

transmits data in analog or digital form over communication links, e.g., a serial link, local area network, wireless link, or parallel link. Also coupled to the I/O bus are a display and a keyboard. Alternatively, separate connections (separate buses) can be used for the I/O interface, display, and keyboard.

[0025] A network **20** connects computers **4** and **6**. The network **20** is any form or medium of digital data communication, e.g., a communication network. Examples of communication network **20** include a local area network (“LAN”) and a wide area network (“WAN”), e.g., the Internet.

[0026] Computer **4** executes instructions of a front end application program **12**. Application program **12** represents a front end component of the business software architecture **2**. Service manager **16**, running on computer **6**, is a service layer between the front end application program **12** and a set of back end service providers **26**. Service manager **16** provides a service interface to front end application program **12** to enable indirect interaction with the set of back end service providers **26** running on computer **6**. This service interface allows for a partial separation of software development for front end application program **12** and the set of back end service providers **26**.

[0027] Computer **6** includes a data storage device **22** that stores a back end database **24** containing data that can be used by the set of back end service providers **26**. Computer **6** also includes a data storage device **8** containing an information repository **18** that defines and describes the services provided by the set of back end service providers **26**. The meta data in repository **18** is organized according to a meta model.

[0028] In general, a meta model is a collection of “concepts” that are the vocabulary with which a certain domain can be described. Meta models typically are built according to a strict rule set, which in most cases is derived from entity-relationship-attribute or object-oriented modeling. The front end application program **12** can access (and interpret according to the strict rule set) the contents of repository **18** via the service manager **16**. These services support the functionality of application program **12** and include retrieving and reading data in addition to modifying stored data. The service providers **26** can access or modify stored data in backend database **24** to provide services to front end application program **12**. To provide the services, the set of back end service providers **26**, upon request from the front end application program **12**, either access or modify stored data in backend database **24** or calculate new data.

[0029] The repository **18** defines a syntax for requesting services provided by the set of back end service providers **26** and semantically describes the services. As front end application program **12** executes, front end application program **12** can use this syntax and semantic description from the repository **18** (accessed through the service manager **16**) to determine what services front end application program **12** can use to meet its requirements. This syntax and semantic description for stored or computed backend data can be referred to as “meta data”. This stored or computed backend data is conceptually organized using object-oriented terminology in terms of business objects, where each business object is an instance of a class or data entity type. In one

example, a class of business objects refers to a relational database table where each row of data in the table represents the data for a particular business object. In this example, each field in the table represents an attribute of the business object class. In another example, there is a class of business objects that partially refers to a relational database table such that some of the fields in the table represent attributes of the business object class and other fields are computed upon request.

[0030] In the business software architecture **2**, services provided to front end application program **12** are focused on data (i.e., data-centric) so the description of these services in repository **18** is also data-centric. Thus, the meta data in repository **18** is structured around representations of classes of these business objects. This meta data includes aspects, or descriptions of these representations of business object classes, and descriptions of available operations on aspects such as select, insert, update, delete, select by relation, and update fields that are provided by service providers **26**. Each description of these aspects includes data attributes as well as actions that can be requested to be executed by the set of backend service providers **26** on instances of these aspects.

[0031] Classifications of data, relations between data classes, prebuilt queries for accessing data, and other descriptions of data provided by the set of backend service providers **26** are represented by repository **18**. This representation, or meta data, of data (e.g., stored in backend database **24**) provided by the set of backend service providers **26** describes different abstract types or classes of data in backend database **24** and how different data classes relate to each other. Objects are instances of these different abstract types. Meta data is information about data rather than content of the data. The meta data also defines a set of pre-built queries that can be executed on the data in database **24**.

[0032] The semantic description in repository **18** can enable front end application program **12** to determine which services to request from service manager **16**. These services often take the form of requesting data to display. Front end application program **12** reads the meta data in repository **18** and can flexibly request data organized in different ways that are specified by the meta data. For example, two service managers **16** with two different repositories **18** handle services that determine prices of books for companies A and B. For A and B, book prices are represented by different aspects with different data fields. Front end application program **12** reads A’s repository **18** to obtain descriptions of data (including a price) concerning a particular book from A’s service providers **26**. Front end application program **12** reads B’s repository **18** to obtain descriptions of data (including a price) concerning a particular book from B’s service providers **26**. Front end application program **12** is able to request and display the information from A’s service provider **26** and the information organized differently from B’s service provider **26** to present the book price information to a user.

[0033] For requesting the services described by the semantic description in repository **18**, service manager **16** provides a canonical interface for services on the business objects in the backend. This canonical interface includes a set of standard operations on the business objects. Such standard operations on the business objects include select, insert,