Applications of Object-Oriented Approaches to Expert Systems in the Earth Sciences

Beth L. Duff
Christopher K. Carlson
U.S. Geological Survey
U.S. Department of the Interior
Reston, VA 22092

Abstract

Combining object and rule processing in knowledge-based systems provides a firm foundation for addressing more complex problems such as planning, scheduling, resource allocation and configuration. The features of object-oriented analysis, design, and knowledge representation advantageous to current expert system applications are abstraction, encapsulation, and hierarchy. Examples of object-oriented expert systems in the earth sciences at the U.S. Geological Survey are a knowledge-based geographic information system (GIS), an integrated GIS and expert system, an enhanced digital line graph (DLG-E) extraction advisor, and a sedimentary basin analysis system.

1 The Object Model

The maturation of software engineering has led to the development of object-oriented analysis, design, and programming approaches. These approaches are fundamentally different from traditional design and will require training in object-oriented techniques for software designers and developers. These approaches also require different ways of thinking about decomposition, and they produce software structures that are largely outside the realm of the structured analysis and design culture. The object model represents things in a problem domain area as objects with inherent state (attributes or properties), behavior (actions or services), and identity (uniqueness). It provides expert systems development with the important principles of abstraction, encapsulation, and hierarchy. These principles are not new. What is important is that these principles are brought together in a synergistic way.

1.1 Abstraction

Abstraction is one fundamental way that humans cope with complexity. In software engineering, abstraction of both data and procedural complexity is a necessity for representing problem domains. Booch defines an abstraction as "the essential characteristics of an object that distinguish it from all other kinds of objects and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer." [3] An abstraction focuses on the outside view of an object (and the class it is a member of; see section 2 below) and so serves to separate an object's essential behavior from its implementation. Abstractions have static as well as dynamic properties. For example, a file object takes up a certain amount of space on a particular memory device; it has a name; and it has contents. These are all static properties. The value of each of these properties is dynamic, relative to the lifetime of the object. A file object may grow or shrink in size; its name may change; its contents may change. The operations that can meaningfully be performed upon an object (i.e., the services it provides) and the ways that the object reacts constitute the behavior of the object.

1.2 Encapsulation

Abstraction and encapsulation are complementary concepts. Encapsulation, also known as information hiding, is the process of hiding all the details of an object that do not contribute to its essential characteristics.
1.3 Hierarchy

Hierarchy is a ranking or ordering of abstractions. The two most important hierarchies in a complex system are its class structure (i.e., generalization-specialization relationships) and its object structure (i.e., whole-part relationships). Inheritance is an essential element of object-oriented systems and represents the most important hierarchy. It allows specialized classes or subordinate parts to obtain some of their properties or behaviors from generalized classes or superior parts or wholes. Inheritance defines a relationship among classes, wherein one class shares the structure or behavior defined in one or more other classes (termed single and multiple inheritance, respectively).

An example of inheritance from the mapping community is the superclass "stream/river" with subclasses being "perennial stream/river" and "intermittent stream/river." Some of the attributes of streams and rivers, such as "flow from a lake, pond, spring or seep" or "are braided" are of interest whether or not the stream/river is perennial or intermittent. Other characteristics are of interest only for perennial streams, such as "channel width" or for intermittent streams, such as "is in an arid region."

2 Object-Oriented Modeling

The object model introduces several novel elements that build upon the more traditional methods of structured analysis, design, and programming. The object model has significant benefits over other models; the most important being that the use of the object model leads to systems that embody the five attributes of well-structured complex systems (hierarchy, randomness, intracomponent linkages, subsystems, and origination from simple systems).

Booch [3] cites five other practical benefits to be derived by using the object model. The expressive power of all object-based and object-oriented programming languages is exploited. The object model encourages the reuse not only of software, but of entire designs by using class libraries of previous analysis and design work. Booch has found that object-oriented systems are often smaller than the equivalent nonobject-oriented implementations due to the inheritance principle and more compact representation of problem domains. The object model produces systems that are built on stable intermediate forms, and thus are more resilient to change. The object model reduces the risk of developing complex systems primarily because implementation is spread out across the life cycle rather than occurring as one big event. Last and most important, the object model appeals to the workings of human cognition [12].

Booch refines the definition of an object as a tangible entity that exhibits some well-defined behavior to a more formal definition - an object has state, behavior, and identity; the structure and behavior of similar objects are defined in their common class; and the terms instance and object are interchangeable [3]. State encompasses all of the properties of the object plus the current values of each of these properties. The behavior of an object is completely defined by its actions or the services it provides. Identity is that property of an object that distinguishes it from all other objects; that is, all objects are distinguishable from one another [7].

Coad and Yourdon [4] define object as an abstraction of something in a problem domain that reflects the capabilities of a system to keep information about it, interact with it, or both; or as an encapsulation of attribute values and their exclusive services. Class instance is a synonym for object. They go on to define a class as a description of one or more objects with a uniform set of attributes and services, including a description of how to create new objects in the class.

3 Applications of Object-Oriented Methods in Earth Sciences

Examples of expert systems at the U.S. Geological Survey that have used object-oriented knowledge representation techniques follow.

3.1 KBGIS-II

A recently prototyped knowledge-based geographic information system (KBGIS-II) was designed to satisfy several general criteria for geographic information systems (GIS). The system has several functions that include query answering, learning, and editing. The main query finds constrained locations for spatial objects that are describable in a predicate-calculus (PROLOG) based spatial object language. The main search procedures include a family of constraint-satisfaction procedures that use a spatial object knowledge base to search efficiently for complex spatial objects in large, multilayered spatial data bases. These data bases are represented in quadtree form. The search strategy is designed to reduce the computational
cost of the search in the average case. The learning capabilities of the system include the addition of new locations of complex spatial objects to the knowledge base as queries are answered and the ability to inductively learn definitions of new spatial objects from examples [14].

A spatial object knowledge base stores both definitions of, and useful information about, all objects known to the system. This knowledge is implemented in terms of a "slot and filler" structure and a discrimination net data structure. Information concerning search heuristics, object definitions, classification, and complexity, as well as low level search procedures that may be invoked in searching for spatial objects, is stored in the knowledge base. Information concerning known locations of spatial objects previously found are stored in the discrimination net data base.

3.2 Integrated GIS/Expert System

The U.S. Geological Survey is developing the specification for an interface between Environmental Systems Research Institute's ARC/INFO GIS and Neuron Data's NEXPERT OBJECT development shell. NEXPERT OBJECT provides the capability to represent class structure models with attached expert system-like rules. This research involves the provision of a "loose coupling" between the two software packages. The project's intent is to provide an enhanced tool for modeling and solving earth science problems [15].

A prototype application involving a rule-based map of ground-water contamination potential was created to develop the interface specification. NEXPERT OBJECT allows the representation of hydrologic features of interest in a class structure for this problem domain. To apply rules in the map model, NEXPERT OBJECT needs to read and write records in an ARC/INFO data base. The interface will allow NEXPERT OBJECT to interact with ARC/INFO and vice versa. It would also be possible to execute a set of ARC/INFO commands to display spatial data from a NEXPERT rule "firing." The interface and interactive execution between the two packages will assist earth scientists in solving spatial problems by employing expert knowledge in conjunction with spatial display and analysis.

3.3 DLG-E 1:24,000 Extraction Advisor

The Geological Survey has also designed an enhanced digital line graph (DLG-E) representation of maps to meet the increasing information demands for digital cartographic data. In this model, phenomena are represented by geographic and cartographic data entities. Entities represent individual phenomena in the real world. A feature is an abstraction of a set of entities, with the feature description encompassing only selected properties of the entities (typically the properties that have been portrayed cartographically on a map). Buildings, bridges, roads, streams, grasslands, and counties are examples of features. A feature instance, that is, one occurrence of a feature, is described by feature objects and spatial objects. A feature object, in the DLG-E model, identifies a feature instance and its nonlocational attributes. Location aspects of the feature instance are represented by spatial objects: points, nodes, chains, and polygons. Representation rules describe all feature object types that can be used to represent a feature.

To establish a domain of features, classes of spatial entities, or views in Survey terminology, were used. Based on common defining characteristics, the five views are cover, division, ecosystem, geoposition, and morphology. Because each view is independent, a single point on the Earth's surface can be represented under multiple views, and features within any one view may coexist with features in the same or any other view. For example, if a monument occurs at a boundary point, two features will be recorded, "boundary point" in the "division" view and "point monument" in the "cover" view.

Operational considerations and user requirements have led to features being grouped into themes. Thirteen themes are identified: bathymetry, boundaries, built-up, census, hydrography, hydrologic units, hypsography, morphology, nonvegetative surface cover, Public Land Survey System, survey control and markers, transportation, and vegetative surface cover. (See Figure 1.) A feature may occur in more than one theme. The theme in which a particular feature object should be included is determined by its nonlocational attributes. For example, if a tunnel provides passage for a road or railway, it is in the "transportation" theme. However, if a tunnel provides passage for an aqueduct or a canal/ditch, then it is in the "hydrography" theme.

Using Information Builders' object-oriented expert system shell, LEVELS OBJECT, a prototype knowledge base was developed that includes some DLG-E features, their attributes, and extraction specifications for 1:24,000-scale Survey products. Extraction rules describe the characteristics of the feature that must be considered in the decision to capture the feature.
3.4 Sedimentary Basin Analysis

Research efforts by Miller [11] also at the Geological Survey are currently directed at exploring the feasibility of applying expert systems and knowledge acquisition techniques to the design and development of a system of classification and geological analysis of sedimentary basins. The primary objective is the design of an object-oriented expert system, interfaced with a GIS, that captures both the logic used to define the geologic concepts and the uncertain reasoning that enables geologists to understand and reconstruct the geologic history of a sedimentary basin. This system will provide these capabilities through documentation of major basin analysis components such as stratigraphy, structural geology, and sedimentology. This system is being designed to analyze the traditional concepts of source, reservoir, and trapping mechanism; to help in the diagnosis of geological conditions favorable for the occurrence of petroleum or other energy resources; and to aid in the assessment of these resources.

The NEXPERT OBJECT development shell is being used to develop applications for sedimentary basin studies. This development system is a hybrid rule- and object-based expert system software program that provides a graphic environment for applications development. It has the ability to support both reasoning and knowledge representation; rules represent reasoning and objects describe the conditions upon which the reasoning is performed.

A second and similar expert system shell, MAHOGANY, is being investigated for use in developing a knowledge base for sedimentary basin analysis techniques. One of the promising features is

Figure 1. - Example of theme (hydrography) and view (cover/water) in the enhanced digital line graph.
Figure 2. - Part of a Basin Analysis Class Network.

that MAHOGANY permits the geologist to assign certainties directly to the rules and data within the knowledge base. This capability allows a user to deal with the uncertainties that are an important part of the geologic knowledge base.

Figure 2, taken from part of the basin analysis object network, illustrates basin stratigraphy as a class of five objects (principal reservoirs, major stratigraphic units, lithology of major stratigraphic units, trapping mechanisms, and source beds). The properties for two of the objects are shown for the principal reservoirs and lithology of the major stratigraphic units.

4 Conclusion

Object-oriented methodologies and computer systems will continue to grow in use because they are easily understood by managers, analysts, designers, programmers, and users. Programmer productivity increases during development, and maintenance effort is reduced. However, to realize these benefits, most software development organizations need to undertake extensive training programs in object-oriented techniques. Yourdon [16] points out that an organization cannot make effective use of new tools or methodologies unless it already has a formal documented process for developing systems in place. However, a survey of data processing organizations in the United States showed that less than 5 percent had instituted a formal methodology for developing systems.

Linking object-oriented methods to knowledge-based technology is a recent and powerful innovation in information technology. Each of these technologies has many advantages over traditional programming techniques; when used together, their advantages are enhanced even further. Although rule-based systems have been used extensively to solve such problems as diagnosis and assessment, they are often inadequate when used on more complex problems such as planning, scheduling, resource allocation, and configuration [10]. The combination of object and rule processing provides a strong base for investigating more complex issues.
References


