

全社プロジェクト管理システムの提案と セルラーデータベースの利用

児玉 敏男[†] 國井 利泰[‡] 関 洋一[†]

† 前田建設工業(株)情報システム SC 〒179-8903 東京都練馬区高松 5-8
‡ 金沢工業大学 IT 研究所 〒150-0001 渋谷区神宮前 1-15-13 K I T ビル 1F
E-mail: † {kodama.ts, yseki}@jcity.maeda.co.jp, ‡ tosi@kunii.com

あらまし 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毎の部分最適を指向し開発されているため、全社的に柔軟な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開発した。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メンテナビリティを高め、同時にプロジェクト内の一連の明細データのトラッキング機能を実現する。結果として、開発コスト、メンテナンスコストの大幅な削減、全社的な総合的プロジェクト管理や頻繁な組織構成の変更への柔軟な対応を可能にした。さらに、本システムの開発と機能拡張へは既存のデータベースよりも現在筆者らが開発中のセルモデルを採用したセルラーデータベースが有効であることを示した。

キーワード プロジェクト管理, 統合データベース, セルラーデータベース

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

Toshio Kodama[†] Toshiyasu L. KUNII[‡] and Yoichi Seki[†]

† Information System Company of Maeda Corp. 5-8 Takamatsu, Nerima-ku, Tokyo, 179-8903 Japan
‡ IT Institute, Kanazawa Institute of Technology 1-15-13 Jingumae, Shibuya-ku, Tokyo, 150-0001 Japan
E-mail: † {kodama.ts, yseki}@jcity.maeda.co.jp, ‡ tosi@kunii.com

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.

Keyword project management, cellular database, integrated database

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プロセス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毎の最適を指向してい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、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(“ 全社プロジェクト管理システム ”)を開発した。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. プロジェクト内の各明細データの全社的なトラッキング機能、から構成される。1, 2では、各業務においてプロジェクト予算管理に共通するスキーマを抽出、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、システムのメンテナビリティを高める。結果としてデータベーススキーマ間のプライマリーキー構造の変化、現実的にプライマリーキーの値が重複するようなケースに対する柔軟な対応が可能である。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。また、本システムの開発・機能拡張には、既存データベースでも可能であるが、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)がより有効である。これについては6章で詳述する。

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。

プライマリーキー構造の変化への柔軟性: 組織構成や業務プロセスが変更されるとき、各テーブルのプライマリーキーの構造が変化する。このときにシステムの変更を最小限にする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のではなく、業務システム全体で経営管理に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。(図 2.2-1) このとき管理対象の各プロジェクトが上流の計画段階から下流の経理段階まで推移するときの各業務処理がなされるフェーズ(以下、業務ステップ)とそれらが実行される順序(以下、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る。また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、業務プロセスの変更等、データベース設計のキー構造に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を行う必要がある。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ので、現実的には頻繁なシステム変更の要求に対応できていない場合が多い。

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図 2.2 に示す。その中では、業務(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、それぞれ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は、開発前に開発者がユーザー側と綿密に協議し、各業務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その業務プロセスを決定する。

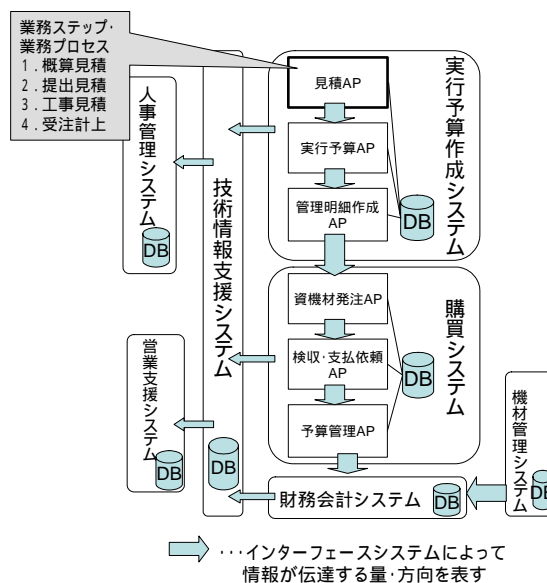


図 2.2 企業の業務システムの例

このとき、組織構成変更、業務改革等でDB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、前述の要求事項に応えるための業務システムである“ 全社プロジェクト管理システム ”を提案する。

3.1. システム概要

本稿で提案するシステムは、業務毎の個別最適を志向する既存の一般的な業務システムとは異なり、プロジェクト予算の計画管理という側面において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経理業務までの一連のプロジェ

クト内の明細データの推移において、各業務に共通なスキーマを抽出したデータベースの設計、共通な処理を統一して開発することにより、業務ステップと業務プロセスの変更への自由度が高めシステムのメンテナビリティを高める。また、プロジェクト内の明細データのトラッキング機能を設けた。結果として、業務ステップの実行とそのタイミングの決定（プロセスコントロール）またプロジェクト管理の判断（フローコントロール）が可能になり、全社的なプロジェクト管理を実現する。

3.2. 統合データベースと全体管理コアシステム

3.2.1. プロジェクト共通スキーマの抽出と自動発番属性の設定

本システムでは、データベース設計に際しプロジェクト予算管理に関し各業務共通のスキーマを抽出し業務固有なスキーマと区別したデータベース設計を行う。（統合データベースの構築）また、共通スキーマに業務ステップを一意的に識別する属性{業務ステップコード}を含め業務ステップの設定を可能にする。また、属性{記録コード}は値の自動発番を行い、明細コードの値が重複するような現実のケースへの対応を可能にする。これは 5.5 に例を示した。

- ・ プロジェクト共通スキーマ
{ 記録コード, 明細コード, 明細名, 開始日, 終了日, 業務ステップコード, プロジェクトコード, 業務, ... }
- ・ 各業務固有スキーマ
 - 営業固有スキーマ
{ 担当者コード, 取引先コード, ... }
 - 生産固有スキーマ
{ 作業所 NO, 担当者 ID, 支店コード, ... }

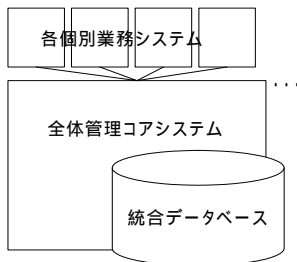


図 3.2 全社プロジェクト管理システム

表 3.2 業務ステップを共通処理別に分けた例

	計画処理	確定処理	実績処理
全社管理	事業全体計画 施工計画		
営業	概算見積 工事見積 提出見積	受注計上	入金処理
社内業務	組織別計画 管理費予算 稟議	発注, 検収	支払査定
施工	実行予算	発注, 検収	現場支払い査定
財務	資金計画		財務支払い査定

3.2.2. 各業務の共通処理の抽出

アプリケーション設計に際し、全体管理コアシステムとして、各業務に共通な案件一覧画面（図 4-1）、明細データ一覧・入出力汎用画面を設けるとともに、各業務ステップを計画・確定・実績に区別し計画処理、確定処理、実績処理を共通機能として開発する。また、各業務ステップを共通処理別に区分した例を表 3.2 に示した。他は業務毎に固有な機能の個別システムとして全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、開発コストの削減、メンテナビリティの向上を実現する。これらは、経営環境の変化に対して組織構成や業務プロセスの変更等、予測できないユーザーニーズの変化が頻繁にあり、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考慮すると非常に優位な点であるといえる。

3.3. プロジェクト内の明細データのトラッキング機能

プロジェクトの全社的推移において、業務ステップ、各明細データの変化を記録する機能を設ける。業務ステップ、各明細データの変化を記録することにより、プロジェクトが全社的に推移するときのトラッキングが可能になる。（図 3.3-1）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、全社的に特定のプロジェクトのデータ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。

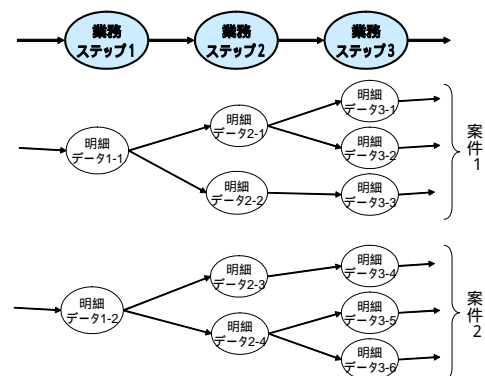


図 3.3-1 プロジェクト内の業務ステップ・明細データのトラッキング

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と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

使用効率が高くなり、開発コストの削減に貢献する。

4. システムの実装

3章の設計に基づき前田建設工業情報システム SC が著者の一人（関）を中心に本システムの実装を行った。本システムは、Web アプリケーションであり実装には JSP1.4、Web サーバーには Tomcat4.2、データベースには MySQL4.0.4(RDB)を用いた。クライアントとサーバーは同一の PC (OS: Windows2000, CPU: Intel Pentium 1.2GHz, メモリ: 512Mbyte, ハードディスク: 20Gbyte) とした。

4.1. 案件表示機能

ユーザーはログイン後、画面左側の業務ステップを選択するとその業務ステップに既登録の案件（プロジェクト）一覧が出力される。案件を新規登録時には、登録案件が前業務段階の明細データを引き継ぐかどうかを指定する。引き継ぐ場合、該当の案件を一覧から選択（元案件の選択）する。

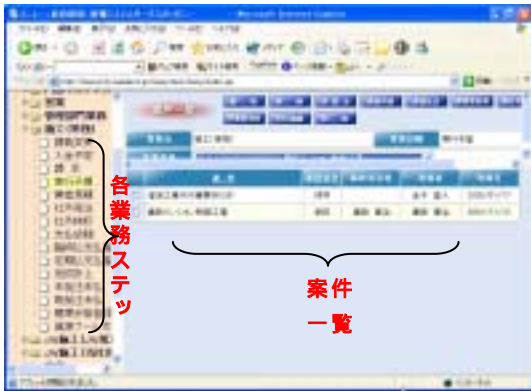


図 4.1 案件一覧画面の例

4.2. 明細データ表示・入出力機能

案件一覧から一つの案件を指定するとそこに登録されていた各明細の一覧が出力される。この一覧画面より各明細データを直接入出力する。

4.3. 明細データのトラッキング機能

明細一覧から一つの明細を指定しトラッキング機能を選択すると、指定の明細に対して下流の各業務段階の各明細が出力される。この結果の表示について、現在は明細データを表形式で出力しているが、グラフ構造等での出力表示については今後の検討課題である。

5. 事例研究

本システムにおいて、建築プロジェクト管理を例に上げ、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。

5.1. 業務ステップ・業務プロセスの設定

建築プロジェクトにおいて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 5.1、図 5.1）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、業務ステップ・業務プロセスをシステム開発後に登録可能である。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“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 5.1 業務ステップの登録

ID	業務ステップ	業務
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

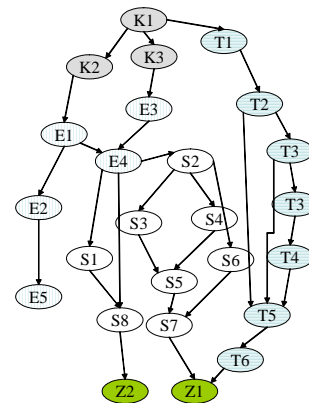


図 5.1 業務プロセスの決定

5.2. 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。その登録内容の一部を表 6.2 に示した。またその際、プロジェクトの推移のトラッキングを可能にするため、各明細間の変化の関連性を記録する。

表 5.2 プロジェクトの明細の登録例

明細コード	明細名	業務ステップ
k1	2002 年度経営計画	経営計画
k2	2002 年度受注計画	受注計画
e1	A ビル建築工事見	概算見積
e2	A ビル建築工事提	提出見積
e4	A ビル建築第 1 期	受注計上
s2	A ビル工事実行予	実行予算
s4-1	発注 1	発注
s4-2	発注 2	
...	...	
s5-1	検収 1	検収
s5-2	検収 2	
...	...	
s6-1	電気代	支払依頼
s6-2	水道代	
s6-3	雑費	
s7-1	支払査定 1	支払査定
s7-2	支払査定 2	
...	...	
s8	入金請求 1	入金請求
z1-1	支払 1	支払処理
z1-2	支払 2	
...	...	
z2	入金	入金処理

5.3. プロジェクトのトラッキング機能の利用例

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、本システムではプロジェクト推移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 5.3 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を持ってないので、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。

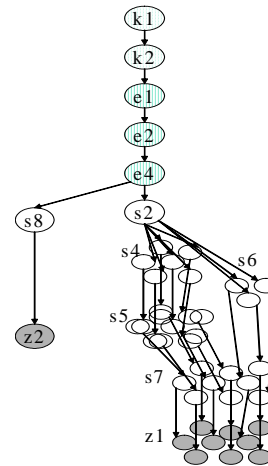


図 5.3 プロジェクトの全明細のトラッキング

5.4. 組織構成の変更への対応例

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・業務プロセスを柔軟に変更可能であり、組織の変更時には変更後に合わせて再調整するだけでよく、システムアーキテクチャに変更はない。よってメンテナンス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「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。

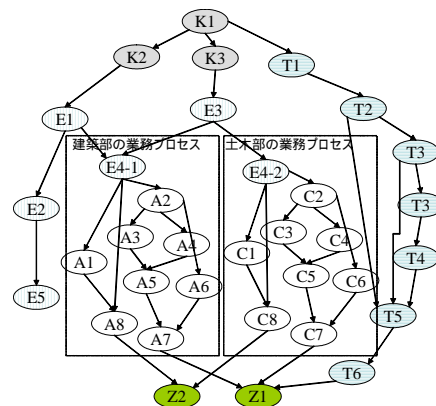


図 5.4 組織変更に対する業務プロセスの決定

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のでシステム再開発の大きなコストが必要である。

5.5. プライマリーキーの値が重複するようなケースの対応例

データベース設計時に定めたプライマリーキーの属性が現実的に重複するような、データベース設計時には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	

このとき、記録コード”1000” “1001” の変化の関係性も同時に記録される。このように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}をプライマリー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。このような場合には、ユーザーの判断でインスタンスをどれかに特定するよりないが、これは明細変更を正確に記録したことにはならない。またデータベースの再設計を行えばメンテナンスコストが大きくなる。

6. セルラーデータベースの利用

6.1. 機能拡張への課題

本システムを使用した複数のユーザーに、今後のシステム機能拡張に関する課題を調査したところ、データモデルにおける課題として、主に以下の点が明らかになった。

“プロジェクト共通スキーマ変更への対応”

システム設計時に定めたプロジェクト管理共通スキーマを変更する必要が生じたとき、データベース設計全体を見直す必要があり大きな手間がかかる。

“他の管理情報との連携機能”

本システムで実装した金額ベースのプロジェクト管理情報と他システムの管理情報(人材管理情報、資材管理情報等)を関連づけ、情報を連係させる機能

が必要である。この課題を実現するには、データモデルの理論拡張が必要であり、筆者の一人(國井)が開発した情報空間構築モデルであるセルモデル^[2]が有効であると考えられる。セルモデルはスキーマの変化を表現する機構(スキーマのホモトピー保存機能)、スキーマ属性間・インスタンス間の関連を表現する機構(同値関係によるセル接合機能)をサポートするからである。

6.2. セルラーデータベース(CDB)

セルモデルは、数学的にはセル構造空間に基づきホモトピー理論が適用されたローカルな情報とグローバルな情報とを統合可能な情報空間構築モデルである。セルモデルの詳細については[2][3][4][5][6]を参照されたい。また数学的な不変量を表す同値関係によって同値類が形成されるが、[2]によれば同値関係には数学的な抽象階層が存在する。

1. ホモトピーレベル
2. 集合論レベル
3. トポロジ空間レベル
4. 接着空間レベル
5. セル空間レベル
6. 表現レベル
7. ビューレベル

この抽象階層は下層に向かうに従い増加的に各階層の機構を継承していく。このセルモデルをデータモデルとして採用したセルラーデータベース(以下 CDB)は現在筆者らが開発中である。その CDB の特徴を以下に上げる。

セル

セル(空間)とは属性とインスタンスを区別する空間である。RDB のテーブルに相当する。

オープンセルとバンダリー

オープンセルとは、そのクローズドセルのインスタンスを必ず一意に識別するセルであり、バンダリーはその属性情報である。オープンセルはそれ自体一つのセルとして扱われ、バンダリーを 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 は以下である。

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

セル変化のホモトピー保存機能

セルが変化するとき、スキーマ、インスタンスの両レベルでのホモトピーを保存する。このホモトピー保存機能により、スキーマ、インスタンスの変化を記録することができる。また、ホモトピー同値性を利用してセルの再利用が可能になる。

セル接合機能

複数のセル間でスキーマ・インスタンスレベルにおいて同値関係の対応をとることによりセルの接合を表現可能である。このセル接合により、双方向のデータ連携を可能にする。

このセルモデルとしてデータモデルに採用した新しいデータベースが CDB である。CDB を本システムの実装、機能拡張時の実装に利用すれば、既存のデータベース(RDB, ODB) 使用時よりも、開工工

数の削減が可能であることを次に示す。

6.3. オープンセル

セルモデルには、RDBとは異なり、インスタンスを必ず一意に識別する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工夫したアプリケーションプログラムとして改めて設ける必要がない。

6.4. 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、DB設計者・アプリケーション開発者がプロジェクトのトラッキング処理の実装が必要である。また、ODBでは、プログラム中でクラスを生成できないという決定的な問題点と共に、クラスを継承し属性追加を表現するとき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変化のホモトピー保存機能がサポートされている。よって、本システムにおける明細データのトラッキング機能はインスタンス変化のホモトピー保存機能としてCDBでサポートされているので改めて開発する必要はない。また、機能拡張への課題“プロジェクト共通スキーマ変更時への対応”に対して、スキーマ変化のホモトピー保存機能としてCDBでサポートされているので機能拡張が容易である。よって、RDB、ODBと比較してアプリケーション開発工数の削減が可能である。

6.5. セル接合機能

課題“他の管理情報との連係機能”を実現するには、双方向のデータ関連付けを表現する機構が必要である。RDB、ODBにはスキーマ、インスタンスレベルともそのような機構がモデルレベルに存在しない。よって、要求を満たすにはアプリケーションレベルで機能を実装する必要がある。これに対して、CDBではモデルレベルでスキーマ属性間(それに伴うインスタンスの値間)での同値関係の対応を取るというセル接合機能がサポートされている。(図7.5)これにより、RDB、ODBと比較して、CDBを使用すれば課題の機能拡張が容易になり、開発工数・メンテナンス工数の削減が可能になる。以下に簡単な事例を上げる。

CDBを使用した業務システムで、プロジェクト管理セル(セルはRDBのテーブルに相当)と資機材管理セルが、以下のように別々に設計されていたとする。

- プロジェクト管理セル (\mathcal{B}_p^n)
- { 記録コード, 明細コード, 明細名, 開始日, 終了日, 業務ステップコード, プロジェクトコード, 資機材コード, ... }
- 資機材管理セル (\mathcal{B}_q^m)
- { 記録コード, 明細コード, 明細名, 開始日, 終了日, 業務ステップコード, プロジェクトコード, ... }
- そして、2つのセルは \mathcal{B}_p^1 : [資機材コード], \mathcal{B}_q^1 : [明細

コード]を同値セルとして次のようにセル接合される。ここで \sqcup は排他和を、 \sim は同値関係を表す。

セル分解写像 f_p, f_q とその等化写像 g, h は、

$$\begin{aligned} f_p: \mathcal{B}_p^n &\rightarrow \mathcal{B}_p^{n-1} \sqcup_g \mathcal{B}_p^1 \\ f_q: \mathcal{B}_q^m &\rightarrow \mathcal{B}_q^{m-1} \sqcup_h \mathcal{B}_q^1 \\ g: \partial^{n-1} \mathcal{B}_p^n &\rightarrow \mathcal{B}_p^1 \\ h: \partial^{m-1} \mathcal{B}_q^m &\rightarrow \mathcal{B}_q^1 \end{aligned}$$

セル接着写像 p は、

$$p: \partial^{m-1} \mathcal{B}_q^m (= \mathcal{B}_q^1) \rightarrow \partial^{n-1} \mathcal{B}_p^n (= \mathcal{B}_p^1)$$

等化写像 g は、

$$g: \mathcal{B}_q^m \sqcup \mathcal{B}_p^n \rightarrow \mathcal{B}_q^m \sqcup \mathcal{B}_p^n / \sim$$

統合セルは、

$$\begin{aligned} \mathcal{B}_q^m \sqcup_f \mathcal{B}_p^n &= \mathcal{B}_q^m \sqcup \mathcal{B}_p^n / \sim \\ &= \mathcal{B}_q^m \sqcup \mathcal{B}_p^n / (x \sim f(y)) \mid \exists x \in \mathcal{B}_p^n, \forall y \in \mathcal{B}_q^m \end{aligned}$$

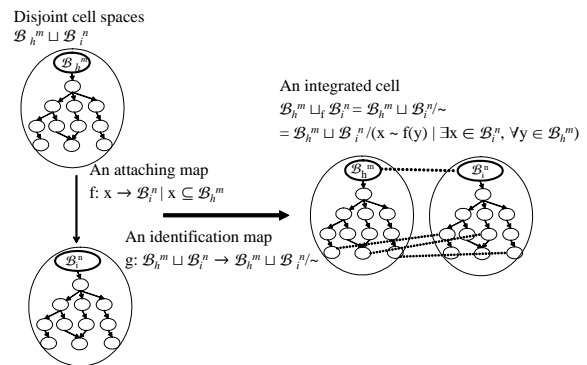


図 6.5 セル接合

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メンテナビリティの向上、総合的なプロジェクト管理のためのプロジェクト内の明細データの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。また本システムの開発と機能拡張には既存のデータベースよりも現在筆者らが開発中のセルモデルを採用したセルラーデータベースが有効であることを示した。

謝辞

本稿中の各実例の作成にあたり、貴重なご意見を頂戴した前田建設工業(株)情報システムSCの池上茂雄氏(副部長), 久村賢一氏(課長), 廣田健治氏(主任)に衷心より御礼申し上げます。

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), pp. 89-99, 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] Michael J. Franklin, "Middle-Tier Extensible Data Management", Vol. 4, No. 3, pp209-229, 2001, Kluwer academic publishers.
- [11] Ilya Zaslavsky, "XML-based Spatial Data Mediation Infrastructure for Global Interoperability", http://www.npaci.edu/DICE/Pubs/gsdi4-mar00/gsdi_iz.html.
- [12] Asuman Dogac, Yusuf Tambag, "An ebXML Infrastructure Implementation through UDDI Registries and RosettaNet PIP's", Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data (SIGMOD'02), pp.512-514, Madison, Wisconsin, U.S.A., June 4 - 6, 2002, ACM Press, New York, U.S.A.
- [13] Roy Goldman, Jennifer Widom, "WSQ/DSQ: a practical approach for combined querying of databases and the Web", Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD'00), pp.285-296, Dallas, Texas, U.S.A., May 15 - 18, 2000, ACM Press, New York, U.S.A.