Chapter 10 Database accounting systems

Guido Geerts and William E. McCarthy

Almost fifteen years ago, McCrae (1976) spoke of the difference between systems and technology in making a case for the development of newer and more sophisticated accounting models:

> We must be careful to differentiate between accounting technology and accounting systems.
>
> The technology of accounting is concerned with the physical artifacts which are employed to process accounting data. These artifacts range all the way from quill pens to remote controlled computers. Accounting systems are concerned with the classifying and structuring of accounting data . . . It is possible to change the accounting technology without changing the accounting system and vice versa. This is illustrated by the . . . example set out later in this chapter. An unchanged system is processed on a changed technology.
>
> The clear distinction between system and technology is important since many accountants suffer from the delusion that because they have changed the accounting technology they must automatically have effected dramatic changes in the accounting system. This is not so. The new computer technology provides a dramatic improvement in the speed of data processing and automatic control but this potential cannot be realized without affecting major alterations in the accounting system. In other words the accountant must develop more sophisticated accounting models to benefit from computer technology (p. 39).

Specially commissioned for *IT and Accounting: The impact of information technology*.
Edited by Bernard C. Williams and Barry J. Spaul.
Published in 1991 by Chapman & Hall, London ISBN 0 412 39210 0.

The major alterations called for by McCrae have, by and large, not yet materialized in the modern EDP environment. Current systems (mainframe accounting software as well as PC accounting packages) consist simply of computerized versions of journalizing and posting routines which use double-entry principles to define procedures and the chart of accounts to classify meaning of economic events.

Most of the research aimed at better compatibility between accounting systems and information technology has concentrated on integrating 'events' accounting theories (Sorter, 1969) with *database* approaches to information management, approaches that assume that an enterprise chooses to manage its data as a centrally-controlled resource to be shared among a wide range of users with highly diverse needs. Accounting systems built with this type of orientation have included hierarchical models, network models and relational models (McCarthy, 1981).

A more generalized approach to the task of constructing events accounting systems in a database environment was proposed by McCarthy (1979) who built an accounting system using a conceptual schema – expressed in terms of Chen's (1976) entity relationship model – that could be mapped into any of the more specific approaches mentioned above. In 1982, McCarthy extended this approach by using data abstraction to develop a generalized semantic representation of accounting phenomena: the REA framework (McCarthy, 1982).

This chapter explains database accounting – as defined by McCarthy (1979, 1982) and later extended with implementation work (McCarthy, 1990) – and contrasts the use of such alternative accounting models with conventional book-keeping-oriented systems.

AN ALTERNATIVE VIEW OF ACCOUNTING REPRESENTATION

A chart of accounts and its accompanying double-entry procedures might be viewed simply as a scheme for organizing, classifying and aggregating financial data. Additionally, however, it represents the imposition upon an accountant of a particular mode of thinking about the economic affairs of an entity. For example, when queried about the 'things' that accounting deals with, accountants might list items such as 'prepaid revenues', 'retained earnings', or 'liabilties' because these, among others, constitute the elements in their predefined world of interest. Collectively, these account names and double-entry procedures represent a data model of an enterprise's economic aspects.

This predisposition toward certain types of 'book-keeping things' of interest will be discarded here. Instead this chapter will view an object financial system without 'traditional-accounting-coloured' glasses and use a *semantic model* for the creation of an accounting information system. Use of such a semantic framework assumes that an accounting system designer is trying to capture explicitly the object structure of the corporation being modelled. This object structure is normally reflected in a 'conceptual schema' of the targeted enterprise, and the purpose and architecture of these conceptual schemas is examined next.

CONCEPTUAL SCHEMAS OF BUSINESS ENTERPRISES

Conceptual schemas have claimed an increasingly important role in the development of information systems. But what are conceptual schemas and what is their function in the development of integrated databases for business enterprises? Consider first the classical ANSI/SPARC three-level architecture illustrated in Fig. 10.1 (Date, 1986, Chapter 2).

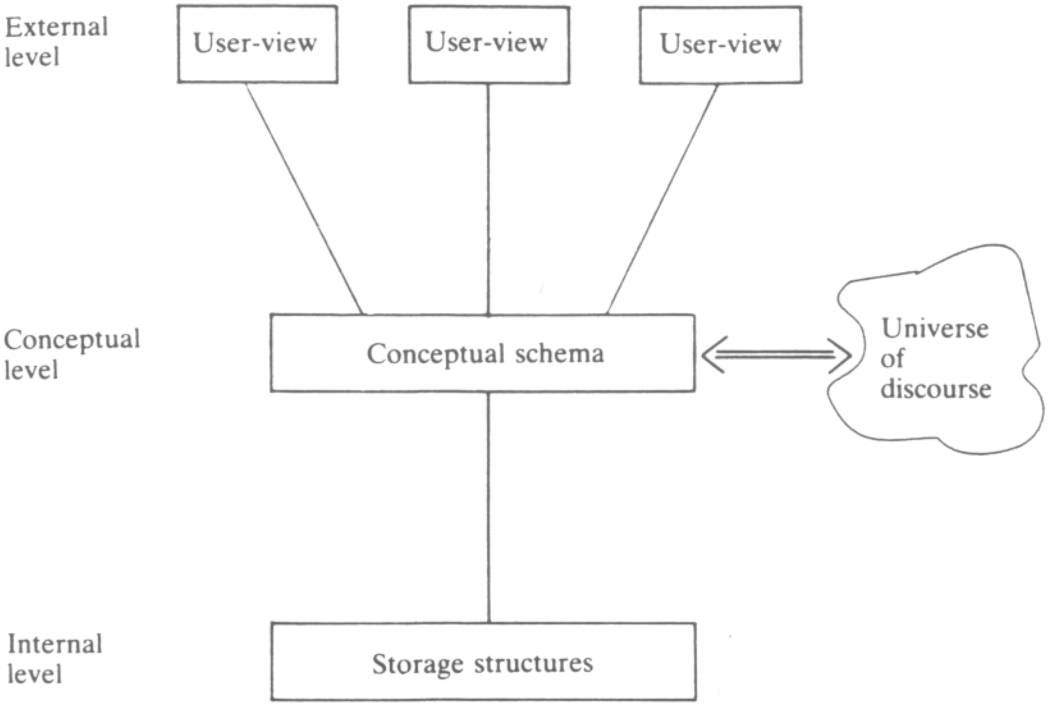


Fig. 10.1 ANSI/SPARC three-level architecture. (*Source*: Date, 1986, Chapter 2.)

While the external level presents the individual data views of various enterprise decision-makers, the conceptual level groups these views to a common enterprise-wide perspective called the conceptual schema. This global view has the explicit goal of modelling enterprise facts in a consistent and non-redundant fashion. The internal level corresponds with the more technical aspects of data storage such as the lengths of records and pointers. While the conceptual and external levels need to capture the problem-oriented aspects of the Universe of Discourse (UoD) – the part of reality being represented in the information system – the internal or physical level models the efficiency-oriented structures of computer storage (Falkenberg, 1981).

The conceptual schema is actually a model of the accounting UoD. Objects of interest – such as transactions, people, and financial resources which exist independently in the application world – should be represented directly in the schema. Identifying and structuring these representations correctly is the most important activity in the development of accounting information systems. Once the conceptual schema has been specified, designation of individual views for decision-makers and computer-oriented views for implementation can follow.

Different strategies exist for the specification of the conceptual schema. In the top-down approach, objects in the accounting world are identified first and modelled directly, while the bottom-up approach concentrates first on individual data needs and then integrates those views into a global perspective. Realistic design methodologies actually use these two approaches together (McCarthy, Rockwell and Armitage, 1989). For the sake of simplicity in the rest of this chapter, however, we will use only the more direct top-down approach. Our primary task in accounting system design then becomes one of identifying and modelling accounting objects directly. Toward this purpose, we will avail ourselves of two bodies of theory:

1. The NIAM model which provides a well-accepted framework of semantics and notational constructs for an accounting conceptual schema.
2. The REA accounting model which provides a semantically-oriented interpretation of the requisite objects in an accounting Universe of Discourse.

Each of these theories is addressed in a section that follows.

INGREDIENTS OF A NIAM CONCEPTUAL SCHEMA

NIAM stands for Nijssen's Information Analyses Methodology and can be considered as an integrated methodology for the development of an information system. A full treatment of NIAM is beyond the scope of this chapter, but interested readers may consult Verheijen and van Bekkum (1982), Wintraecken (1986) or Nijssen (1989). We will limit ourselves to an explication of NIAM's basic ideas. Before that explanation commences, however, we would like to stress some important overall characteristics of Nijssen's method:

1. NIAM provides a natural language interface for system analysis and design. Well-formed sentences are used during requirements analysis to aid communication between end-users and analysts. This interface is one of the keystones of the fact-based approach to system design (Kent, 1986; Leung and Nijssen, 1988).
2. A rich set of graphical constructs for schema specification is a key component of the methodology.
3. An algorithm is provided for the translation of a NIAM information structure diagram (ISD) to a fully normalized and optimal relational database specification. Fully normalized means that all relations are at least in 5th Normal Form (Date, 1986) while optimal means that the number of relations in the generated schema is minimal (Nijssen, 1989).

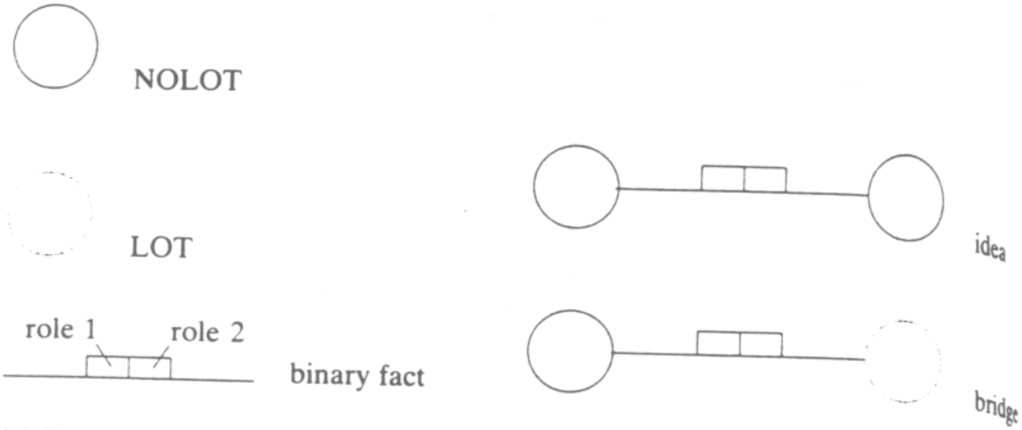
These characteristics provide most of the rationale for NIAM's widespread use in database design, especially in Europe and Australia.

BASIC CONCEPTS

The symbols in Fig. 10.2 cover the major aspects of a conceptual scheme in NIAM. A NOLOT expresses a set of non-printable objects with similar characteristics. Non-lexical objects refer to individual things in the UoD that are not utterable. A LOT by contrast contains a set of utterable objects. These can be used to describe and identify non-lexical objects. A binary fact is an association between two NOLOTs or between a LOT and a NOLOT. An association between two LOTS is never allowed.

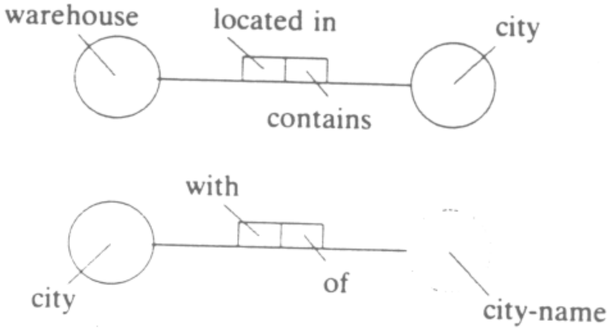
The distinction between LOTS and NOLOTS leads to the two different kinds of binary association shown in Fig. 10.2(b): ideas and bridges. A bridge-type represents a set of similar binary associations between a NOLOT and a LOT. This association can be considered as a bridge between two different worlds. An idea-type connects two NOLOTS, and it is best considered as an abstract association.

Figure 10.2(c) illustrates two example associations. The first expresses the idea that warehouses are located in cities (or conversely that cities contain warehouses). The second association illustrates a bridge between the real world (where cities exist) and the representation world (where those cities are referred to by their names).



(a) Basic elements

(b) Binary associations



(c) Example

Fig. 10.2 NIAM constructs.

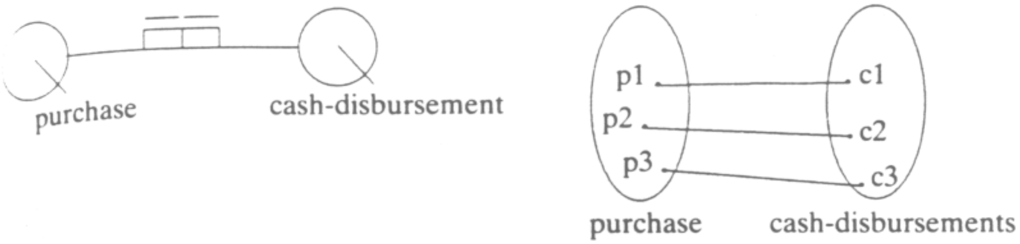
Consistent use of these concepts keeps clear the distinction between reality and a representation of reality. We consider this an important distinction that some data models fail to enforce.

CONSTRAINTS

Constraints specify the allowable ways in which data model constructs can be used to represent the real world. They are methods that enforce the semantic integrity of a conceptual schema. For our purposes in this chapter, consideration of a subset of available NIAM constraints suffices. We will treat successively cardinality and totality constraints.

Cardinality Constraints

A cardinality constraint expresses a restriction with respect to role populations. We limit ourselves first to the three examples presented in Fig. 10.3. Each example is followed by an equivalent set representation.



(a) One-to-one

chen:
many-to-one

(b) One-to-many

(c) Many-to-many

Fig. 10.3 Cardinality constraints.

Fig. 10.3(a) illustrates a *one-to-one* correspondence between two NOLOTs. Suppose for example a business enterprise has an operational rule that all purchases are to be paid for, in full, exactly five days after a receipt. Further, a payment corresponds with exactly one purchase. Each purchase event would correspond then to only one cash disbursement event and vice versa.

A *one-to-many* relationship specifies a correspondence between just a single occurrence in one role population and many occurrences in another role population. Suppose a situation where an enterprise bills its customers once a month for all sales during the preceding month and where customers must pay in full shortly thereafter. In this case, each cash receipt event would correspond to multiple sale events as illustrated in Fig. 10.3(b).

Finally, a *many-to-many* relationship specifies not only a possible correspondence between one occurrence of the first object-type and many occurrences of a second object-type, but also a possible correspondence between one occurrence in the second object-type and many occurrences in the first object-type. To illustrate in the case of a relationship between the NOLOT's sale and inventory as shown in Fig. 10.3(c), suppose not only that each sale consists of many products, but also that each product participates in many sales.

Additionally, we can illustrate the use of a special kind of cardinality designation called the **uniqueness constraint** (portrayed with a capital 'U'). Such a constraint asserts that a combination of role occurrences uniquely identifies a certain non-lexical object. This interrelation constraint can best be explained by means of the purchase-line example of Fig. 10.4.

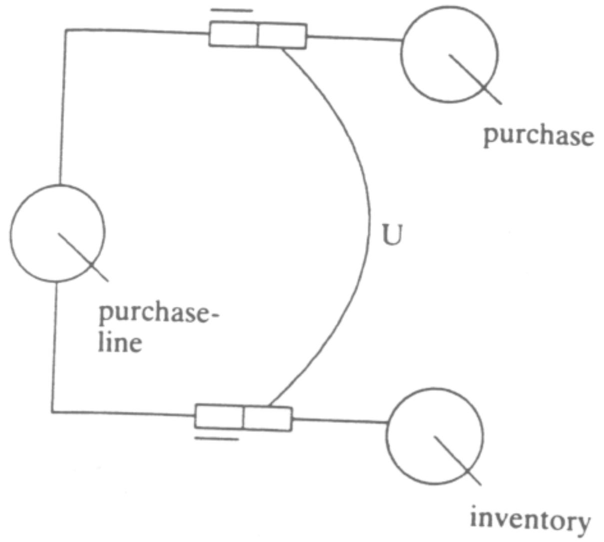
The ISD of Fig. 10.4(a) indicates that the occurrence of a particular purchase of a certain type of inventory uniquely identifies an instance of a purchase line. The meaning of this construct can be elucidated further with the 'population diagram' of Fig. 10.4(b) which shows that the combination of the elements in role 1 (r1) and role 4 (r4) are unique. Such combinations can be used to identify occurrences of the object purchase line.

The Totality Role Constraint

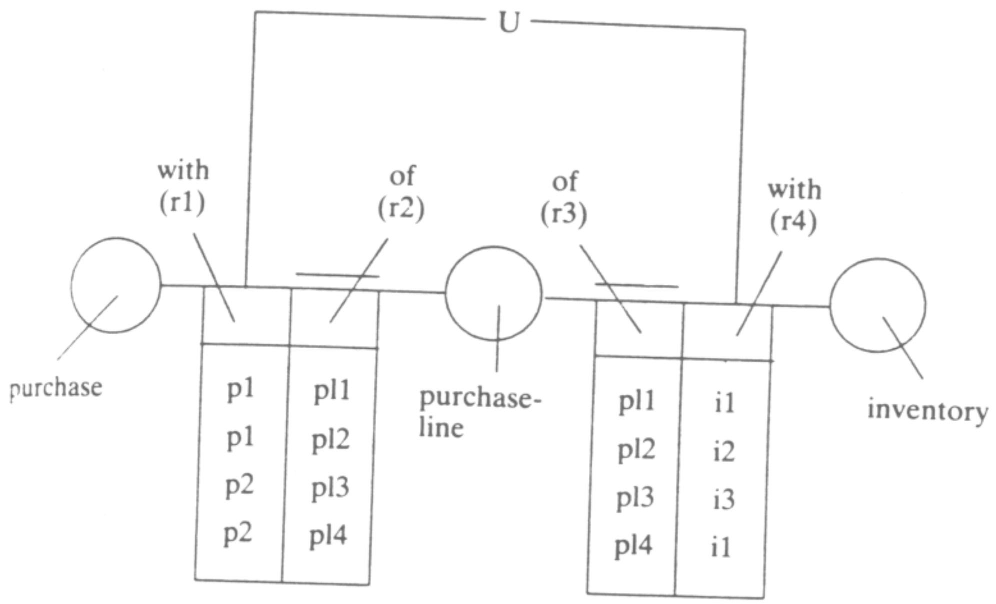
A **totality role constraint** requires that every instance of an object-type participate in a certain role. This construct is designated by the arrowhead symbol (>) drawn perpendicular through the line. The totality role constraint is illustrated in Fig. 10.5 for the association of two lexical object-types with vendors. The first of these LOTs is required, while the second is not. From a practical viewpoint, these diagrams express the following integrity maintenance rule:

> If information concerning new vendors is added to the database, their names 'must' be added while their phone-numbers 'may' be omitted.

NIAM includes other types of integrity constraints dealing with semantic concerns of equality subsetting and exclusion (Nijssen, 1989). However, for our purposes here, the simple set we have explained thus far will suffice.



(a) Information structure diagram



(b) Population diagram

Fig. 10.4 Uniqueness constraint.

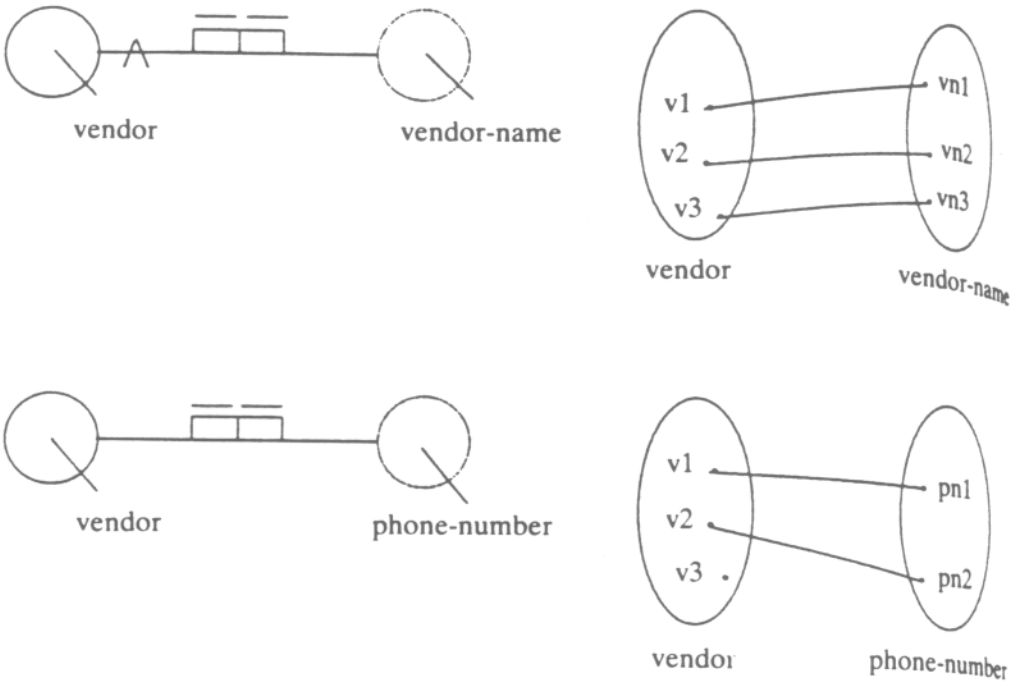


Fig. 10.5 Totality constraints.

A FINAL NIAM EXAMPLE

To summarize the meaning of the basic concepts presented thus far and to emphasize two important aspects of NIAM, we will use a simple set of accounting facts about customers and sales. Consider the following pair of well-formed sentences and note their expression in the ISD of Fig. 10.6.

FACT-type:	A Sale with Sale-Number Is made by A Customer with Customer-Number.
FACT:	A Sale with Sale-Number Sn1 Is made by A Customer with Customer-Number CN1.

This example illustrates first of all the close relationship that exists between natural language expression and the ISD notation. This correspondence between these two representations makes natural language communication during systems analysis natural.

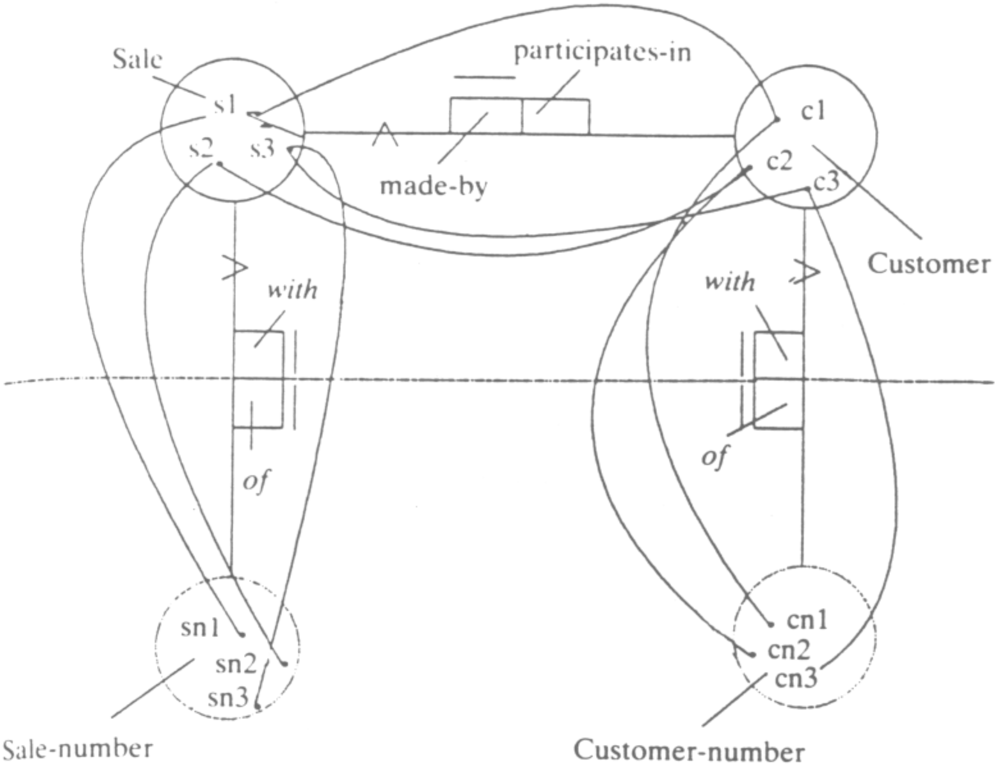


Fig. 10.6 Two world representation.

Secondly, the dotted line of the figure illustrates the distinction between the real world and the representation of the real world. This distinction in NIAM is sometimes obscured in database analysis and design. It is very important to proper design and use of accounting information systems, a fact noted well by Ijiri (1975) in his discussions of the differences between accounting principals and surrogates.

THE BOOK-KEEPING APPROACH TO ACCOUNTING vs AN REA SEMANTIC MODEL

Semantic models like NIAM are a recent addition to the science of representing real-world phenomena and making those representations available to decision-makers. What did people do before the advent of such models or even before the advent of computerized information systems?

In the case of accounting decision-makers, the answer to this question is obvious. Corporate book-keepers used charts-of-accounts and double-entry procedures to record phenomena such as sales with customers and purchases with vendors. However, as noted in the quote by McCrae at the beginning of this chapter, such book-keeping systems have not evolved along the path suggested by semantic modelling and database theory but have instead taken more the tack of simply running old systems on faster storage and processing technologies.

In this section, we will highlight the implications of such choices by showing a simple book-keeping example of an accounting UoD first and by then contrasting that example with a NIAM representation. In between the construction of these two systems, we will illustrate a theory of database accounting – the REA model – which was specifically designed to facilitate the construction of enterprise-wide accounting information systems.

A DEBIT–CREDIT EXAMPLE

Consider the following description of a fastener retailer:

> We purchased £200 worth of nuts and bolts from one vendor on Wednesday, then £300 worth of screws and nails from another vendor, and finally £25 worth of baling wire from a third supplier. We turned around on Thursday and sold half of the nuts and bolts plus all of the screws and nails to a customer for £500. Later that day, we sold the rest of the nuts and bolts to another customer for £150. On Friday, our first customer paid in full while the second one sent us £50, and on Saturday, we squared our accounts with the first two vendors.

Book-keeping analysis of the first Wednesday transaction would result in the following journal entry:

Debit: Inventory	£200	
Credit: Accounts payable		£200

to record Wednesday's purchase of nuts and bolts

In Fig. 10.7, we illustrate how the rest of this retailer's weekly business would be transcribed with book-keeping terms. Readers should note that these T-accounts are a modelled representation of the enterprise's activities, albeit a very restricted one intended for a very limited audience of predefining accounting users. Most modern computerized general ledger systems retain this somewhat insular mindset. In McCrae's terms, they have switched technology but not system.

With a database approach, however, we would discard these debit-credit approaches and try to model the Universe of Discourse directly in terms of its facts and ideas. Nonetheless, even in the case where a natural

Inventory	
Wed. 200	
Wed. 300	
Wed. 25	
	Thu. 400
	Thu. 100

Accounts Payable	
	Wed. 200
	Wed. 300
	Wed. 25
Sat. 200	
Sat. 300	

Sales	
	Thu. 500
	Thu. 150

Accounts receivable	
Thu. 500	
Thu. 150	
	Fri. 500
	Fri. 50

Cost of goods sold	
Thu. 400	
Thu. 100	

Cash	
Fri. 500	
Fri. 50	
	Sat. 200
	Sat. 300

Fig. 10.7 Book-keeping model.

language oriented approach such as NIAM is used, modelling a UoD with semantic constructs remains a difficult task. This is true for accounting information systems as well as for other systems. An important difference in the accounting sphere, however, is the availability of a semantically specified domain framework: the REA accounting model (McCarthy, 1982). This framework is discussed next.

DATABASE ACCOUNTING WITH THE REA MODEL

The REA accounting model is best considered as an 'occurrence template' for accounting transactions. It implies that accounting phenomena occur in well-defined constellations of associated objects which can be linked together and which can be used to produce traditional accounting numbers and reports. The obvious difference with conventional accounting systems is that the resulting accounting database is much more flexible and natural. The ISD in Fig. 10.8 reflects the REA framework in terms of binary semantic model. The term 'REA' is deduced from the three basic components of the framework: resources, events and agents. We will explain these terms briefly and then proceed to a discussion of this framework's use.

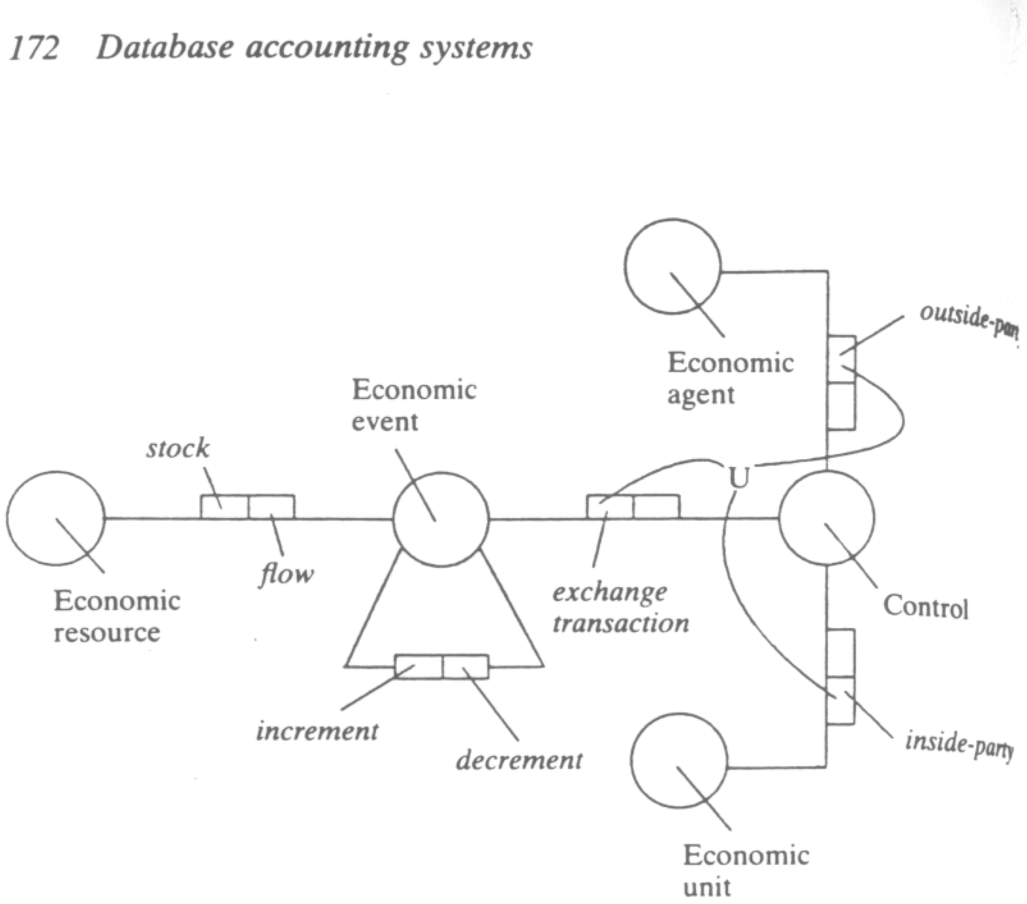


Fig. 10.8 The REA accounting model.

1. *Economic resources* are considered as objects that:
   (a) are scarce and have utility, and
   (b) are under the control of an enterprise
   (Ijiri, 1975, pp. 51–2).
2. *Economic events* are 'a class of phenomena which reflect changes in scarce means [economic resources] resulting from production, exchange, consumption and distribution' (Yu, 1976, p. 256).
3. *Economic agents* include persons and agencies who participate in the economic events of the enterprise or who are responsible for the participation of subordinates.
4. *Economic units* are a subset of agents – they are inside participants who work for or are part of the enterprise being accounted for.

The control object expresses a three-way association among event, unit and agent. There is also a binary association between resources and events which reflects the stock-flow roles that these objects play in enterprise business. Finally, and perhaps hardest to understand in terms of its departure from book-keeping, there is a binary association required of every accounting event with another event: its dual transaction. Accounting theory (Ijiri, 1975; Mattessich, 1964) requires that transactions associated with resource outflows from a company (decrements) be paired with resource inflows (increments) and vice versa. This is the duality principle of accounting.

The REA framework is intended to be used for operational object analysis of an enterprise's information needs during the specification phase of system design. For example, if something is identified as an economic resource, it can be expected that two event sets, one an inflow and the other an outflow, also will be specified. Furthermore, each of these two event sets would require inside and outside participants, and additional events that would be linked via duality relationships, and so on.

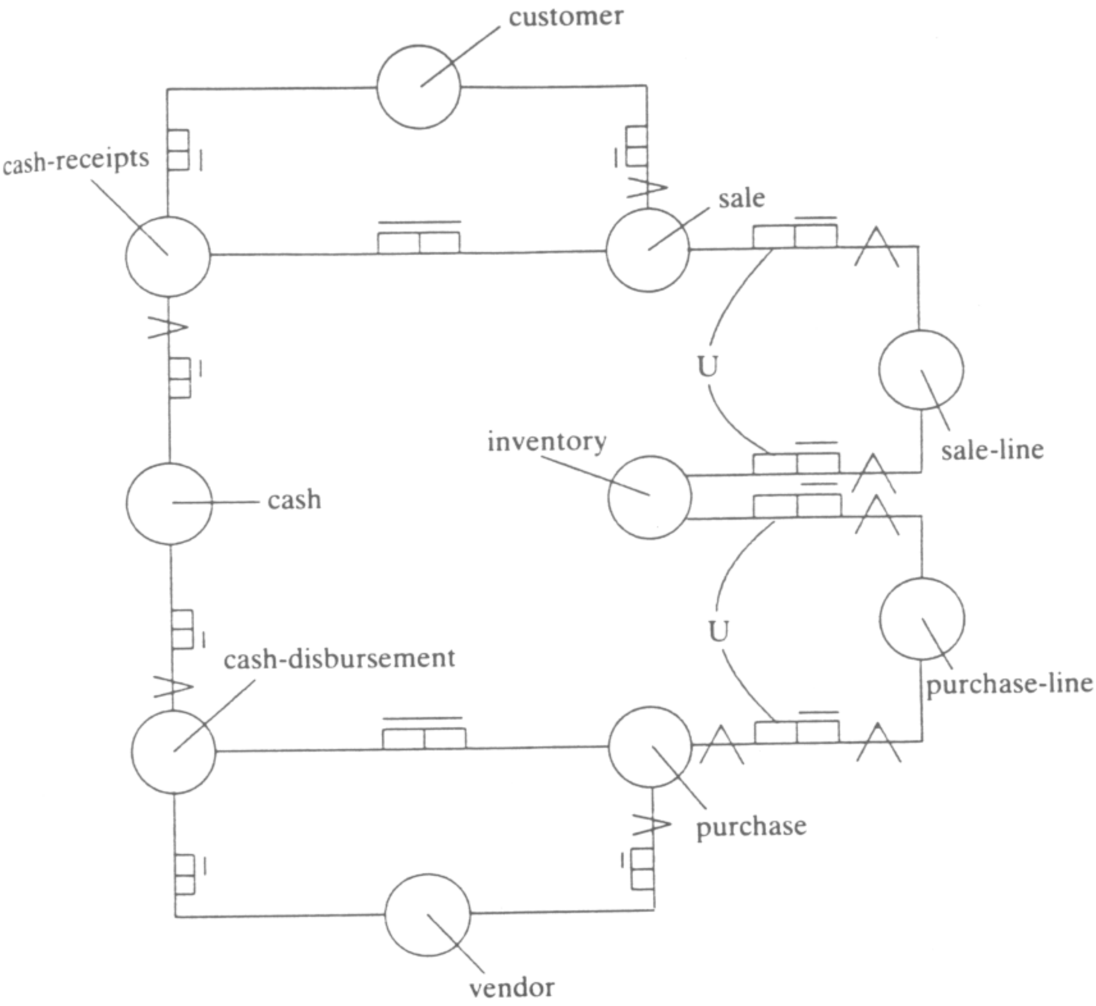


Fig. 10.9 REA instantiation.

In Fig. 10.9, the four transaction types of our fastener company – the Wednesday purchases, the Thursday sales, the Friday cash-receipts, and the Saturday cash-disbursements – are represented in such an integrated manner, beginning with the southeast part of the illustration and proceeding around counter-clockwise. For reasons of simplicity, inside participants (such as buyers, salespeople and cashiers) are neglected in this diagram. However, readers should be able to understand how each REA transaction template can be first instantiated and then linked to the other object constellations using the control, duality or stock-flow associations as the amalgamation points.

Transaction number	*Date*	*Economic event*
5	June 1	Bought on account merchandise from Oliver: 6 000 of A @ £2 £12 000 2 000 of B @ £4 8 000 £20 000 (Purchase Order 1)
6	2	Bought on account merchandise from Williams: 20 000 of E @ £1 £20 000 3 000 of C @ £9 27 000 £47 000 (Purchase Order 2)
7	3	Bought on account merchandise from Smith: 600 of D @ £10 £ 6 600 (Purchase Order 3)
19	9	Purchased on account from Smith: 600 of D @ £11 £ 6 000 (Purchase Order 4)
21	10	Bought on account merchandise from Oliver: 2 000 of B @ £4.30 £ 8 600 2 000 of A @ £2 4 000 £12 600 (Purchase Order 5)
30	17	Purchased merchandise on account from Oliver: 4 000 of A @ £2.30 £ 9 200 2 000 of B @ £4.30 8 600 £17 800 (Purchase Order 6)
33	18	Purchased from Williams on account: 1 000 of C @ £10 £10 000 (Purchase Order 7)
48	28	Purchased on account from Williams: 500 of C @ £9.25 £ 4 625 (Purchase Order 8)

Fig. 10.10 Purchase transactions.

AN EXTENDED REA EXAMPLE

Figures 10.7 and 10.9 contrast the structures of double-entry book-keeping and REA modelling, but a full appreciation of database accounting requires an extended example in which the reader may see final implementation ideas in very specific format. In this section we intend to provide such specifics with the NIAM modelling of purchasing transactions and the subsequent conversion of an ISD to a relational database. Our examples here will not be as broad as the fastener enterprise case, but the narrowing of scope will provide room for more details.

THE WILSON COMPANY – PURCHASE TRANSACTIONS

Figure 10.10 portrays a set of eight purchase transactions for a hypothetical enterprise called the Wilson Company. This example firm has been used previously by Gal and McCarthy (1983, 1986) in actual database accounting implementations. It includes a set of 55 transactions which occur during the first month (June) of a retail company's existence. Full Wilson details are given in McCarthy (1980) and are available upon request.

If we consider the set of purchase transactions as accounting phenomena to be modelled and if we use the REA framework of Fig. 10.8 as an occurrence template (again forsaking inside agents), the NIAM information structure diagram of Fig. 10.11 results (readers should note the central resource–event–agent constellation). With the exception of more detail, this is essentially the same modelling process we went through with the fastener company in Fig. 10.9.

ISD TO RELATIONAL DATABASE

As mentioned previously, a transformation algorithm exists that converts a conceptual NIAM schema into a fully normalized and optimal relational schema (Nijssen, 1989). Full explanation of this algorithm goes beyond the scope of this chapter, so we will limit ourselves to giving a cursory description of how the purchase REA-template in Fig. 10.11 can be translated to the relational model in Fig. 10.12.

1. Every NOLOT is represented as a relation, and a key must be selected to represent the non-lexical objects. That key must have the following characteristics:
   (a) the role at the side of the NOLOT must be mandatory, and
   (b) both roles of the binary association must have cardinality one.
   Keys are shown in Fig. 10.12 as double-headed arrows.

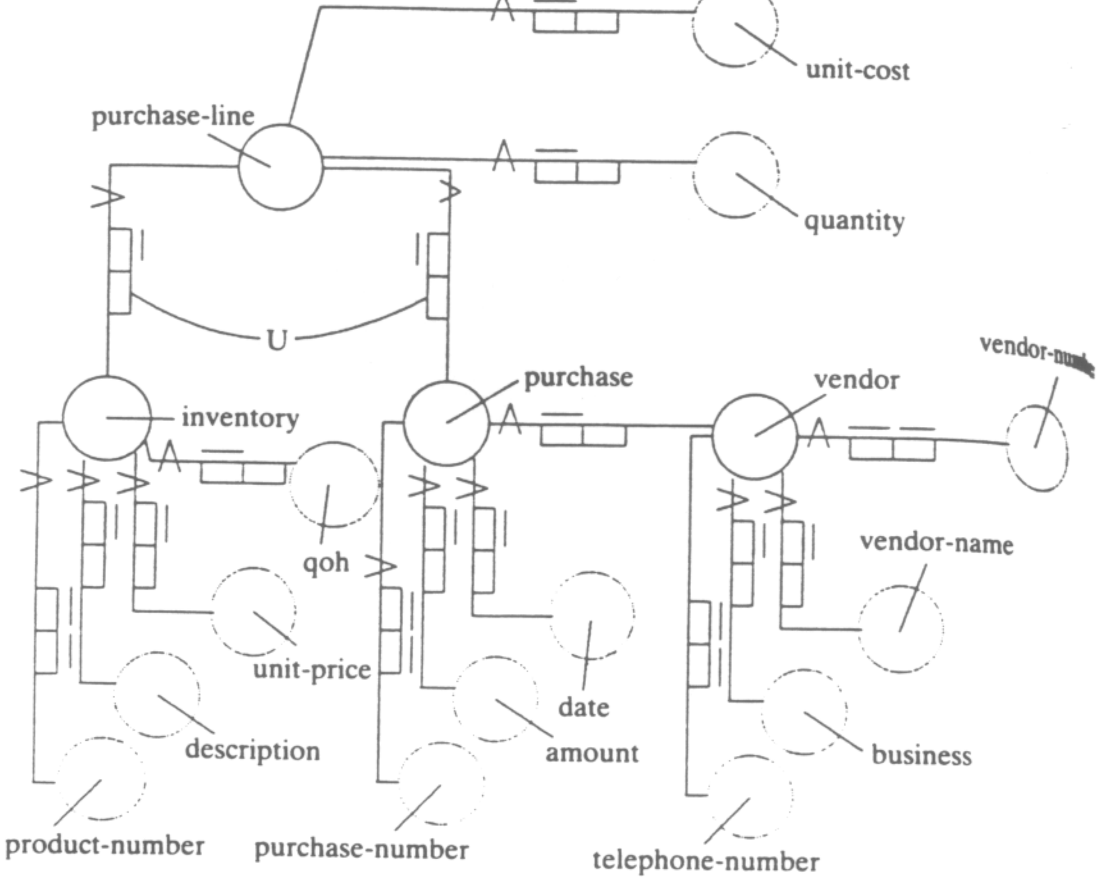


Fig. 10.11 NIAM model for purchase transactions.

2. Only LOTs with a uniqueness constraint on the role at the side of the NOLOT may be incorporated in the relation. Other cases are neglected in the example.
3. Non-mandatory roles are indicated by the symbol 'O' in the relational data model. Telephone_number in the relation customer is an example.
4. Finally, the associations between NOLOTS must be translated. Two kinds of interrelation connections are expressed in Fig. 10.12: equality (with an 'E') and subset (with an arrowhead). In our case these connections are implied by the mandatory constraints of Fig. 10.11. In simpler terms, the equality and subset symbols in the Wilson example equate to the following:

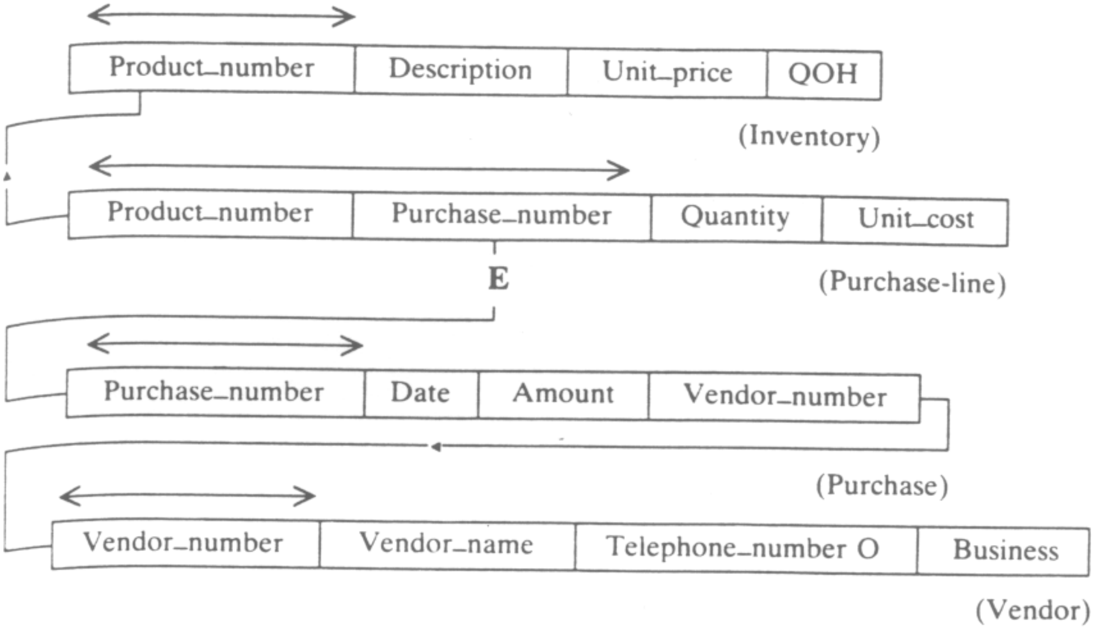


Fig. 10.12 Relational model for the purchase transaction.

(a) when a purchase occurs; it has to have designations of products, quantities and unit costs; and

(b) products and vendors exist independent of purchases, but when a purchase does occur, its vendor and products must already be known to the database.

Constraints like these enforce what database theorists call 'referential integrity'.

Fig. 10.13 shows the full Wilson extension for the database of Fig. 10.12. In examining the individual rows or tuples of these four tables, readers should remember again the purpose of conceptual schemas and databases – to represent to potential users the facts of a certain Universe of Discourse. By following around the individual values in these tuples, one can recreate part of the Wilson story for the month of June. Database users can do the same which was, of course, our primary goal in designing this type of accounting system.

PROCEDURAL SPECIFICATION FOR WILSON

In an operational enterprise, the structural components of a relational database are augmented by procedures that maintain data and aggregate it for eventual decision use. Two such procedures are shown in SQL (Date, 1986) in Fig. 10.14.

INVENTORY

Product_number	Description	Unit_price	QOH
7432	A	3.00	2000
8519	B	5.00	1000
6784	C	12.00	1500
5862	D	15.00	200
4888	E	1.50	7000

PURCHASE-LINE

Product_number	Purchase_number	Quantity	Unit_cost
7432	1	6 000	2.00
8519	1	2 000	4.00
4888	2	20 000	1.00
6784	2	3 000	9.00
5862	3	600	10.00
5862	4	600	11.00
8519	5	2 000	4.30
7432	5	2 000	2.00
7432	6	4 000	2.30
8519	6	2 000	4.30
6784	7	1 000	10.00
6784	8	500	9.25

PURCHASE

Purchase_number	Date	Amount	Vendor_number
1	June 1	20 000	200
2	June 2	47 000	201
3	June 3	6 000	202
4	June 9	6 600	202
5	June 10	12 600	200
6	June 17	17 800	200
7	June 18	10 000	201
8	June 28	4 625	201

VENDOR			
Vendor_number	Vendor_name	Telephone_number	Business
200	Oliver	3244	wholesaler
201	Williams	1204	wholesaler
202	Smith	4502	wholesaler
203	Hodge	9861	trucker
204	Simmons	1243	cleaner
205	Green	7743	realtor
206	McKenzie	1288	pub-rel
207	Horvath	1324	off-equip
208	Shore	1917	car-dealer

Fig. 10.13 Relational database.

For data entry of the first purchase transaction of 1 June the SQL statements in Fig. 10.14(a) can be used. In real applications, such line-by-line coding would not be used with large amounts of new data. Instead, customized forms (in SQL-forms) would be provided. Notice that the application must compel the defined semantics. For example, no product_number can be inserted in the purchase-line relation if this number is not an occurrence of the inventory relation.

Once transactions are registered, information must be derived from stored data. In the case of the traditional book-keeping model, this occurs by aggregation after posting. This procedure is well-known and can be found in classical accounting textbooks.

The same information can easily be obtained in accounting databases by using conclusion-materialization hierarchies (McCarthy, 1984). Gal and McCarthy (1986) demonstrate these with a QBE implementation for the general ledger while Geerts (1990) illustrates an SQL implementation for a traditional report in current accounting environments: age-listings. A much simpler example of information retrieval with SQL is illustrated in Fig. 10.14(b): list the name(s) of the vendors who are wholesalers.

CONCLUSIONS AND DISCUSSION

In this chapter an alternative accounting model – called database accounting – was discussed and compared with the existing double-entry approach. We demonstrated first the importance of conceptual schemas in information systems design and use, and we proceeded to introduce a widely accepted methodology for schema specification: the NIAM model. We proceeded

```
Insert into PURCHASE values
(‘1’, ‘June 1’, 20 000, ‘200’);
```

```
Insert into PURCHASE-LINE values
(‘7432’ ‘1’, 6000, 2.00)
```

```
Insert into PURCHASE-LINE values
(‘8519’, ‘1’, 2000, 4.00)
```

```
Update INVENTORY
set QOH = QOH + 6000
where product_number = ‘A’;
```

```
Update INVENTORY
set QOH = QOH + 2000
where product_number = ‘B’;
```

(a) Date entry operations

```
QUERY
Select Vendor_name
From VENDOR
Where Business = ‘wholesaler’
```

```
OUTPUT
Oliver
Williams
Smith
```

(b) Retrieval operations

Fig. 10.14 Database operations.

next to document how a semantically oriented domain theory – the REA accounting model – could be used to facilitate the development of accounting databases. These databases were then compared with more traditional methods in the context of two transaction sets, the first very simple and the second loaded with more detailed facts.

Out of this comparison of two approaches to accounting follows the conclusion that the database approach leads to a more natural representation of the Accounting Universe of Discourse. This is true for a variety of reasons documented by McCarthy (1990), the most prominent of which is that semantically developed accounting systems aspire to assist a much wider group of decision-makers than do debit–credit frameworks. They therefore cannot become encumbered with book-keeping artifacts which make little sense to non-accountants. While representation of disaggregate data and richer semantics requires more complex integrity-maintenance procedures, very diversified sets of information patterns can be followed, and the extra cost seems well worth it.

DISCUSSION QUESTIONS AND EXERCISES

1. Prepare the four-day financial statements for the fastener company. Make any assumptions about incomplete transactions that make sense.
2. Fill in the other three types of transactions for Wilson using your imagination for the facts. If you assume no other expenses and cash payments, you might have close to a whole system, and you should be able to prepare income and balance sheet estimates. Alternatively, you could look at a fully specified data model for Wilson (McCarthy, 1979, p. 675) in entity–relationship form and try to translate it to NIAM.
3. What kind of operations would be needed to produce financial statements from a full set of relations. If you have access to a relational DBMS on a micro or mainframe computer, you might try a partial implementation of these ideas. If your DBMS is strictly relational, however, you might have problems with LIFO or FIFO inventory costing as did Gal and McCarthy (1986). Why?
4. Why are debits and credits not needed in the REA model? Do you think accounting systems like this would ever win enthusiastic approval among accountants enamoured with more traditional methods?
5. Why would an REA system work better than book-keeping for a marketing or transportation decision-maker who was interested in data about inventory sales and shipments? Would this generalize to other non-accountants?
6. If you have had any exposure to expert systems or AI programming, try to discuss how semantic systems might interface with such tools.

ACKNOWLEDGEMENT

A significant portion of this chapter has been written directly from prior work of the authors (McCarthy, 1979, 1980, 1982). For readability purposes, quotations from those sources are not identified explicitly. Support for this work was provided by IBM-Belgium and Arthur Andersen & Co.

REFERENCES

Akoka, J. and Augustin, G. (1983). 'Le modèle comptable multidimensionnel l'approche entité-relation', *Revue Française de Gestion,* Juin–Juillet–Août, pp. 73–88.

Chen, P. P. (1976). 'The entity-relationship model – toward a unified view of data', *ACM Transactions on Database Systems,* 1 March, pp. 9–36.

Date, C. J. (1986). *An Introduction to Database Systems Volume I.* Addison Wesley, Reading, Mass.

Falkenberg, E. (1981). 'Some aspects of conceptual data modelling'. In Holsapple, C. W. and Whinston, A. B. (eds), *Data Base Management: Theory and Applications.* pp. 19–35.

Gal, G. and McCarthy, W. E. (1983). 'Declarative and procedural features of a CODASYL accounting system'. In Chen, P. (ed.), *Entity–Relationship Approach to Information Modeling and Analysis.* North–Holland, Amsterdam.

Gal, G. and McCarthy, W. E. (1986). 'Operation of a relational accounting system'. In Schwarz, B. N. (ed.), *Advances in Accounting.* JAI Press, Greenwich, Conn, pp. 83–112.

Geerts, G. (1990). *Het Gebruik van Tweede Generatie Gegevensmodellen bij het Ontwerp vaneen Events Accounting Information System.* Working Paper BEIF/ 23, Free University Brussels.

Ijiri, Y. (1975). *Theory of Accounting Measurement.* American Accounting Association, Sarasota, Florida.

Leung, C. M. R. and Nijssen, G. M. (1988). 'Relational database design using the NIAM conceptual schema', *Information Systems,* Vol. 13, No. 2, pp. 219–27.

Kent, W. (1986). 'The realities of data: basic properties of data reconsidered'. In Steel, J. R. and Meersman, R. (eds), *Database Semantics (DS–1).* North-Holland IFIP, Amsterdam, pp. 175–88.

Mark, L. (1983). 'What is the binary relationship approach?' In Davis, C. G., Jajodia, S., Ng, P. A. and Yeh, R. T. (eds), *Entity Relationship Approach to Software Engineering.* ER Institute, pp. 205–20.

Mark, L. (1987). 'E–R modeling versus binary modeling. A message from the church of binary methodists'. In *Proceedings of the Sixth International Conference on Entity–Relationship Approach,* New York, pp. 67–8.

Mattessich, R. (1964). *Accounting and Analytical Methods.* Richard D. Irwin, Homewood, Illinois.

McCarthy, W. E. (1979). 'An entity–relationship view of accounting models', *The Accounting Review,* Vol. 54, October, pp. 667–86.

McCarthy, W. E. (1980). *A Case Study Demonstrating the Applicability of Data Modeling to Accounting Object Systems.* Presentation for the Southeast Regional Meeting of the American Accounting Association, South Carolina. (Working Paper available.)

McCarthy, W. E. (1981). 'Multidimensional and disaggregate accounting systems', *MAS Communication,* July, pp. 7–13.

McCarthy, W. E. (1982). 'The REA accounting model: a generalized framework for accounting systems in a shared data environment', *The Accounting Review,* Vol. 57, July, pp. 554–78.

McCarthy, W. E. (1984). *Materialization of Account Balances in the REA Accounting Model.* Presentation to the British Accounting Association, University of East Anglia. (Working Paper available.)

McCarthy, W. E. (1990). *Once Upon a Time, at the Agleclap Store. ...* Presentation to the 1990 AAA Senior Faculty Consortium, Phoenix, AZ. (Working Paper available.)

McCarthy, W. E., Rockwell, S. R. and Armitage, H. (1989). 'A structured methodology for the design of accounting transaction systems in a shared data environment', In *Proceedings STA 5,* Chicago, pp. 194–207. (Working Paper available.)

McCrae, T. W. (1976). *Computers and Accounting.* John Wiley and Sons.

Nijssen, G. M. (1989). *Conceptual Schema and Relational Database Design: A Fact Oriented Approach.* Prentice Hall.

Sorter, G. H. (1969). 'An events approach to basic accounting theory', *The Accounting Review,* Vol. 44, January, pp. 12–19.

Verheijen, G. M. A. and van Bekkum, J. (1982). 'NIAM: an information analysis method'. In Olle, T. W. and Sol, A. A. (eds), *Information Systems Design Methodologies: A Comparative Review.* North–Holland, Amsterdam, pp. 537–89.

Wintraecken, J. J. (1986). *Informatieanalyse volgens NIAM in Theorie en Praktijk.* Academic Service.

Yu, S. C. (1976). *The Structure of Accounting Theory.* University Press of Florida, Grainesville, Florida.