

## 全社プロジェクト管理システムの提案と セルラーDBMSの利用

児玉 敏男<sup>1</sup> 國井 利泰<sup>2</sup>

### 要旨

企業の経営環境が様々に変化する今日、企業の既存の一般的な基幹業務システムは業務機能毎の部分最適を指向し開発されているため、変化への対応が困難な状況にある。そこで業務システムの全体最適を志向した全社プロジェクト管理システムを提案する。本システムでは、統合データベース・全体管理コアシステムを構築することでシステムのインターオペラビリティを高め、開発コスト、保守運用コストの大幅な削減を可能にする。同時に、ホモトピー理論を適用することによりプロジェクトのトラッキング機能を実現する。結果として、全社的な総合的なプロジェクト管理や頻繁な業務組織の変更を可能にした。さらに、本システムの開発には既存のデータベースよりも現在筆者らが開発中の新型データベースであるセルラーデータベースが有効であることを示す。

## Development of a project management system of the whole company and Use of a Cellular Database Management System

Toshio Kodama<sup>1</sup> Tosiyasu L. Kunii<sup>2</sup>

### Abstract

The business environments of companies are changing in varied ways today. As for the common basic operation systems of companies, the systems are being developed to optimize each operation function. Although it is very hard to cope with various changes, we have developed a project management system of the whole company which aims at optimizing the entire operation system by building an integrated database and the core of the operation systems that can improve the interoperability of the operation systems and enables a great reduction of the development cost and the maintenance operation cost. At the same time, this project management system achieves the tracking of the project by applying the homotopy theory. Consequently the advantages of this project management system are: 1) Systematic in planned management of a project as much as possible; 2) Flexibility in changing operation organizations and business processes. In addition, we show that the cellular database we are developing is more effective than existing traditional databases in the development of this project management system.

### 1. はじめに

今日、インターネット上で様々なコンピュータ機器がグローバルに常時接続され、莫大な情報が常時処理されるような情報ハイウェイ時代に入った。とともに、企業の経営環境においては、企業間の合併・吸収、市場ニーズの多様化、コスト競争の激化、企業評価視点の変化、会計制度の変更等、予測のつかない変化が頻繁に起こる時代になった。今後、これらの環境の変化に対し、企業の業務システムにおいては、全社的に柔軟な対応が可能な機能を備えることが急務の課題である。例えば、経営トップの経営戦略のために全社的なプロジェクト管理情報を速やかに提供できる機能や業務仕様の変更時に柔軟な対応が可能なシステムアーキテクチャ、等である。

しかし、企業の既存の業務システムの大多数は、これまで業務機能毎に用途を想定して設計し個別に開発されているため、それらの要求に応えることが非常に困難である。これは、システム開発の指向性として個々の業務機能毎の最適を指向して全社の全体最適を指向していないこと、またシステム開発後のシステム統合には非常にコストが大きく現実的に不可能であることに起因する。

そこで、本稿では全社の業務システムの全体最適を指向した業務システムを提案する。本システムは、1. 全社業務の統合データベース、2. 全体管理コアシステムと個別業務システム、3. プロジェクトの全社的フローのホモトピー保存機能、から構成される。1, 2では、各業務機能において共通となるデータ属性を同一のデータベースで管理し、また各業務に特有な機能と共通な機能を区別しアプリケーションを開発することにより、システムのインターオペラビリティを高める。結果としてデータベーススキーマ間のキー構造の変化、キー属性値の同一のケースに対し柔軟に対応が可能である。3では、プロジェクトの系統的なトラッキングによって計画から実績までの総合的なプロジェクト情報の管理と、組織構成・業務プロセスの変更に対する柔軟な対応を可能にする。これらは3, 4章で詳述する。

また、本システムの開発には、既存データベースを使用するよりも、現在著者らが開発中の新型データベースであるセルラーデータベース（以下CDB）が有効である。なぜなら、CDBには既存DBとは異なり、データベース統合を一般化するセル接合機能、スキーマ・インスタンス両レベルのホモトピー保存機能がモデルレベルでサポートされているので、アプリケーション開発工数の大幅な削減とデータ処理速度の向上を実現するからである。これらは5章で詳述する。

<sup>1</sup> 前田建設工業（株）情報システム SC  
Maeda Corporation

<sup>2</sup> 金沢工業大学 IT 研究所  
IT Institute, Kanazawa Institute of Technology

## 2. 現状調査

### 2.1 要求事項

著者らは、業務システムへの今後のニーズを把握するため、情報系企業数社へヒアリング調査を行った。その結果から、今後の業務システムは次のような要求を満たすべきであると考えられる。

**開発工期の短縮：**システムの利用年数が今後もより短くなることが予想されるので、開発コスト削減と共に開発工期の短縮が急務の課題である。

**組織構成の変更・業務のプロセス変更への柔軟性：**組織構成の変更やそれに伴う業務プロセスの変更に対して、業務システムの変更が不必要な機能の柔軟性が求められる。

**業務仕様の変更に対する柔軟性：**業務仕様が変更した際には、変更箇所が極力少なくて済むような機能の柔軟性が求められる。

**全社的なプロジェクト管理：**企業の経営者、プロジェクトマネージャーが必要とするのは、計画から実績までの全社を通じた一連のプロジェクト管理情報である。つまり、各業務システムが個々の業務要求に応えるだけでなく、業務システム全体で有用な情報を速やかに提供できる機能が必要である。

### 2.2 企業の業務システムの現状

一般的な既存の業務システムを概観すると、データベースとその業務アプリケーションは、業務機能毎に個別にシステムが開発されている。その大きな理由として、多くのユーザー、システム開発者が、全社業務の全体最適よりも各業務機能の部分最適を目指した開発方法を指向していることによる。

このとき、業務ステップ・業務プロセスはシステム開発者の設計意図によって決定され、運用後はシステム全体の依存関係を把握したシステム管理者の存在のもと、ユーザーはシステム設計に従いシステムを利用する。よって、業務ステップ・業務プロセスに少しでも変更があった場合は、システム管理者はシステム全体を考慮しながら再設計する必要がある。よって、開発後のシステム変更のコストは必然と大きくなる。

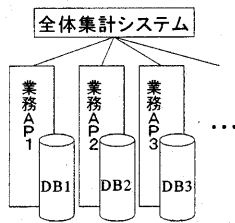


図 2.2-1 既存DBを用いた一般的な業務システム

ここである一企業（非製造業）の業務システムを調査した例を以下に上げる。

その中では、業務機能（実行予算作成・購買・機材管理・財務会計・技術情報支援・営業支援・人事）毎に業務システムを開発し、それらをインターフェイスシステムにより接続している。この各業務システム開発前に、開発者がユーザー側と綿密に協議し、各業務機能を果たすための業務ステップ・業務プロセスを決定する。例えば、実行予算作成システムの見積システムでは、{1. 概算見積、2. 提出見積、3. 工事見積、4. 受注計上}という4つの業務ステップと、業務プロセス（業務ステップを実行する順序とタイミング）を決定する。

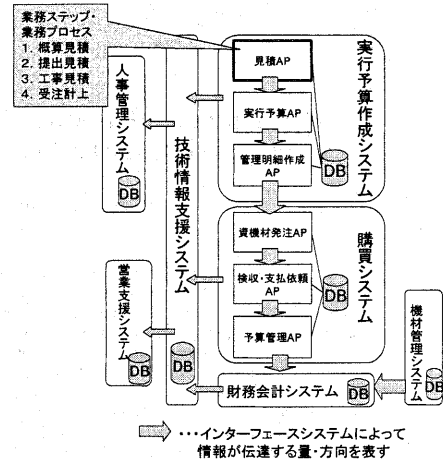


図 2.2-2 企業の業務システムの例

このとき、組織構成変更、業務改革等でデータベーススキーマ間のキー構造が変化（つまり業務ステップ・業務プロセスが変化）したときに、データベース設計、アプリケーション共に修正作業を行う必要がある。

## 3. 全社プロジェクト管理システム

今回提案するシステムは、プロジェクトが全社的にフローするときの業務ステップ・業務プロセスをユーザーが任意に決定可能であり、プロジェクトのトラッキング可能な柔軟なプロジェクト管理の手法を提案するものである。

### 3.1 システム概要

本章で提案するシステムは、前述の一般的な業務システムとは異なり、全社業務の全体最適を指向する。

前述の通り、既存の一般的な業務システムは業務機能毎に分かれて開発され、インターフェイスシステムを別に開発し業務システム間のデータ処理連携を行う。（図 3.1-1）これに対し、本システムでは計画から財務業務までの一連のプロジェクトのフローにおいて、各業務機能に共通の機能を統一して開発し、システムのインターオペラビリティを高めた。（図 3.1-2）また、ユーザーに



体管理コアシステムとは分けて開発を行う。(図3.2)

各業務機能を通して統一したモデルにより共通機能を統一して開発するので、開発工期・コストの大幅な削減が可能である。また、システム運用後については、ユーザーがプロセスコントロール・フローコントロールを自由に行えること、また業務仕様変更に対し既存システムのようにシステム全体を考慮する必要が無く個別業務システムを部分的に変更すればよいことにより、保守・運用コストの大幅な削減が可能である。

これらは、経営環境の変化に対して業務組織や業務仕様・プロセスの変更等の予測できないユーザーニーズの変化があり、システム耐用年数が年々短くなっている業務システムの現状を鑑みると、非常に優位な点であるといえることができる。

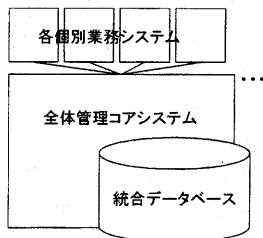


図3.2 全体プロジェクト管理システムのアーキテクチャ

### 3.3 ホモトピー保存機能

プロジェクトの全社的フローにおいて、業務ステップ、各明細データ変化のホモトピーを、記録する属性を設ける。

ホモトピー保存スキーマ

{業務ステップコード, 元業務ステップコード}  
{記録コード, 元記録コード}

このようにして、保存されたホモトピーにより、プロジェクトが全社的にフローするときのトラッキングが可能になる。(図3.3-1) また、同じプロジェクトの各業務段階、各明細はホモトピー同値であることを利用し、プロジェクトの逆トラッキングを行う機能も設ける。(図3.3-2)

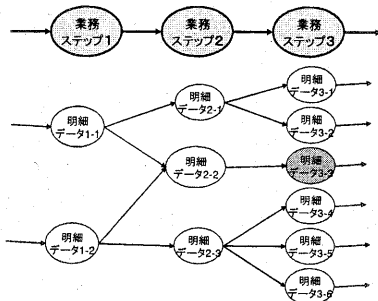


図3.3-1 保存されたホモトピーによるプロジェクトの明細のトラッキング

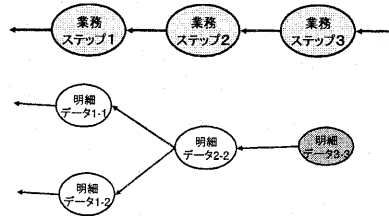


図3.3-2 保存されたホモトピーによるプロジェクトの明細のトラッキング

また、既存のシステムでは業務毎にシステムが開発されているので、受注計画等の上流の段階から支払処理等の下流の段階までのプロジェクトを計画的に管理することは困難である場合が多い。これに対し、本システムでは、プロジェクトのトラッキング機能により、プロジェクトの予定・確定・実績管理をより系統的に行うことができる。この機能は、経営管理機能等において、全社的にプロジェクトを把握する必要があるときに非常に有効である。また、そのトラッキング機能により、明細処理を行った担当者の業務責任もより明確になるという利点もある。

さらに応用として、グループ企業経営において持ち株会社の経営計画から各企業の事業計画へのホモトピーを保存すれば、経営管理が非常に効率的になると考えられる。

### 3.4 業務共通サーバーの構築

既存システムでは、業務システムに共通した統合データベース、全体管理コアシステムを構築するため、ハードウェアも統一したサーバーを利用する。システム毎にサーバーを立てる場合に比較して、サーバーの使用効率が高くになり、開発コストの削減に貢献する。

## 4. 事例研究

前述のシステム設計に基づき、プロトタイプシステムを実装した。プロトタイプシステムにおいて、筆者の一人(児玉)が所属する前田建設工業(株)のプロジェクト管理を例に取り、業務ステップ・業務プロセスの設定を行い、ユーザーのシステム利用例を通してシステムの検証を行う。利用例では、既存のシステムにおいて非常に困難であった問題点に対する本システムによる解決の事例に重点をおいた。

### 4.1 業務ステップ・業務プロセスの設定

ヒアリング調査により、使用される全ての業務ステップ・業務プロセスを決定し、登録を行った。(表4.1, 図4.1) このように、本システムでは既存の一般的なシステムとは異なり、ユーザーが業務ステップ・業務プロセスを自由に作成することが可能である。

また、各ユーザーは自分の業務に合わせて所属部門の業務機能に関する業務ステップ・プロセスを設定する。例えば、営業部門に所属するユーザーは、業務機能「営業」

の業務ステップ{概算見積・提出見積・工事入札・受注計上・失注処理}から必要な業務ステップを選択し、業務プロセスを定める。

表 4.1 業務ステップの登録

ID	業務ステップ	業務機能
K1	経営計画	管理
K2	受注計画	管理
K3	施工計画	管理
E1	概算見積	営業
E2	提出見積	営業
E3	工事入札	営業
E4	受注計上	営業
E5	失注処理	営業
S1	請負変更	施工
S2	実行予算	施工
S3	業者見積	施工
S4	発注	施工
S5	検収	施工
S6	支払依頼	施工
S7	支払査定	施工
S8	入金請求	施工
Z1	支払処理	財務
Z2	入金処理	財務
T1	部門計画	共通
T2	経費予算	共通
T3	稟議	共通
T4	契約	共通
T5	支払検収	共通
T6	支払依頼	共通
T7	支払査定	共通

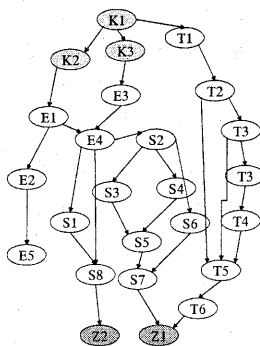


図 4.1 業務プロセスの決定

#### 4.2 明細の登録

典型的なプロジェクトを想定して、管理部・営業部・建築部・土木部・財務部に所属する各ユーザーに各明細の登録を求めた。その際、システムのホモトピー保存機能を使用して、各明細間の変化のホモトピーを保存する。

#### 4.3 プロジェクトのトラッキング機能の利用例

##### 利用例 1：プロジェクトの全社管理

経営者やプロジェクトマネージャーは、経営計画から支払・入金処理までの全社を通じた一連のプロジェクト管理情報を必要とする。このとき、本システムではホモトピー保存機能によりプロジェクトフローのトラッキングが可能なので、プロジェクトの全社管理に必要とする情報を速やかに得ることができる。以下に、実際に登録された明細データにより利用例を示す。

ユーザーはプロジェクトの全明細を図 4.3-1 のようにトラッキングが可能である。ここで、例えば経営者がプロジェクト工程の進捗を確認する必要があるときは、トラッキング機能によって、プロジェクトの上流から明細データを上から順にたどればよい。また、プロジェクトマネージャーが一連のプロジェクトにおける営業部、施工部がそれぞれ確保した利益を知りたいという要求があるときは、プロジェクトの上流から明細データを上から順にたどり、営業部の見積明細 (e2) と受注計上明細 (e4)、施工部の実行予算明細 (s2) と発注明細 (s4) のデータを探し検討すればよい。

これに対し、既存のシステムでは、業務機能毎に業務システムが異なり系統的にプロジェクトをトラッキングする機能が無いので、上記の要求に対して経営者、プロジェクトマネージャーは各部門に該当の明細データを問い合わせる必要があり、非常に時間と手間のかかる作業になる。

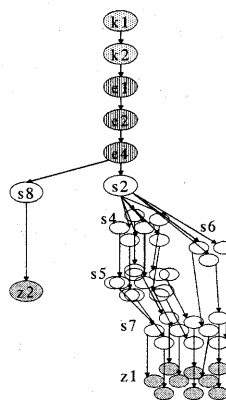


図 4.3-1 プロジェクトの全明細のトラッキング

##### 利用例 2：プロジェクトの逆トラッキング

登録された明細データにおいて、明細コード z1 を登録した財務部のユーザーである支払担当者が、プロジェクトを上流にさかのぼり、施工部での発注に関する情報を知る必要があるとする。このとき、ホモトピー保存機能により施工部が記録した明細と財務部が記録した明細の間でホモトピー同値性が保証されているので、プロジェクトの上流への逆トラッキングが可能であり、財務部のユーザーが発注情報を得ることができる。(図 4.3-2)

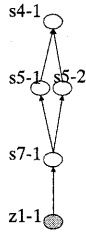


図 4.3-2 プロジェクトフローの逆トラッキング

これに対し、既存のシステムでは、一般的に複数の部門にまたがったプロジェクトの情報は一方向のみに伝達されるよう設計される。よって、上記のような要求に対しては支払担当者が施工現場に連絡を取り、支払担当者、検収担当者、発注担当者の順にプロジェクトについて問い合わせなければならない、大きな手間がかかる。

#### 4.4 業務組織変更への対応例

予測不可能な経営環境の変化に従い企業の業務組織も頻繁に変更されるが、業務システムもそれらの変更に対して柔軟に対応可能であることが望ましい。ここで、本システムでは業務ステップ・業務プロセスをユーザーが決定可能であり、組織の変更時には変更後の組織に合わせて業務ステップ・業務プロセスを決定するだけでよく、システムアーキテクチャに変更はない。よって、保守コストを低く抑えることができる。

以下に、例を挙げる。業務組織の変更で、施工部が建築部と土木部に分割されたとする。このとき、業務機能「施工」を「建築」「土木」に区別し、それぞれの機能に応じて業務ステップ・業務プロセスを決定する。(表 4.4, 図 4.4-1)

表 4.4 組織変更に対する業務ステップの決定

ID	業務ステップ	業務機能
E4-1	建築受注計上	営業
E4-2	建築受注計上	営業
A1	請負変更	建築
A2	実行予算	建築
A3	業者見積	建築
A4	発注	建築
A5	検収	建築
A6	支払依頼	建築
A7	支払査定	建築
A8	入金請求	建築
C1	請負変更	土木
C2	実行予算	土木
C3	業者見積	土木
C4	発注	土木
C5	検収	土木
C6	支払依頼	土木
C7	支払査定	土木
C8	入金請求	土木

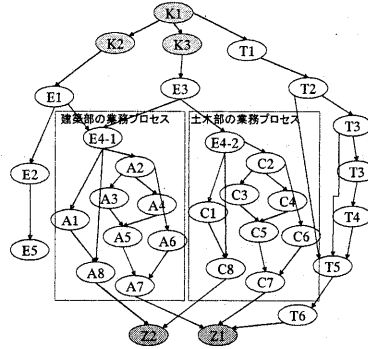


図 4.4-1 組織変更に対する業務プロセスの決定

これに対し、既存システムでは、システム設計の際に各組織に対して業務ステップ・業務プロセスが決められている。よって左記の要求に対しては組織変更後、全体のシステム設計を考慮しながら、新たに建築部生産管理システム、土木部生産管理システムを開発する必要がある。システム再開発の大きなコストが必要である。(図 4.4-2)

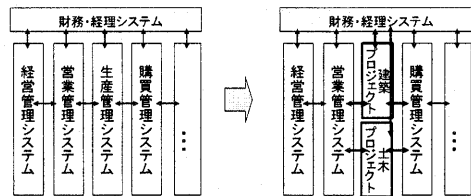


図 4.4-2 業務組織の変更に対する既存の業務システムの再開発

#### 4.5 キー属性値が同一のインスタンス変化への対応例

データベース設計時に定めたキー属性がインスタンスを一意に識別しないという予測不能なケースがしばしば発生する。本システムでは、インスタンスを必ず一意に識別する自動発番属性(記録コード)を設けることによりこの問題を解決可能とする。以下に例を挙げる。

生産管理において、発注担当者が資材 A を発注時、明細を以下のように記録した。

プロジェクト管理共通テーブル

記録コード	明細コード	明細名	金額	...
1000	S100	資材 A 発注	100	

その後、金額が変更になり再度、明細の記録を行った。このとき、当然同じ発注案件なので明細コードは変わらないが、記録コードには新しい値が自動発番される。

プロジェクト管理共通テーブル (変更後)

記録コード	明細コード	明細名	金額	...

1000	S100	資材 A 発注	100
1001	S100	資材 A 発注	120

また、明細の変更は以下のように記録される。

ホモトピー保存テーブル

記録コード	元記録コード
1000	-
1001	1000

このようにして、自動発番属性を設けることにより、インスタンスの変更は正確にデータベースに記録される。

これに対し、既存のシステム設計では、属性（明細コード）をキー属性として設計していれば発注記録の変化を記録できない。このようにキー属性がインスタンスを一意に識別しない場合は、ユーザーの判断でインスタンスをどれかに識別するよりないが、これは変化を記録したことにはならない。またデータベースを再設計すれば開発工数が大きくなる。

## 5. セルラーデータベースの利用

セルラーデータベース（以下CDB）は、著者の一人（國井）が開発したセルモデル<sup>[1]</sup>を採用した新しいデータベースであり、現在筆者らが開発中である。全社プロジェクト管理システムを開発するに当たり、既存のメジャーなデータベースであるリレーショナルデータベース（RDB）、オブジェクト指向データベース（ODB）と比較し、CDBが有効であることを示す。

### 5.1 オープンセルとクローズドセル

セルモデルには、RDB、ODBとは異なり、インスタンスを必ず一意に識別する（自動発番を行う）属性であるオープンセルの概念があるので、4.2のように自動発番する属性を改めて設ける必要がない。

オープンセルはそれ自体一つのセルとして扱われ、ハンダリーを $S^{n-1}$ 、クローズドセルを $B^n$ とすると、オープンセル $e^n$ は以下のように表せる<sup>[1]</sup>。

$$S^{n-1} = \partial B^n = B^n - \text{Int } B^n = B^n - e^n.$$

### 5.2 ホモトピー保存機能

RDBではスキーマ、インスタンスの変化を表現する機構がモデルレベルに存在しないので、4.2、4.3のようなホモトピー保存スキーマをDB設計者が設計して、アプリケーション開発者がそれに基づいたトラッキング処理を実装する、というような工夫が必要である。

また、ODBを使用すれば、クラスを継承し属性追加を表現するときに、インスタンス間の関連が表現できないので、インスタンスの変化を追跡するときにはアプリケーション開発での工夫が必要になる。

これに対し、CDBではモデルレベルでスキーマ、イ

ンスタンス両レベルでのホモトピー保存機能がサポートされていて、本システムにおける明細のトラッキング機能はインスタンスの変化として、明細属性追加時のトラッキング機能はスキーマの変化として表現可能である。よって、RDB、ODBと比較して開発工数が少なく済む。（図 5.1）

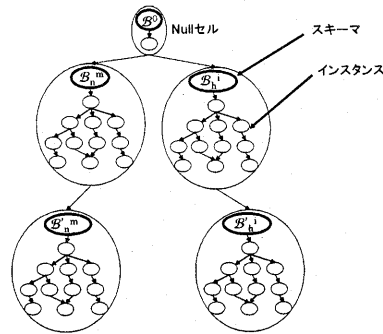


図 5.1 セルの変化構造

### 5.3 セル接合機能

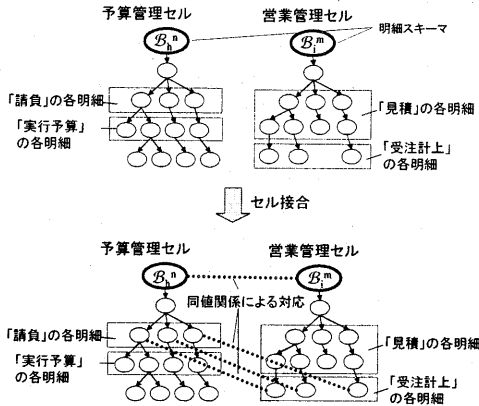
RDB、ODBには同値関係によりスキーマ、インスタンス間の対応を取るという機構がモデルレベルに存在しない。よって、全社プロジェクト管理システムの統合データベースと、それとは別に設計・開発された新たな業務システムのデータベースとを統合するときには、それぞれのシステム全体のデータ依存関係を知っているDB管理者による対応が必要である。そのとき、対応を取ろうとする属性またはその要素数の組み合わせは指数関数的に増加し「開発工数爆発」が生起する<sup>[6]</sup>。よって、統合コストは非常に大きくなる。

これに対し、CDBではセル接合関数により、モデルレベルでスキーマ、インスタンス間の同値関係の対応を取るが、その組み合わせ数は線形関数になる。つまり、セルの各属性を接着空間モデルに一回変換することでセル間の不変量を定義でき、インターオペラビリティが保証される<sup>[6]</sup>。よって、ユーザーによるデータベース統合が可能であり、統合コストは非常に小さくて済む。以下に、CDB使用時のデータベース統合の実例を上げる。

業務システムとして、営業管理システムと予算管理システムがCDBを使用して、個別に開発されていたとする。営業管理セル $(B_h^n)$ 、属性数が $n$ 、インスタンス数が $s$ のインスタンスには、業務ステップが見積・受注計上の各明細が記録されている。また、予算管理セル $(B_i^m)$ 、属性数が $m$ 、インスタンス数が $t$ のインスタンスには、業務ステップが請負・実行予算の各明細が記録されている。

ここで、2セルをセル接合により統合する。まず、2つのセルの各属性を目的同値関係の選択により接着空間モデルに変換する。そうすれば、インターオペラビリティが保証されるので、自動的に属性（明細ID、業務ステップ名、案件名）で同値関係の対応を取ることがで

きる。このときの属性変換の組み合わせ数は、 $n+m$  (属性数の和) であり、線形的に増加する。各インスタンスに対しても同様に、接着空間モデルに変換する。このときのインスタンス変換の組み合わせ数は、 $s+t$  (インスタンス数の和) であり、線形的に増加する。



$$B_n^m \sqcup_i B_n^i = B_n^m \sqcup B_n^i / \sim = B_n^m \sqcup B_n^i / (x \sim f(y) \mid \exists x \in B_n^m, \forall y \in B_n^i).$$

図 5.2 セル接合

結果として、業務ステップ〔見積・受注計上・請負・実行予算〕の順に業務プロセスが作成され、プロジェクトの明細のトラッキングが可能になる。

この例において、既存のデータベースを使用すれば、セルの各属性の組み合わせ数は  $n \cdot m$ 、各インスタンスの組み合わせ数は  $s \cdot t$  であり、指数関数的に増加するので工数爆発が生じ、統合コストが非常に大きくなる [6]。

## 6. おわりに

本稿では、業務システムの全体最適を志向した全社プロジェクト管理システムを提案した。本システムの特長は、統合データベース・全体管理コアシステム構築によるインターオペラビリティの増加、総合的なプロジェクト管理のためのプロジェクトの明細トラッキング機能である。その有効性を幾つかの実例をあげて検証した。また、本システムの構築においては、既存のデータベースよりも現在筆者らが開発中の新型データベースであるセルラーデータベースが有効であることを実例により示した。

### 参考文献

- [1] C. J. Date, "The Database Relational Model: A Retrospective Review and Analysis", Addison Wesley Publishing Company, 2000.5.
- [2] T. L. Kunii and H. S. Kunii, "A Cellular Model for Information Systems on the Web -Integrating Local and Global Information-", Proceedings of 1999 International Symposium on Database Applications in Non-Traditional

Environments (DANTE'99), November 28-30, 1999, Heian Shrine, Kyoto, Japan, Organized by Research Project on Advanced Databases, in cooperation with Information Proceeding Society of Japan, ACM Japan, ACM SIGMOD Japan, pp. 19-24, IEEE Computer Society Press, Los Alamitos, California, U.S.A.

- [3] T. L. Kunii, "Creating a New World inside Computers -Methods and Implications-", Proc. of the Seventh Annual Conference of the Australian Society for Computers in Learning in Tertiary Education (ASCILITE 89), G. Bishop and J. Baker (eds.), pp. 28-51, Gold Coast, Australia, December 11-13, 1989, [also available as Technical Report 89-034, Dept. of Information Science, The University of Tokyo].
- [4] T. L. Kunii, "Homotopy Modeling as World Modeling", Proceedings of Computer Graphics International '99 (CGI99), (June 7-11, 1999, Canmore, Alberta, Canada) pp. 130-141, IEEE Computer Society Press, Los Alamitos, California, U. S. A.
- [5] T. L. Kunii, "Web Information Modeling: The Adjunction Space Model", Proceedings of the 2nd International Workshop on Databases in Networked Information Systems (DNIS 2002), pp. 58-63, The University of Aizu, Japan, December 16-18, 2002, Lecture Notes in Computer Science, Subhash Bhalla, Ed., Springer-Verlag, December, 2002.
- [6] T. L. Kunii, "What's Wrong with Wrapper Approaches in Modeling Information System Integration and Interoperability?", Proceedings of the 3rd International Workshop on Databases in Networked Information Systems (DNIS 2003), in press, The University of Aizu, Japan, September 22-24, 2003, Lecture Notes in Computer Science, Nadia Bianchi-Berthouze, Springer-Verlag, December, 2003.
- [7] T. L. Kunii, "Web Information Modeling: The Adjunction Space Model", International Journal of Shape Modeling, World Scientific, December 1999.
- [8] Setrag Khoshafian, "Object-Oriented Databases" pp. 132-142, John Wiley & Son, 1993.
- [9] E. F. Codd, "A Relational Model for Large Shared Data Banks," Communications of the ACM, Vol. 13, No. 6, pp.377-387, June 1970.
- [10] Yousuke Watanabe, Hiroyuki Kitagawa, "Integration of Multiple Dissemination-Based Information Sources Using Source Data Arrival Properties", Proceedings of the 2nd International Conference on Web Information Systems Engineering (WISE2001), pp21-30, Kyoto, Japan, December 3-6, 2001, IEEE Computer Society Press, Los Alamitos, California, U.S.A.
- [11] Michael J. Franklin, "Middle-Tier Extensible Data Management", Vol. 4, No. 3, pp209-229, 2001, Kluwer academic publishers.