

# FileMaker 7のキーコンセプト 新たな領域の地図を描く

マイケル・ハリス

Cerné Systems, Inc. シニアパートナー

FileMaker 7は天才の発明かもしれません。時がたてば分かるでしょう。天才の特徴を備えているのは間違いありません—シンプルさと、複雑さと。問題は複雑さの方ですが、ほとんどの人の場合、FileMaker 7は使いこむと印象がガラリと変わります。シンプルさを楽しめるのは少し先へ進んでから、複雑なカーブをいくつか曲がった後というわけです。

本稿では、FileMaker 7を使いこなす上での最大の難所、新しくなったリレーションシップ構造を取り上げます。この構造は、要するにゲシュタルトだと考えてください。さまざまな新しいコンセプトが組み合わさって機能していく、そのパターンを観察することで理解に近づくしかありません。みなさんは、部分的な理解をいくつも積み重ねて、徐々に全体が像を結ぶという道のりをたどることになるでしょう。

この文章は、その道のりを歩むための地図です。真っ黒になるまで書き込みをして、くしゃくしゃになるまで使いこんでください。しわが多いほどみなさんの役に立ったという証拠です。FileMaker 7のキーコンセプトを「用語解説」の項で定義し、「新しい世界のルール」にまとめました。みなさんの旅の道しるべとなるでしょう。また、新しいルールは本免許取得に向けての路上試験にもなっています。ルールを完全に理解できたら、地図はダッシュボードに放り込んで、出発です。

## 従来のバージョンとの違い

FileMaker 7の構造で従来のバージョンとの最大の違いは、ファイルとテーブルの関連づけがゆるやかになったことです。従来は、ファイルとテーブルは固く結びつき、1つのファイルに1つのテーブルが大原則でした。そのため、リレーションシップ理論の奥義を扱うような場合をのぞいて、「テーブル」という言葉自体ほとんど使われませんでした。この構造の利点は、スクリプト記述やフィールド定義のときに「どのテーブルについて話しているのか」を意識しなければならないという問題がまったく発生しないことです。従来のバージョンにおけるこうした作業は、そのデータを保有しているファイルでのみ可能でした。そして、他のファイルすなわち、他のテーブルのデータにリレーション経由でなく直接アクセスするには、そのファイルに移動しなければなりませんでした。

FileMaker 7では、ファイルとテーブルの関係はゆるやかにはなりましたが、完全に切れたわけではありません。テーブルはファイルからしか作成できませんし、

データベースには必ずファイルが1つは必要です。テーブルにとっては、自分の生まれたファイル、ホームファイルこそが「わが家」です。テーブルの定義や基本的なセキュリティの管理はホームファイルでしかできません。

一方、それ以外の場面では、ファイルとテーブルの結合は解消されました。1つのファイルが使用できるテーブルには制限がなくなり、他のファイル、とくにリモート位置にある他ファイルのテーブルも使えるようになりました。その代わりに、FileMakerに対してあちこちで「どのテーブルについて話しているのか」を明らかにしなくてはなりません。具体的にはレイアウト、スクリプト、フィールド定義などで毎回、リレーションシップグラフ上のいずれかのテーブルオカレンス (TO) を指定して、参照テーブルを明確にします (図3a)。

ほとんどの人の場合、FileMaker 7は使いこむと印象がガラリと変わります

TOを指定する意味は、それだけではありません。そのTOの

リレーションシップを介して他にどんなテーブルをどんな条件で使用できるかも明らかになります。つまりFileMaker 7では1つのテーブルを他ファイルを含めたさまざまなTOから参照できますが、同じ主テーブルのデータを参照しつつ、関連テーブルはTOごとに一新できるのです。図3aの例では、「DM有権者」「テレコール有権者」「送迎有権者」という3つのテーブルオカレンスはすべて同じ主テーブルを参照していますが、アクセスできる関連テーブルはそれぞれ全く異なります。

従来のバージョンの1ファイル=1テーブル構造では、リレーションシップにはある意味で普遍性がありました。業務プロセスの中で、あるリレーションシップを使用するには、そのリレーションシップがつくられたファイル=テーブルに移動しなければなりません。逆に移動さえすれば、そのファイル内の全リレーションシップを利用できました。一方、FileMaker 7ではファイルごとにリレーションシップグラフがあり、他ファイルのテーブルを使ってリレーションシップを作成することもできます。ただし、そのリレーションシップは、そのファイルにしか適用しません。

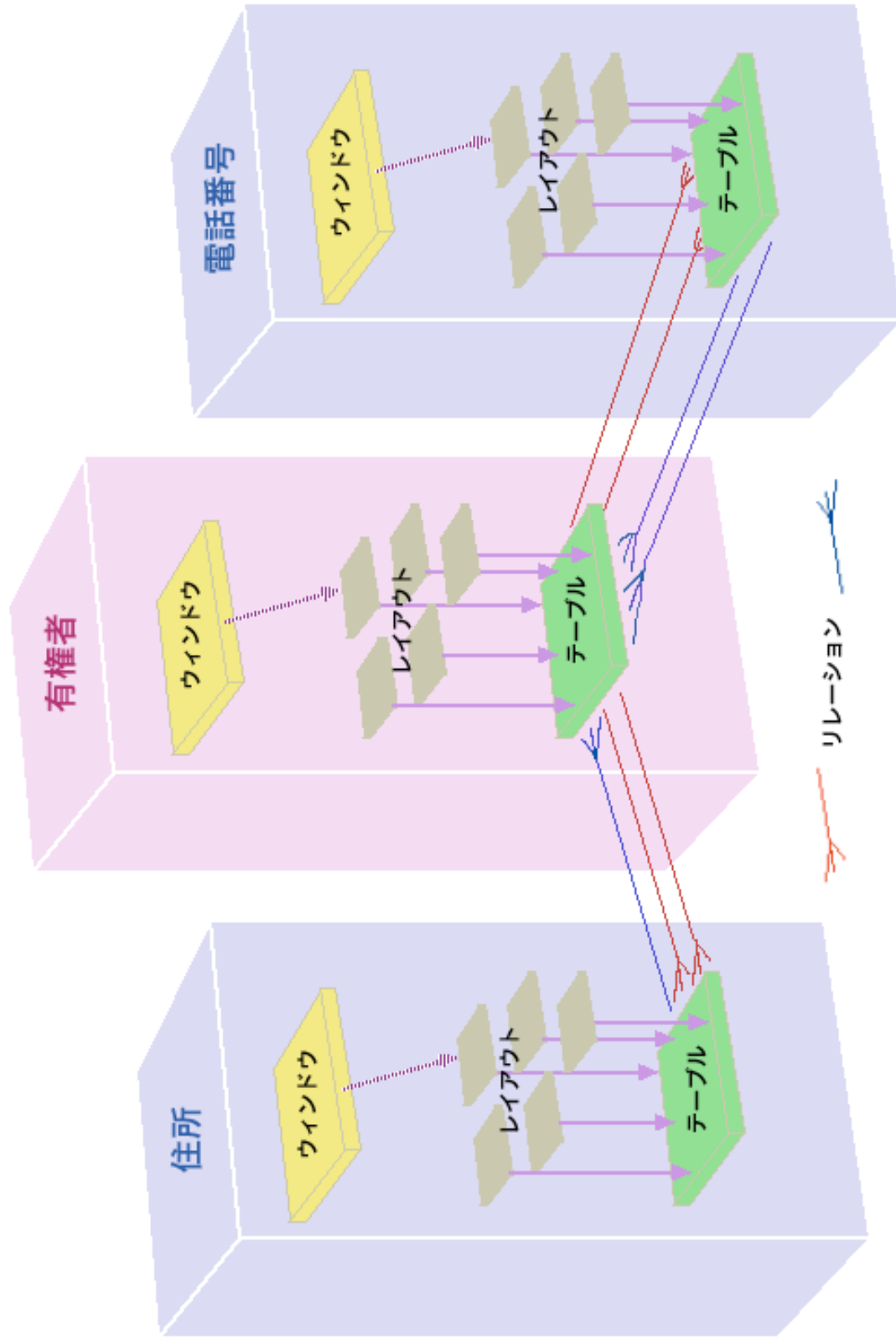


図0a FileMaker 7以前の構造。1つのファイルに対して、ウィンドウ1つ、レイアウト多数、テーブル1つ。リレーションシップはファイル同士をつなぐ一方向のオブジェクト。

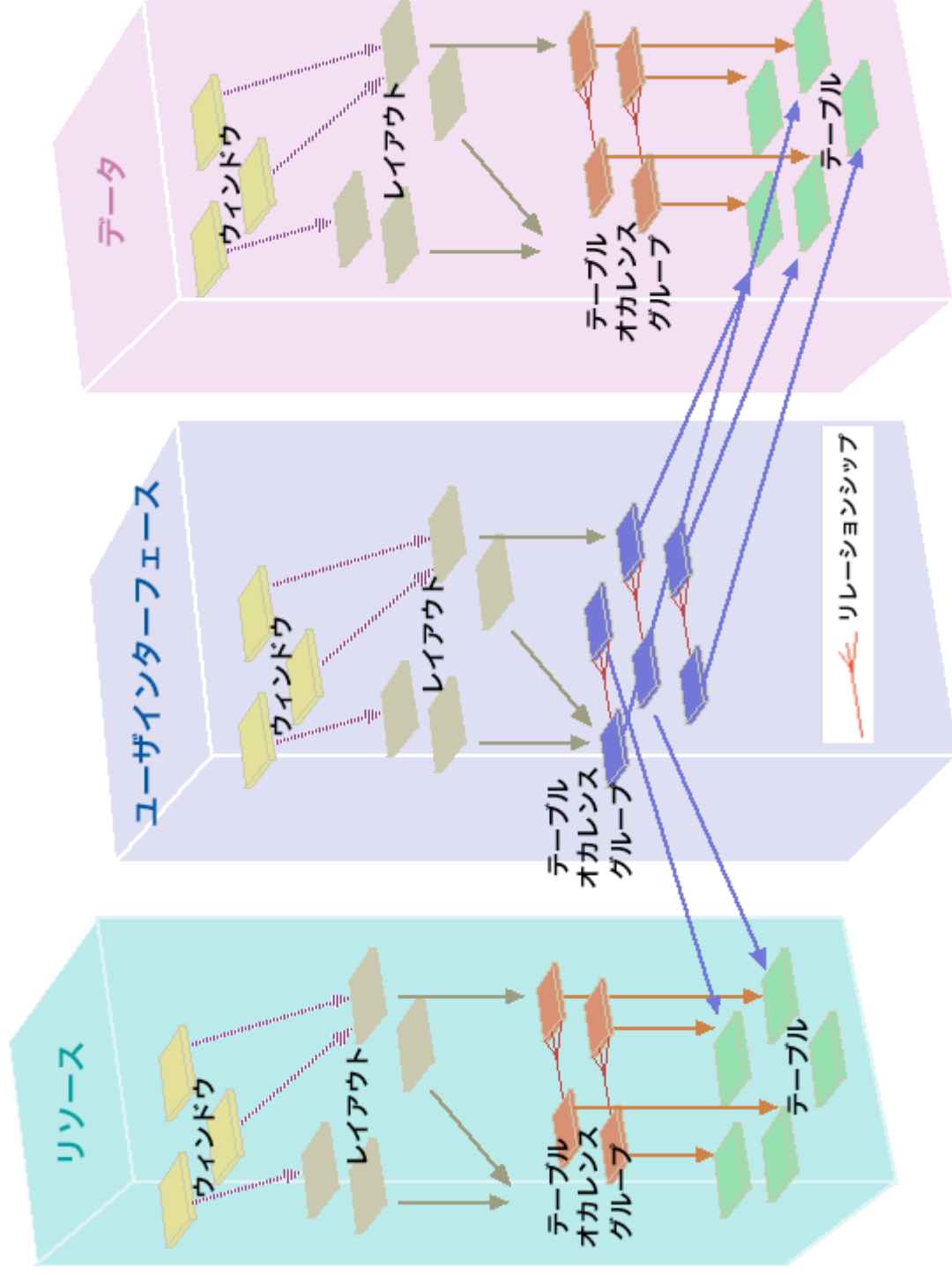


図0b FileMaker 7の構造。1つのファイルに対して、同時に多数のウィンドウ、レイアウト多数、テーブル多数、テーブルオカレンス、どのファイルのテーブルでも使用可。リレーションシップはテーブルオカレンス・グループ (TOG) と呼ばれるデータ環境についての、双方向の設定オブション。1ファイルに複数のTOG。(注0、18ページ)

たとえば、人物テーブルと住所テーブルのリレーションシップを3つのファイルで使うには、各ファイルでリレーションシップの定義が必要です。その代わり、リレーションシップの内容をファイルごとに変えられるので、踏み込んだレベルの管理が可能になります。

このようにFileMakerの構造にゆりみが生まれた結果、ユーザはコンテキストを明確にするよう求められることになりました。「何について話しているのか」というFileMakerの問いに何度も答えねばなりません。最初のうちはめまいがしたり、人によっては実存主義的嘔吐にみまわれることもあるでしょう。

## ケーススタディ

デニスは選挙に立候補することにしました。選挙運動には有権者のデータベースが必要です。運動の柱となる活動は、有権者へのダイレクトメール、テレコール、投票所への送迎の3つですが、活動を担うボランティアは素人も多く、永続的な組織でもありません。そのため、データベースはシンプルかつ堅固、しかも柔軟なものが求められます。

DM発送では、ボランティアは住所という切り口でデータを見ます。1つの住所に有権者が何人いても、DMはまとめて1通にしたいからです。1通のDMには複数の有権者の名前を印字することから、住所テーブル

は有権者テーブルに対して一对多のリレーションシップを持つことになります。次にテレコールでは、できるだけ気分を害すことのない有権者の番号を選んで電話をします。つまり切り口は有権者です。投票所への送迎では、集合場所の住所を切り口にして電話番号と有権者を見ます。電話番号テーブルは住所テーブルと直接リンクされ、有権者テーブルとは間接的につながります。

従来のバージョンでは、このデータベースにはファイルが3つ必要でした。ファイルとテーブルは不可分で、実質的には同じものだったからです。しかしFileMaker 7では、必要なテーブルの数は変わりませんが、ファイルの数は自分で決められます。このような小規模なデータベースでは、複数のファイルをつくる理由もありませんし、セキュリティなどメンテナンスの手間からいっても1つで十分でしょう。そこで「選挙」ファイルを作成します。FileMaker 7ではフィールドはファイルではなくテーブルに帰属するので、テーブルごとにフィールド定義をおこないます。また、レイアウトは3つの活動に対して1つずつ作成します。

次に、3つの用途ごとにデータ環境を構築します。作業はリレーションシップグラフ上でおこないます。

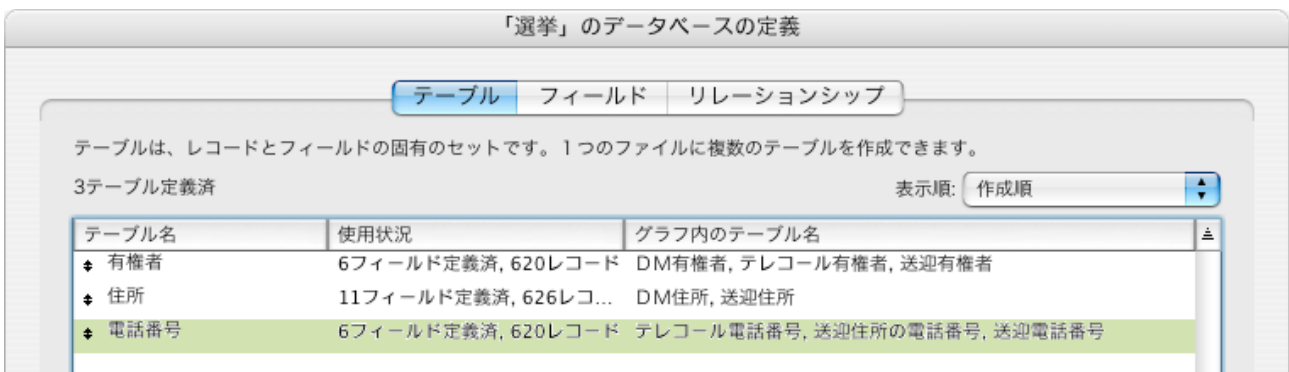


図1 1つのファイルに3つのテーブル。テーブルごとにフィールドが定義されます。

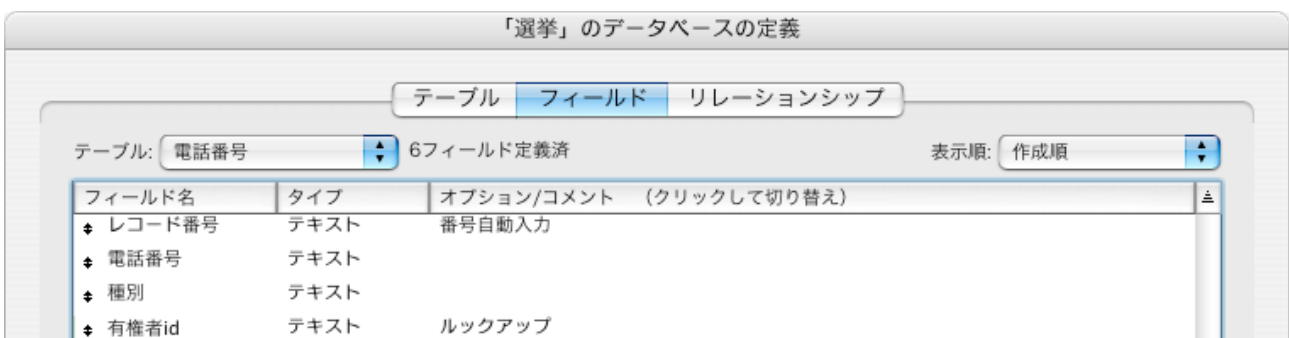


図2 「電話番号」テーブルのフィールド

リレーションシップグラフでは複数のテーブルオカレンス（TO）がリレーションシップで結ばれ、グループ1つで1セットのデータ環境をあらわしています。これをテーブルオカレンス・グループ（TOG）と呼びます。図3aは「選挙」ファイルのリレーションシップグラフです。

分かりやすいように、3つのTOGにはそれぞれ空のTOを用いてタイトルをつけ（DM、テレコール、送迎）、さらに色分けしてあります。このリレーションシップグラフを理解できれば、FileMaker 7という未知の世界への第一歩となるでしょう。

## リレーションシップグラフ

「地図は現地ではない」

—アルフレッド・コージブスキー

コージブスキーの有名な言葉は、個人の心の中にある世界と現実の世界を混同することへの警告です。もっと言えば、地図に山が描いてあるからといって、現実には存在しない山がそこにあるはずだと言い張る、そんな手合いを諷めたものです。似たような思い込みによる失敗は残念ながらだれでも経験があるでしょう。この箴言をFileMaker にあてはめるなら、すでに現実合わなくなった地図、不適切な地図をFileMaker 7に持ち込むなということです。自分がFileMakerについて何事かを知っているとんでも、一時保留にしてください。忘れるのではなく保留にする、禅の世界という「無心」が肝要です。

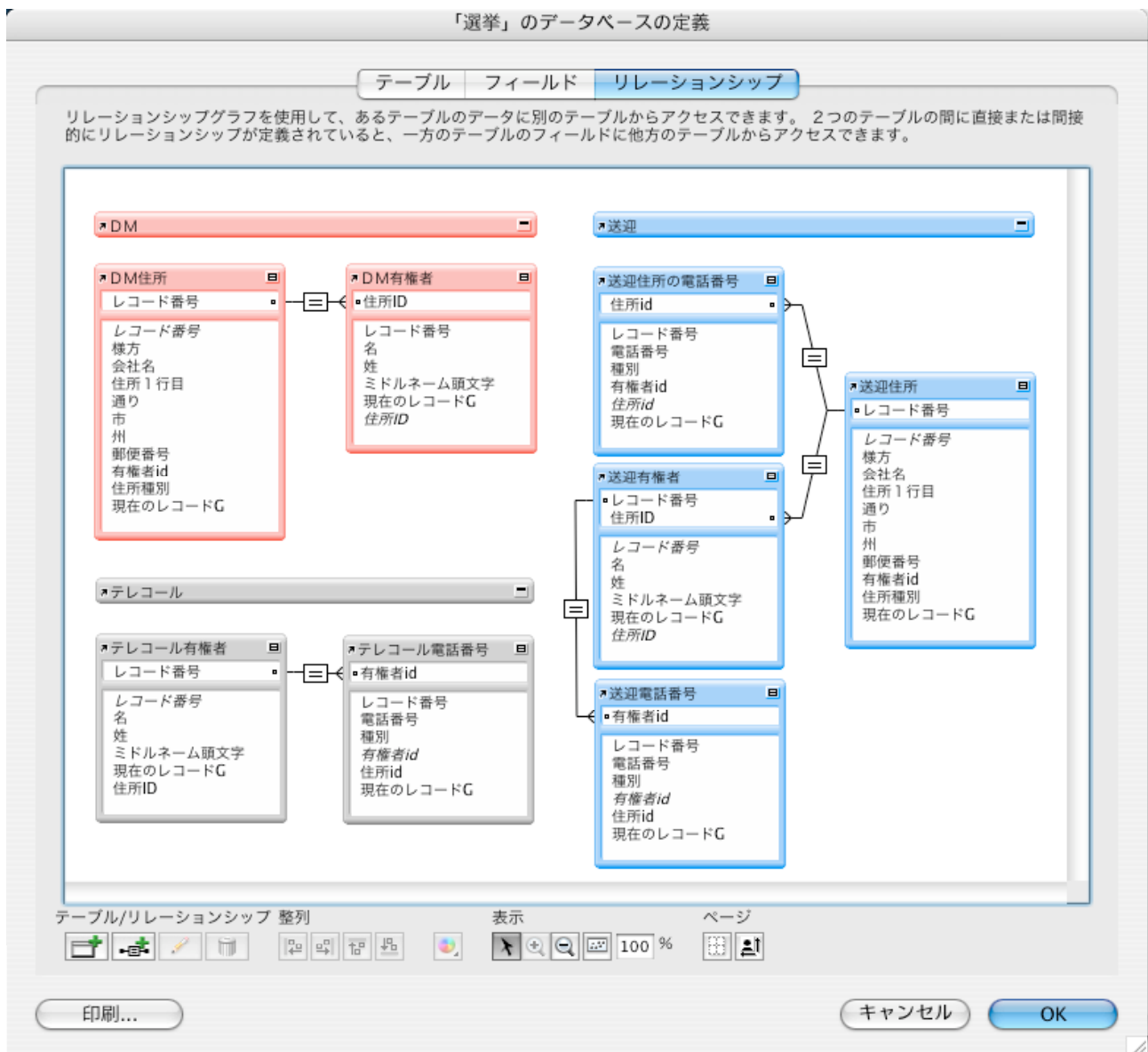


図3a 「選挙」ファイルの3つのデータ環境（TOG）

それでは、スタートとしては少々奇妙ですが、図3aのリレーションシップグラフが「～ではない」という観点から見てみましょう。

- ・ 実体間の関係をあらわすいわゆるER (entity-relationship) 図ではありません (図3b)。
- ・ TOはその主テーブルの別名ですが、テーブルと同じではありません。
- ・ リレーションシップは、従来のバージョンにあったオブジェクトとしてのリレーションシップではありません。したがって名前をつけられず、リスト表示もできません。
- ・ リレーションシップグラフはこのファイル内でのみ有効です。テーブル間に永続的、普遍的な結合をつくるわけではありません。

ER図では、テーブル (長方形のボックス) とリレーションシップ (照合フィールドを結ぶさまざまな線) を使って、データベース内のテーブル群が持つ基本的なデータ構造をあらわします。リレーションシップグラフとは異なるものです。

## リレーションシップはリレーションにあらず

FileMaker 7のリレーションシップは、これまでのファイルメーカーでリレーションと呼ばれていたものとはまったく違います。たしかに利点が多いのですが、主要なFileMaker用語がここまで大きく意味を変えると、混乱を招かざるをえません。たとえば、FileMaker 7ではリレーションシップには名前をつけられず、リスト表示もできません。最初は皆、FileMaker, Inc.が馬鹿げたミスをしでかしたのだと思いました。しかし今では、新しい構造の根本にかかわる変更であり、FileMakerの変化を象徴するものだと考えています。

それでは、説明の前にみなさん考えてみてください。FileMaker 7のリレーションシップには名前がつけられません。これは何を意味するのでしょうか。リレーションシップは、もはや「オブジェクト」ではないということです。では一体何なのでしょう。むしろ「プロパティ」であり、環境設定やセットアップなどに近いものです。リレーションシップの重要性が低下したわけではありません。以前よりも強化されたと言えるでしょう。ただ、FileMakerの構造におけるその位置づけは大きく変わりました。

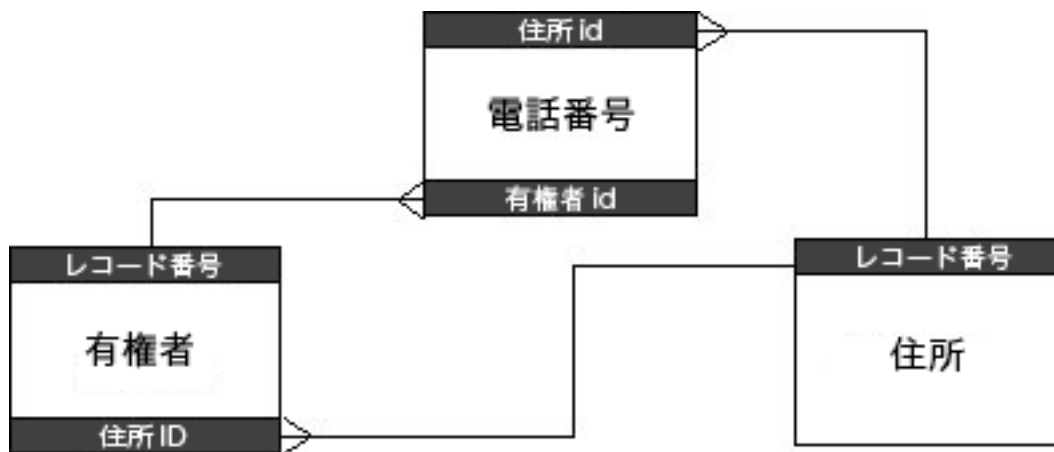


図3b 「選挙」のデータベースを使った、典型的なER図の例

リレーションシップグラフ (図3a)	ER図 (図3b)
<ul style="list-style-type: none"> <li>・ 長方形のボックスはテーブルオカレンス (TO) をあらわす。</li> <li>・ 1つのテーブルが多数のTOから参照される。</li> <li>・ すべてのTOは、同じ主テーブルを参照したものであっても、それぞれ固有の名前を持つ。</li> <li>・ リレーションシップの線は、データ構造、プロセスロジック、インターフェースなど多様なものをあらわす。</li> <li>・ リレーションシップの線は、複数の照合フィールドの組み合わせや条件を持つことがある。</li> </ul>	<ul style="list-style-type: none"> <li>・ 長方形のボックスはテーブルをあらわす。</li> <li>・ 1つのテーブルはふつう一度しか参照されない。</li> <li>・ テーブルの名前は1つしかない。</li> <li>・ リレーションシップの線は、データ構造をあらわす。</li> <li>・ リレーションシップの線は、1組の照合フィールド間をつなぎ、1つのテーブルには1つの照合フィールドしかない。</li> </ul>

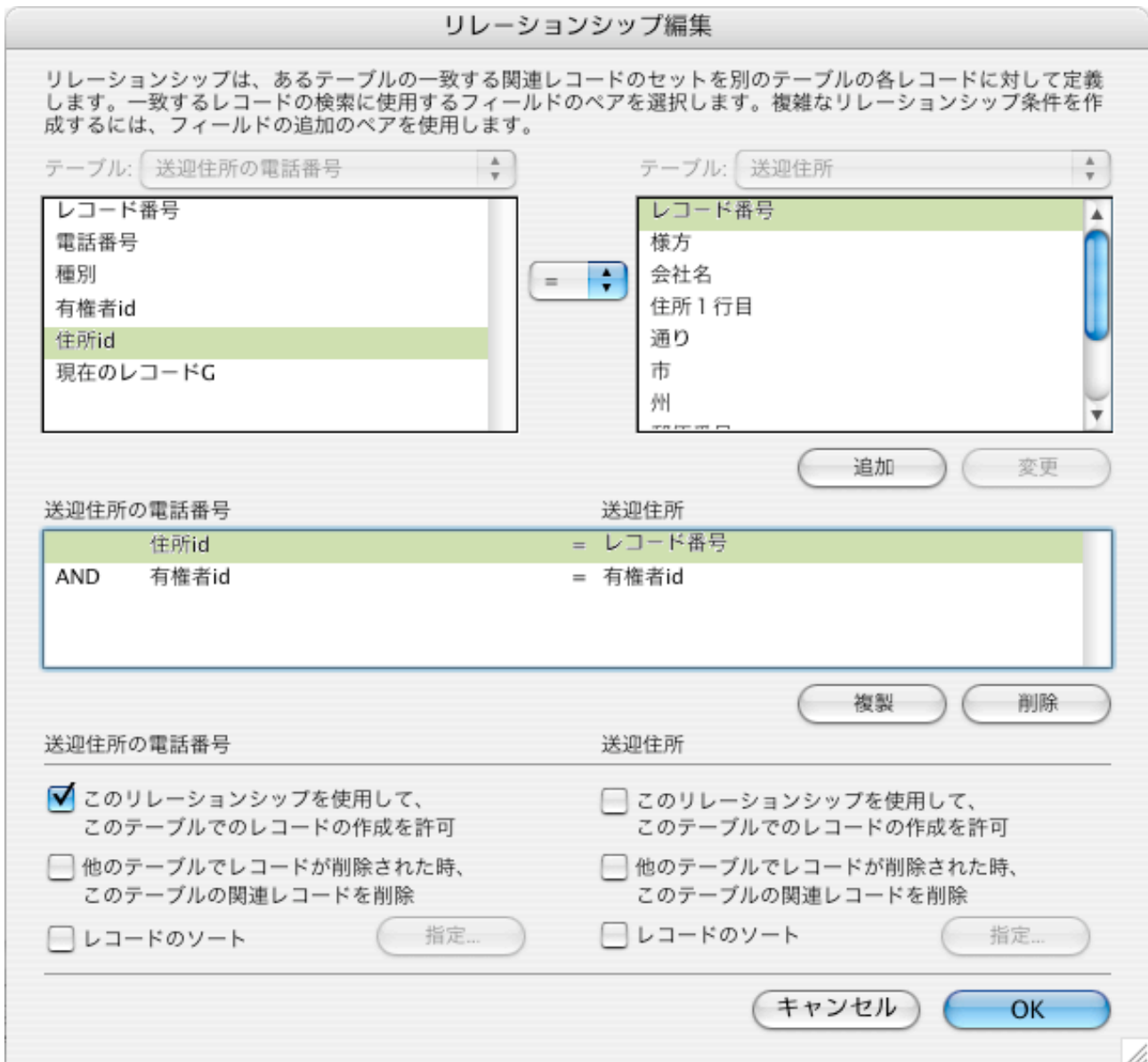


図4 この[リレーションシップ編集]ダイアログでは、「選挙」ファイルを少し変えて2つの条件を設定しています。

リレーションシップは、リレーションシップグラフ上で1つのTOから別のTOへ線をドラッグするか、[リレーションシップ編集]ダイアログで作成します。線をドラッグすると、そのTOの1つのフィールドと別のTOのフィールドが結ばれます。もし片方が番号自動入力のフィールドであれば、FileMakerは一对多のリレーションシップを想定し、線の一方の端だけを鳥の足のような形にします。場合によって線の端の形や、線の中央に表示されるボックスの記号は変わってきます。線をクリックすると[リレーションシップ編集]ダイアログが開き、リレーションシップの修正や設定のやり直しができます。図4は、「選挙」ファイルで作成したリレーションシップの編集ダイアログです。

このダイアログを見ると、FileMaker 7のリレーションシップが従来のバージョンと大きく異なる2つのポイントが分かります。第一に、複数の照合フィールドの組み合わせを使って複数の条件を設定できます。つまり、従来なら複雑な計算式の照合フィールドが必要だったところを、リレーションシップの設定だけでほぼ済ませることが出来ます。機能的にも拡充します。第二に、リレーションシップは双方向の性質を持ちます。従来のバージョンなら往路・復路と2つのリレーションシップが必要だったところが、1つでよいわけです。ただし、復路を設定する必要はないものの、リレーションシップをオフにすることもできません。それは常に存在し、データベースのどこかが要求すれば、ユーザの意図にかかわらずそのリレーションシップが使われます。これも理由あってのことです。

## TO 滴 テーブル

このタイトルをTシャツにプリントしようかと考えています。「TO ≠ Table!」、これこそFileMaker 7という禅の公案への答えです。

TOはテーブルを参照する、テーブルの別名です。ではなぜ、TOとテーブルを明確に区別することがそれほど重要なのでしょうか。それはTOが、テーブルオカレンス・グループ (TOG) というもっと大きな機能グループに属しているからです。TOGによって、そのTOのコンテキストつまりデータ環境が決まります。つまり、テーブルにはファイルの内外にいくつもの別名 (TO) があり、それぞれ主テーブルに対して同レベルのアクセスを持っています。しかし各TOは異なるTOGに属するため、主テーブル以外のテーブルとの関係はそれぞれまったく違うものになるのです。

「選挙」の例を考えてみましょう (図3a)。「有権者」テーブルには3つの別名があり、各TOは「DM有権者」「テレコール有権者」「送迎有権者」と名前がついています。それぞれにレイアウトを割り当てると (一例として図5a)、各レイアウトは「有権者」テーブルのデータに対して同レベルの直接アクセスが可能となり、すべてのフィールドとレコードを閲覧できます (セキュリティ設定による制限を除く)。しかし、「テレコール有権者」のTOに帰属するレイアウトからは、「住所」テーブルのデータは見られません。同じく、「DM有権者」TOに帰属するレイアウトからは、電話番号は一切閲覧できません。このように、TOにはそれぞれ独自の

データ環境があり、その環境は帰属するTOGによって規定されます。

図5aは、「送迎有権者」TOに帰属するレイアウトをレイアウトモードで表示したものです。「有権者」テーブルのフィールドであれば直接ドラッグして配置できます。しかし、ここから「住所」テーブルのフィールドを見るには、フィールド名の前に何らかの「識別コード」を指定する必要があります。従来のバージョンでは、この識別コードはアクセスするテーブルへのリレーションシップ名でした。しかしFileMaker 7では、リレーションシップは名前がつけられません。そこで、代わりにTO名を使います。ただし、選択できるTOは「送迎有権者」TOと同じTOGに属するものだけです。外部のTOGからは選べません。(注1)

レイアウトに新しいフィールドを置くと、フィールド指定の選択肢がリストとして表示されます。現在のテーブル (この例では「有権者」)、関連テーブル、非関連テーブルの3種類があり、さらに「データベースの定義」から新たに作成することもできます。しかし、ここで注意してください。選択肢に表示されているのは実はテーブル名ではなく、TO名です。TOだと何が違うのでしょうか。「DM有権者」「テレコール有権者」「送迎有権者」の3つのTOはどれも「有権者」テーブルを参照しています。しかし、「非関連テーブル」の欄にあるテーブルに「DM有権者」「テレコール有権者」を経由してアクセスすることはできません。これら非関連テーブルのフィールドをレイアウトに配置しても、<非関連テーブル>のエラーが出てしまいます (図5b)。

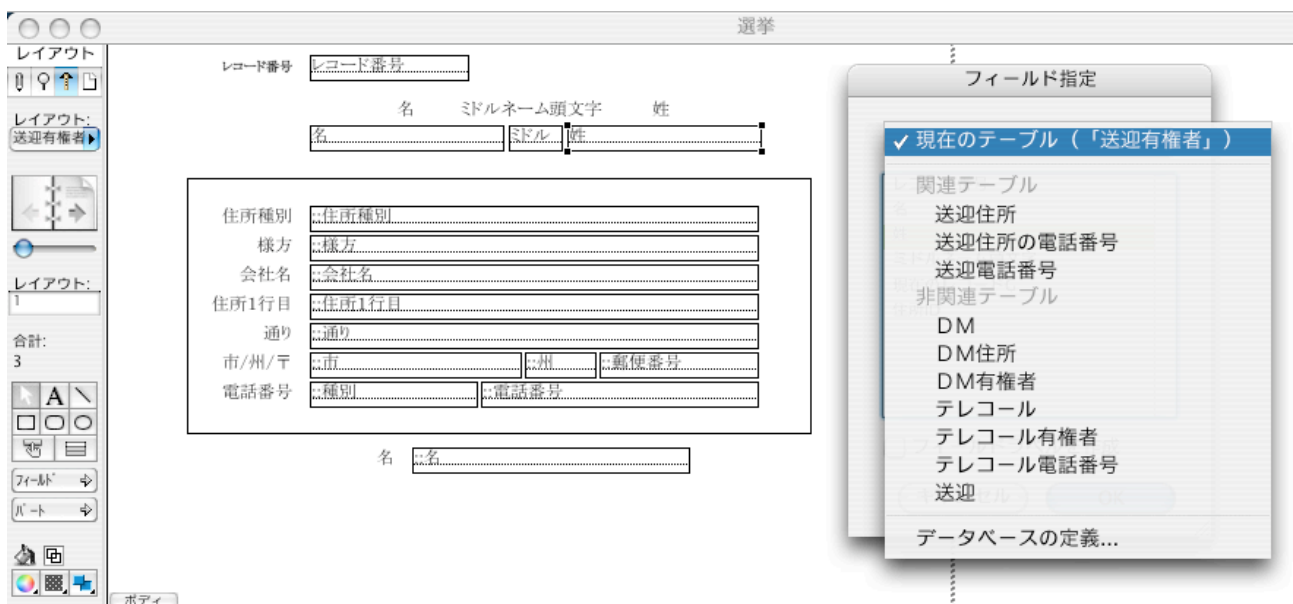


図5a [フィールド指定]ダイアログ。ポップアップリストの中で「DM」「テレコール」「送迎」の3項目はTOGのタイトルとして使われる空のTOであり、有効な選択肢ではありません。

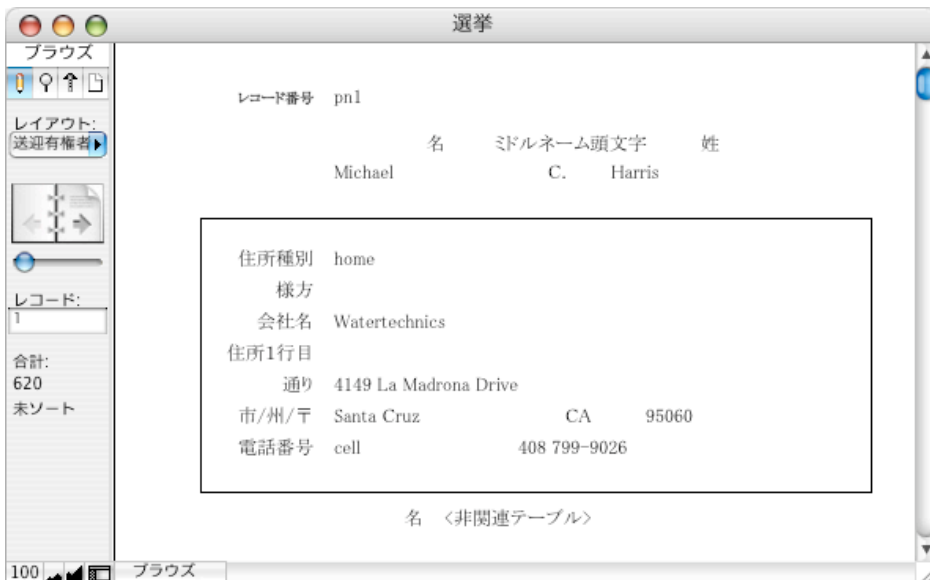


図5b 非関連テーブル（現在のTOG外）のフィールドは、レイアウト上に配置はできますが、値はエラーになります。

この[フィールド指定]ポップアップ・ウインドウを解読してみましょう。まず、リレーションシップグラフ上でTOGのタイトルとして使用される空のTOは無視します。「データベースの定義」も同様です。残りはどうでしょうか。「関連テーブル」の実体は、このレイアウトのTOと同じTOGに属している、他のTOです。つまり、現在のデータ環境あるいはコンテキストの中にあるTOが表示されています。これに対して「非関連テーブル」の方は、現在のTOGの外にあるTOをあらわしています。

TOとは、あるテーブルのデータを他のテーブルのデータと関連づけるために作成される、テーブルの別名です。ですからレイアウトやスクリプト、フィールド定義でTOを指定するのは、主テーブルのデータに直接アクセスを得るだけでなく、リレーションシップというコンテキストを獲得することでもあるのです。テーブル自体は他のテーブルとは何の結びつきもありません。単なるデータの入れ物です。だからこそ、「TO ≠ Table !」

### 新スキーマ登場

ここまで説明してきたFileMaker 7の特徴は、それらが組み合わさると単なる個々の総和以上のものになります。FileMakerのまったく新しい構造、スキーマ、パラダイム…呼び方は何でもよいでしょう。この新スキーマの中心となるのが、FileMaker 7ではじめて登場した設計オブジェクトであるテーブルオカレンス・グループ（TOG）です。TOGがデータ環境を規定し、すべての作業はその環境の中でおこなわれます。

TOGは適切に設計し、注意深く使用し、TOをとおして常に参照しなければなりません。必要なだけのTOGつまりデータ環境をリレーションシップグラフ上で作成し、その中から1つのデータ環境を「現在のTOG」として選びます。他のファイルのTOGは使えませんが、逆に、他ファイルのTOGに影響されることもありません。

先ほど紹介したコージブスキーの引用には続きがありません。

「地図は現地ではない。しかし、正しい地図は現地と同じ構造を持っている。だから使えるのだ」

リレーションシップグラフにおける「正しい地図」（TOG）とは、FileMakerが正しく機能できるデータ環境が描かれた地図と言えるでしょう。1つのファイルにはたくさんのこういった環境が作成でき、それらが集まって1つのファイルになります。

もう一つの見方として、TOGをレイアウトという切り口から考えてもよいでしょう。レイアウトは特定の目的をもって作成されます。データ入力、リスト・レポートの作成、データ分析、ユーティリティ作業などさまざまな業務目的があります。例えば「選挙」ファイルなら、宛名ラベル印刷用のレイアウトがあるのではないのでしょうか。宛名ラベルを出すボランティアは、有権者の電話番号を知る必要はありません。情報保護の面からもそうしておいた方がよいでしょう。そのためには、宛名ラベルレイアウト用のデータ環境を用意します。有権者の電話番号は見られないがラベルレイアウトの調整はできる、という環境を与えることです。この例では、レイアウトに「DM住所」TOを指定します。このTOの属するTOGは「電話番号」テーブルのTOとはつながりがありません（図3a）。

リレーションシップグラフ上で抽象的な作業をするよりも、まずは必要なレイアウトの目的にもとづいて、リレーションシップの具体的な仕様を決める方が簡単かもしれません。そのあとで、適切なTOGがない場合は、レイアウトの目的にあわせて新しくつくればよいのです。



### TOGの構造

「… (アリスは) 2、3ヤードさきの木の枝に坐っているチェシャ・ネコが目にはいつて、ちょっとぎくつとしました。

ネコは、アリスを見て、にやりと笑っただけでした。気だてのよさそうなネコなこと、とアリスは考えました。でも、なにぶん、つめはととも長いし、やたらにたくさんの歯があるので、

これは敬意を表してあつかわなければいけないな、と感じました。

“チェシャ・ネコさん”と、はたしてこの名前が気に入るかどうか、まったく見当がつかないので、アリスはいくらかおずおずとよびかけました。でもネコは前より少々大口をあけてにやりとただけでした。“しめた、ここまではまずごきげんらしいわ”とアリスは思い、話しつづけました。“ここからどの道をゆけばよいのか、教えてはいただけませんか？”

“そりゃ、まったくあんたのゆきたい方角次第だよ”とネコは言いました。

“わたし、とくべつにどこへということも—”とアリスは申しました。

“それじゃあ、あんたがどの道をゆこうと、かまわんじゃないか”とネコは言います。

“どこかに出られるものなら”とアリスは説明をつけ足しました。

“なあに、どこかに出られるにきまつてるさ”とネコは言いました。“たっぷり歩きさえすればね！”

—ルイス・キャロル『ふしぎの国のアリス』生野幸吉訳 より

アリスはどこかへ出られればよいのですから、どの道を通ってもかまいません。しかしFileMaker 7の道つまり経路は、そう簡単にはいかないのです。システムの中核をなすオブジェクトとして、TOGには固有の性質、ルール、構造要件があります。中でも次のルールは重要です。

リレーションシップグラフ上の2つのTOを結ぶ経路は1通りであるか、経路なしとする。

これはなぜでしょうか。関連フィールドの指定ではTO名しか使いません（「DM住所::通り」）。つまりFileMaker 7が指定されたTOの主テーブルからデータを取り出すためには、たどるべきリレーションシップのつながり（経路）が自明でなければならないのです。2つのTO間の経路を開発者がリレーション名を指定して選択する余地はありません。FileMakerがたどるべき経路をはっきり認識できるように、経路は1通りにします。

アリスは自分がどこへ行きたいのか言わなければなりません。行き先となるTOを指定し、データを取り出す主テーブルを明確にします。従来のバージョンでは、経路（別ファイル=テーブルへのダイレクトなリレーションシップ）を指定していました。FileMaker 7では、行き先を指定します。そうすれば経路は1通りしかないので、自動的に決まります。

TOGを理解するには次の用語が必須です。

・経路 — 複数のTOを結ぶリレーションシップのつながり。1つのTOから始まり、1つまたは複数のリレーションシップを経由して、別の1つのTOで終わります。TOのグループにおいては、リレーションシップが枝分かれすることもあります。2つのTOを結ぶ経路は1通りでなければならず、枝分かれはできません。

経路グループ — 経路によって結ばれた2つ以上のTO。より大きなTOGの下位グループであることが多い。

### TOGサーフィン

TOGはそれぞれ固有のデータ環境ですが、業務プロセスの中では1つのデータ環境から別の環境へ、ユーザには分からないように移動する必要も出てきます。その場合、業務プロセスの段階によってレイアウトを切り替えることになります。これがTOGサーフィンです。

選挙の例で、あるボランティアは70歳以上の有権者を送迎の集合場所にあつめる仕事を担当しています。「送迎住所」TOに属するレイアウトを使って、集合場所となる家に電話をかけ、全員あつまったかどうか確認します。到着していない有権者がいる場合は、「テレコール有権者」TOのレイアウトに「サーフィン」すれば、携帯電話を含めてその人に関連するすべての電話番号が見られます。「送迎住所」のレイアウトでは、携帯電話の番号までは分かりませんでした。

従来のバージョンでは経路を指定しました。  
FileMaker 7では、行き先を指定します。

業務プロセスによっては、TOGサーフィンの活用が役立つどころか不可欠になることも十分考えられます。一方、TOGサーフィンなどしなくても他の方法で十分同じ結果が得られるのかもしれませんが。これは実際に現場でやってみなければ分からないことです。いずれ結論が出るでしょう。

## コンテキスト

「バラはバラはバラ」

ーガートルード・スタイン

TOGサーフィンなどせずにFileMakerの作業が進められる場合でも、複数のファイルやテーブルの中で自分の位置をはっきりさせておくのは、やはり困難なことでしょう。「何について話しているのか」を何度も尋ねられる上に、ダイアログによっては選択時にさらなる混乱が生じます。指定するのがファイルなのか、レイアウトなのか、テーブルなのか、あるいはテーブルオカレンスなのかがはっきりしないのです。異なるオブジェクトに同じ名前をつけられるので、非常にやっかいなことになります。

異なる階層のオブジェクトに分かりやすく名前をつけるため、いくつか命名法が提案されています。もちろん、個人あるいは個々の課題に適したやり方でなければなりません、必ず何らかの命名法を決めるようにしてください。ファイルなのか、テーブルなのか、テーブルオカレンスなのか、名前を見れば分かるような方法が必要です。レイアウトとTOGについても別途ルールを決めるとよいでしょう。

スタインの引用は、次のように解釈することもできます。すなわち「バラ（花）はバラ（言葉）はバラ（経験）」。しかし、どの「バラ」がどれを意味しているのでしょうか。そこがこのフレーズのうまいところですが、FileMaker 7の場合は区別できた方がよいでしょう。たとえば、

「バラ[花]はバラ[言葉]はバラ[経験]」。

## 用語解説

FileMaker 7という未知の国では、話す言葉もこれまでとは違います。使われなくなった語や新たな意味を与えられた語があり、まったく新しい語彙も覚える必要があります。従来から使われていた基本用語のうち、FileMaker 7で大きく意味が変化したのは、「ファイル」「レイアウト」「リレーションシップ」「ウインドウ」「グローバル」の5つです。また、FileMaker 7を理解するのに必要な新しい語彙としては、「テーブル」「テーブルオカレンス（TO）」「テーブルオカレンス・グループ（TOG）」「TOGサーフィン」「経路」「経路グループ」「コンテキスト」があります。本稿で使用するこれらの用語のうち、「経路」「コンテキスト」などはFileMaker 7をめぐる議論の中で多くの開発者から提案されたものであり、一方「TOG」「TOGサーフィン」「経路グループ」「設計空間」などは本稿独自の用語となっています。今後半年のうちには、用語についてFileMakerのコミュニティの中でコンセンサスが形成されると思いますが、ここでは筆者が有効と考える用語を紹介することにします。（なお、FileMaker, Inc.は同社独自の用語を使用しており、筆者のものと一致しない場合があります）

ホームファイル（母ファイル、作成元ファイル）  
home file (mother file, creation file)

FileMaker 7は新しい言葉話します。使われなくなった語や新たな意味を与えられた語があり、まったく新しい語彙も覚える必要があります。

テーブルが作成された元のファイル。フィールド定義の追加・編集、基本的なセキュリティ設定はこのファイルでしか行えません。

ファイル（ホームファイルも参照） file

(1) FileMakerのデータベースの基本となるドキュメント。データベースの作成は必ずファイルの作成に始まり、すべてのデータベースには少なくとも1つのファイルが必要です。テーブルに対してリレーションシップを指定する場所でもあります。

(2) 有能な管理者であり、テーブルの入れ物でもあります。テーブル管理、ビジネスロジック構築、ユーザへのプレゼンテーションなど、さまざまな目的に使用されます。

(3) テーブル、フィールド、スクリプト、リレーションシップを作成、使用する場所。

テーブル table

FileMakerデータベースのデータが行と列の形式で保存される場所。FileMaker 7のダイアログボックスでは「テーブル」という語が「テーブルオカレンス」の意味で使われている場合があります（一例として図5a）。

バラ[FILE]≠バラ[FILEREF]≠バラ[FIELD]≠バラ[TABLE]≠バラ[TO]≠バラ[TOG]≠バラ[LAYOUT]≠バラ[WINDOW]

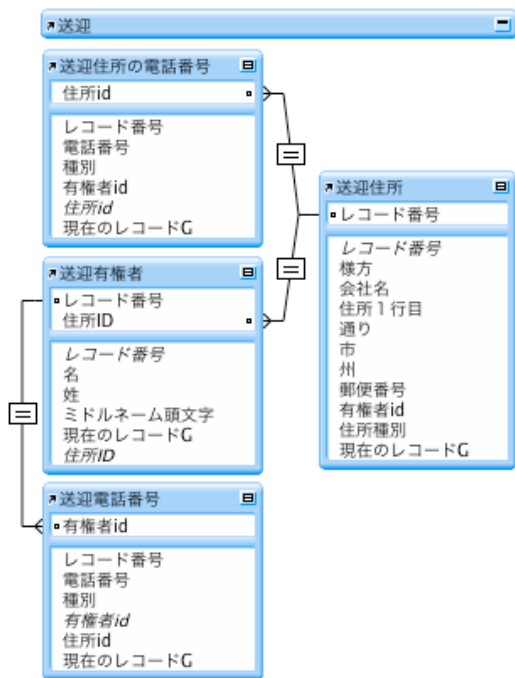


図6a 図3aで取り上げた3つのテーブルオカレンス・グループの1つ。「送迎」TOはTOGのタイトル表示用で、TOとしては機能していません。

### テーブルオカレンス (TO、テーブルの別名) table occurrence (TO, table alias)

リレーションシップグラフの中でテーブルを表す別名。あるレイアウトに対してどのTOを指定するかで、そのレイアウトからアクセスできるデータが決まります。TOを指定すると、そのTOの主テーブルが持つすべてのデータ、およびそのTOと同じグループ (TOG) 内の関連テーブルのデータが見られます。

### リレーションシップグラフ relationships graph

ファイル内のデータの関係 (リレーションシップ) を定義する場所。なお、ここで言う「ファイル」は、データベースが1つのファイルのみで構成されている場合を除き、「データベース」とは異なる点に注意してください。リレーションシップグラフは1つまたは複数のTOGで構成されており、各TOGはリレーションシップでリンクされた1つまたは複数のTOから成っています。1つの主テーブルに対して複数のTOが存在する場合もありますが、各TOにはそれぞれ固有の名前をつけます。

### リレーションシップ (またはリレーション) relationship

リレーションシップグラフにおける、2つのTO間の関係とその条件 (TOはそれぞれテーブルに帰属する)。一方のテーブルからもう一方のテーブルのどのデータを表示するかを指定するフィルタとして機能します。

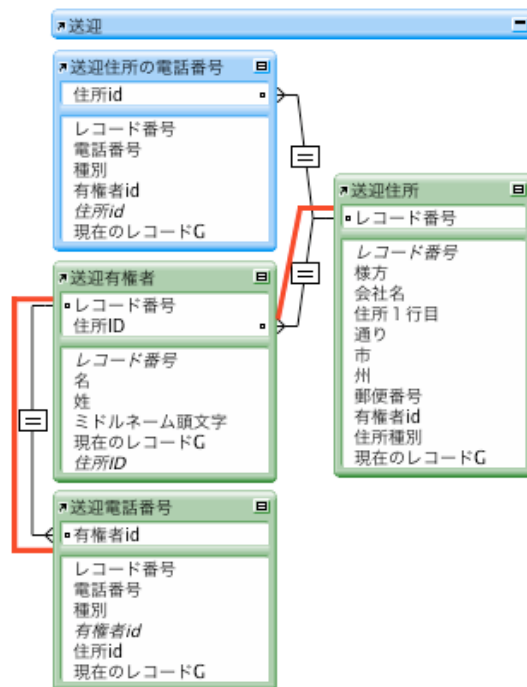


図6b 「送迎電話番号」TOと「送迎住所」TOを結ぶ経路 (赤) は、「送迎住所」「送迎有権者」「送迎電話番号」の3つを経路グループ (緑) としています。

す。FileMaker 7では、リレーションシップは必ず双方向となり、複数の条件および照合フィールドの設定が可能です。さらに、いくつかのリレーションシップがつながって1つの「経路」として機能します。経路でつながれた別のテーブルのデータにアクセスする場合、途中で経由する全リレーションシップの条件の総和がデータ閲覧の条件になります。

なお、筆者の知るかぎり「リレーション」と「リレーションシップ」は同じものを意味します。個人的には短い言葉の方がよいですが、通常はFileMaker, Inc.の用語に従って「リレーションシップ」を使うようにしています。開発者のみなさんからは、「リレーション」は2つのテーブル間のリンクに対して用い、「リレーションシップ」はTOGの中で複数のTOを結ぶ経路に対して使ってはどうかという意見や、TOに対しては「リレーションシップ」を使うことにしてはどうかという意見もありました。誤解を避けるという意味では、「リレーションのつながり」に対しては「経路」の方がよいように思います。

### 主テーブル (またはベーステーブル) primary table (also base table)

1つのTO (あるいはレイアウト) が無制限のアクセスを持つデータ。そのTOが別名としてあらわしている元のテーブル。

## コンテキスト (切り口、データ環境)

context (point-of-view[POV],data environment)

現在どのような切り口でデータやスクリプトステップ、計算式、レイアウトなどを見ているかを示します。通常、TOを指定 (つまりTOGを指定) することによって明らかにされます。

## テーブルオカレンス・グループ (TOG)

Table occurrence group (TOG)

複数のテーブルオカレンスをリレーションで結びつけたもの。TOG作成のルールとして、リレーションシップグラフにあらわれる2つのTOの間には、経路は1つのみ、あるいは経路なしでなければなりません。あるTOGがより大きなTOGの一部である場合もあります。

## レイアウト layout

データベースのテーブルの表示方法。レイアウトは特定のTOに帰属し、複数のTOに使用されることはありません。1つのレイアウトからは、帰属するTOの主テーブルが持つ全データが見られます。また、そのTOと同じTOG内の他のテーブルについても、経路を通じてデータが

FileMaker 7における「ファイル」とは何か、  
1文で定義しなさい。

れば、より明晰な思考が可能となるはずで  
す。筆者は「データベース」を上記のように定義して使っています。

レイアウトのデータ環境 (同義語「コンテキスト」も参照) layout data environment

1つのレイアウトから見られる各テーブルのデータと、データ閲覧のための各条件 (各経路に含まれるリレーションシップの条件)

## 経路 path

複数のTOを結ぶリレーションシップのつながり。1つのTOから始まり、1つまたは複数のリレーションシップを経由して、別の1つのTOで終わります。TOのグループにおいては、リレーションシップが枝分かれすることもあります。2つのTOを結ぶ経路は1通りでなければならず、枝分かれはできません。

## 経路グループ path group

経路によって結ばれた2つ以上のTO。より大きなTOGの下位グループであることが多い。

## TOGサーフィン TOG surfing

業務プロセスを実現するためのテクニックの一つ。レイアウトを次々と移動しつつもユーザの見ていたデータは同じであり、ただしレイアウトごとにデータの切り口が変わり、異なるTOGつまり異なるデータ環境に属するTOが使われます。これにより、ユーザが見る主テーブル以外のデータ (関連テーブルのデータ) を業務プロセスの段階ごとに管理できます。

## ウインドウ window

ファイルにおいて、レイアウトにもとづいてデータを表示したもの。FileMakerのファイル (またはデータベース) では、複数のウインドウが同時に開けます。各ウインドウは別々のレイアウトを表示するか、同じレイアウトの異なる検索結果を表示します。ウインドウには個別に名前がつけられます。

## 設計空間 design space

想像上の空間で、FileMaker 7において可能な設計上の選択肢を、複数の次元をもつ座標軸であらわしたもの。物理上の空間に似ていますが、次元数は3次元あるいは4次元を超えることもあります。

## データベース database

共通の目的をもって使用されるテーブルの集まり。少なくとも1つのファイルで構成されます。

なお、よく知られているとおり、FileMakerのコミュニティでは非常にいい加減な用語の使い方をすることがあります。「データベース」もその一つで、しばしば「ファイル」の意味で使われてしまっていますが、「テーブル」「ファイル」「データベース」をはっきり使い分け

## 新しい世界のルール

FileMaker 7の構造を理解するには、その新しいルールをよく考えることです。これらのルールは、正しく理解されれば、FileMaker 7の構造をはっきりと照らし出します。

1. フィールド定義の追加・削除・編集はそのテーブルが作成されたホームファイルでのみ可能。
2. 基本的なセキュリティ管理はそのテーブルが作成されたホームファイルでのみ可能。
3. テーブルは、そのホームファイルを離れては存在しない。テーブルはファイル内にしか作成できません。別ファイルへの移動や複製はできず、ファイルが移動すればそのテーブルも一緒に移動します。
4. リレーションシップはオブジェクトではなくTOGのプロパティ、設定オプションである。TOGは、リレーションシップで結ばれたTOの集まりである。

5. リレーションシップグラフで作成されたデータ間の関係は、そのファイルおよびファイル内のフィールド、レイアウト、スクリプト、設定にしか適用しない。(注2)
6. すべてのTOは、そのファイル内で固有の名前を持つ。
7. リレーションシップグラフ上の2つのTOを結ぶ経路は1通りであるか、経路なしとする。
8. FileMakerのファイル(データベース)では、複数のウインドウが同時に開ける。各ウインドウは別々のレイアウトを表示するか、同じレイアウトの異なる検索結果を表示できる。
9. レイアウトは常に1つのTOにのみ帰属する。レイアウトからは、そのTOの主テーブルのデータはすべて見られます。また、他テーブルのデータもそのTOが属するTOGの経路を通じて閲覧できます。この場合、閲覧できる他テーブルのデータは、レイアウトの主テーブルと関連TOをつなぐ経路によって規定されます。
10. ファイルで使用するすべてのテーブルについて、少なくとも1つのTOをリレーションシップグラフに置く。  
テーブルデータをインポートまたはエクスポートする、またはODBCおよびJDBC機能を使用する場合に、TOが必要です。  
またインポートの際は、レコードを受け取るテーブルのTOに帰属するレイアウトが少なくとも1つ必要です。計算式などでフィールドの値を使用する場合も、そのテーブルのTOを少なくとも1つはリレーションシップ上に置いてください。TOがないとそのフィールドは指定できません。計算式、スクリプト、自動入力設定、値一覧、レコード単位でのアクセス権設定などすべて同様です。テーブルのTOがすべて削除されると、そのテーブルのすべての計算フィールドは失われます。
11. ポータルにポータルのTO以外からフィールドを表示する場合、そのフィールドのTOとレイアウトのTOを結ぶ経路上にポータルのTOがなければならない。  
そうでないと、逆向きの経路を指示する必要があります。
12. リレーションシップは双方向の性質を持つ。実際に双方向で使用するか、使用できるかに関わらずこの性質は維持される。

13. 検索結果はテーブルではなくTOに帰属する。  
同じ主テーブルを持つTO間では、1つのTOでおこなった検索結果を保持したまま別のTOへ移動できます。[関連レコードへ移動]スクリプトステップで、行き先のテーブルに検索元のTOを指定し、使用するレイアウトにもう一方のTOのレイアウトを指定します。

## FileMaker 7の黄金律

### TO満テーブル

#### 一歩ずつ進む～設計空間～

「ヨシはFileMakerという禅を学ぶため、和尚を訪ねた。“テーブルオカレンスはいつテーブルになるか?”と和尚は問いを与えた。“師よ、感謝します”。ヨシは和尚の前を辞し、与えられた公案を考えはじめた。それから2年のあいだ、ヨシは週に一度師のもとへ戻って同じ問いを受けたが、満足な答えはできなかった。そしてある晴れた春の日、師はいつものように問うた。“テーブルオカレンスはいつテーブルになるか?”。ヨシは答えた。“水もボードもなしでサーフィンができる時です”。師はヨシの頬を張り、弟子は気づきを得た。師はつぶやいた。“では、4次元の設計空間に寺院を建てる正しい方法は?”。ヨシは帰って修行をつづけた」

最初のステップとして、まずはFileMaker 7にどんな使い方があるかを学ばなければなりません。しかしこれはまだ簡単な方です。FileMaker 7のベストプラクティス、つまり「いかに使うべきか」を理解するのはずっと困難な過程になるでしょう。まずは、FileMaker 7の設計にどのような選択肢があるかをつかむことです。顧客のビジネスの構造をFileMaker 7で描き出すためには、何通りのやり方があるのでしょうか。

FileMaker 7によって可能となる、膨大な設計の選択肢を把握するために筆者が考えたのが「設計空間」です。この空間では、相互に関連するさまざまな選択肢が多次元の座標軸上に配置されており、FileMaker 7をどう使えば顧客の問題を解決できるか、考えるヒントになります。

筆者は、FileMakerはそれ自体の自然な構造を持っていると考えてきました。システム開発者の仕事は、顧客の用途に合った形で、FileMakerの持つすぐれた構造を描き出すことです。このプロセスがプロジェクトの初期段階にきちんとおこなわれれば、後の開発作業は楽にすすみます。反対に最初に誤った構造を設計してしまうと、ぶつからなくてもよい困難に見舞われ、システムがある程度複雑になった段階で自滅します。

ですから、FileMaker 7で何ができるかを学んだら、次はこの問いに取り組んでください。「FileMaker 7で顧客の問題の構造を描き出す方法が何通りもあるなかで、設計の主要素をどんな設計空間に、どのように配分するか」(図7)

FileMaker 7での設計にどんな選択肢があるかをつかむには、何年もかかるでしょう。試行錯誤を繰り返し、二度としたくないような失敗を積み重ねて、やっと見えてくるのではないのでしょうか。ここで設計空間という考え方が助けになります。手始めに、設計の選択肢を3次元の座標軸で示した空間を思い浮かべてください。第1軸は、どんなテーブルをいくつ作成するか。第2軸は、どんなファイルをいくつ作成するか。第3軸は、レイアウトをファイル間でどのように配分するかです。

まず、テーブルの問題は簡単です。従来のバージョンでは、「どんなレコードが必要か」を考えれば作成するファイル=テーブルがだいたい決まったものですが、この点はFileMaker 7でもあまり変わりません。従来のバージョンを使った経験がある人、リレーションシップの理論を多少なりとも知っている人なら、必要なテーブルはすぐに分かります。プロの開発者同士なら、同じ問題に対して考え出すテーブルはほとんど同じでしょう。そうすると、設計空間ではこの座標軸はかなり短くなります。なお、FileMaker 7では1ファイル=1テーブルの制約がないので、従来のバージョンを使うよりも多少テーブルの数が増えると思われます。次にファイルの座標軸は、テーブルよりもずっと長くなります。FileMaker開発者待望のファイル数「1」を最小値に、従来のバージョンにならって「テーブルと同じ数」、さらにはテーブルよりはるかにたくさんのファイル数も可能です。1つのデータベース・エンジンを使用するユーザが多様で、それらのユーザに合わせて多数のユーザインターフェイス（UI）ファイルを作成する場合などはファイル数がテーブル数より多くなるでしょう。第3軸のレイアウトについても、実際に可能かどうかは別として、理論上はたくさんの選択肢があります。すべてのレイアウトを1つのUIファイルにまとめる、ファイルの機能グループごとにレイアウトを配分する、FileMaker 6のようなレイアウト配分にする、などが考えられます。FileMaker 6のような配分とは、FileMaker 6のファイルをFileMaker 7に変換したときのような状態です。

図7はこの設計空間を図示したものです。厚切りのトーストのような形をしているのは、具体的なプロジェクトでは「テーブル」の選択の幅が実質上ほとんどないからです。大まかに、4つの象限で異なる選択肢の組み合わせをあらわしていると考えてください。1つのファイルでデータベースを作る場合、レイアウトも当然1ファイルにおさまるので、下側手前の象限のどこかになります。機能によってテーブルとレイアウトをいくつかのグループに分けるなら、上側手前の象限です。

ファイルメーカー 6のようなレイアウト配分であれば、上側奥の象限でしょう。100の作業グループそれぞれにUIファイルを作成するような場合は、ほとんど同一のレイアウトを大量につくり、各UIファイルに割り当てることになります。これは下側奥の象限です。

ここまで設計空間の各象限を簡単に説明してきましたが、これでは選択肢を示しただけです。無数の選択肢があることはわかりましたが、実際にデータベースをこの空間のどこにつくるべきか、何にもとづいて決定すればよいのでしょうか。

---

複雑なプロジェクトでは、  
テーブルオカレンス・グループをうまく使える  
かどうかの結果を左右します。

---

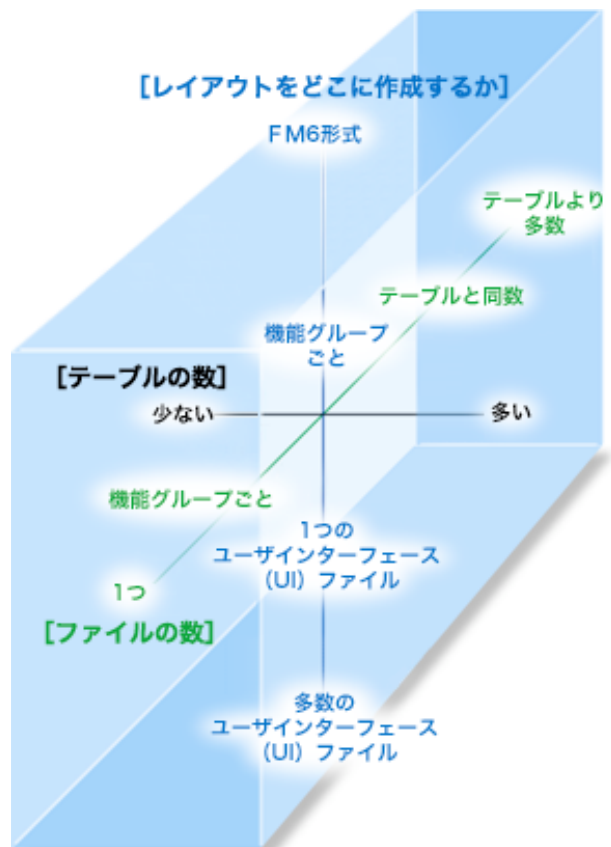


図7 シンプルな3D設計空間の例

当社では、FileMaker 7という新しい領域を注意深く探検していきます。多くのデータベースは1つのファイルで間に合うでしょう。一部については、テーブルを機能ごとにグループ分けして別々のファイルに入れます。こうすれば、各グループの修正やセキュリティ管理、開発作業はまとめておこなえます。当社のデータベースでいえば、氏名や住所関連のテーブルは1つの機能グループにまとめられそうです。「個人」「住所」「通信会社」「企業」「社員」などのテーブルを「関係者」または「顧客」などの名前で1つのファイルにまとめます。オンライン・アプリケーションサービスの事業では、複数のファイルからなる機能グループをつかって「ソフトウェア・エンジン」とし、これを中心にかくさんのUIファイルを配置します。各UIファイルは特定の顧客専用で、ユーザインターフェースとビジネスロジックの両方が入ります。

以上は大変シンプルな例ですが、それでFileMaker 7の登場によって、開発者にとってはまさに「富めるが故の惑い」の時代に入ったことが分かります

## 探検の旅へ

半年か1年もすれば、FileMaker 7で何ができるかについては皆が了解するところとなるでしょう。

FileMakerのコミュニティには各種講座、ワークショップ、解説本、論文、プレゼン、認定試験などがあふれかえり、インターネットではあらゆる情報の断片が手に入るはずですが、しかしそのベストプラクティス、つまりパワーアップしたFileMaker 7で「何をなすべきか」となるとそうは行きません。これを理解するには、開発の現場という荒野で長い旅路を行くことになるでしょう。

しかし、FileMaker, Inc.でソフトウェア開発という苦役に従事する天才たちは、わたしたちにフロンティアの探検旅行に加わるチャンスくれたのです。こんな魅力的な話があるでしょうか。筆者はすでに準備万端です。旅の途中では熊に遭うこともあるでしょうが、このチャンスは絶対に逃せません。

<p><b>テーブル空間</b></p> <ul style="list-style-type: none"> <li>・どんなフィールドが必要か</li> <li>・どんなテーブルを作成するか</li> <li>・どんな照合フィールドを使うか</li> <li>・グローバルフィールドをどこに作成するか</li> <li>・計算フィールドをどこに作成するか</li> </ul>	<p><b>レイアウト空間</b></p> <ul style="list-style-type: none"> <li>・各レイアウトの業務目的は何か</li> <li>・どのレイアウトがどのTOに帰属するか</li> <li>・どのレイアウトをどのファイルに作成するか</li> <li>・各レイアウトはどんなデータ環境にあるか</li> <li>・新規ウインドウを開く、または閉じるのはいつか</li> </ul>
<p><b>ファイル空間</b></p> <ul style="list-style-type: none"> <li>・ファイルとは何か。どんな機能があるか</li> <li>・ファイルは1つか、多数か。それぞれどんな得失があるか</li> <li>・どのファイルがどのテーブルのホームファイルか</li> <li>・業務プロセスはファイル間をまたいでおこなわれるか</li> <li>・ファイルは機能別に階層化させるか。どんな構成にするか</li> </ul>	<p><b>TO空間</b></p> <ul style="list-style-type: none"> <li>・どんなTOを作成するか</li> <li>・どんなTOGを作成するか</li> <li>・業務プロセスにTOGサーフィンが必要か</li> <li>・どのファイルにどのTOおよびTOGを作成するか</li> <li>・どんなリレーションシップを作成するか。どこまで複雑にするか</li> </ul>
<p><b>超空間</b></p> <ul style="list-style-type: none"> <li>・FileMaker 7で問題の構造を描き出す方法が何通りもあるなかで、設計の主要素をどんな設計空間に、どのように配分するか</li> </ul>	

「わかりにくく話せたと思うよ」

戯曲『わが命つきるとも』  
ロバート・ボルト作  
トマス・モア卿のセリフより

「ある大学院の研究室で、教授はひとりの新生に歩み寄ると新鮮な魚を1匹手渡し、言った。“その魚を説明しなさい”。新生は図書館へ行って長文のレポートを書き上げた。魚の種、亜種、進化の歴史、現在の個体数から寄生虫まで網羅して、教授に提出する。ところが教授はちらっと見ただけで、レポートをぼいと放ってしまった。“再提出”。それから何度か提出と不合格を繰り返すと、教授はうんざりした顔でこう言った。“魚を見なさい”。2週間後、魚はすっかり腐っていたが、魚を見て書いた新生のレポートは合格した」

—生物学の大学院で一年生がよく聞かされる話

「無知な人間が理解すれば教師になれる。だが教師が理解するにはもう一度無知な人間になるしかない」

—百丈懐海 禅師の言葉より

「量子力学を理解できたと思うなら、それは量子力学を理解していない」

—リチャード・ファインマン

