Meeting the additional demands placed on cost accounting by the Total Quality Management philosophy requires computerization. Activity-based cost accounting is designed to satisfy the additional demands of TQM. But maintaining cost data at the activity level involves at least an order of magnitude more data that does cost center accounting. Thus, activity-based cost accounting demands a redesigned, computerized cost accounting system. An object-oriented approach to the design and specification of activity-based cost accounting systems is outlined and illustrated. Three phases in the approach are first described. Then the approach is illustrated by applying it to the design of an data processing system for activity-based cost accounting. As with all management information systems and most decision support systems, an activity-based cost accounting system is seen to be highly dependent on the transaction processing systems of the business. The point is made that this makes the clean interfaces between objects in object-oriented systems particularly advantageous, if both the transaction systems and the costing systems are based on a common set of business objects, defined in a business-wide repository.

Many businesses today are making a concerted effort to ensure their survival by matching or exceeding the product and service quality of their competitors. The Director of the Malcolm Baldrige National Quality Award recently reported that his office has distributed more than 700,000 copies of the Criteria for that award during the past six years. For the first time, two of the five 1992 awards went to service companies (AT&T Universal Card Services and The Ritz-Carlton Hotel Company) which indicates that the total quality management philosophy is spreading beyond manufacturing companies.

The significance of quality is always relative to the cost of producing it. It is conceivable that an incompetent organization could produce excellent quality at a high cost, for a while. Only a very competent organization can produce high quality at low cost and it is this type of producer that survives.

Thus, a continuous striving to improve quality should be done with an eye on the cost impacts of the process and product changes considered for adoption. The cost accounting system should provide the data required for this analysis. It should provide a basis for estimating the marginal change in cost that change in a process or product design will cause.

Unfortunately, traditional cost accounting systems often hide this information in a cost aggregation called overhead. All costs except direct labor and materials are accumulated in cost center overhead accounts. These overhead costs can be allocated to products and services produced during a period, based on some measure of output but this does not allow the marginal cost of the product or service to be determined. If one unit less is produced there is no basis for estimating how much overhead charges will be reduced.

To obtain this marginal cost information, many of the companies striving for continuous quality improve-
ment are turning to Activity-Based Costing (ABC). (Turney, 1992, cites a number of companies. Kaplan, 1988, describes dissatisfaction with conventional cost accounting.) In this approach, cost centers are subdivided into activities. (An activity is a unit of work, the most detailed process or procedure of interest; it is the atomic unit in the ABC model.) Costs are accumulated by activity and classified as either variable with some unit of measure for the activity or fixed per time period.

Computerization of the cost accounting system makes the ABC approach feasible. Using activities instead of cost centers as the basic unit of cost accumulation explodes the amount of data in the system. The ABC model also includes non-financial data needed to assess process improvement ideas which increases the amount of data still more. Without a well-designed computerized system, ABC is not likely to be cost-effective.

The purpose of this paper is to explain an object-oriented approach to designing the application software for an ABC system. The object-oriented approach is described in general terms in the next section. This is followed by a section on the details of the ABC model. Then the first step in the object-oriented approach is taken in the section entitled, Objects and Relationships Identified. The identified objects are classified in the Kinds of Objects section based on the program modules they require. This is followed by a section on the system processing specifications required for business and report objects. The final section summarizes the design approach.

Object-Oriented Approach

An object-oriented approach to computer programming has proven to be superior to other approaches for the development of interactive personal computer software. It provides a way of modularizing software development projects that creates better module interfaces than other methods. This makes it possible to manage the extreme complexity that can arise when attempting to build “user friendly” systems that allow the user great flexibility.

Given that the application software for a system is going to be written using the object-oriented approach, there are several compelling reasons to organize the software design specifications in terms of objects. Firstly, if they are not so organized, the programmer will convert whatever specifications are given to an object-oriented form. Secondly, verifying that the software created satisfies the specifications is simplified if both are object-oriented. Thirdly, an object-oriented system design effort is easier to manage for the same reasons as were cited in the case of programming, namely, effective module interfaces.

A number of object-oriented approaches to application software design have been proposed in recent years. Peter Coads (1991), David Taylor (1992) and John Gessford (1992) have all proposed object-oriented systems development methodologies. The Gessford approach, which will be used here, takes full advantage of the fact that an object-oriented system is a simulation system. Objects are simulated by the computer in such systems. The approach divides the work of designing the simulation into three phases. In the first phase, the business objects to be simulated are identified through the use of entity-relationship analysis. In the second phase, entity types and relationships identified in the first phase are used to define classes of business objects to be simulated. These classes are then categorized as either permanent, event, or dependent business objects. In the third phase, the program modules required to simulate each class of business objects are specified. Each of these phases is briefly described in the following subsections.

Entity-Relationship Analysis

The design of business databases requires that each item of data used in the business be classified according to the type of object it describes. For example, the customer address is classified as data about customer objects. The weight of a product is classified as data about product objects.

Defining the information requirements of a business in terms of the things the business needs to know about is quite different from defining them in terms of the accounting transaction documents, reports, messages, and querying capabilities needed to effectively run the business. These “things” (objects) can be derived from the documents and forms with which business people are familiar but they are not the same.

Entity-Relationship Analysis (ERA) is a well-established and widely used method for setting up the classifications of business data for database design purposes.. The classifications are called entity types in ERA and the specific items of data about an entity type are called attributes. Customer and Product are two entity types that ERA usually identifies.

Relationships between entity types define linkages between instances of the types that have meaning (they hold information). For example, a business in which
Related Content

Neurocomputing Approach to Residential Property Valuation
[www.igi-global.com/article/neurocomputing-approach-residential-property-valuation/55685?camid=4v1a](www.igi-global.com/article/neurocomputing-approach-residential-property-valuation/55685?camid=4v1a)

Self-Determined Adoption of an ICT System in a Work Organization
[www.igi-global.com/article/self-determined-adoption-ict-system/46969?camid=4v1a](www.igi-global.com/article/self-determined-adoption-ict-system/46969?camid=4v1a)

A Path Analysis of the Impact of Application-Specific Perceptions of Computer Self-Efficacy and Anxiety on Technology Acceptance
[www.igi-global.com/chapter/path-analysis-impact-application-specific/62804?camid=4v1a](www.igi-global.com/chapter/path-analysis-impact-application-specific/62804?camid=4v1a)

Should we Take Disintermediation in Higher Education Seriously?: Expertise, Knowledge Brokering, and Knowledge Translation in the Age of Disintermediation
[www.igi-global.com/chapter/should-take-disintermediation-higher-education/69754?camid=4v1a](www.igi-global.com/chapter/should-take-disintermediation-higher-education/69754?camid=4v1a)