

IV ERダイアグラム

1. エンティティ

(1) エンティティ・オカレンス・ID

DOAの基礎技法として、「データ」についての要件をモデル化するデータモデリング技法において扱う「エンティティ」について説明します。

エンティティとは、特定の目的に有効な情報となるように関連づけられたデータ項目のグループのことです。エンティティ及びデータ項目の例を以下に記載します。

例1)

エンティティ	データ項目
従業員	従業員番号、従業員名、部門コード 職種コード、入社年月日
部門	部門コード、部門名
職種	職種コード、職種名
技能	技能コード、技能名
保有技能	従業員番号、技能コード、経験年数、技能レベル

例2)

エンティティ	データ項目
従業員	従業員番号、従業員名、住所、電話番号、 職能資格、・・・・・・・・
店	店コード、店名、住所、電話番号、 総売場面積、・・・・・・・・
受注	受注日、受注商品、受注数量、納入期限、 納入先

エンティティによって表現される具体的な情報のことをオカレンスといいます。
 データ項目はエンティティのオカレンスを表現する属性となるので、エンティティを構成する
 データ項目を、エンティティの属性といいます。

例2の従業員のエンティティより、例3としてエンティティの属性、オカレンスの例を記載
 します。

例3)

エンティティ「従業員」

属 性	従業員コード	氏 名	住 所	電話番号	・ ・	
属性の値によって 表現されるオカレ ンス	001	中村 良夫	神奈川県 …	なし		オカレンス
	002	奥山 一藤	長野県 …	111(111)1111		オカレンス
	003	串田 恭子	神奈川県 …	222(222)2222		オカレンス

エンティティには、必ず複数の属性が存在します。一つの属性しか存在しない場合でもそれはエンティティではなく属性と考えられます。まったく属性が存在し得ない場合、それはエンティティでも属性でもないはずで

す。エンティティのオカレンスを特定することの出来る属性をID（アイデンティファイア）キーあるいは識別子と呼びます。通常、エンティティの属性のうちIDとなるのは属性一つとは限りません。IDが複数存在する場合や、IDが複数の属性によって構成される場合（連結ID）があります。

例4)

	エンティティ	IDとなる属性
単一の属性で構成されるID	従業員	従業員コード
連結ID	口座	銀行番号+支店番号
複数のID	図書	書名+著者名+出版社名

(2) エンティティの見つけ方

エンティティを見つけるときは、物理的な処理手順にとらわれることなく、「何をしたいのか」「何に関心があるのか」といった視点で分析する必要があります。

エンティティの見つけ方（データ分析）には、トップダウン・アプローチ、ボトムアップ・アプローチ、トップダウン+ボトムアップ・アプローチの3種類の方法が存在します。

① トップダウン・アプローチ

ユーザのニーズ、対象業務領域の範囲に関する概念的な情報（キーワード）を基に、ブレインストーミングを通じて管理対象となる情報を洗い出し、それらのオカレンス、属性、IDを検討することによってエンティティを定義していく方法です。

簡単に説明すると、問題記述から名詞の抽出を行い、誤った抽出を排除することによりエンティティを見つけ出せるということです。「いつ、どこで、だれが、何を、どれだけ・・・」というようにして抽出を行います。

トップダウン・アプローチによるエンティティ抽出の例を記載します。

例5)

「Aさんが1月9日に、東京秋葉原店でノート型パソコンを一台カードで購入した。」

WHO : Aさん

WHEN: 1月9日

WHAT: ノート型パソコン

顧客の情報
氏名
住所
電話番号
生年月日
:
:

営業日の情報
曜日
気候
イベント
:
:
:

商品の情報
商品名
販売単価
仕入単価
仕入先
:
:

WHERE: 東京秋葉原店

HOW:カード(金に変わる手段)の情報

店の情報
所在地
電話番号
地域特性
:
:
:

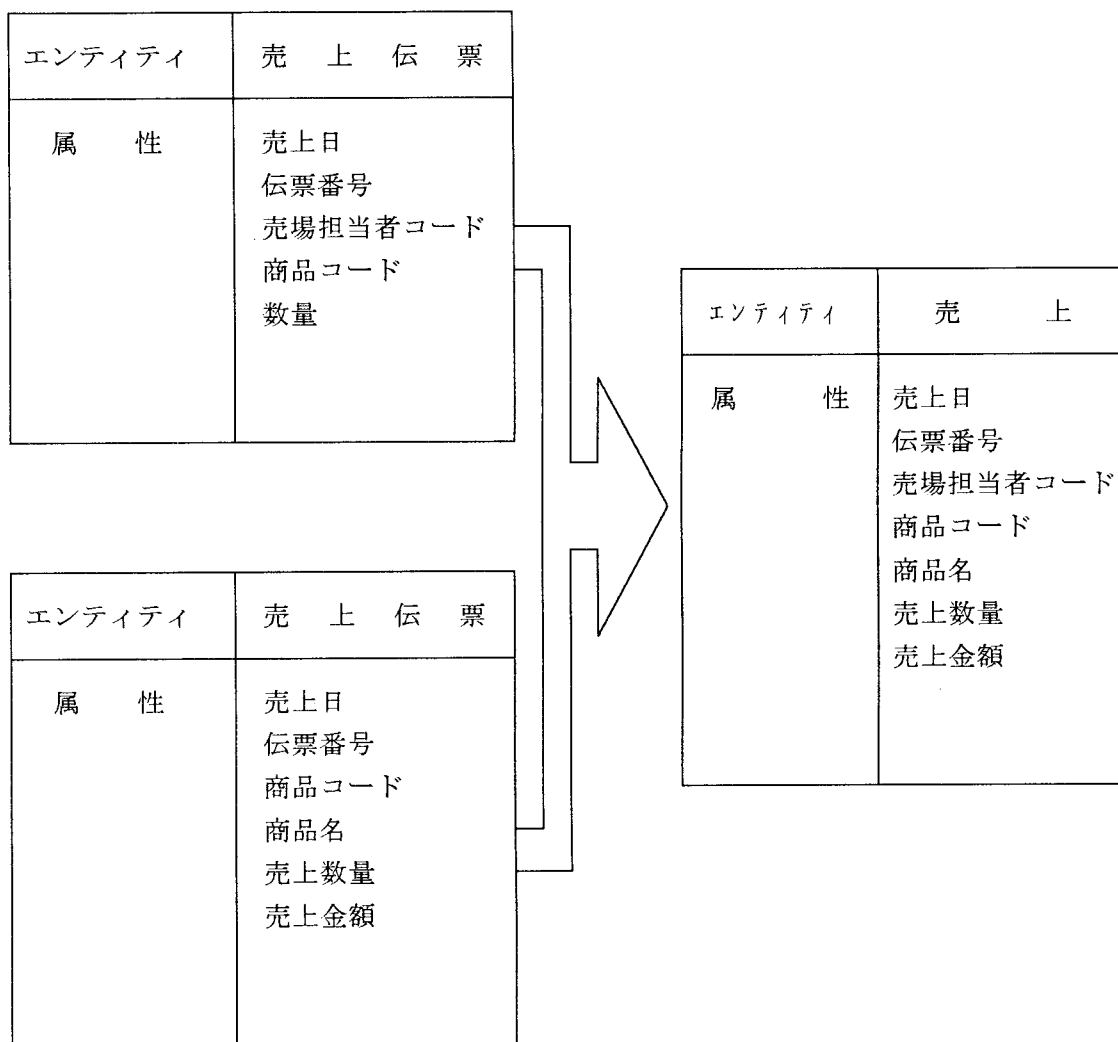
カードの情報
カード番号
金融機関口座
有効期限
:
:
:

② ボトムアップ・アプローチ

現在、帳票やファイルなどで管理しているデータ項目及び今後必要なデータ項目を洗い出し、それらのグループ化の作業を通じてエンティティを定義していく方法です。

ボトムアップ・アプローチによるエンティティ抽出の例を記載します。

例6)



以上までに説明した二つのアプローチにおいて、前者（トップダウン・アプローチ）は、理想型の概念データモデルを作成するのが狙いのときに、また後者（ボトムアップ・アプローチ）は、現実型の概念データモデルを作成するのが狙いのときにそれぞれ適しています。

③ トップダウン+ボトムアップ・アプローチ

トップダウン・アプローチの全体の整理のしやすさ、ボトムアップ・アプローチの精度の高さの両者を採用した手法が、トップダウン+ボトムアップ・アプローチです。

トップダウン・アプローチでおおまかなエンティティを定義し、平行してボトムアップ・アプローチでデータ項目を洗い出しておきます。エンティティとデータ項目が、ある程度出そろったところで、データ項目を属性として適切だと思われるエンティティに対応づけ、エンティティの属性として定義を行います。

①から③の3種類の手法を通じて、エンティティとその属性が洗い出され、データモデルの概略ができあがります。

その後、正規化などの手法によってデータの性質を分析することで、データの本質的な分類を明確にし、冗長性のないデータの管理ができるデータ構造になるように、データモデルを改訂していきます。

2. エンティティ関連ダイアグラム

(1) エンティティ関連ダイアグラム：ER図

これまではエンティティを1つの独立した単位としてとらえてきました。しかし正規化によって1つのエンティティが複数のエンティティに分解されていく過程をみてわかるように、エンティティ同士はまったく独立しているのではなく互いに関係があります。このようなエンティティ間の関連を表現した図をエンティティ関連ダイアグラム（ERD：Entity Relationship Diagram）・ER図といいます。

二つのエンティティ間の関連（リレーションシップ）には名前が付けられます。また名前は、双方向につけられることが多いです。

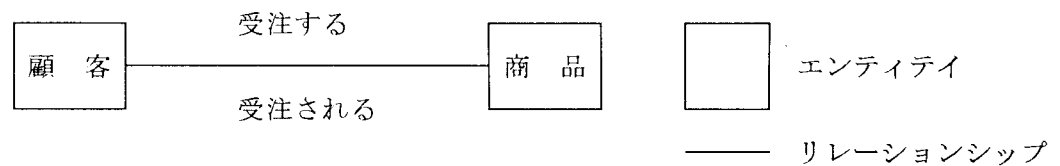


図9 ER図

上記の図9は、次のことを表現しています。

- ・「顧客」は「商品」を受注する。
- ・「商品」は「顧客」から受注される。

リレーションシップの名称は動詞で表現します。

3. リレーションシップ

(1) M : Mのリレーションシップ

正規化されたエンティティ間に「M : M」の関係がある場合、重要な情報が喪失している可能性があります。

例えば、M : Mの関係を持つエンティティ、「講師」と「科目」について見てみましょう。このままでは、「どの講師がどの科目をいつ、どの教室で教えているか」という情報が喪失していることに気付くでしょう。これらの情報を、どのエンティティの属性としたらよいのでしょうか？ どちらのエンティティの属性にすることも可能ですが、正規化された状態ではなくなってしまいます。

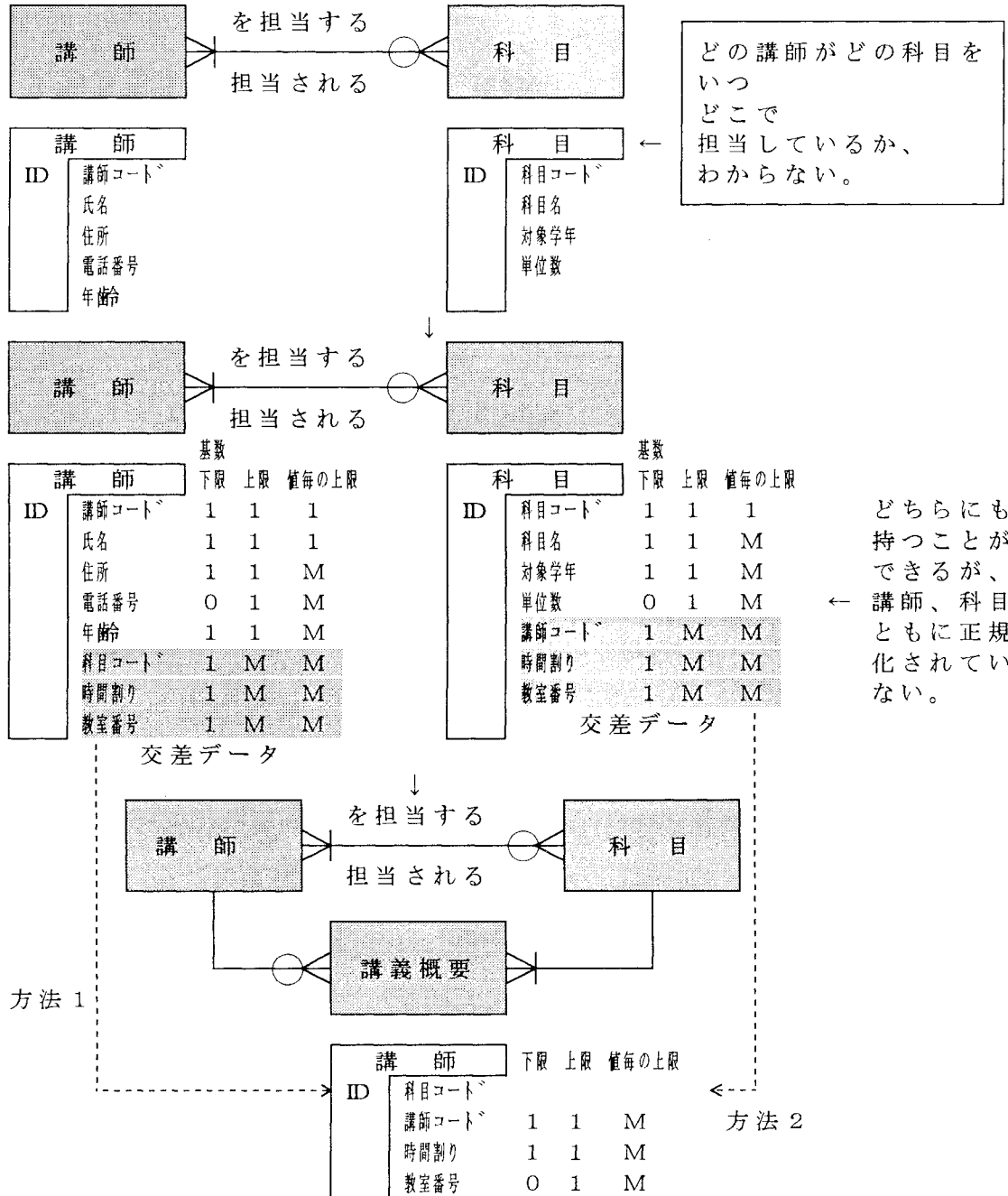


図 10 M : Mのリレーションシップ

このような、どちらのエンティティの属性にもなりうるデータを交差データ項目ということにします。次のように、交差データ項目をどちらかのエンティティの属性とすると、そのエンティティは非正規形の状態となるので、交差データ項目群を別のエンティティとして定義します。（第1正規化）。

- ① 「講師」に交差データ項目群を定義した場合、それらの上限はMになるので、それらを別のエンティティ、「講義概要」の属性として定義することができます。（第1正規化）
- ② 「科目」交差データ項目群を定義した場合も同様です。

(2) 複数のリレーションシップ

二つのエンティティ間には、二つ以上のリレーションシップが存在する可能性もあります。この場合、リレーションシップによる関連づけの目的の相違が明確になるように、それぞれのリレーションシップの名称を定義します。

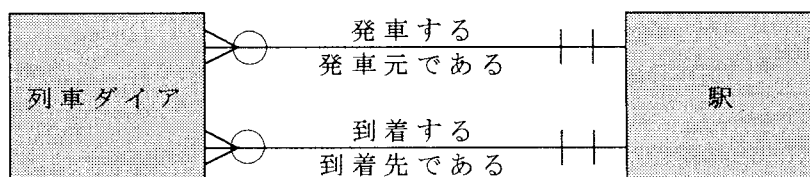


図 1 1 複数のリレーションシップ

(3) リレーションシップの冗長性排除

エンティティ間に同じ目的のリレーションシップが複数存在する場合、リレーションシップに冗長性があります。余分なリレーションシップを削除して冗長性を排除します。

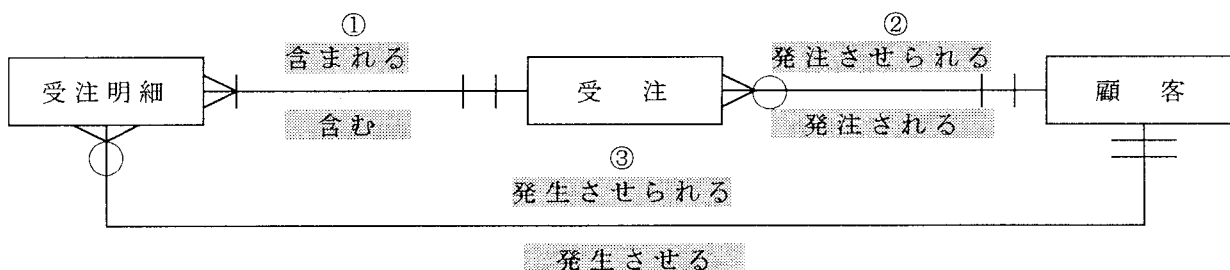


図 1 2 リレーションシップの冗長性排除

- a) ②と③はリレーションシップの名称・基数ともに同じ（類似）である。
- b) ①、②のリレーションシップを通じて、エンティティ「受注明細」に関連する「顧客」の情報を一意に特定することができる。
- c) ①、②ともに必要なリレーションシップである。

以上の点から、③が冗長性のあるリレーションシップとして排除することができます。

(4) 同一エンティティ間のリレーションシップ

リレーションシップは二つのエンティティ間だけでなく、同一エンティティ間にも存在することがあります。

エンティティ「組織」において、
「組織」の1オカレンス『営業部』は、「組織」のオカレンス『営業第1課』、『営業第2課』、『営業第3課』を統括する。
「組織」のオカレンス『営業第1課』は、「組織」のオカレンス『営業部』に統括される。というように、1つのエンティティにおいて、あるオカレンスと他のオカレンスとの間に主従関係を持つような場合が例としてあげられます。

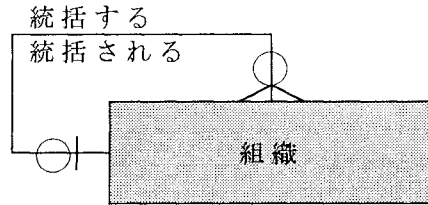


図 1 3 同一エンティティ間のリレーションシップ

(5) 長期的な視点の反映

エンティティ間のリレーションシップの基数は、時間の推移とともに変化します。

すなわち、短氣的視点に立った場合と長期的視点に立った場合とでは、リレーションシップの上限・下限が異なる場合が出てきます。リレーションシップは長期的に見て一貫性安定性が保たれている必要がありますが、どちらの視点から見た方がより関心が高いかを考えて、適切なリレーションシップを選択してください。

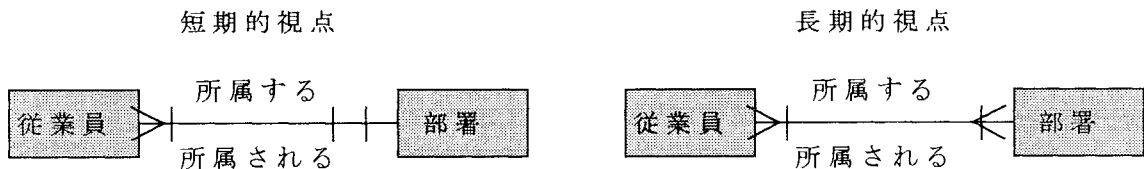


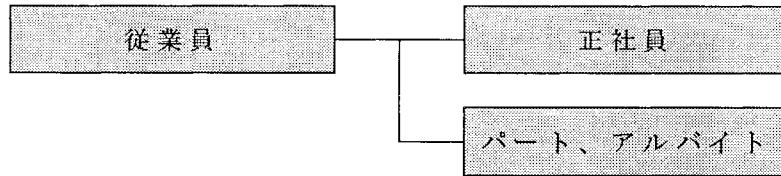
図 1 4 長・短期的視点のリレーションシップ

通常、従業員は1つの部署にのみ所属します。しかし、長期的にみると複数部署の業務というのもあります。

(6) スーパータイプ、サブタイプ

あるエンティティの中の特定の性質を持つ属性群を、別々の下位概念のエンティティとして定義した場合、これらのエンティティを「サブタイプ」、元の上位概念のエンティティを「スーパータイプ」といいます。スーパータイプとサブタイプとは親子関係にあります。

これによって、情報の管理目的がより細分化され、明確になります。スーパータイプ、サブタイプに共通の属性群はスーパータイプの属性として定義し、サブタイプ特有の属性群がサブタイプの属性として定義します。



従業員	下限	上限	値毎の上限	
従業員氏名コード	1	1	1	← 共通の属性
従業員氏名	1	1	M	← 共通の属性
住所	1	1	M	← 共通の属性
生年月日	1	1	M	← 共通の属性
健康保険証番号	0	1	1	← 正社員特有の属性
職能資格	0	1	M	← 正社員特有の属性
勤労時間帯	0	1	M	← パート、アルバイト特有の属性
時間給	0	1	M	← パート、アルバイト特有の属性

従業員	下限	上限	値毎の上限	
従業員氏名コード	1	1	1	
従業員氏名	1	1	M	
住所	1	1	M	
生年月日	1	1	M	
健康保険証番号	0	1	1	
職能資格	0	1	M	
就労時間帯	0	1	M	
時間給	0	1	M	

正社員	下限	上限	値毎の上限	
従業員氏名コード	1	1	1	
健康保険証番号	1	1	1	
職能資格	1	1	M	

パート、アルバイト	下限	上限	値毎の上限	
従業員氏名コード	1	1	1	
就労時間帯	1	1	M	
時間給	1	1	M	

図 1 5 スーパータイプ、サブタイプ

スーパータイプに定義された属性群は、各サブタイプにも共通の属性群ですから、スーパータイプとサブタイプの間で定義されるリレーションシップ（関係）は、それらの属性群が各サブタイプにも継承されることを意味しています。

エンティティ記述で次のような点をチェックすることにより、サブタイプを見つけることができます。

- a) 下限 0 の属性に注目します。
- b) 下限 0 になる場合の法則性を発見します。
- c) その法則性が情報管理上、重要な意味を持っている場合、下限 0 の属性群をサブタイプのエンティティとして定義します。

(7) 広い視野によるモデリング

もともと別のエンティティとして定義したが、それぞれ同じオカレンスが存在するような場合、広い視野で考えると、それらを1つのエンティティとしてまとめて定義することもできます。

そして、まとめたものをスーパータイプ、もともと別々のエンティティをサブタイプとすることができます。

これを「汎化」といいます。

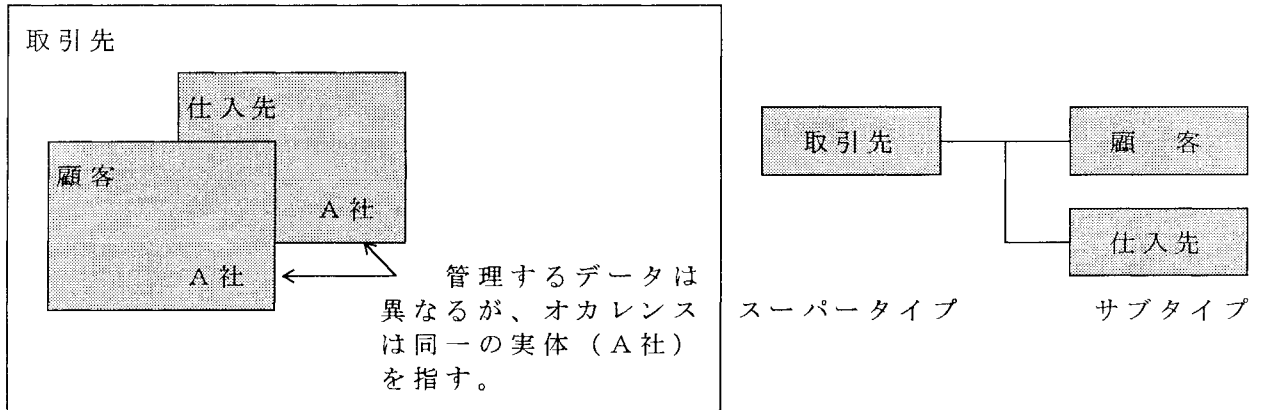


図 1 6 広い視野におけるモデリング

(8) データモデルとしてのERD

ERDは関係型データベースの設計モデル(関係データモデル)としてよく利用されているが、階層型データベースや順編成ファイルの設計にも利用することができます。

ERDは、ファイル編成に依存せず、必要な「データ要素」をモデル化したものなのです。