ソフトウェア工学 90−19
（1993. 2. 5）

PCTEスキーマ設計にみる発想機構

沢田 篤史　鯵坂 恒夫　松本 吉弘
京都大学工学部

次世代の CASE プラットフォームとして PCTE が注目されている．本稿では PCTE リポジトリのスキーマに反映されるソフトウェア（ツール）設計者の発想機構を分類し，PCTE におけるツールの設計方法論の構築に向けての考察を行う．PCTE リポジトリのスキーマは SDS と呼ばれる拡張 ERA モデルで規定される．SDS はツールの利用するデータ構造を規定するものであり，ツール設計者による意志決定の多くが反映される．本稿では，例題として与えられた仕様を概略設計した段階での SDS と，プログラミングを念頭に置いた詳細設計後の SDS とを比較することにより，詳細設計段階での設計者の発想を抽出，類型化する．この結果に基づき PCTE によるツール構築プロセスの定式化の可能性について考察する．

The Design Decisions Reflected on PCTE Schema

SAWADA Atsushi　AJISAKA Tsuneo　MATSUMOTO Yoshihiro
Faculty of Engineering, Kyoto University
Yoshida honmachi, Sakyo-ku, Kyoto 606-01, JAPAN

The SDS's (Schema Definition Sets) define the schema of data repository which is used by the PCTE softwares (tools). The SDS's reflect much of the design decisions which are made during the tool development on PCTE. In this paper, we extract the typical decisions which are made by the tool designer of detailed design step. Two kinds of example SDS's are compared: one is a result of conceptual design and the other is precisely designed for the following programming step. Based on the result of extraction, the design method for PCTE tool is also discussed.

1 はじめに

本稿では PCTE (Portable Common Tool Environment) [1], [2] でのツール開発方法論構築の基礎として，リポジトリのスキーマに反映される設計者の発想機構について考察する．ここでいう PCTE ツールとは PCTE で動作するソフトウェア一般を指し，CASE ツールに限ったものではない．

PCTE リポジトリのスキーマは SDS (Schema Definition Set) と呼ばれる拡張 ERA モデルにより規定される．本稿では PCTE によるツール設計を二つの例題について行い，SDS に反映された発想機構の類型化を試みる．

以下，2節で PCTE と SDS に関する概説を行い，ついで3節で二つの例題の設計過程における発想機構について類形化を行う．4節では PCTE ツール設計プロセスの定式化と計算機支援に関する考察を行い，5節でまとめをする．

2 PCTE と SDS

PCTE は，ソフトウェア・エンジニアリングを支援する CASE 環境のプラットフォームとして，リポジトリ・インタフェースの標準化に基づくデータ統合とツール可搬性の保証とをその目的としている．

PCTE では，SDS と呼ばれる拡張 ERA モデルによりリポジトリのスキーマを定義し，そのスキーマに従ってリポジトリに保持されるデータを利用するツールの構築が可能である．

SDS の要素となるデータ型には，**オブジェクト型**，**リンク型**，**関係型**，**属性型**がある．以下，各データ型について説明する．

SDS によるスキーマの記述例を図 1に示す．

四角形で示されている**オブジェクト型**はツールによる種々の操作の対象となる基本的なエンティティを表すデータ型であり，内部の文字列は型名を示す．新たな型は既存の型のサブタイプとして定義される．この継承木の頂点として 'object' と呼ばれるオブジェクト型が与えられ

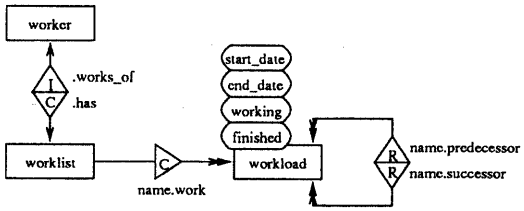

図 1: SDS によるスキーマ記述例

ているほか，'file'，'dir' など，システムに既存の型が与えられており，これらを SDS に**移入** (import) して用いることもできる．なお，継承関係は図には表されていない．

三角形で示されている**リンク型**は，オブジェクト型間に存在する有向関係を表すデータの型である．内部の文字はリンクの**カテゴリ**であり，**C: Composition** は構成複合関係を，**R: Reference** は参照関係を，**I: Implicit** は暗示的関係を示す．傍らに示される '.' で区切られた文字列のうち，最右の文字列がリンク型名を表し，左側のものはキー属性型名を表す．キー属性型を指定することで，リンク型は 1 対多の関係を表すことができる (Cardinarity Many: 二重の矢印)．指定しない場合は 1 対 1 の関係を表す (Cardinarity One: 一重の矢印)．例えば，図中 'name.work' で示されるリンク型は，一つの 'worklist' 型オブジェクトが複数の 'workload' 型オブジェクトをその構成要素として持つことができ，それらは 'name' 型のキー属性により一意に識別されることを示している．

関係型はオブジェクト型間に定義され，互いに逆方向を持つ二つのリンク型の対で表されるものである．図中では，'.has' + '.works_of' などが定義されている．

長円形で示されている**属性型**は，オブジェクト型あるいはリンク型に適用され，文字列，整数，真偽値，日付の値を持ちうる．前述のキー属性型は属性型の特別な場合である．内部の文字列は属性型名を示す．

3 SDS に反映される設計者の発想機構

本節では，二つの例題を用い，概略設計段階での SDS と詳細設計後の SDS とを比較する．できる限り広い範囲のツールをカバーすべく，例題としては制御系システムであるエレベータ制御の例題 [3]，事務処理系システムである酒類販売会社の例題 [4] を選んだ．

概略設計段階においては，プログラム論理やシステム機能の利用などは念頭におかず，仕様に示された対象領域の物理的，論理的構造をできるだけ忠実に SDS として表現する．

一方，詳細設計では，プログラム論理や利用するシステム機能などが決定され，それに応じて概略設計で得られた SDS が改変される．

このようにして得られた二つの SDS の差分に詳細設計段階に介在する設計者の発想機構が反映されるはずであり，本節ではこれらの類型化を試みる．

3.1 問題 A: エレベータ制御の例題

例題の仕様の概略を以下に示す．

> n 基のエレベータが m 階建てのビルに設置されており，搬器とその駆動制御機構は与えられているものとする．与えられた制約の下で各搬器の上下動作を決定するための制御論理のプログラミングを行え．

この例題に対する概略設計段階の SDS を図 2に示す．システム機能の利用を前提としないため，オブジェクト型はすべて 'object' 型をそのスーパタイプとしている．PCTE システムに既存のオブジェクト型の移入も 'object' 型を除いて行われていない．

次に詳細設計後の SDS を図 3に示す．ここでは，エレベータの制御システムが，搬器や各種ボタンからのイベント系列を入力し，それに応じてリポジトリの内容を書き換え，搬器，各種ボタンへの指令を出力するものとして設計を行った．また，エレベータ・システムの管理者 ('site_manager') は PCTE のユーザであると仮定し，警告信号 ('warn_signal') はそのユーザへのメッセージ (電子メイルまたは，端末への出力) であるとしている．

詳細設計段階を経て，SDS には，

- PCTE システム既存型の新規移入.
- PCTE システム既存型の移入と代用.
- 関係型の付加.
- 逆リンク型の付加.
- リンク型のカテゴリ変更.
- スーパタイプの変更.
- 属性型の付加.

といった変更が加えられた．

3.2 問題 B: 酒類販売会社の例題

例題の仕様の概略を以下に示す．

> 酒類販売会社のコンテナ入庫処理，酒の受注，出庫依頼表作成，在庫不足連絡，在庫不足リスト作成のための計算機プログラムを作成せよ．

この例題に対する概略設計段階の SDS を図 4に示す．リポジトリへの保存が必要な伝票は在庫不足リスト ('shortage_list') のみであることが仕様からわかるため，他の伝票はオブジェクト型として表されていない．PCTE システムの既存型は 'object' のみが移入されている．

詳細設計後の SDS を図 5に示す．ここでは，受付係プログラムが標準入力から積荷表あるいは出庫依頼を受け，出庫指示書あるいは在庫不足連絡を標準出力に行うものとして設計した．このため，問題 A での 'sys_IO' のような入出力のためのオブジェクト型は移入されていない．

詳細設計段階を経て，SDS には，

- PCTE システム既存型の新規移入.

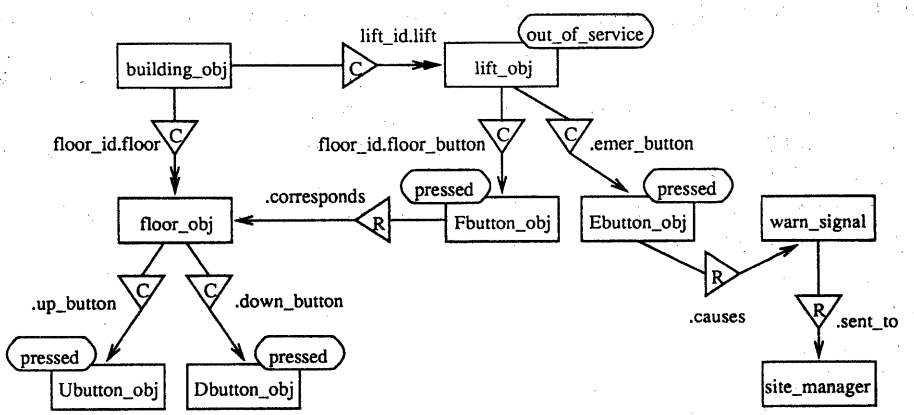

図 2: エレベータ制御の例題: 概略設計

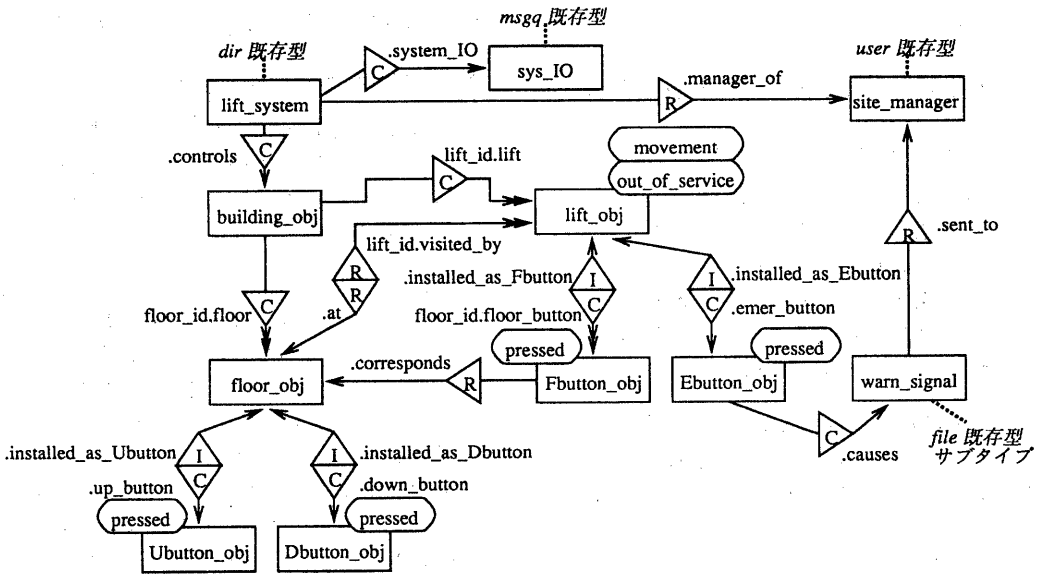

図 3: エレベータ制御の例題: 詳細設計

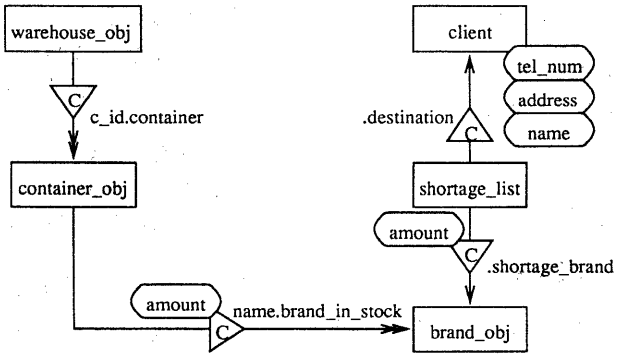

図 4: 酒類販売会社の例題: 概略設計

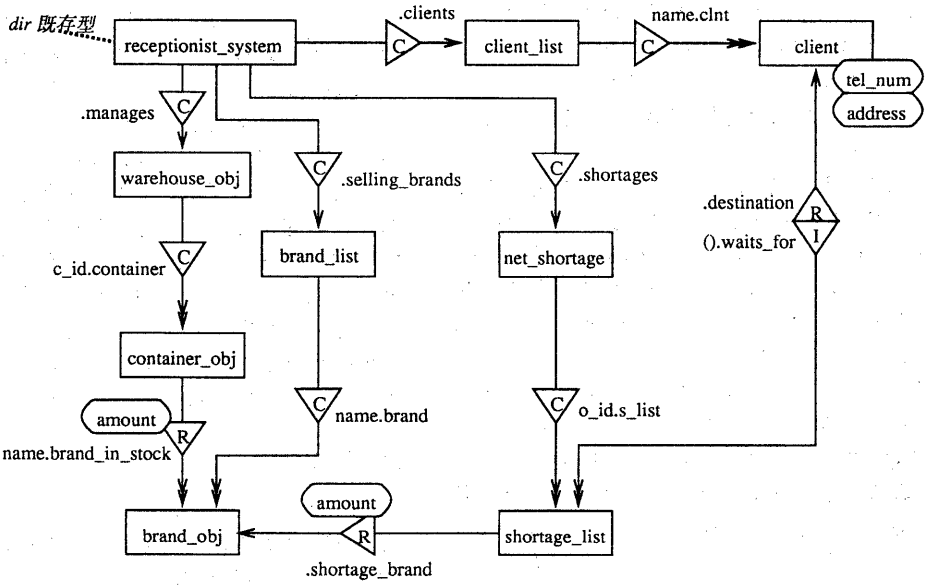

図 5: 酒類販売会社の例題: 詳細設計

- オブジェクト型の新規作成.
- 属性型のキー属性型化.
- リンク型のカテゴリ変更.
- 逆リンクの付加.

といった変更が加えられた.

3.3 発想機構の類型化

以下，詳細設計の段階を経て SDS に加えられた変更の種類毎に，その変更を行った設計者の動機 – 発想機構 – を類型化する.

PCTE システム既存型の新規移入

第一に，PCTE システムの制約，すなわち「オブジェクトがリポジトリ上に存在するためには，既存のオブジェクトから少くとも一つの Composition リンクが流入できなければならない. 」を満たすための変更であると考えられる. 例として問題 A の 'lift_system' 型などが挙げられる. これは既存型 'dir' を新規移入したものである.

第二に，PCTE システムによって提供されている機能を利用するための変更であると考えられる. 例として問題 A の 'sys_IO' 型が挙げられる. これは，既存型 'msgq' の移入により PCTE システムのメッセージ通信機能を利用しようとするものである.

第三に，リポジトリ上での管理の視点を変えるための変更であると考えられる. これはツールが複数の同一型のオブジェクトを扱う必要がある場合，それらを一括して管理するための変更である. 例としては現れないが，問題 A でシステムが扱うビルが複数であると仮定すると，それらを一括して取り扱う目的で 'dir' 型として移入された 'lift_system' 型がこの例となる.

PCTE システム既存型の移入と代用

PCTE システムの制約を満たすための変更，PCTE システムによって提供される機能を利用するための変更の二つの場合が考えられる. 例として問題 A の 'site_manager' 型が挙げられる. これは既存型 'user' を移入することにより，リポジトリ上で存在するための制約を満たし，同時に PCTE システムの提供するユーザ関連の機能を利用しようとするもので，両方の場合の例となっている.

スーパタイプの変更

PCTE システムによって提供される機能を利用するための変更であると考えられる. 例として問題 A の 'warn_signal' 型が挙げられる. これは，'site_manager' に送られる警告メッセージの保持に，PCTE ファイル ('file') を利用しようとするものである.

リンク型のカテゴリ変更

第一に，PCTE システムの制約を満たすための変更であると考えられる. 例として問題 A の '.causes' リンク型が挙げられる. この変更により，'warn_signal' オブジェクトがリポジトリ上に存在することが保証される.

第二に，リポジトリ上での管理の視点を変えるための変更であると考えられる. 例としては問題 B の '.destination' 型が挙げられる. これは，'client_list' 型の新規作成 (後述) に呼応したものである. この種の変更は，同一の目的で行われるオブジェクト型の新規作成やリンクのカテゴリ変更に呼応して行われる.

関係型の付加

第一に，PCTE システムの制約を満たすための変更であると考えられる. 変更は同一の目的で行われる既存型の移入などと呼応して行われ，通常 Composition リンク型を含む関係型が付加される. 例として問題 A の '.controls' 関係型 (リンク型) などが挙げられる.

第二に，ある型のオブジェクトからある型のオブジェクトを直接参照する手段を提供するための変更であると考えられ，通常 Reference リ

ンク型を含む関係型が付加される．例としては問題Aの'.at'＋'lift_id.visited_by'関係型が挙げられる．これは，搬器が現在どのフロアにあるのか，フロアにどの搬器があるのかを相互に直接参照するためのものである．

オブジェクト型の新規作成

リポジトリ上での管理の視点を変えるための変更であると考えられる．例としては問題Bの'client_list'型などが挙げられる．これは，出庫依頼主を表す'client'型のオブジェクトを一括し，依頼主リストとして管理するものである．一般にオブジェクト型の新規作成は，それに関連する関係型の付加など，他の変更を波及する．

属性型のキー属性型への変更

リポジトリ上での管理の視点を変えるための変更であると考えられる．例としては問題Bの'client'型に適用されていた'name'属性型が挙げられる．これは'client_list'型の新規作成に呼応したものである．

逆リンク型の付加

第一に，複数オブジェクトからの共有を抑制するための変更であると考えられる．例として，問題Aの'.installed_as_Fbutton'型などが挙げられる．ここでは，逆リンクをCardinarity Oneとすることで複数の搬器オブジェクトが同一の行き先ボタンオブジェクトを共有する状況を抑制している．

第二に，明示的直接参照を可能にするための変更であると考えられる．例として，問題Bの'().waits_for'型などが挙げられる．ここでは，顧客側からそれに関する在庫不足リストの明示的な参照を可能としている．

属性型の付加

オブジェクト型の内部状態を表現し，利用するための変更であると考えられる．例として問題Aの'lift_obj'型に搬器の動作状況表すために付加された'movement'型が挙げられる．

なお，例題には現れなかったが，その他の分類に属するSDSの変更として，類似した意味関係を表す関係型をまとめて扱うために行われる**関係型の合併**，類似したオブジェクト型をまとめて扱うために共通のスーパタイプを新設する**抽象オブジェクト型の作成**などが考えられる．

4 設計プロセスの定式化へ向けて

前節で抽出，類型化された発想機構とSDSに加えられた変更との因果関係は，図6のようにまとめられる．これにより，ツール設計者の発想に対してどのような変更がSDSに行われるかを知ることができ，結果としてSDSの設計過程における計算機支援として，次のような機能の提供が有効であることがわかる．ただし，図中の**A**, **B**に関してはPCTEの利用を前提とした設計上の発想であり，他とは区別して扱うべきであろう．

- SDSの形状からシステムの制約に抵触する可能性のある部分を設計者に指摘する機能．(図6の**A** → **1**, **2**, **4**, **5**)
- PCTEシステム機能に関するヘルプ機能とSDS設計過程とを有機的に組合せて設計者を支援する機能．(**B** → **1**, **2**, **3**)
- リンク型によって明示的に連結されていないオブジェクト型間の直接アクセスを設計者との対話によって実現する機能．(**D** → **5**, **8**)
- 複数のオブジェクトによる同一オブジェクトの共有を可能とするようなスキーマをSDSより抽出し，設計者に指摘する機能．(**E** → **8**)
- オブジェクト型に連結するリンク型の状況からそのオブジェクトに一般的な内部状態を導出し，設計者に提示する機能．(**F** → **9**)

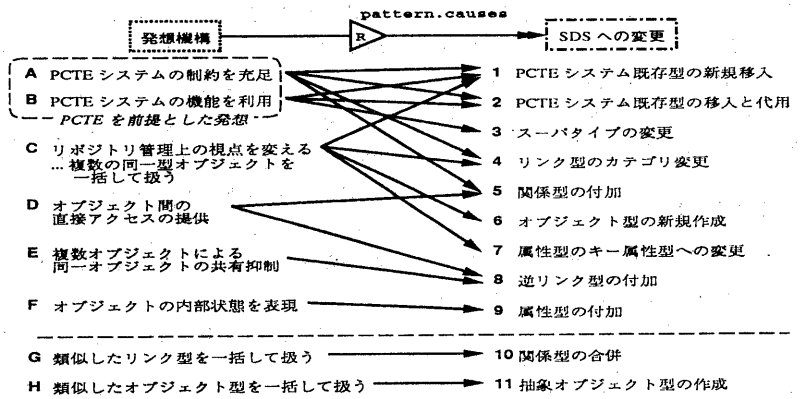

図 6: 発想機構と SDS への変更との因果関係

- SDS の形状から同一リンク型としてまとめることのできるリンク型の組を設計者に指摘する機能. (G → 10)

設計プロセスの定式化に関しては，関連する一まとまりの SDS 変更作業によって一つの発想を実現する場合のプロセスを定式化することにより，設計プロセスの一断片を定義することが可能になると考えられる．しかしながら，実際の設計においては一つの発想に対して行われた変更に応じてさらなる変更の要求が生まれるはずで，このような発想機構の間の因果関係を定式化することが PCTE ツール設計の方法論構築につながると考えられるが，本稿の範囲ではそのために十分な結果は得られなかった．

5 おわりに

本稿では，PCTE の SDS に反映されるツール設計者の発想機構を例題の設計過程から抽出し，PCTE ツールの設計方法論構築の可能性について模索した．

今後の課題としては，さらに多くの例題についてこのような検討を行い，一般的な設計プロセスの定式化と支援システムの構築へ発展させて行くことが挙げられる．

謝 辞

本研究に関して文部省科学研究費重点領域研究「高度ソフトウェア」 B01 班 (ソフトウェア構成法における発想機構の研究) 参加メンバの方々に御討論を頂いている．ここに謝意を表す．

参考文献

[1] "Portable Common Tool Environment (PCTE) Abstract Specification," *Standard ECMA-149,* (1990).

[2] 鯵坂 恒夫，沢田 篤史，満田 成紀: "ソフトウェア評論: Emeraude PCTE," コンピュータソフトウェア, Vol. 10, No. 2, (1993), (to appear).

[3] "Problem Set for the 4th International Workshop on Software Specification and Design"

[4] 山崎 利治: "共通問題によるプログラム設計技法解説，" 情報処理, Vol. 25, No. 9, (1984), pp. 934.