

many. They also can be implemented by any other programming language on any other platform, e.g., Java on Linux or C# on Windows.

[0042] Repository service provider 30 handles requests to get or modify meta data from repository 18. Query service provider 32 handles queries on data in backend database 24 from front end application program 12. Aspect service provider 34 handles accessing and modifying data, navigation through relations, and calling actions. The aspect service provider 34 has a standard set of methods that correspond to the standard operations on aspects that can be requested from the service manager 16. These standard operations include select, insert, update, delete, select by relation, and update fields. Transaction service provider 40 allows business logic to act on different states of a transaction between front end application program 12 and service providers 26. Locking service provider 42 enables separation of concurrent accesses on data types in backend database 24. Action service provider 44 enables execution of actions on aspects. Query relation service provider 46 is the interface for the target aspect of a relation. In some examples, service manager 16 can have different multiple instances of service providers 32, 34, 40, 42, 44, and 46 for different elements in repository 18 representing services. Upon receiving a request for a service represented by an element in repository 18, the service manager 16 can look up a name of a service provider (e.g., 32, 34, 40, 42, 44, and 46) in the meta data for the element in repository 18. For example, the meta data describing an aspect in repository 18 defines which aspect service provider 34 is designed to handle services for the aspect. The service manager 16 uses this information in the meta data to direct requests from the front end application program 12 to the appropriate aspect service provider 34. Similarly, the meta data describing a query in repository 18 defines which query service provider 32 is designed to handle services for the query.

[0043] The interface provided by the service manager 16 provides requests or commands to front end application program 12. As mentioned previously, standard commands select, insert, update, delete, select by relation, and update fields are standard operations on aspects in the repository 18. These standard operations are provided by aspect service provider 34 and correspond to some of the requests or commands available to front end application program 12. A "Select" command provides a capability such that if the identifiers (or keys) of instances of a data type (e.g., stored in database 24) provided by aspect service provider 34 are known, front end application program 12 can select and read the attributes of these instances. An "Insert" command allows front end application program 12 to add new instances of a data type (e.g., stored in database 24) provided by aspect service provider 34. A "Select By Relation" command provides a capability that if a data type is known, front end application program 12 can find other data types that have relations to this data type as defined in repository 18. An "Update" command provides a capability to modify instances of data types (e.g., stored in backend database 24) provided by aspect service provider 34. A "Delete" command provides the capability to delete one or more selected instances of one or more data types (e.g., stored in backend database 24) provided by aspect service provider 34.

[0044] An "Execute" action command provides a capability to execute a semantically defined action on one or more

instances of one or more data types (e.g., stored in database 24) provided by aspect service provider 34. Either the aspect service provider 34 or the action service provider 44 executes the Execute action command. A "Query" command provides a capability to search and find particular data of interest. The Query command is a method with a fixed set of search parameters and a result set with a defined structure. Queries are defined for particular service modules, or clusters of aspects in the meta data of the repository 18. The query service provider 32 executes a Query command.

[0045] The meta data in repository 18 is classified into data types or classes. The names of meta model classes representing the data type classifications in repository 18 have the suffix "descriptor" to express their belonging to the meta model and to differentiate them from runtime classes used by service manager 16. Descriptors of classes of the meta data of the repository 18 and their class relations are illustrated using an Unified Modeling Language (UML) class diagram 50 in FIG. 4.

[0046] Comparing the meta data to data described by relational database terminology, an aspect in the repository 18 can represent a class or an entity type fully or partially stored in backend database 24 and an aspect descriptor 56 includes attributes for the entity type in addition to other information about the entity type. The meta data in the repository 18 also can include relations descriptors 84 defining relations between aspects that can be implemented in database 24 as relationships using foreign keys in relational databases. The meta data also can include service modules descriptors 54 representing service modules that are aggregations of aspects and have predefined queries for accessing data in database 24.

[0047] The service modules defined in repository 18 are the building blocks for a set of applications (e.g., front end application program 12) in business software architecture 2 for a particular application area or industry. The service modules encapsulate the implementation and business logic and provide access to data and functionality in a unified canonical way. Examples for service modules in repository 18 are "business partner", "employee", "sales order", or "business activity". Service module descriptor 54 describe services modules in the data model of the meta data of the repository 18 and how the service modules can be accessed by queries from application program 12.

[0048] In repository 18, each defined query is an entry point to search instances of a data type (represented by an aspect) provided by service providers 26 via service manager 16. A "key" is an identifier of an instance of a data type provided by service providers 26. An "action" is a specialized method on one or more instances of an aspect. A "structure" is the aggregation of attributes representing the data of an aspect. A "relation" is the relation between objects of a source and a target aspect. A service module group is associated with a service module and is an aggregation of aspects, relations, and queries. An aspect group is associated with an aspect and is an aggregation of relations, aspect actions, and field descriptors 86. The meta data in the repository 18 also includes a text description of each aspect, query, key, action, structure, relation, service module group, and aspect group that is included in the available back end (e.g., backend database 24). So, the organization of the meta data in the repository 18 can be described in terms of those