Towards Integrating Advanced Database Concepts Into Accounting

David H. Olsen, Utah State University
Sharon L. Kimmell, (skimmell@uakron.edu), University of Akron

Abstract

This paper provides a review of several recently developed, advanced database technologies which could dramatically affect current accounting practice. These technologies are discussed in the context of a proposed taxonomy along four specified dimensions. The paper also provides a description of each of these technologies and an associated table of key concepts as well as a definition or description. The paper discusses the advantages and disadvantages of each of these newer technologies and their applicability to the accounting function in companies. Where appropriate, concrete examples of accounting applications for these technologies are provided.

Introduction

Changes in the area of information technology are occurring at an unprecedented rate. Just five years ago, almost no one in accounting research or practice used any part of the Internet. Now, features such as e-mail, document transmission, and on-line searches are integral to several basic accounting functions. Similarly, active databases, object-oriented databases, and data warehousing are newer database technologies that will likely have substantial impacts on the way accounting processes are accomplished.

In this paper, we review seven newer database-related technologies that are likely to have an impact on accounting practice. The paper continues with the presentation of an advanced accounting database taxonomy and the four dimensions that compose the taxonomy in section

II. A review of conventional database technology follows in section III, and in section IV we review active databases, Intranet databases, data warehousing, data mining, object-oriented databases, business rules, and temporal databases. We conclude in section V.

Advanced Accounting Database Taxonomy

A number of new database technologies are appearing or are becoming more mainstream in business. Some are designed to support day-to-day operations while others are specific to decision support. Some of the new database technologies are appropriate for end users while others are primarily targeted for application developers. In Tables I and II, we introduce an advanced accounting database taxonomy and four dimensions to classify these newer database technologies.

Readers with comments or questions are encouraged to contact the authors via e-mail.

Table 1
Advanced Accounting Database Taxonomy

Advanced Database Technology	Work Type	Appropriate Systems Development Life Cycle Phase	User Type	Specificity of Purpose
Conventional Relational Databases	Operational and Decision Support	Implementation	Developers	General Purpose
Object-Oriented Databases	Operational and Decision Support	Implementation and Conceptual	Developers	General Purpose
Data Mining	Decision Support	Implementation	End Users	Specialized Applications
Data Warehouses	Decision Support	Implementation	Developers	Specialized Applications
Business Rules	Operational and Decision Support	Conceptual	Developers	General Purpose
Intranet Databases	Operational and Decision Support	Implementation	Developers and End Users	General Purpose
Temporal Databases	Operational and Decision Support	Implementation	Developers	Specialized Applications
Active Databases	Operational	Implementation	Developers	General Purpose

The four dimensions that we use to classify these newer database technologies are: (1) Work Type, (2) Appropriate Systems Development Life Cycle Phase, (3) User Type, and (4) Specificity of Purpose

Work type is the notion that business and accounting tasks either primarily support day-to-day operations or longer-term decision support. Even though the dividing line between operations and decision support may not always be entirely clear, we make the distinction because some database tools are clearly designed for one purpose or the other. We classify all seven technologies as either operational, decision support, or both.

Appropriate systems development life cycle phase (SDLC) is binary in that it refers to only two alternatives in the SDLC: either implementation or conceptual. Some tools are designed for application implementation while others are designed to support the conceptual design phase. All of the technologies except object-

oriented databases are either implementation or conceptually oriented. Object-oriented databases not only support implementation, but also come with a conceptual design methodology.

Appropriate users are also binary in that we classify users as developers or end users. Some tools like temporal databases are clearly used by developers, while others like data mining are used primarily by end users. Some tools probably have primary users in one group, but users in the other group may also use the tool. Still, we try to classify a tool as appropriate for one group or the other.

Specificity of purpose refers to whether a tool is appropriate for a wide variety of applications or whether it is used for one specific purpose. Data mining tools are used primarily to extract information from large sets of possibly unstructured data. Data mining tools would not be appropriate for inventory management or payroll whereas an object-oriented database could be used to build a wide variety of applications. We

Table 2 Advanced Database Practices in Accounting Systems Dimensions and Technologies	
Dimensions	
1. Work Type:	(a) Operational and (b) Decision Support.
2. Appropriate Systems Development Life Cycle Phase:	(a) Conceptual and (b) Implementation
3. User Type:	(a) Developers and (b) End-Users.
4. Specificity of Purpose:	(a) General Purpose Applications and (b) Specialized Applications.
Technologies	
1. Conventional Relational Databases:	(a) Operational and Decision Support, (b) Implementation, (c) Developers, and (d) General Purpose.
2. Object-Oriented Databases:	(a) Operational and Decision Support, (b) Implementation and Conceptual, (c) Developers, and (d) General Purpose.
3. Data Mining:	(a) Decision Support, (b) Implementation, (c) End Users, and (d) Specialized Applications.
4. Data Warehouses:	(a) Decision Support, (b) Implementation, (c) Developers, and (d) Specialized Applications.
5. Intranets:	(a) Operational and Decision Support, (b) Implementation, (c) Developers and End Users, and (d) General Purpose.
6. Active Databases:	(a) Operational, (b) Implementation, (c) Developers, and (d) General Purpose.
7. Business Rules:	(a) Operational and Decision Support, (b) Conceptual, (c) Developers, and (d) General Purpose.
8. Temporal Databases:	(a) Operational and Decision Support, (b) Implementation, (c) Developers, and (d) Specialized applications.

therefore classify these technologies as either special or general purpose.

Conventional Databases

Conventional database management systems (DBMS) used in business and accounting for the past 15 years are largely passive, functioning as data repositories with appropriate tools for querying and building large-scale applications. These conventional DBMS function very well in an environment characterized by simple, short-lived transactions that have been common in business and accounting systems. Indeed, many business and accounting systems have been characterized by very large volumes of short-lived, simple transactions. For example, recording a transaction such as the receipt of payment for a given invoice would include a debit to cash and a credit to a receivable. Such a transaction is considered computationally simple

and would take nanoseconds to record. DBMS tools, therefore, have been built to manage these types of transactions much like a high volume assembly line in a manufacturing environment. Table 3 provides a summary of some salient conventional database concepts.

Although conventional database technology is currently used for many standard accounting applications such as general ledger support and valuation of inventory, its efficiency and reliability with advanced transactions leaves a great deal to be desired. For example, although conventional databases function almost solely as efficient storage and retrieval systems, no intelligence with regard to the data is built into the core database itself. This means that integrity constraints such as "we do not ship products to anyone with receivables that are aged over 60 days" be discussed later in the paper, the ability to support complex transactions using

built-in database tools is superior to simply building functions in each individual application.

Conventional databases also lack the necessary features to support the following kinds of transactions that are becoming more common: (1) Transactions involving complex data types such as graphics and multimedia, (2) Transactions involving time-oriented data (where different dimensions of time are critical), (3) Transactions where data must be available on the World Wide Web or integrated with Internet / Intranet technologies, (4) Transactions where deadlines are critical, and (5) Transactions where decision support is the main focus as opposed to operations.

Within the taxonomy listed in Tables I and II, conventional relational databases can be

classified as appropriate for operations and decision support. Relational databases are implementation and developer oriented as well as general purpose. Indeed, the fact that relational databases are so widely used demonstrates their versatility.

Advanced Database Topics

Active Databases

Although only limited work has been done in the accounting area (Olsen and Sproveri, 1996), active databases are very useful for accounting systems because of the inferential power associated with *event-condition-action* (ECA) rules otherwise known as *production* rules (Widom and Ceri, 1996). *ECA rules* (Dayal, Hanson, and Widom, 1995), which have

Table 3
Conventional Database Concepts

Database Term	Definition
Data Dictionary	The data dictionary system stores information about the database design, constraints, standards, application programs, user information, and information about the stored data.
Data Model	An underlying model to guide access, storage, and retrieval. Some examples include the network, hierarchical, relational, object, and temporal data models.
Data Repository	A standard database stores data in a logical, accessible manner.
E/R Model	A popular high-level conceptual model that is used in the design phase of database applications.
Graphical Design Tools	Entity-relationship diagrams and object-oriented modeling tools that aid in the database design phase.
Normalization	In a relational database, the normalizing process transforms a given relation, through a series of tests, into a relation guaranteed to be in a certain normal form. An objective of the normalizing process is to ensure that relations are free of the update, insert, and delete anomalies.
Query	Automated mechanisms for extracting information from a database. SQL and QBE are the two popular query methods in current DBMS's. A major concern in the query area is optimization which improves performance.
Relational Data Model	Introduced by Codd in 1970, this data model is very popular because of its single, intuitive data structure - the relation. Several commercial products are based on this data model.
Report Creation Tools	Report creation tools that require minimal programming effort.
Transaction Processing Capabilities	The ability to include related operations into a unit called a transaction. Transactions have ACID properties that stand for Atomicity, Correctness, Isolation, and Durability.
Tuple	A tuple is a relational database occurrence in a table. A tuple is similar to a record in a file-based system.

their origin in the fields of artificial intelligence and expert systems rule languages such as OPS5 (Brownston et al., 1985), have the ability to provide intelligence to an otherwise passive database. ECA rules are defined to be an event that triggers a rule and checks a condition. If the condition is satisfied, the active database executes an action. Table 4 lists the prominent active database concepts including ECA rules.

Dayal, et al. (1995) and Widom (1994) list a number of useful tasks that can be accomplished through the use of ECA rules: (a) En-

force integrity constraints for specific applications, (b) Implement triggers based on the conditions, (c) Maintain derived data, (d) Enforce access constraints, (e) Implement version control policies, and (f) Gather statistics for query optimization or database reorganization.

In contrast to a passive database system, an active database monitors pre-specified situations and, when the given conditions are met, triggers an appropriate response (Brownston et al., 1985). For example, when sales are made on account, it is axiomatic that some will be-

Table 4
Active Database Concepts

Term	Description
Production Rules	Take the form: On Event : If Condition : Then Action : They are the key to responding appropriately to given situations. The roots of production rules are in the expert systems field of artificial intelligence (Brownston, 1985).
Rule Analysis	A technique used to examine the validity of rules. For example, rule analysis can be used to identify rules that trigger each other indefinitely.
Integrity Constraint Maintenance	Production rules are often triggered in response to an integrity violation. Two possible responses include: a. Correct the violation by producing a consistent final state taking into account the violation. b. Simply rollback the transaction.
Rollback	To completely undo the effects of a transaction by an abort. A rollback may require subsequent transactions to be aborted if they conflicted with any aborted transactions.
Rule Management Tasks	Tasks that result from the presence of rules including condition checking tasks and responses to true conditions.
Forward Chaining	Data and facts are known, and after some processing, higher level truths are determined.
Backward Chaining	Higher level truths are known, and data and facts are examined to see if they support the higher level truths.
Desired Functions	a. All of the standard functions of a standard passive database system are included. b. System performance should not be negatively affected because the database system is active. c. Active databases must provide mechanisms for developers to specify active behavior. These specifications must become a permanent part of the database. d. An active database must monitor the database system and automatically respond in an appropriate manner. e. An active database must provide database design and debugging tools that are extended for active behavior.

come bad debts. It is also a well established fact that, as accounts move farther beyond their due date, the probability of collection declines progressively. Typically, an accounts receivable aging list is prepared at specified intervals after which a person in accounting reviews the list to identify specific receivables that are still unpaid after some arbitrary period, often 60 days. These overdue accounts then receive attention, generally a letter from Credit and Collections.

Alternatively, generalized rules can be implemented in an active database system that creates a *collection risk report* whenever a receivable hits the 60-day mark. The sequence of rules could send e-mail to the Credit and Collections Manager, generate correspondence to the delinquent customer, and start a review of the customer's credit history. A questionable credit history could execute yet another series of actions.

To summarize, an active database makes use of pre-specified rules that are "triggered" on a given event and the related conditions. Thus, rules can be used to automate certain tasks and also to augment the manual completion of other tasks.

Two additional advantages of active databases include the reusability of rules and specialized tools for rule creation. First, a number of commercial databases such as Sybase and Oracle include triggers that aid in rule creation. These specialized mechanisms aid the process of rule creation. Rules created in Sybase are reusable for related situations. For example, a rule may state that "a receivable cannot be eliminated unless inventory is returned or cash is received." The syntax for that rule can be reused and modified slightly for a similar rule that states "a payable cannot be eliminated unless the inventory is returned or cash is disbursed." Though the rules are semantically different, the structures are very similar and are in fact, reusable. Rule reusability is very attractive because of reduced time for creation and maintenance.

Active databases are classified in Tables I and II as appropriate mainly for operations and implementations. It is conceivable that active databases could be used in a decision support role, but they are used mainly in operations because decision support applications are often very specific and do not lend themselves to general purpose. Active databases are also developer oriented and are general purpose in nature.

Intranet Databases

Business computing is in the midst of its third major phase (Gilder, 1995). The first was the advent of the computer used for business purposes in the 1950's. The second was the widespread use of the personal computer that started in the early 1980's. This third fundamental change is marked by the growing use of Internet and Intranet technologies. Indeed, the increased reliance on networked computing is nothing short of remarkable.

Many corporations are using Internet technologies to support their internal business processes. In fact, corporate *Intranets* are in use by 22% of America's top 1,000 companies (Jones, 1996). Using Internet technology internally in an organization is the essence of an Intranet. Though there are no fundamental technological differences between the Internet and Intranets, there is certainly a difference in focus. Table 5 illustrates some of the differences between the Internet and Intranets while Table 6 lists some basic definitions related to Intranet technology.

A technical definition of an Intranet might be "... a heterogeneous computing environment connecting different hardware platforms, operating system environments, and user interfaces in order to seamlessly communicate, collaborate, transact, and innovate" (Hinrichs, 1997). An organizational definition of an Intranet might be "...a learning organization, capable of integrating people, processes, procedures, and principles to form an intellectually creative culture dedicated to implementing total organiza-

Table 5
Intranet Database Concepts

Term	Description
CGI Scripts	Common Gateway Interface - A method for making simple SQL database queries from a web page. Any application that requires high levels of performance or security will require a more powerful solution.
ODBC	Open Database Connectivity - A Microsoft-developed standard that links conventional databases such as Oracle, Sybase, or Informix to the World Wide Web. Using the ODBC drivers is thought to be more powerful than using CGI scripts.
JDBC	Java Database Connectivity - A Sun Microsystems-developed standard that links conventional databases such as Oracle, Sybase, or Informix to the World Wide Web. Using the JDBC drivers is also thought to be more powerful than using CGI scripts.
Client	A personal computer or workstation with browser software, TCP/IP connection, and other Intranet software as needed. Web browsers are specific to each platform so that a different browser is used for an Apple Macintosh, a Pentium, or a SPARCStation.
Server	There are multitudes of different types of servers that are appropriate for different applications. A standard Intranet server might consist of a high end pentium personal computer running Windows NT and the included web server.
Cookies	Small pieces of information that a CGI program can store on a user's web browser.

Table 6
Differences between the Internet and an Intranet

The Internet	Intranet
Public Information	Confidential Information
A Focus on Customers	A Focus on Employees
Financial Transactions	Collaboration Transactions
Designed for Customer Interaction	Designed for Workflow
Sales Oriented	Process Oriented
Required Connection to the Internet	Optional Connection to the Internet
Transforms the Marketing Process	Transforms Organizations
Emphasis on Single Point of Contact with the Organization and User Profiling	Emphasis on Work Groups, Teams, and Intradepartment Flow
Creates Brand Awareness and Net-citizen Status	Creates a Learning Organization
Overall Focus on Making Money and on Educating People	Overall Focus on Improving Business Processes and Reducing Development Life Cycles

(Hinrichs, 1997)

tional effectiveness" (Hinrichs, 1997).

Intranet technologies have been shown to support internal business processes in an effective manner. Calderon and Olsen (1997) report on a case where an Intranet database system was developed to support mutual funds creation. Eight different phases needed to be completed to

create a new mutual fund. An application using Intranet technology was developed to replace a manual system that was plagued by delays.

Even though the future of Intranet usage is very bright, several issues must be addressed so that the viability of Intranets remains strong. Security is a concern because traditionally, data-

base security has relied on the underlying operating system and physical environments for guarantees of security. A scheme such as conventional user identification and authorization is no longer possible because clients and servers must be able to interact without prior knowledge of each other (Jones, 1996). We cannot allow a server to collect social security numbers or bank account numbers merely to confirm the identity of a customer. Personal privacy must be maintained, both for the customer and for the requests and responses from the server. A client must also be able to authenticate servers with which it communicates in order to avoid fraud. In summary, client software must be able to deal gracefully with a wide variety of types of credentials obtained from different sources.

Another major issue is standards. Currently, the two different driver standards that can be used for connecting conventional databases to an Intranet include ODBC and JDBC (see Table 5). Standards are usually created so that competing products are able to seamlessly work together, but in this case, it seems as though the competing products were purposefully designed to be incompatible for purposes of trying to dominate the technology. Though third party products are starting to appear that bridge the differences between the two standards, true standards will have to emerge for Intranets to achieve their full potential.

Another issue concerns the basic nature of the World Wide Web (WWW). More specifically, each time a different document is accessed, a new connection is created which complicates typical database tasks. For example, once a user submits a database request, how does that user know it has been received? One solution might be a user ID and a password. However, without considerable up front planning, a user might have to enter the ID and password each time a new document is accessed which would clearly be unacceptable. Current solutions to this problem include using "cookies" and hidden tags to capture the current state of a form.

Another major issue that must be addressed is integrating legacy and warehouse database systems to the corporate Intranet. Currently, simple connections between the corporate Intranet and legacy systems can be made using CGI scripts that are usually customized C++ programming code. More complex integration tasks require ODBC or JDBC drivers. Nonetheless, easier methods of integrating data warehouse and legacy systems must be developed for the corporate Intranet to be a viable, long-term solution.

Intranet databases are classified in Tables I and II as appropriate for both operations and decision support. They are also implementation oriented and are appropriate for both developers and end users and are general purpose in nature.

Object-Oriented Databases

Object-oriented databases have not only provided a fertile area of research for academics, they are also starting to fill a void in fields where conventional database technology is deficient. Some deficiencies of conventional databases include an inability to manage complex types of data and more realistically model real world data. Areas such as computer aided design (CAD), software engineering, and advanced financial management are benefiting from the object-oriented database model because it is able to efficiently manage more complex data (Cattell, 1994).

Unlike the relational database model, however, object-oriented databases are based on neither an underlying, elegant, mathematical model, nor a well established, accepted set of concepts. Because it is a developing field, many systems, products, and research projects that are touted as "object-oriented" unfortunately fall short of expectations. This mislabeling may represent an attempt to appeal to current market trends. Additionally, it is interesting to note that many features of object-oriented databases are being incorporated into what had been tradition-

Objects are simply any real-world entities that a user finds valuable to identify. (See table 7 for some essential definitions). Objects are related to the instance concept in semantic data modeling. The three different types of objects in an object-oriented database are instance objects, class definition objects, and method objects. An instance object may be thought of as being roughly equivalent to a record in a flat file system or a tuple in a relational database system. A specific accounts receivable recorded in an object-oriented database is an example of an instance object. Obviously then, a given class might be associated with thousands of instance objects. Additionally, an instance object belongs to *one* class and is an instance of the class in which it belongs.

An object can be a member of only a single class. It is possible, however, that a class could inherit its attributes and methods from more than one super class which is known as multiple inheritance. Though multiple inheritance introduces complexity, there are many situations where it would facilitate representation of a given state as well as reduce the amount of development time.

In an object-oriented database management system (OODBMS), each object is assigned a unique ID number. The ID number is theoretically never assigned to another object even if the original object is deleted from the system. The assignment of this unique OID is advantageous because it eliminates the need for relational database table normalization as well as primary and foreign keys. These two areas are often major issues in relational database table design. In general, an object can be an instance, a class definition, or a method, and it is generally advisable to specify the object type for clarification.

The class concept is another important concept of object-oriented databases. A class collects a group of logically similar objects with a defined set of attributes and all objects that are an instance of the class have the defined structure. Each class and its subclasses are individually a class. A rigid methodology does not exist for determining what classes should be modeled in a given situation. Though some heuristics exist, modeling is usually done by an experienced designer who has the foresight to plan an acceptable system. An example of a class in an accounting system might be "accounting transactions." Two subclasses of "transactions" might

Table 7
Object-Oriented Database Concepts

Term	Description
Objects	Any real-world entity which is associated with a system-wide unique identifier. Also includes method and class definition objects.
Operations	Commands that access a database, the most basic of which are read and write operations.
Class	All objects which share the same set of attributes and methods may be grouped into a class. An object belongs to only one class as an instance of that class.
Attributes	An object has one or more attributes. The value of an attribute of an object is also an object.
Methods	An object has one or more methods which operate on the values of the attributes.
Persistence	Data remains after an application program or a user session is executed.
Inheritance	Corresponds to the semantic modeling notion of generalization and specialization. Methods and class definitions can be inherited.
Encapsulation	Provides data independence through the implementation of methods, allowing the private portion of an object to be changed without affecting programs that use that object type.
Transactions	Though transactions are not native to object-oriented databases, they are critical with regard to concurrency control. Transactions are comprised of methods.

be “revenue transactions” and “balance sheet transactions.” A subclass is a specialized case of its superclass. A subclass can inherit all of the properties of its superclasses if needed. This can be a substantial time savings in software engineering, as only the attributes and methods specific to a new subclass need be defined.

In summary, object-oriented databases are general-purpose databases that are becoming more popular for business applications. Even traditionally relational database vendors such as Oracle and IBM have incorporated some object-oriented features into their products.

Object-oriented databases are classified in Tables I and II as appropriate mainly for operations as well as decision support. The object-oriented database area includes a design methodology, so it is used in both the conceptual as well as the implementation phases of the systems development life cycle. Object-oriented databases are used mainly by developers and are used for a wide variety of purposes.

Data Warehousing

Data warehousing is a recently emerging topic in the database area that is becoming increasingly popular. In Inmon and Hackathorn, (1994), a data warehouse system is defined as “...a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management’s decision making process.” Alternatively, Gupta (1997) defines a data warehouse as a “structured extensible environment designed for the analysis of non-volatile data, logically and physically transformed from multiple source applications to align with business structure, updated and maintained for a long time period, expressed in simple business terms, and summarized for quick analysis.” Both definitions make several key points that must be understood to fully appreciate the benefits of data warehousing.

To understand data warehousing, one must contrast standard database processing known as on-line transaction processing (OLTP)

versus on-line analytical processing (OLAP)(see table 8 for a listing of essential concepts). Standard OLTP systems are used to create operational data and reports in the traditional business areas. For example, month-end profit and loss statements, inventory reports, production reports, and sales reports are often created in standard OLTP database systems.

In contrast, OLAP systems are used primarily for decision support purposes. Companies often combine internal operational data and external data with an extensive database design² to build the data warehouse. Though there are some common objectives between OLTP and OLAP systems, there are also some real differences. For example, standard OLTP systems are marked by the majority of transactions being updates. A standard accounting transaction might record a payment on an invoice. Normally, a debit to cash and a credit to the appropriate receivable is recorded. This transaction is written to the database and is considered to be an update transaction. In contrast, many OLAP systems do not allow for any updates to the warehouse once the external and internal existing data is loaded. Instead, numerous queries and statistical programs are executed against the data warehouse in order to support better decision-making. In fact, data mining (discussed in the next section) includes advanced techniques that are applied against the data in a data warehouse to improve decision making.

Size is another distinctive characteristic of OLTP and OLAP systems. OLTP systems are characterized by megabytes to gigabytes of data. In contrast, OLAP systems are characterized by gigabytes to terabytes of data. This is not surprising given that OLAP systems might include several years of operational data as well as data external to the organization. In contrast, standard OLTP systems usually include data from a relatively short time period. After the useful life of operational data, it is often archived and purged from the operational system.

Because of the enormous size of OLAP

systems, many design considerations that are standard in OLTP systems must be revised. Specifically, relational database tables are usually normalized in OLTP systems to minimize redundancy and eliminate update and delete anomalies. This normalization process usually yields a greater number of tables with fewer fields. OLAP systems, however, are characterized by a smaller number of denormalized tables for two main reasons. First, OLAP systems have very few updates during normal operations. Instead, data loading usually takes place at off times, during planned sessions. Almost all transactions in a data warehouse will be reads. Because there are very few updates, there are few benefits to normalizing. Second, joining tables to perform queries is computationally expensive. To normalize and create more tables that must be joined to support advanced queries would be misguided. Rather, database designers in an OLAP system denormalize and reduce the number of tables to enhance performance. This performance consideration is exacerbated by the extreme size of OLAP systems.

Data warehouse systems could definitely benefit accounting systems. First, good data warehouse systems support both "hard" and "soft" data (Brackett, 1996). Hard data are the historical facts and figures that the accounting function produces in abundance. Examples might include expenses, revenues, net income, inventory valuation, etc. Soft data on the other hand are explanations, comments, opinions, and evaluations about the business.

In summary, data warehouse systems are used primarily for decision support and not for operations. Numerous data warehouse applications are starting to appear in the marketplace, and the combination of accounting and data warehouses seems like a natural "marriage" because of the volumes of financial information accounting systems generate. Proponents of data warehouse systems say that the vast amounts of financial data organizations generate are not used for decision support so it would seem that this is an area with great potential.

Data warehouses are classified in Tables I and II as appropriate mainly for decision support in the implementation phase. Additionally, developers mainly build data warehouses for specialized purposes.

Data Mining

Data mining is a process that attempts to reveal predictions and hidden information from a collection of data (Red Brick, 1996). The purpose of data mining is to use the data that an organization accumulates to help the organization grow and become more effective. Data mining achieves this objective by finding relationships between the data that are not apparent without a great deal of analysis. Furthermore, data mining often uncovers relationships between the data that were never known to exist.

Data Mining can be thought of as a three step process (Bigus, 1996). First, data is gathered and prepared. Second, the data mining algorithm processes, compresses, transforms, and identifies valuable data (Bigus, 1996). Third, the user performs data analysis.

In the first step of data mining, gathering and preparation of data, data must be cleansed (Bigus, 1996). Cleansing data ensures that complete and accurate data is entered into the database (see Table 9 for essential concepts). This process is very important because if the data entered is incomplete or inaccurate, the data mining effort will likely yield questionable results.

The optimal place to store data that will be mined is in a data warehouse (Red Brick, 1996). The data warehouse is normally a complete set of data containing data that are not modified after entry. Use of a data warehouse often helps achieve proper data cleansing because the data should be complete and accurate when entered.

The second step of data mining, the data mining algorithm, involves several issues. First,

Table 8
Data Warehouse Concepts

Term	Description
OLTP	On-line transaction processing. Traditional database processing that supports operational systems. Most standard business systems are supported by OLTP.
OLAP	On-line analytical processing. OLAP refers to an advanced data analysis system that supports decision making and business modeling. OLAP tools are usually used in conjunction with the data warehouse.
MOLAP	Multidimensional on-line analytical processing. OLAP tools that specifically support multidimensional data analysis. Multidimensional data refers to the fact that business data is often related in many different ways, some obvious and some not.
ROLAP	Relational on-line analytical processing.
Data Warehouse	"A data warehouse is a structured extensible environment designed for the analysis of non-volatile data, logically and physically transformed from multiple source applications to align with business structure, updated and maintained for a long time period, expressed in simple business terms, and summarized for quick analysis" (Gupta,1997).
Data Mart	"A subset of the data resource, usually oriented to a specific purpose or major data subject, that may be distributed to support business needs. The concept of a data mart can apply to any data whether they are operational data, evaluational data, spatial data, or meta data" (Brackett, 1996).
Decision Support	"A methodology (or a series of methodologies) designed to extract information from data and to use such information as a basis for decision making" (Rob and Coronel, 1997).
Decision Support System	"An arrangement of computerized tools used to assist managerial decision making within a business. A DSS usually requires extensive data 'massaging' to produce information" (Rob and Coronel, 1997).
Operational Data	"Data used in the operational processing of business transactions that support day-to-day business operations. They are detailed, largely primitive data necessary to keep the organization operating" (Brackett, 1996).

Table 9
Data Mining Concepts

Term	Definition
Associations	Information generated by data mining where two or more events are linked together to reveal affinities or patterns in the data set.
Classification	Information generated by data mining that states if one side is true, then the other side is true.
Cluster	Information generated by data mining where subsets are established and descriptions of the subsets are defined to reveal associations among data points.
Data Cleansing	Assuring that data is accurate, concise, and complete before entering it into the database (often a data warehouse) that will be mined.
Data Mining	Uncovering predictions and hidden information from a large set of data.
Data Mining Algorithm	The second step in the data mining process where the algorithm processes, compresses, transforms, and identifies data.
Data Visualization	An approach to data mining where the user receives a picture as output.
Decision Tree	An approach to data mining where values and variables of the data define how the data is divided into groups.
Forecast	A type of information generated by data mining.
Model	The model is a mathematical formula that is applied to the data that explains the effect inputs have on outputs.
Neural Net	An approach used in data mining.
Rule Induction	An approach used in data mining.
Sequences	Information generated by data mining where events are related by time.

a model of a current situation is created to apply against the data (Pilot Software, 1997). This application results in a prediction of a likely outcome. IBM identifies two types of models: verification and discovery models (Bigus, 1996). A verification model verifies or negates a hypothesis; here, no new data is generated. A discovery model discovers hidden information in the data.

In the data mining algorithm step, there are four types of approaches used. Neural networks can be used. Neural networks are collections of connected nodes. There are inputs and outputs at each node, as well as processing (Edelstein, 1996). In addition, there is a hidden layer that processes inputs (Edelstein, 1996). Decision trees can also be used. In a decision tree, values and variables of the data are used to group the data (Edelstein, 1996). Rule induction is also a tool. In rule induction, a non-hierarchical set of conditions is created (Edelstein, 1996). Finally, data visualization is a useful tool. Data visualization allows the user to view a picture representation of certain data (Edelstein, 1996).

Finally, in the data algorithm step, there are five general types of information generated by the tools discussed above. Classifications of data can be generated (Queen's University, 1997). Associations between data can be found. When associations are discovered, affinities or patterns in a given set of data are revealed (Queen's University, 1997). Clusters of data can be identified. To identify clusters, subsets of data are established and descriptions of the subsets are defined (Queen's University, 1997). The data algorithm can also produce forecasts. Finally, Sequences, which use time to relate events, can also be established; this is also called time series analysis (Red Brick, 1996).

There are many areas in accounting where data mining can be very useful. Some examples of applicable areas are: aging accounts receivable, cost accounting and manufacturing (to predict costs and schedule jobs), auditing and fraud detection, accounting information given to

support the marketing department, sales forecasting, and inventory management.

An example of data mining, which is applicable to virtually all businesses, is aging accounts receivable. Most companies have systems of aging accounts receivable which identify receivables that are late, how late the receivables are, and who the customer is. Data mining can take this information further.

Imagine a company that can predict which customers are likely to be late by forty-five days, but are never late by sixty days. Or imagine being able to associate certain products or times of year with late payments. Also imagine a company that can predict when a previously reliable customer is experiencing financial problems and is likely to become an overdue payer.

For example, Company X (seller) and Company Y (buyer) have been doing business for ten years. Company Y always pays within twenty days. In the eleventh year of business, company Y starts to pay within twenty-five days, then thirty days, and so on, until the thirteenth year when company Y's receivables are in the sixty day file. In many accounts receivables aging systems, no attention would be given to company Y until it reached the forty-five day file. By employing data mining, company X could have detected company Y's problems when company Y strayed from its usual practice of paying within twenty days. Company X could then have monitored Company Y's receivables, using its observations to make better business decisions and possibly prevent uncollectible receivables.

Being able to identify these facts, predictions, and associations can greatly reduce resources spent on receivables collection. These facts, predictions, and associations can also give a company the information it needs to take steps to prevent future uncollectible receivables.

Data mining has many potential ac-

counting applications. By using the vast amount of accumulated accounting data, data mining can help in making predictions and associations, and revealing useful facts. A company can use the information produced in data mining to make more cost effective decisions, create optimal plans, market products more effectively, and identify potential problems that may need to be monitored.

Data mining is classified in Tables I and II as definitely a decision support tool in the implementation phase. End users use data mining tools to learn the story the data is telling for specialized purposes.

Business Rules

In many business organizations, business managers and data administrators struggle to understand each other's areas. Data modeling and the conceptual design process are supposed to help these two groups communicate more effectively. Additionally, data administrators are often not aware of business processes and the critical aspects of the business.

The concept of the "business rule" was identified to represent a communication methodology that can bring the goals of the business managers and the data administrators together (Von Halle, 1997). A business rule is defined as "a statement that defines or constrains some aspect of business" (Von Halle, 1997). It is fur-

ther stated that "business rules must relate in obvious ways to important aspects of the business, specifically, business knowledge and behavior" (Von Halle, 1997). Business rules allow data administrators to translate the identified needs of the business managers into a form that can be modeled and later implemented into the system.

There are four categories of business rules: definitions, facts, constraints, and derivations (Von Halle, 1997). Definitions are used to define different aspects of the business (Von Halle, 1997). For example, a definition could be "the accounting department is where all of the accounting is done." A fact is a property or role of an object or a relationship that an object has with another object (Von Halle, 1994). An example of a fact would be "accounts receivable clerk enters receivables." A constraint is a "condition about the data that must be true" (Von Halle, 1997). For example, a constraint could be "an overdue receivable must have been outstanding for forty-five days." Finally, a derivation is a business rule that is derived from existing information (Von Halle, 1997). An example of a derivation would be "the amount of a discount is the invoice amount multiplied by the discount percentage."

Identifying business rules begins with natural language statements made by business management (members of the business not involved with data administration). These statements are termed "business ramblings" (Von

Table 10
Business Rules Concepts

Business Rule Term	Definition
Atomic Units	An atomic unit is a statement that cannot be broken down without losing its meaning.
Business Ramblings	Business ramblings are natural language statements made by business management that may contain business rules.
Business Rule	A business rule is "a statement that defines or constrains some aspect of business" (Von Halle, 1994).
Constraint	A constraint is a "condition about the data that must be true" (Von Halle, 1994).
Definition	Definitions describe different aspects of the business.
Derivation	A derivation is a business rule that is distilled from existing information.
Fact	A fact is a property or role of an object or a relationship that an object has with another object.

Halle, 1994). Business ramblings can be very structured and easy to understand, or they can be a jumble of facts and ideas. The duty of the data administrator is to record the business ramblings so that they can be used to identify business rules. The recorded business ramblings are broken down into "atomic units" (Von Halle, 1994). Atomic units are then used to create a model to eventually implement the business rules into the system.

By identifying a methodology that data administrators can follow to capture business rules, a line of communication has been opened between business managers and data administrators. This occurrence is very important, as businesses rely more and more on information technology to assist in data retrieval, decision making, and business projections. For an information system to assist in a business's success it is imperative that the data administrators understand the goals and requirements of the business. This understanding will give the data administrators the ability to design the optimal system for their company.

Accounting is an area in every business that requires a huge amount of information support. Accounting is also an area in every business that can be distinct from other businesses, necessitating customized information support. This is very apparent in cost accounting systems for manufacturers.

Most production systems are different, setting each manufacturer's needs apart from other manufacturers. These different needs can be viewed at two levels. First, there are differences in materials, production, and labor. Second, there are the differences in what business managers expect from production.

The second need is where business rules are of particular importance. What managers expect and require is what really defines the business. Data administrators can understand the need to calculate materials, labor hours, and production time, but they might not understand

the need to design a system that provides more.

For example, Company S manufactures ice cream makers. Company S has a cost accounting system that calculates labor time, materials, and production time for each ice cream maker. Company S's management receives a report each month identifying these figures. Recently, management has noticed that material costs have gone up and it seems to be taking longer to produce each ice cream maker. Management is frustrated -- management wants to find out what is causing the problem, but its information system is not capable of producing such extensive data.

When the data administrators implemented the system, they were unaware that management wanted to be alerted when it took more than five minutes to manufacture an ice cream maker. Data administration was also unaware that more than ten wasted material parts per day was unacceptable and was cause for alarming management. Further, data administration did not know that production time had to be greater than three minutes per ice cream maker to produce a safe product.

The above are examples of business rules. These rules make Company S unique. The integration of these rules into the information system is necessary for the creation of an optimal information system for Company S. Company S's data administrators will have to listen to management's needs, break their ramblings down to atomic units, model the business rules, and finally, implement the business rules. Thereafter, management will be provided with the information it needs to run its business as efficiently and as effectively as possible.

Business rules are a way of linking management and data administrators. Communication between management and data administrators is necessary for the creation of a supportive information system. Accounting information systems require much customized data. By data administrators identifying and implementing

business rules, accountants will be provided with more useful data.

Business rules are classified in Tables I and II as appropriate for both operations and decision support because they are a conceptual design tool. Developers mainly use business rules, although they must communicate with end users as a part of the process. Business rules are used for a wide variety of purposes.

Temporal Databases

According to *A Consensus Glossary of Temporal Database Concepts*, "A temporal database supports some aspect of time, not counting user-defined time" (Jensen et al., 1992). A temporal database relates data according to the time assigned to the data. Temporal databases give the user the ability to recapture the status of data as it existed at a requested time. Further, temporal databases are very attractive because they allow the user to go back in time and recreate the circumstances that existed at a particular time.

When data is entered into a temporal database it is timestamped. This means that the

data is assigned a time (Jensen et al., 1992). There are different classifications of time that can be assigned to data. Valid-time is the time when the data event actually happened or was actually true (Jensen et al., 1992). Transaction-time is the time when the data was entered into the database (Jensen et al., 1992). User-defined-time represents a date or time. User-defined-time is uninterpreted, as it represents something like a birthday that the user defines and is always true (Jensen). Some temporal databases support only one kind of time. A bitemporal database supports both valid-time and transaction-time (Jensen et al., 1992).

A temporal database contains relations between data that are based on time. In a temporal database, each tuple (roughly a record) is matched with its attributes (columns). Tuples must be homogeneous, meaning that every attribute of the tuple must have the same time attached to it (Jensen et al., 1992). Once data is entered into the database and timestamped it cannot change. This ensures that the integrity of the data will not be compromised (Jensen et al., 1992).

Databases that are not structured to sup-

Table 11
Temporal Database Concepts

Temporal Database Term	Definition
Attribute	A column in a temporal database.
Bitemporal Database	A database that supports both valid-time and transaction-time.
SQL	The general language used in temporal databases.
Temporal Database	A database that supports some aspect of time, not counting user-defined time.
Timestamp	Attaching a time to data when it is entered into a temporal database.
Transaction Time	The time when the fact is stored in the database. A transaction time is fixed and cannot be modified once the time is recorded because it is impossible to change the past.
TSQL	One of the two most common query languages used for temporal databases.
TQUEL	One of the two most common query languages used for temporal databases.
Tuple	A row in a temporal database.
User Defined Time	A combination of the date and time domains. There is no special query language support for user-defined time.
Valid Time	The time when a fact is true. A fact may be associated with any number of events and intervals or with just a simple event and interval.

port temporal theories do not reflect the history of data (Jensen et al., 1992). These databases are static, meaning that when queries are run, the output represents data at some point in time (Jensen et al., 1992). In a temporal database, temporal constraints are used to restrict the data that are queried to "how the data changes in time" (Abiteboul, 1995). When the queries are run, the output reflects the history of data as it existed at specific points in time.

To function as a temporal database, queries must be developed to extract the needed information. SQL is the general language used in temporal databases. However, in the temporal database field several languages have developed from SQL to support temporal queries. The two most common languages used for temporal queries are TQUEL and TSQL. Although this is a developing field, the development seems to be producing languages and modifications of languages that are not compatible with one another. Currently researchers are working to advance temporal databases and to make the languages more uniform.

The business world could benefit greatly from the use of temporal databases. With the use of temporal databases, a company could go back in time and recreate the circumstances that existed at a specific point in time. This could give the company the ability to detect problems that existed in the past, detect patterns, and predict future events.

There are a variety of accounting applications that could be enhanced by the use of temporal databases. Some of the possible accounting applications include auditing and fraud detection, inventory estimation with FIFO and LIFO, accounts receivables, and cost accounting. With the use of temporal information and queries, an accountant can know exactly what data looked like at a specific time in history or at a specific time in data entry. Further, with the use of a bitemporal database, an accountant could view data at its transaction time and its valid time to detect any differences.

One area in which temporal databases could greatly assist accountants is in auditing. With the use of a bitemporal database, a query can be structured to compare the valid-times and transaction-times of specific data. The result could identify a number of transactions, which have suspiciously different valid-times and transaction times.

In a regular audit, the accountant would statistically devise a sample of transactions to examine. The temporal database, on the other hand, can examine every transaction for time-related errors. Imagine being able to examine every record, whether there are hundreds, thousands, or even millions of records. Examination of every record could conceivably uncover every suspicious transaction. An audit sample may not include these transactions, and possible fraud may go undetected.

For example, imagine company C and its employee, Max. Company C does all of its selling on account. Therefore, Company C has many accounts receivables. Max has been at company C as its accounts receivables clerk for twenty years. Company C has been satisfied with Max's performance.

Every so often Max is short on money. Over the years Max has devised a scheme that allows him to cash a check that is received by the company and later repay the amount that he "borrowed."

Every year company C is audited. Because company C has a large amount of receivables only a sample is examined during audit. Luckily for Max none of the receivables in his scheme have ever been part of the sample. Company C has noticed that every so often a reliable customer will pay very late, but the company just assumes that the customers are experiencing short-term cash-flow problems. Company C couldn't imagine that its trusted employee, Max, could ever steal.

With the use of a bitemporal database,

suspicion could have been sparked when Max's scheme began. A bitemporal database would record when the company actually received the receivable check. This would be the valid-time. A bitemporal database would also record when the receivable was actually posted in the system. This would be the transaction-time. By running a query on all receivables to detect a certain disparity between valid and transaction times, Max's transactions would have been detected from the beginning.

The temporal database is emerging technology that should not go unnoticed. Temporal databases can assist the business community in recreating the status of data at a specific time. Querying relationships between data and time may uncover fraud, detect patterns, and help generate predictions. There are also possible applications that temporal databases have to accounting, including the areas of auditing, cost accounting, inventory, and accounts receivables.

Temporal databases are classified in Tables I and II as appropriate for both operations and decision support in the implementation phase. Additionally, developers are almost exclusively the persons that interact with temporal databases. As has been demonstrated in the examples, temporal databases are used for specialized purposes though it is conceivable that a general purpose temporal database could be built.

Conclusion

The purpose of this paper was to provide the reader with an overview of seven recently developed database technologies in the context of a proposed taxonomy along four specified dimensions. The seven technologies included in this paper are not exhaustive of this area. They were chosen based upon their applicability to accounting practice, and the likelihood that readers do not have an extensive knowledge of these applications due to their recent development.

Implications for Future Research

Since the technologies discussed in this paper are fairly new, future research efforts should be directed toward investigating the extent to and the ways in which they have been integrated into accounting practice. Along those lines, it would be interesting to know whether or not the use of these various database technologies is divided by industry, type of business (manufacturing, retailing, etc.), firm size, and various other dimensions. This paper suggested accounting applications for some advanced database technologies, but was, by no means, exhaustive. A more in-depth exploration of potential accounting applications for the specific technologies would also be of interest. □

Endnotes

- 1 ODBC and JDBC drivers are used for more complex connections, or larger systems. Simple connections can be made using a standard HTML form with CGI scripts.
- 2 Many database texts refer to the database design as a schema.

References

1. Bigus, Joseph P., *Data Mining With Neural Networks*, McGraw-Hill, New York, New York, 1996.
2. Brackett, M.H., *The Data Warehouse Challenge*. John Wiley & Sons, New York, New York, 1996.
3. Brownston, L., R. Farrell, E. Kant, and N. Martin, *Programming Expert Systems in OPSS: An Introduction to Rule-Based Programming*. Addison-Wesley, Reading, Mass., 1985.
4. Burke, S., "Net emerging as next 'mass media'," *Computer Reseller News*, December 16, pp. 39-40, 1996.
5. Calderon, T., and D. Olsen, "Leveraging Internet and Database Technologies to Enhance Internal Business Processes," *Review of Accounting Information Sys-*

- tems, Fall, pp. 25-39, 1997.
6. Cattell, R.G.G., *Object Data Management*, Addison Wesley, Reading, Mass., 1994.
7. Chen, P. P., *The Entity-Relationship Model: Toward a Unified View of Data*, ACM Transactions on Database Systems, January, pp. 9-36.
8. Dayal, U., E. Hanson, and J. Widom, "Active Database Systems," in *Modern Database Systems*, ed. Won Kim, Addison Wesley Publishing Co., ACM Press, New York, New York, pp. 434-456, 1995.
9. Edelstein, Herb, *Technology How To: Mining Data Warehouses*. TechWeb, CMP Media Inc., January 1996.
10. Gilder, G., "The Coming Software Shift," *Forbes*, ASAP, August, pp. 147-162, 1995.
11. Gupta, V. R., *An Introduction to Data Warehousing*, White Paper, System Services Corporation, Chicago, Illinois, August, 1997.
12. Hay, D., and K. A. Healy, *Guide Business Rules Project*, White Paper, Guide International, <http://www.guide.org/ap/apbrules.htm>, October, 1997.
13. Hinrichs, R. J., *Intranets: What's the Bottom Line*. Prentice-Hall, New Jersey, 1997.
14. Holsheimer, M., M. Kersten, H. Mannila, and H. Toivonen, *A Perspective on Databases and Data Mining*. First International Conference on Knowledge Discovery and Data Mining, Montreal, Canada, 1995.
15. Inmon, W.H., and R. D. Hackathorn, *Using the Data Warehouse*. John Wiley & Sons, New York, New York, 1994.
16. Jensen, C.S., J. Clifford, S. K. Gadia, A. Segev, and R. T. Snodgras, *A Glossary of Temporal Database Concepts*. SIGMOD Record 21(3), pp. 35-43, 1992.
17. Jones, V. E., *Dissertation Proposal*, Department of Computer Science, University of Illinois, 1996.
18. Olsen, D. and D. L. Sproveri, "An Active Database Approach for Supporting Accounting Information Systems," American Accounting Association National Meeting, Chicago, Illinois, 1996.
19. Pilot Software, *An Introduction To Data Mining: Discovering Hidden Value In Your Data Warehouse*, White Paper, 1997.
20. Queen's University of Belfast, *Data Mining*, Technical Report, 1997.
21. Queen's University of Belfast, *Data Mining Functions*, Technical Report #2, 1997.
22. Red Brick Systems, Inc., *Data Mining: The Power Of Integration*, Red Brick Systems, Inc., White Paper, October, 1996.
23. Rob, P., and C. Coronel, *Database Systems: Design, Implementation, and Management*, International Thompson Publishing Company, Cambridge, Massachusetts, 1997.
24. Von Halle, B., "Lessons to learn from Tee-ball," *Database Programming & Design*, December, 1994.
25. Von Halle, B., "Back to Business Rule Basics," *Database Programming & Design*, November, 1994.
26. Von Halle, B., "Explaining the Business Connection," *Database Programming & Design*, December, 1996.
27. Von Halle, B., "Introducing Natural-Language Modeling," *Database Programming & Design*, September, 1997.
28. Von Halle, B., "Living By The Rules," *Database Programming & Design*, November 1994.
29. Von Halle, B., "The Art Of Letting Go," *Database Programming & Design*, February 1997.
30. Widom, J., *Research Issues in Active Database Systems: Report from the Closing Panel at Ride-Ads*, in ACM SIGMOD Record, 23(3), September, pp. 41-43, 1994.
31. Widom, J, and S. Ceri, editors, *Active Database Systems: Triggers and Rules for Advanced Database Processing*, Morgan Kaufmann Publishers, Inc., San Francisco, California, 1996.
32. Zaki, M. J., M. Ogihara, S. Parthasara-

thy, and W. Li, *Parallel Data Mining for Association Rules on Shared-memory Multi-processors*, The University of Rochester, Computer Science Department, Technical Report 618, May, 1996.