and interaction language SQL, remains the global leader among database paradigms. However there are problems with more complex forms of data that are difficult to handle using standard relational technology. Database construction is an art supported by many tools and methodologies. We have spent some time describing the data modeling approach provided by the entity-relationship model. This modeling approach has the advantage of simplicity and ease of transformation to an appropriate database scheme. However, there are other approaches, referenced in the bibliographic notes at the end of the chapter.

2.4 OBJECT-ORIENTATION

object-oriented

This section introduces the object-oriented approach as an alternative to the relational model paradigm for information systems. The staged representation of the information system development process (section 1.3.4) leaves us with an important difficulty that motivates the object-oriented approach. The gap between the constructs that are available at different stages of the development process makes the transition from one stage to the next inefficient; information may be lost or a simple concept may get hidden in a complex modeling paradigm. The term *impedance mismatch* is often used to describe this problem of translating information from one modeling stage or level of abstraction to another. An important instance of impedance mismatch occurs if an application domain model, expressed using high-level and domain-specific constructs, must be translated into a low-level computational model in order that it can be implemented. In a nutshell, object-orientation attempts to raise the level of the logical and physical computational modeling environment so that the problem of impedance mismatch is lessened.

impedance mismatch

2.4.1 Foundations of the object-oriented approach

object

As might be expected, the concept of an *object* is central to the object-oriented approach. Rather like a tuple of a relation in the relational model, an object models the static, data-oriented aspects of information. For example, a **city** object might have **name**, **center**, **population** among its attributes. A particular **city** object might take the value 'Newcastle-under-Lyme' for the **name** attribute. The totality of attribute values for a given object at any time constitutes an object's *state*.

state

Unlike relations in the relational model, an object also aims to model the dynamic *behavior* of the system. The behavior of an object is expressed as a set of operations that the object can perform under appropriate conditions. For example, the idea of a region is captured not just by specifying a set of points or curves giving the boundary of its extent (the data), but also the operations that we can expect a region to support. Such operations might include the calculation of the region's area and perimeter; plotting the region at different scales and levels of detail; the creation and deletion of regions from the system; and operations that

behavior

Copyright © 2004 Mike Worboys and Matt Duckham