Business Object Reference Ontology Program

simplifying semantics

Working Paper

MA2

Methodology: Application - 2

Using Business Objects to Re-engineer the Business

Issue: Version - 4.01 - 01-July-2001

Contact

For queries regarding this document, or the BORO Program in general, please use the following email address:

contact@BOROProgram.org

MA2

METHODOLOGY: APPLICATION - 2

USING BUSINESS OBJECTS TO RE-ENGINEER THE BUSINESS

CONTENTS

BORO

CONTENTS

MA2

MA2

METHODOLOGY:
APPLICATION - 2

USING BUSINESS OBJECTS TO RE-ENGINEER THE BUSINESS

1 Introduction

The BORO Working Papers explain how the business object paradigm is starting to revolutionise the way computer systems are built. They describe how business objects are not only making systems simpler and functionally richer and so cheaper to build and maintain. This inevitably leads to big changes, such as substantially increased levels of automation. However, it would be a serious mistake for us to think that business objects will only change computer systems. We would be missing their far more exciting potential for re-engineering the business.

1.1 Missing the wider potential

It often happens that the wider potential of a radical innovation is missed. History is littered with examples. For instance, Western Union, the telegraph company, turned down the chance to buy Alexander Graham Bell's 1876 telephone patent for a small sum. It thought that it was thinking strategically when it offered to stay out of telephones if Bell stayed out of telegraphy. (Bell also missed the point: he entitled his patent *Improvements in Telegraphy*.)

More recently, the inventor of the transistor, one of the 20th century's most important innovations, thought it might be used to make better hearing aids. Even more recently, when the laser was invented at Bell Labs, its lawyers were initially unwilling even to apply for a patent on the invention, believing it had no possible relevance to the telephone industry.

In the computing industry, the founder of IBM, Thomas J. Watson, Senior, originally declared in 1948 that as many as 12 companies might some day have their own computers (a few years later he revised this figure to 50). He anticipated that scientists and engineers would use them as improved calculating machines—replacements for their log tables and slide rules. He had no idea that business people and accountants might be a market.

1.2 Business objects' wider application

We would be making a similar mistake to Thomas J. Watson, if we expect business objects to be only used to build better computer systems that automate more of the business. Like other radical innovations, they have a wider potential than their obvious application. Surprising as it may seem, I expect that their most significant impact will not be on computer systems, but on the businesses underlying those systems. They will play an important part in the industrialisation of business's information. This will have its biggest impact on those businesses (or parts of the business) that work with information, for example:

- Information industries—such as banking and insurance, and
- Information professions—such as accounting and law.

2 The accounting paradigm's debit and credit pattern

We can get some idea of how far reaching the effects of this industrialisation will be by looking at an example of how business objects are going to change an area of the business. The spatial, temporal and naming patterns that we re-engineered in

MW—The BORO Methodology: Worked Examples are too general for this. We use as our example a paradigm that is central to the management of most businesses—the accounting paradigm. Re-engineering this is a substantial task; all we do here is outline how a core pattern—the accounting transaction's debit and credit pattern—can be re-engineered.

The accounting paradigm is a pertinent example. Its current framework is the accounting ledger, whose columns and rows are designed for paper's two-dimensional surface. *MW—The BORO Methodology: Worked Examples* showed us the extent to which the paper-bound entity paradigm constrains and distorts patterns. The example of an accounting pattern that we are going to look at—accounting transaction – has been distorted in a similar way to fit into the constraints of paper's two-dimensional surface. The re-engineering will free it from those constraints.

2.1 From journal transaction to debit and credit movements

Accounting transaction's debit and credit movement pattern is well over five hundred years old, but was effectively standardised in the 15th century. This happened when the invention of printing lead to the publication and wide distribution of a number of books describing the process of bookkeeping.

The first, and most famous, book was by a Franciscan monk, Fra Luca Pacioli. In 1494 he published a book on mathematics (*Summa de Arithmetica, Geometria, Proportioni et Proportionalita*), which contained a treatise on bookkeeping (*Particularis de Computis et Scripturis*, which translates as 'Details of Accounting and Recording').

In his treatise, Pacioli described the book-keeping method used by the merchants of Venice (which was then one of the most powerful city states in Europe); hence, he called it the Method of Venice. The method was not new; the merchants of Venice had been using it for centuries. However, once Pacioli's book was published, bookkeepers across Europe started to standardise on it. The Method of Venice

has proved to be extremely durable; accountants and bookkeepers still use something similar today.

A central feature of the Method of Venice is double entry bookkeeping. It is called double entry because a transaction is, in general, entered twice, firstly into a journal and secondly into a ledger. It is entered into the journal in the format of an accounting transaction. It is then divided into a debit and a credit movement and these are entered into different parts of the ledger.

This is an example of how resorting and reformatting is done within the constraints of paper and ink technology. Each 'book' contains the same information, but in a different format and order; each gives us a view of the business. Paper and ink technology sets a limit on the number of possible views. For example, taking any more than these two book-keeping views would involve significant extra effort.

Nowadays, most computing systems automate this manual resorting and re-formatting. When a transaction is entered, they first store it on a transaction file. Then, they automatically re-format it as a debit and credit movement and 'post' it to an account movements file, updating the relevant ledger balances.

The way in which the book-keeping process divides the transaction into a debit and credit movement for the ledger view, suggests that it sees the transaction as having the two movements as components—as illustrated schematically in *Figure MA2–1*.

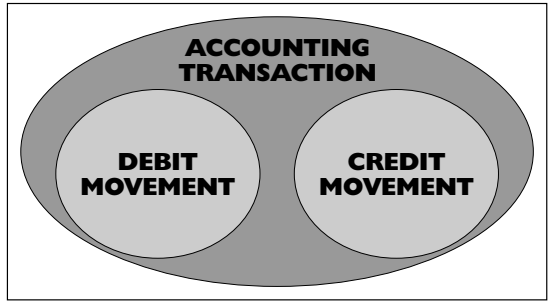


Figure MA2–1
An accounting transaction and its component movements

2.2 The accounting transaction and movements entity formats

We now re-engineer the accounting transaction and its two components. This should unwind any distortions imposed by pen and paper technology—revealing the objects that the transaction refers to.

We follow the process used in the worked examples. We look at a listing of the entities and then their entity formats. The entities are shown in *Table MA2–1* and *Table MA2–2*, their formats in *Table MA2–3* and *Table MA2–4*. You can see how neatly accounting's paper-based rows and columns map into the similarly paper-based entity formats.

Table MA2–1 Partial accounting transactions listing

Transaction Code	Transaction Date	From Account	To Account	Amount
#101	25-Apr-94	Joe Bloggs	Me	£10,000

Table MA2–2 Accounting transaction entity format

Entity type	Accounting Transaction
Attribute type #1	Transaction code
Attribute type #2	Transaction date
Attribute type #3	From Account
Attribute type #4	To Account
Attribute type #5	Amount

Table MA2–3 Partial accounting movements listing

Entry Code	Transaction Code	Transaction Date	Account	Debit/credit Indicator	Amount
#10	#101	25-Apr-94	Joe Bloggs	Debit	£10,000
#11	#101	25-Apr-94	Me	Credit	£10,000

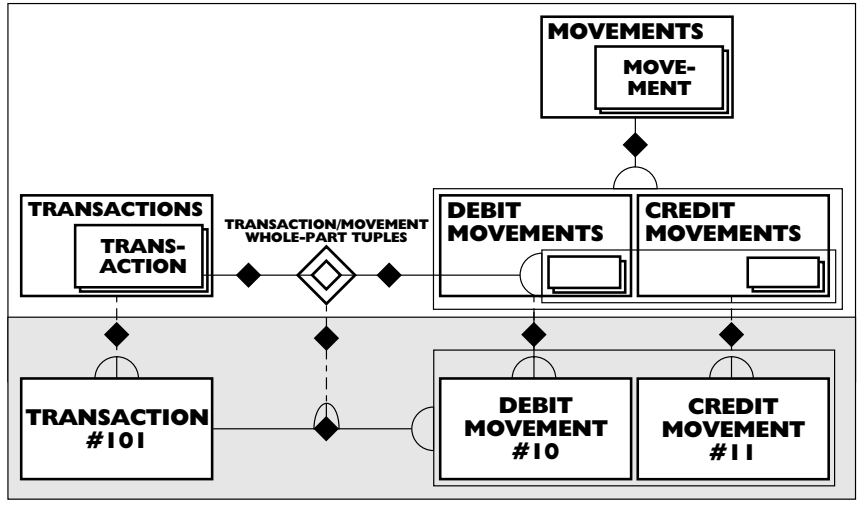
Table MA2–4 Accounting movement entity format

Entity type	Accounting movement
Attribute type #1	Entry code
Attribute type #2	Transaction code
Attribute type #3	Transaction date
Attribute type #4	Account
Attribute type #5	Debit/credit indicator
Attribute type #6	Amount

2.3 Re-engineering the accounting transaction pattern

If we assume that the movements are components of the transaction, we might re-engineer the two entity formats into the model in *Figure MA2–2*.

Figure MA2–2 Accounting transaction model



2.4 Re-engineering a transaction event

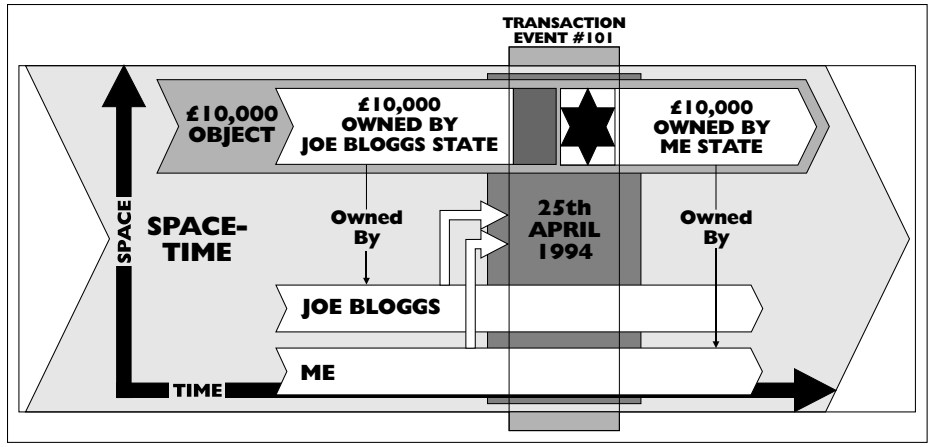
We begin to realise that *Figure MA2–2* is not accurate, when we ask, from an object point of view, what the accounting entities refer to. We realise that the model is describing patterns in the information rather than in what the information refers to—the business.

A big clue that this is happening is the type of sign used for the transactions and movements. They are modelled with physical body signs, suggesting that they are physical bodies. But in the business neither the transaction nor the movements persist through time. This makes them events, not physical bodies. (We looked at this type of mis-classification for accounting movements in *AS4—Focusing on the Things in the Business*, when we considered the data–process and things–changes distinctions.)

If we now look at the actual transaction event in the business, we get a very different pattern from *Figure MA2–1* and *Figure MA2–2*. We start by asking what the event happens to. The answer is the £10,000—it changes owner. We came across this pattern in *OP4—Business Object Ontology Paradigm*, where we looked at the sale of a car (illustrated in OP4's *Figure OP4–30* and *Figure OP4–31*). In that pattern, the car moved from an 'owned by garage' state into an 'owned by Ms Brown' state. We re-use the pattern here on the £10,000. We see it 'moving' from an 'owned by Joe Bloggs' state into an 'owned by me' state.

The transaction event is revealed as a 'change' in the £10,000's states (described in the space-time map in *Figure MA2–3*). You may have noticed that in this revised view there are no debit #10 and credit #11 movement objects (illustrated in *Figure MA2–2*). Debits and credits are ways of looking at the transaction event, not objects.

Figure MA2–3 £10,000 state change event space-time map

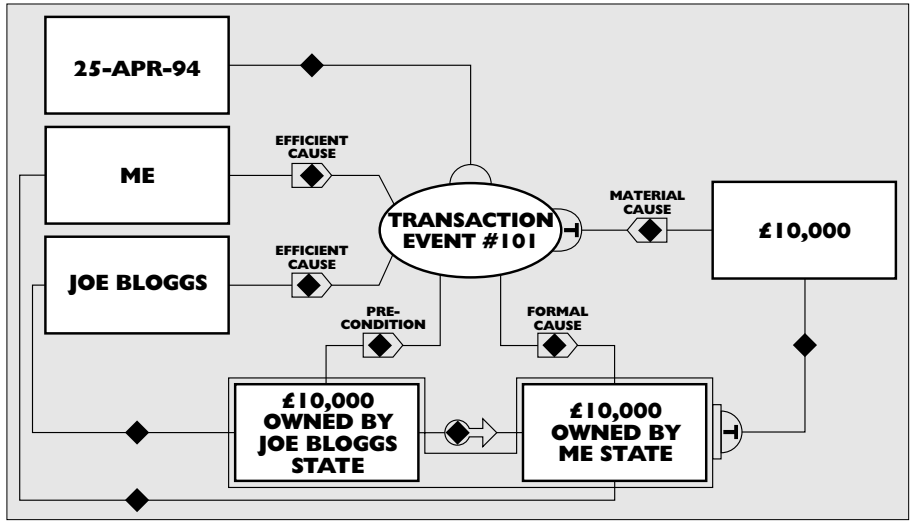


This space-time map helps us see the event's causal connections (as explained in *OP4—Business Object Ontology Paradigm* and illustrated by OP4's *Figure OP4–38*). For example, the two parties to the transaction (Joe Bloggs and Me) are, in Aristotelian terms, the efficient causes of the event (things that make the change happen). In addition:

- The £10,000 is a material cause (what the change happens to), and
- The '£10,000 owned by me' state is a formal cause (what the change results in).

In addition, the '£10,000 owned by Joe Bloggs' state is a pre-condition All these causal connections are modelled in the schema in *Figure MA2–4*. It also describes the structural nature of the transaction event's connection with the 25-Apr-94 day (date) object, which is whole–part.

Figure MA2–4 £10,000 transaction event #101 causes object schema



This is a very different pattern from that in *Figure MA2–2*. Its pattern was moulded by the constraints of paper and ink technology—particularly its re-sorting and re-formatting process. It gave a reflection of how transactions are re-formatted into movements, not a reflection of the business.

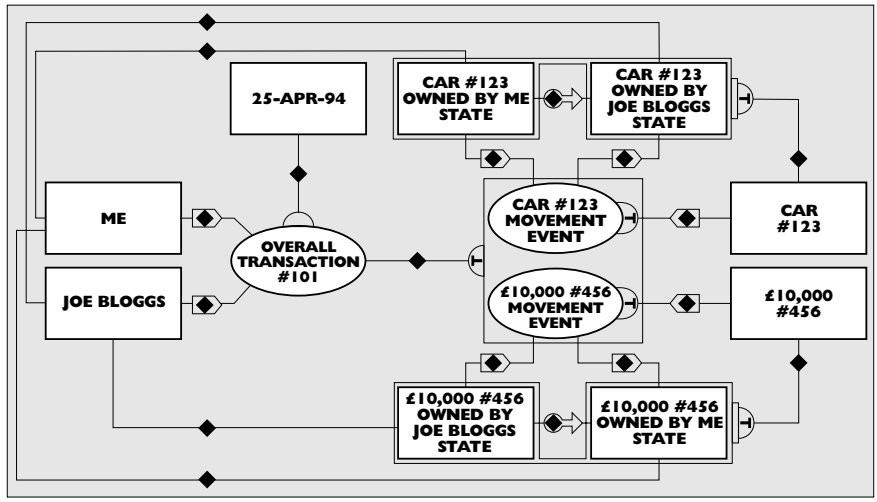
2.5 Re-engineering the overall transaction event

The re-engineered transaction event in *Figure MA2–4* only covers half the transaction. Joe Bloggs paid £10,000 for something, which does not appear in the accounting transaction. This is because accounting transactions only record 'movements' of money. They ignore the non-money element. Once we recognise this non-money element, we can see that the two elements combine to form an overall transaction.

When we analyse the non-money element of the transaction, we see it has the same pattern as the money element. Assume that Joe Bloggs bought car #123 with his £10,000. This car has an owned by me state ending in a movement event followed by an owned by Joe Bloggs state. The car and £10,000 movement events are the encapsulated parts of an overall transaction. Once we recognise

this, we can see that Joe Bloggs and I are parties to the overall transaction rather than the individual movement events. We capture this insight in *Figure MA2–5*.

Figure MA2–5 Overall transaction #101's object schema

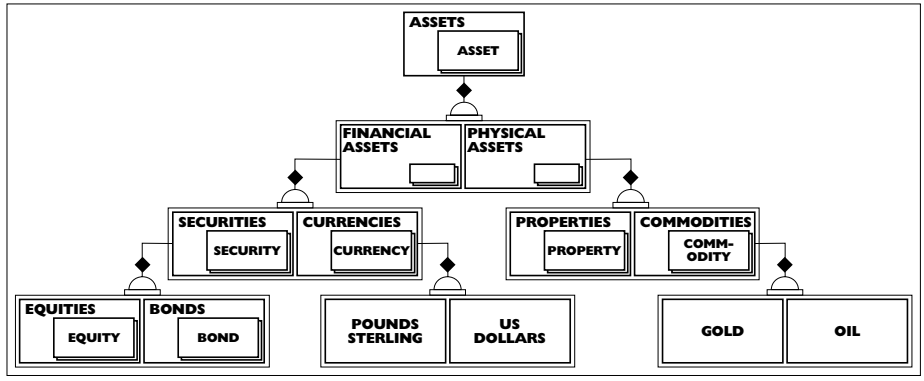


2.6 Generalising to the assets level

In *Figure MA2–5* it is clear that the pattern for the money and non-money movement events are similar. When we generalise this individual transaction pattern to class level, we construct one class-level pattern that has both money and non-money elements as its members. The accounting transaction pattern cannot recognise this generalisation because it is artificially restricted to the money elements only—a result of its origins in paper technology's rows and columns.

Money—or currency—is merely one type of asset. It is not even a major type of asset as the schema of generalised assets in *Figure MA2–6* shows.

Figure MA2–6 Generalised assets



The asset super–sub-class hierarchy is rich. The re-engineerings that I have been involved in have revealed a variety of asset sub-classes. Things such as dividend entitlement coupons and tax credit vouchers turn out to share in the overall pattern of the asset family. However, for our current purposes, the key aspect of this asset hierarchy is that it shows the types of asset our re-engineered transaction can model.

2.7 Generalising transactions to orders/exchanges

There is another direction in which the transaction pattern can be generalised. Transactions are composed of two general patterns—order and exchange. These can be combined in a different way to construct another core pattern—order then exchange.

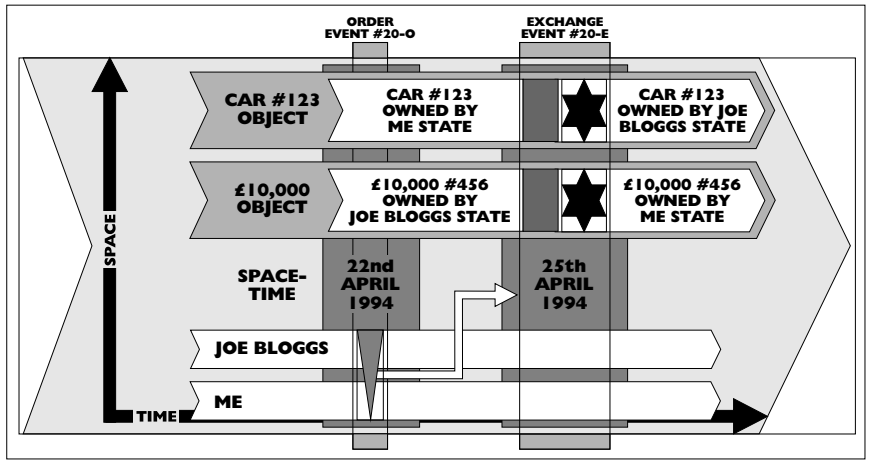
We now assume that Joe Bloggs called up and ordered his car a few days before he came in and exchanged his £10,000 for it. The order then exchange would be recorded as in *Table MA2–5*.

Table MA2–5 Partial simplified orders then exchanges listing

Order Code	Ordered By	On	For	Item Type	Number Of Items	Item Cost	Total Cost
#20	Joe Bloggs	22-Apr-94	25-Apr-94	Car	1	£10,000	£10,000

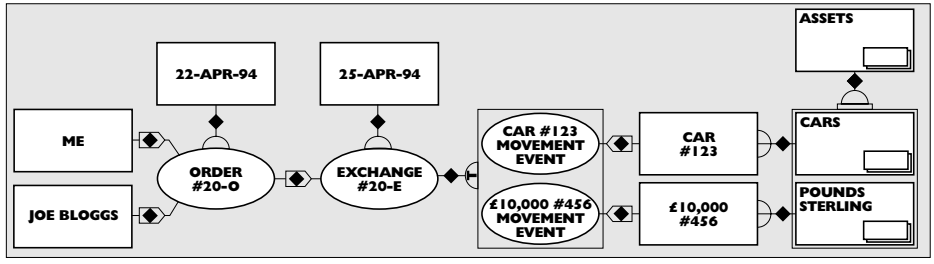
If we re-engineer this order, we find it refers to the objects described by the space-time map in *Figure MA2–7*. Notice that the order and exchange elements look as if they are the transaction illustrated in *Figure MA2–3* divided in half.

Figure MA2–7
Order space-time map



We intuitively understand the order as contracting for the future exchange. This is reflected in the object schema in *Figure MA2–8*. The order event gets its meaning (in Fregean terms, its sense) from its connecting pattern with its exchange. Notice that now we have generalised assets; we show the two amounts as belonging to the asset sub-classes—cars and sterling. Notice also that the underlying pattern has been made clearer by the omission from the schema of the before and after states of the amounts.

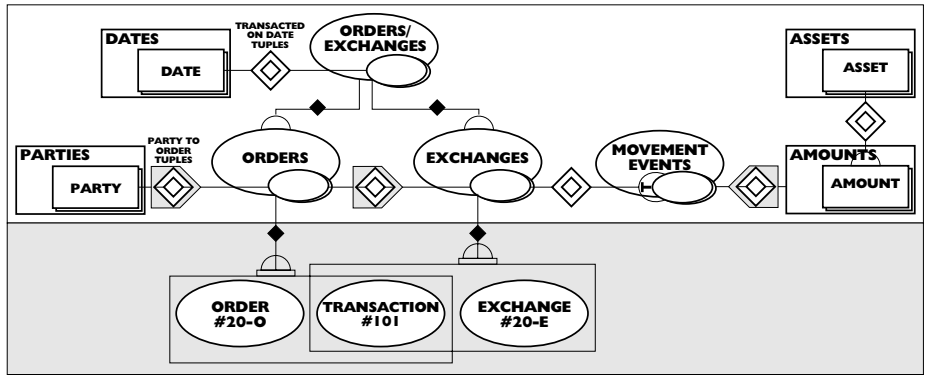
Figure MA2–8
Order object schema



It is interesting to compare this schema with the schema in *Figure MA2–5*. The connecting patterns that 'Me' and Joe Bloggs had with transaction #101 have

moved along to order #20-O. It is as if the standalone transaction in Figure MA2–5 has been divided into two—which it indeed has. Order #20-O has separated the parties contracting to the exchange from exchange #20-E. The individual level order, exchange and transaction patterns generalise into the class level pattern shown in the simplified object schema in *Figure MA2–9*.

Figure MA2–9
General order
object schema



The current accounting paradigm, with its origins in paper and ink technology, cannot accommodate this general order/exchange pattern's shape. It typically works around the problem by treating the order element of an order then exchange as another accounting transaction, which generates debit and credit movements for the order date. The accounting paradigm cannot give any firm guidance about what these movements should be, because they only indirectly refer to the transaction. This has resulted in equally 'valid' but different ways of accounting for the overall transaction. For example, bank's treasury operations can choose between a trade (order) and a value (exchange) date accounting approach.

2.8 Generalising the order pattern

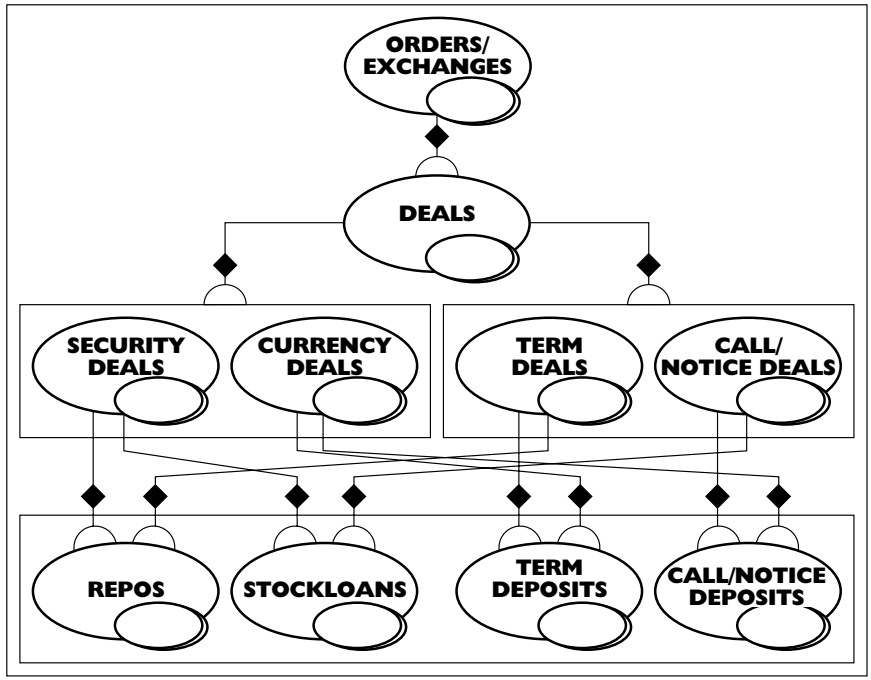
The re-engineering has given us a general order/exchange pattern with the movement event—the object version of accounting movement—at its core. This order/exchange pattern is a basic business pattern. It occurs frequently across a range of businesses. In a re-engineering of an international securities settlement system, we found it in most of the 'transactions', including:

- Security purchases,
- Security sales,
- Dividend entitlements,
- Bonus and rights entitlements,
- Tax entitlements,
- Stock borrowing and lending agreements,
- Term deposits placed and accepted,
- Foreign exchange deals, and
- Call/notice deals.

We found that the generalisation of these classes followed the same pattern as the *MW—The BORO Methodology: Worked Examples*. As higher level classes were constructed, these became redundant and were purged, compacting the model.

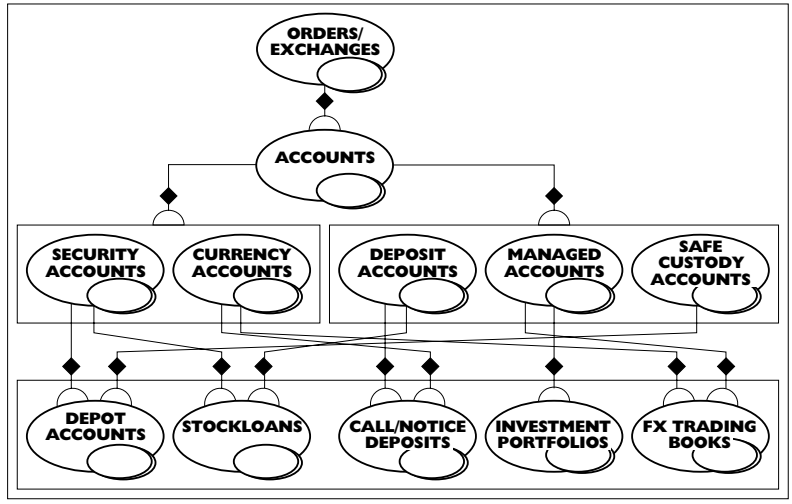
The sub-classes of the general orders/exchanges class fell into a super–sub-class hierarchy similar to the one shown for deals in *Figure MA2–10*. In it, we can see how high level patterns combine to construct new sub-classes. For example, when the term deals pattern is combined with the currency deals pattern, it gives a term deposits pattern. When it is combined with the security deals pattern, it gives a repurchase agreements (repos) pattern.

Figure MA2–10 Deals super–sub-class hierarchy



We also found that at a general level, the order/exchange pattern was a generalisation of the account pattern. So accounts and its various sub-classes are revealed as sub-classes in the orders/exchanges super–sub-class hierarchy. *Figure MA2–11* shows part of the hierarchy and how the higher level accounts classes combine to give lower level classes. It also shows the wide scope of the accounts pattern. This not only covers the more traditional call/notice deposit and stock depot accounts, it also covers investment portfolios and foreign exchange (fx) trading books.

Figure MA2–11 Accounts super–sub-class hierarchy



There is a certain irony in the fact that the scope of the orders/exchanges pattern includes the accounts pattern. Accountants involved in the specification of some of the first generation of international banking systems stretched the account pattern almost to the breaking point. They tried to fit everything, including the order/exchanges pattern, into it.

For example, in one system they created new 'currencies' to accommodate foreign exchange deals (where a sum of money in one currency is exchanged for a sum of money in another). For a US$–Deutsche Mark foreign exchange deal, they would create an accounting movement in a US$–Deutsche Mark 'currency'. This had the advantage of making it easier to fit a foreign exchange deal into the accounting movement pattern. It was soon found that the disadvantages of distorting the deal to fit the accounting mould more than outweighed the advantages, and the 'general' accounting pattern was dropped.

However, this re-engineering shows that the accountants' belief in a general pattern underlying the deals is correct. Unfortunately for them, it is not their accounting transaction/movement pattern!

2.9 Fitting the business into the current accounting paradigm

The accounting transaction pattern is a partial view (more correctly, two partial, distorted views) of the money element of the overall transaction pattern. This is typical of a paradigm based on paper and ink technology. We saw something similar when, in *MW3— Re-Engineering Bank Address*, we re-engineered the address pattern. It was also a partial—and distorted—view.

Being restricted to two partial views creates problems. It is difficult, for instance, to give a full rounded picture. Bookkeepers often massage the chart of accounts so that they can fit more into the two views. For example, they create extra accounts, which do not reflect anything directly. They justify the particular rules they use for generating these accounts and their accounting movements by the way they result in final reports that give 'a true and fair view' of the business. Whether the accounting movements actually reflect the business accurately is often not considered. In this environment, it is not surprising that a number of different accounting practices arise—such as the trade and value date accounting methods mentioned earlier. Without the criteria of reflecting the business accurately and directly, it is impossible to arrive at a definitive accounting practice.

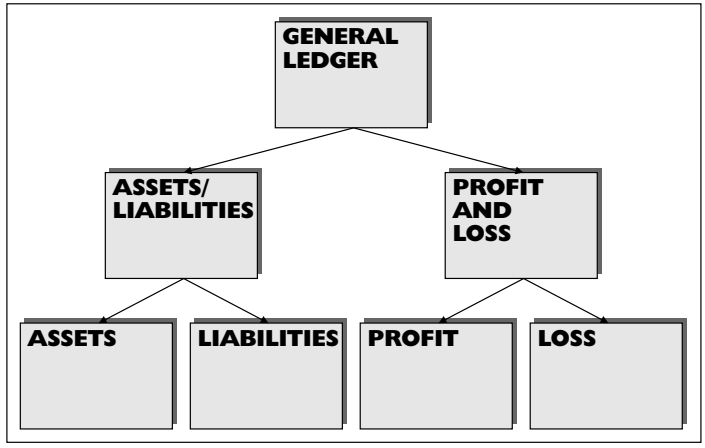
We saw something similar happening in *OP1—Entity Ontology Paradigm*. There we looked at how the entity paradigm was simplified to work within paper and ink technology. We saw how this confused its semantics so that it was no longer able to reflect the real world directly. As a result, people made decisions on whether to use an entity or attribute sign based purely on which made the information processing more effective. Whether the sign directly referred to an entity or attribute in the business was not considered.

3 Accounting's ledger hierarchy

It is not just the underlying accounting movement pattern that is constrained by paper and ink technology. All the patterns in the current accounting paradigm

are. Another good example is the ledger balance hierarchy. This is a hierarchy of the balances created by the debits and credits posted to the ledger book. It is traditionally a structure similar to that shown in *Figure MA2–12*.

Figure MA2–12 The traditional structure of the ledger hierarchy



This is a tree structure—much like the secondary substance hierarchy (shown in OP2's *Figure OP2–15* and *Figure OP2–16*). As we discussed in *OP2—Substance Ontology Paradigm* and *OP3—Logical Ontology Paradigm*, a tree structure means that the hierarchy is constrained (see OP3's *Figure OP3–23*). Classification schemes that reflect the world directly, such as the super–sub-class hierarchy, have a less constrained lattice structure. This ledger hierarchy needs to be liberated from its tree structure constraints by re-engineering.

4 Developing a new business object-oriented accounting paradigm

While companies manage their businesses using paper reports, the current paperbound accounting paradigm will have a use. However, when information is routinely supplied electronically, things should change. There will no longer be any technological reason for supplying managers with partial and distorted views of their business.

To see this, consider a business that is building a computer system based on the general orders/exchanges pattern. At a meeting, the requirement to produce standard accounting reports is raised. The system designer suggests that this is done by taking a partial view of the orders/exchanges pattern and using it to generate the traditional accounting entries. He or she explains that these can then be processed in the traditional way to produce the daily journal, balance sheet and other accounting reports. The business modeller then asks what business objects this new information reflects. The answer is that they do not directly reflect anything.

This raises the question—why should managers use this distorted accounting information? Furthermore, why should they be restricted to two views? Shouldn't they be given a multiplicity of views over undistorted information? They should, and this is why the current accounting paradigm needs a thorough re-engineering. When this is done, managers will have undistorted information.

However, this re-engineering will be a substantial task. The shift to the general orders/exchanges pattern described earlier is only a small part of it. We not only have to re-engineer the foundational accounting transaction/movement pattern—as we have started to do here; but we also have to re-engineer the patterns built from it, such as the ledger hierarchy.

In the re-engineered accounting paradigm, complex notions such as assets, liabilities, profit and loss will be transformed. The new paradigm will use the transformed notions to give a more accurate, more relevant vision of the business. However, to re-engineer these requires a thorough knowledge of accounting. We will not find the insights that we need in the entity formats of computer systems. We need to look at the conceptual patterns of people who understand accounting in depth.

Undoubtedly, when businesses start using the new accounting paradigm there will be resistance. Though some people will welcome the new paradigm, others will oppose it. There was a similar reaction when computers were first introduced. Like computers, the superiority of the new paradigm will ensure that, in the long run, it establishes itself.

5 Industrialising information

The transformation of business paradigms (of which the accounting paradigm is an example) is going to be a vital part of an overall industrialisation of information. This industrialisation will need new skills applied to new standards of accuracy. I expect that a new profession of information engineers will need to be created to do this.

This is a direct parallel with the rise of a 'physical' engineering profession in the 18th century Industrial Revolution. The two revolutions are similar in many ways. *AS4—Focusing on the Things in the Business* compared the physical accuracy that drove the Industrial Revolution's development of interchangeable parts with the conceptual accuracy that is driving the current development of general, re-usable business objects.

5.1 The rise of an engineering profession

Not only was the 'physical' engineer instrumental in making the Industrial Revolution, it can equally be said that the Industrial Revolution created the modern engineering profession. As the revolution emerged, it demanded new technical skills. Ones that were not taught to the pre-revolution craftsmen in their craftshops. When it became apparent that these skills could be codified, it was also realised that the best way to learn them was a formal technical training. Originally this was provided in military academies, but eventually established universities followed suit. This formal training set the 'engineers' apart, and from this, the engineering profession naturally developed.

We can see a similar pattern emerging in business modelling. Currently business modelling is a craft. Modellers are not given much, if any, formal training. They are certainly not given any training in information paradigms and how they work. Most of them are recruited from the ranks of programmers and system analysts. Some are recruited from the operational parts of the business. For all practical purposes, business modelling can be considered a craft carried out by craftsmen.

Business objects require information 'engineers' with a more professional technical training. For a start (as this book has shown), they need to be able to see and model business objects with a high degree of accuracy. For this, they need a good understanding of what they are. This is sufficient for the simple re-engineering of entity formats in existing systems. But, to take advantage of business objects' flexibility to handle far more powerful patterns, information engineers will need to re-engineer the conceptual patterns of experts in the business. This will involve either training the experts in business objects or, more likely, the information engineers developing a deep understanding of the experts' conceptual patterns. In other words, the engineers will have to become business experts.

Business analysts already have to develop a good understanding of the business to do their job. This is typically learnt in a similar informal way to the pre-revolution engineer–craftsmen. Information engineers will need a much deeper understanding. Formal technical training will be the simplest way for them to develop the required in-depth knowledge of the business.

5.2 Where will the information engineers come from?

If information engineering follows the same path as physical engineering, then we can expect information engineering professions to emerge. An interesting question is—where will they come from? One obvious source is the current computer system developers. This will inevitably lead to a segregation of information engineering from the rest of system building. I realised quite early on that this is a natural divide. Business modelling and computer system design are very different kinds of activities.

When we were developing our re-engineering approach, the team distinguished between It (big I for information, small t for technology) and iT (small i, big T). Business modelling was It; computer system design was iT (interestingly, some years ago the Post Office renamed its IT department iT). If you think about it, the scope of most IT department's work does not include all the kinds of information technology used by the business. Information technology covers more than computers; it includes paper and ink technology as well as human brains. What most

IT departments deal in is computing technology. So it is by no means a foregone conclusion that IT departments will supply the professional information engineers of the future.

IT people are not the only candidates for information engineers. Areas of the business that have traditionally belonged to powerful professions will need the information engineer's skills. For example, matters relating to the accounting paradigm will have to be decided by people with information engineering skills. The accounting profession is unlikely to want ex-computer people to take over this task. To make sure that they do not 'miss the boat', accountants will have to develop the information engineering skills needed to manage the re-engineering of the accounting paradigm. This is more of an opportunity than a threat. The re-engineered paradigm will give accountants much more powerful tools than they currently have, with which to help managers run their businesses.

Wherever the professional information engineers are drawn from, whether it is the ranks of accountants, lawyers, systems analysts or business people, the roles that people play within businesses will change. New responsibilities will arise from the industrialisation of information. Old responsibilities will change or become irrelevant. Undoubtedly, when these responsibilities are shared out, information engineers will not only have some of the new responsibilities, but also take over some currently held by other professionals.

6 21st century information industries

Re-engineering entity oriented legacy systems into simpler and better systems is interesting and useful work. But it is using business objects to re-engineer the business that really offers really exciting opportunities for information engineers. It will put them at the centre of a business and social revolution, where they will be shaping the future. They will help change the way businesses work, shifting them from the paper-bound information processing institutions of the 20th century into the industrialised information industries of the 21st century.

BORO Working Papers - Bibliography

The BORO Working Papers

Volume A
A—The BORO Approach

Book AS
AS—The BORO Approach: Strategy

AS1—*An Overview of the Strategy*
AS2—*Using Objects to Reflect the Business Accurately*
AS3—*What and How we Re-engineer*
AS4—*Focusing on the Things in the Business*

Volume - O
O—ONTOLOGY Papers

Book - OP
OP—Ontology: Paradigms

OP1—*Entity Ontology Paradigm*
OP2—*Substance Ontology Paradigm*
OP3—*Logical Ontology Paradigm*
OP4—*Business Object Ontology Paradigm*

Volume - B
B—Business Ontology

Book - BO
BO—Business Ontology: Overview

BO1—*Business Ontology - Some Core Concepts*

Book - BG
BG—Business Ontology: Graphical Notation
Constructing Signs for Business Objects

Graphical Notation I
BG1— *Constructing Signs for Business Objects*
Graphical Notation II
BG2— *Constructing Signs for Business Objects' Patterns*

Volume - M
M—The BORO Re-Engineering Methodology

Book - MO
MO—The BORO Re-Engineering Methodology: Overview

MO1—*The BORO Approach to Re-Engineering Ontologies*

Book - MW
MW—The BORO Methodology: Worked Examples

Worked Example 1
MW1—*Re-Engineering Country*
Worked Example 2
MW2—*Re-Engineering Region*
Worked Example 3
MW3— *Re-Engineering Bank Address*
Worked Example 4
MW4—*Re-Engineering Time*

Book - MA
MA—The BORO Re-Engineering Methodology: Applications

MA1—*Starting a Re-Engineering Project*
MA2—*Using Business Objects to Re-engineer the Business*

Book - MC
MC—The BORO Re-Engineering Methodology: Case Histories

Case History 1
MC1—*What is Pump Facility PF101?*

USING BUSINESS OBJECTS TO RE-ENGINEER THE BUSINESS

A–I

A

B

C

D

E

F

I

M

P

R

S