data about its bikes, customers, hires and payments so that all this information is readily available each time the program runs.

In an object-oriented system, the most obvious way of storing data in such a way as to maintain a seamless development process is to use an object-oriented database. A database stores, organizes and maintains all the data required to support the operations of an organization centrally and in such a way that it can be shared by many different programs. An object-oriented database provides all the storage facilities and functionality of a traditional database, but is specifically designed to implement the types of complex data structure frequently found in object-oriented systems, particularly multimedia, computer-aided design, and geographical information systems.

However, for an information system, such as Wheels, a developer would be much more likely to choose a relational rather than an object-oriented database. Relational database technology has dominated the market for many years; object-oriented databases, on the other hand, are relatively recent. Relational databases are established, flexible and have proved themselves extremely efficient for the sort of data that they were designed to handle. This is precisely the sort of data that information systems deal with. Many business organizations have invested money, time and effort into creating, populating and maintaining their relational database systems, and they would be very reluctant to throw all that away in order to change to an object-oriented database. For the majority of clients, developers have to find some way of allowing new object-oriented programs to share an existing relational database.

Linking an object-oriented program to a relational database poses two problems for developers. First, some means must be provided to allow the program to access the database, and second, the object-oriented analysis and design models have to be modified in some way so that they can be implemented within the relational database framework.

Linking an object-oriented program to a relational database

If an object-oriented program is to access and manipulate data in a relational database, there has to be supplementary code that can establish a connection with the database, relay the program instructions to it, and process the results. JDBC (Java Database Connectivity) is an application program interface (API) that performs these tasks for a program written in the object-oriented language Java. The application program is the Java code and JDBC is the interface that interacts with both the code and the database. The details of the way JDBC works are too complex to include here,