. The query subjects also hold the metadata representing the business concept. They are collections of attributes.

Query items within the combined layer contain expressions related to columns of tables in the underlying database. A query item is usually directly related to a single column of the underlying data sources 100. For example, the query subject “customer” could have query items “customer name”, “customer address”, and the like. In the simplest case, all the query items of a query subject in the combined access and business layer 102, 104 are related one-to-one to the columns of a single table in the underlying data sources 100. However, the relationship is not always a one-to-one relationship. Also, a query item may be expressed as a calculation based on other query items, and constants, (and columns). For example, a query item may be a summary of data in other query items, e.g., a total amount of all the orders placed by customer.

Query subjects are related to other query subjects by joins.

The query subjects, during the importation process are converted to ‘objects’ containing embedded SQL. During use, such objects may be inserted into existing SQL statements – typically using ‘drag and drop’ graphical techniques – and the embedded SQL takes care of ensuring consistent syntax. In some circumstances, guided manual intervention is required during the creation of these new objects, but generally it is an automated process.

The present technique is a significant improvement over that described earlier, in that almost all of the modelling process is automated, and only occasional intervention is required.

Additional levels of abstraction can be built for different levels of reporting by creating model query subjects whose query items are simply references to other query items, which can be in either database queries or model queries. In other words, references to references can be used, to any arbitrary depth.

- 5 In summary, the invention can be considered to involve the following concepts:
- The embedding of SQL in a database query object (query subject) to facilitate access to the underlying databases.
 - The inclusion of query items in the query subjects, and the ability to use them interchangeably.
 - 10 • The inclusion of filter and calculation building blocks capable of acting on query subjects and query items contained in the query subjects through an extended form of SQL (or pseudo-SQL) that is not constrained by the capabilities of the underlying databases and their related database query languages.
 - 15 • The definition of relationships for the various query items so that they are not necessarily based directly on database columns.

The concept of entities (distinguishable objects to be represented in the database – usually by rows in relations) and attributes (columns or fields in a table) - is replaced by the concept of query subjects and query items. This is not just a change in terms. When a database schema is imported, the system creates a set of unified database query subjects that are directly tied to the underlying database, and are also directly usable in creating reports without the need for a business layer as previously. This set of objects bring together all of the abstraction and mapping that the previous invention (Metadata Modelling) handles with ‘connections’ between physical and logical segments of the model. IN contrast to a ‘view’. Which is defined by the database administrator, a query subject is abstracted and separate from the underlying database, is applicable to and able

to translate different kinds of databases, and provides a translation of the data and metadata into the terminology of the user.

In the case of the present invention, these terms entities and attributes are ‘replaced’ by the terms query subject, which is the subject of the relational query, and query item, which is a particular attribute of a query subject (or rather a pointer or something that can be de-referenced as necessary to ultimately provide that data). These new terms may in fact be pointers or references to their underlying query item or query subject respectively. The new terms are not totally analogous to the older terms, since then imply that this extra level of referencing is possible – and thereby provide powerful new benefits.

The query subjects, during the importation process are converted to ‘objects’ containing embedded SQL. During use, such objects may be inserted into existing SQL statements – typically using ‘drag and drop’ techniques – and the embedded SQL takes care of ensuring consistent syntax. In some circumstances, guided manual intervention is required during the creation of these new objects, but generally it is an automated process.

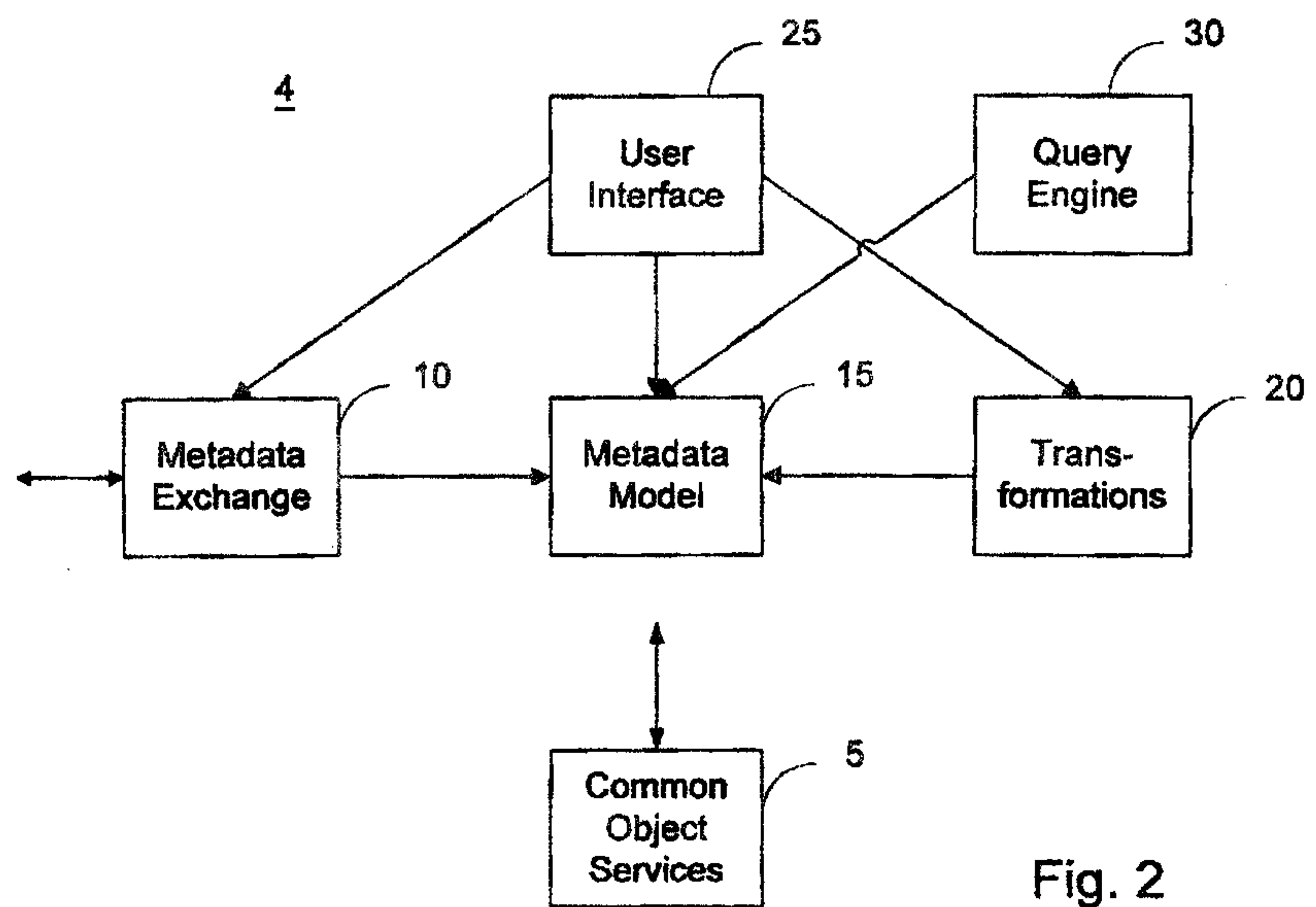
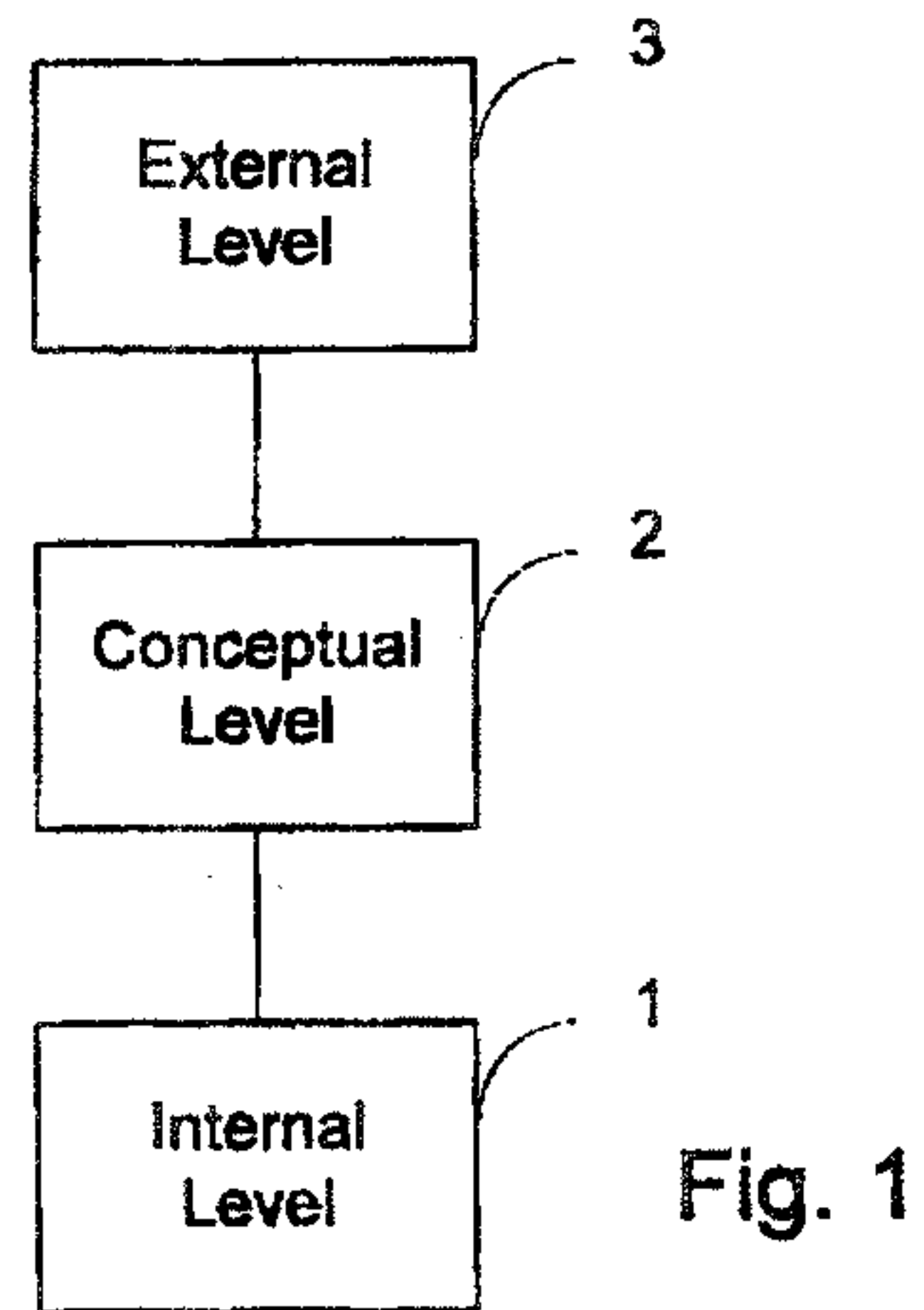
The present technique is a significant improvement over that described earlier, in that almost all of the modelling process is automated, and only occasional intervention is required.

Additional levels of abstraction can be built for different levels of reporting by creating model query subjects whose query items are simply references to other query items, which can be in either database queries or model queries. In other words, references to references can be used, to any arbitrary depth.

What we claim is:

1. A metadata model defining model objects to represent one or more data sources, the metadata model comprising a combined data access and business layer including query subjects that describe how to retrieve data from the data sources, the query subjects containing query items.
2. The metadata model as claimed in claim 1, wherein the query subjects are constructed from metadata in the data sources.
3. The metadata model as claimed in claim 1, wherein the query subjects include a query subject that is imported from a metadata repository.
4. The metadata model as claimed in claim 1, wherein the data sources include a database that contains a table having columns; the query subjects include definitions of the tables contained in the data sources, and query items that describe definitions of the columns of the tables contained in the data sources.
5. A metadata model containing model objects representing one or more data sources, the data sources containing tables having columns, the metadata model comprising a combined data access and business layer containing query subjects, the query subjects including table objects that describe definitions of the tables contained in the data sources, and query items that describe definitions of the columns of the tables contained in the data sources.
6. The invention as substantially described in the specification attached hereto.

1/2



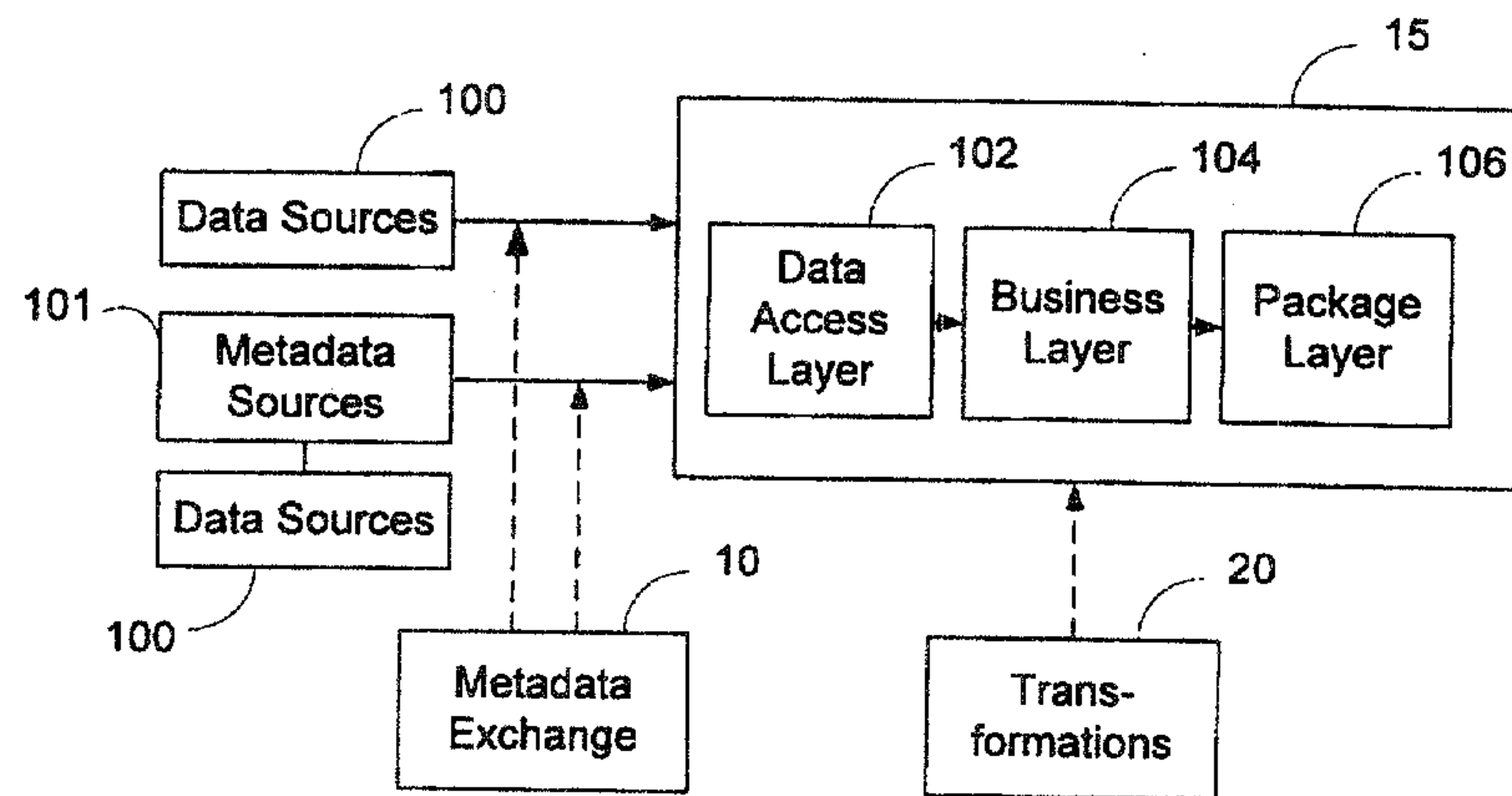


Fig. 2A

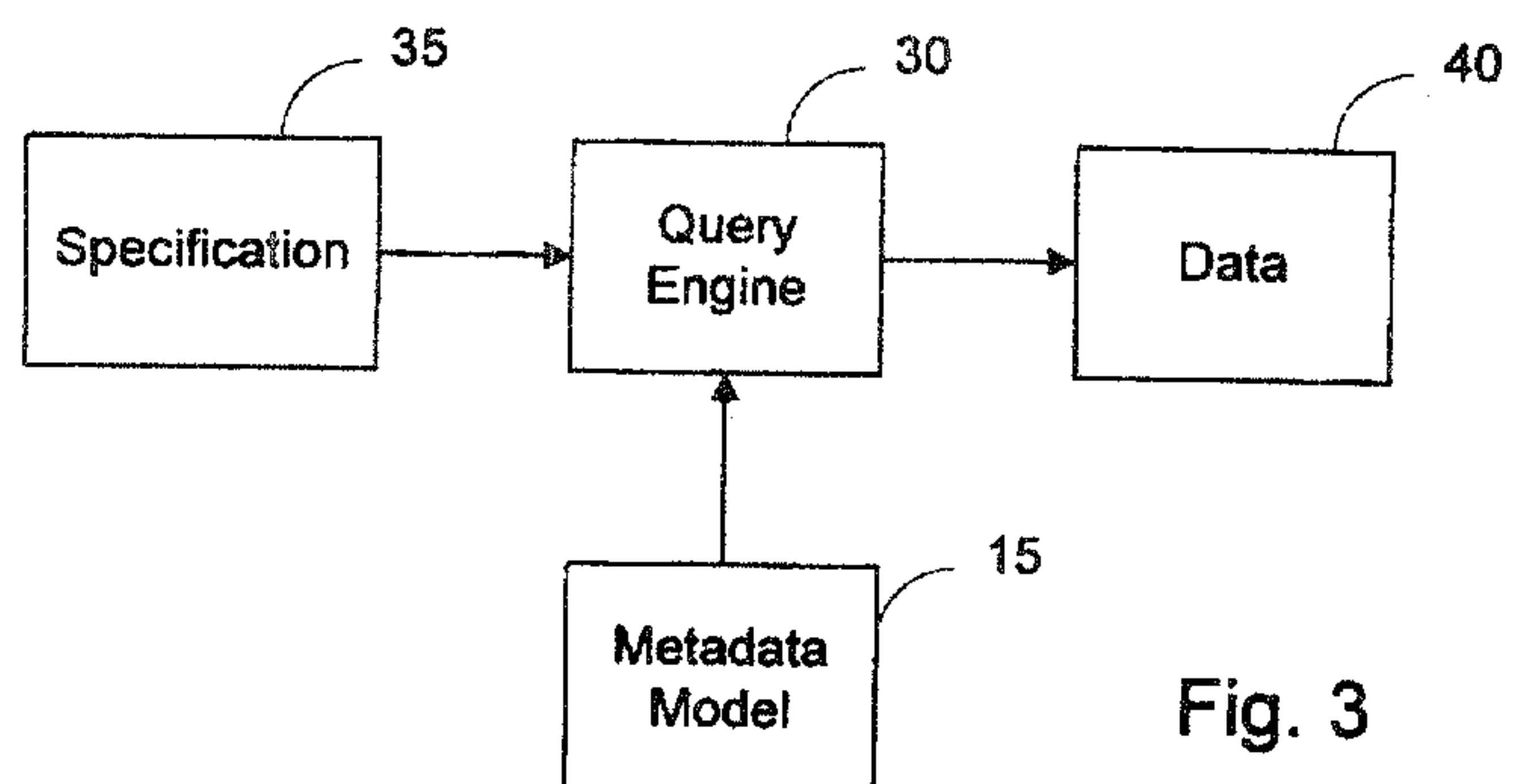


Fig. 3

