

インタフェースソフトウェア部品の再利用モデル

岩根 典之* 木下 哲男**

*広島市立大学 情報科学部

**沖電気工業(株) マルチメディア研究所

ソフトウェアの利用/再利用を支援するための部品化モデルとインタフェースソフトウェアを対象とした部品知識の表現/利用モデルの枠組みについて述べる。部品化モデルは、ドメイン分析の手段を与える表現モデルと段階的知識獲得の手段を与える利用モデルからなる。表現モデルからインタフェースソフトウェアの部品知識の表現形式が構成される。更に、利用モデルに基づいてインタフェースソフトウェア部品の利用/再利用を促進するための部品知識の表現形式が構成される。

REUSE MODEL FOR INTERFACE SOFTWARE PARTS

Noriyuki Iwane* Tetsuo Kinoshita**

*Faculty of Information Sciences, Hiroshima City University

**Multimedia Laboratories, OKI Electric Industry Co., Ltd.

Models for representing and utilizing knowledge of interface software modules, based on a model for formalizing software parts to support use/reuse, will be discussed. The formalization model consists of two models, i.e., the representation model and utilization model of parts knowledge which offer methods for domain analysis and incremental knowledge acquisition. The knowledge representation language for interface software parts is constructed from the representation model. On the other hand, use/reuse oriented knowledge structure is constructed from the utilization model.

1. はじめに

ソフトウェアシステムの生産性を向上するためには、ソフトウェアの部品化と再利用は重要なプロセスである。しかし、いかにして再利用しやすい部品化を行うか、また、いかにして部品の再利用性を上げるかが問題である。こうした中で、ソフトウェア工学的観点からドメインを限定した分析とモデリング¹⁾に基づく系統的再利用の重要性が認識されてきた。ドメイン分析では、対象となるドメインの知識を収集・整理してドメインモデルを構築する。十分な分析に基づいたドメインモデルは、そのドメインを理解したり、そこから直観を得たりするのに役に立つ。すなわち、ドメインモデルはソフトウェアの再利用に有効である。一方、ドメイン分析は知識工学的観点とも密接に関係する。ソフトウェア部品のための知識表現モデルはドメインモデルの一部と考えられる。知識の表現構造と表現形式はドメイン分析により得られる。この時、知識の利用は部品の利用に、知識の獲得は再利用に対応する。そして、知識ベースは部品のリポジトリに対応する。更に、部品は再利用により進化し、それに伴いドメイン分析も繰り返される。その結果、ドメインモデルはより良いものに進化し、知識表現モデルもリファインメントされる。

我々は、知識工学的観点からインタフェースソフトウェア部品の知識ベース化を目指して知識表現モデルの構築に取り組んできた^{2)・3)}。この知識表現モデルの基本となる考えは、再利用と部品化の捉え方である。一般に、再利用は様々な知識レベルで行うことができる。しかし、レベルの違いは意味的なギャップとなり、その対応関係を取る(理解する)ことの難しさが再利用を困難にする。そこで、我々のアプローチでは、再利用に3つの知識レベルに基づく部品化モデルを導入し、上位のレベルから段階的に再利用を試みる。すなわち、部品化モデルは部品の利用に関する知識(部品知識)の構造化と再構造化の2つのモデルからなる。前者は部品知識表現モデル、後者は部品知識利用モデルと呼ばれ、どちらも部品知識の表現形式を与える。部品知識表現モデルは個々の部品の部品知識を形式化するためのものである。また、部品知識利用モデルは部品セットの部品知識を利用/再利用に適した形式にするためのものである。インタフェースソフトウェアの部品化と再利用はこの2つのモデルに基づいて行われる。

以下、2. では我々の提案する再利用とソフトウェアの部品化の基本的枠組みについて述べる。そして、3. でインタフェースソフトウェア部品のための知識表現モデルについて述べ、部品の再利用と知識獲得プロセスについて議論する。

2. ソフトウェアの部品化モデル

2.1 部品とその再利用

部品化手法と再利用手法は密接に関係している。部品化により再利用がはじまり、また様々なレベルの再利用により部品化が見直されることになる。再利用を容易にするには、部品に対して、何ができるかだけでなく、何のために利用できるかという知識を付加して部品を提供すること、また再利用の場面でもそのような知識を獲得蓄積することが必要となる。更に、部品として組み込まれるソフトウェアモジュール群を与えると共に、それらソフトウェアモジュール群を部品として参照・適用するための様々な知識(部品知識)を整備する必要がある。こうした部品知識は、部品の設計や利用経験から得られる知識であり、部品の効果的な利用/再利用を実現するためには必要不可欠である。そこで、ソフトウェア部品を、ソフトウェアシステムに組み込まれる部品本体であるソフトウェアモジュール(SM)と個々のSMの部品知識を提供する知識モジュール(KM)に分けて考える。KMを構成する視点に設計者の視点(dv)と利用者の視点(uv)を設定する。dvは、既存/新規のSMを部品として設計/定義する際の視点であり、uvはソフトウェアシステム設計プロセスにおいてSMの利用/再利用を行う際の視点である。これら視点に基づいたソフトウェアモジュールの構造化プロセスの枠組みは模式的に表すと次のようになる。

部品の構造化プロセス：{目的(uv)+機能(dv)+構造(dv)}

⇒ {<用途(dv)>+<用法(dv)>+<コード・仕様(dv)>}

ここで、「⇒」は部品知識の獲得と構造化プロセスを、「<>」は既知知識の記述を意味する。各項目のuvとdvは、それぞれ利用者視点と設計者視点を表す。すなわち、ソフトウェアモジュールが提供する機能(dv)は、目的(uv)を(設計)根拠として設計され、構造(dv)を持つ部品として提供される。「目的(uv)+機能(dv)+構造(dv)」は、それが明示的もしくは暗黙的に存在する(式は設計時点で存在したはずの)ソフトウェアモジュールに係わる知識を意味する。

一方、部品の構造化プロセスで形式化された部品知識の活用形態は、利用者要求や設計条件などによって与えられる実際の部品利用の目的(uv)に応じて次の4種類のパターンに分類される。まず、目的(uv)と部品の用途(dv)の完全適合によりソフトウェアモジュールの機能が利用される場合は、これを通常利用と呼び、模式的に

通常利用：<目的(uv)>⇔<用途(dv)>→<用法(dv)>.

のように表記する。また、利用目的と用途知識の部分適合によりソフトウェアモジュールの機能を利用する場合は、これを再利用と呼び、

再利用1：<目的(uv)>⇔<用途(dv)>→<用法(dv)>.

再利用2：<目的(uv)>⇔<用途(dv)>→<用法(dv)>.

再利用3：<目的(uv)>⇔<用途(dv)>→<用法(dv)>+<コード・仕様(dv)>.

のように表記する。ここで、「<>」は既知の知識に対する新たな部品知識の追加・修正を表す。また、「⇔」は利用目的(uv)と用途(dv)との適合とを表し、「→」は部品の利用目的に応じた用法(dv)の適用を表す。よって、再利用では、既知の用途(dv)に対し何らかの発見的知識の補完により目的(uv)と適合することを意味している。再利用1は、想定された用途(dv)以外に既存の利用形態でソフトウェアモジュールが用いられる場合、再利用2は用法(dv)の追加・修正により、目的(uv)に適したソフトウェアモジュールの利用形態が実現できる場合、また、再利用3は用法(dv)とコード・仕様(dv)の追加・修正により、目的(uv)に適したソフトウェアモジュールの利用形態が実現できる場合に対応している。このとき、再利用1と再利用2では部品をブラックボックスと看做して再利用を行う(狭義の再利用と呼ぶ)ことに対応している。また、再利用3は、部品をホワイトボックスと看做して再利用を行う(広義の再利用と呼ぶ)ものであり、再利用3ではソフトウェアモジュール自身の再設計と再実装が行われる。一般に、こうした再利用が行われたソフトウェアモジュールに関する新たな部品知識が既知の知識として適宜獲得・追加されるならば、部品の利用形態は通常利用のクラスに縮退する。

2. 2 部品知識の表現モデル

部品知識の表現モデルは部品の構造化プロセスの考え方に基づいた部品知識の構成手段を与える(図1)。本モデルに基づいた部品知識の構成法では、各ドメイン毎に部品知識の構造を分析する。そこでは、SMの目的(uv)と機能(dv)と構造(dv)に対し、それぞれ用途(dv)と用法(dv)とコード・仕様(dv)の枠組で分析が試みられる。ただし、用法やコード・仕様は、普通はすでに形式化されているので、用途に関する知識の分析が中心となる。また、SMの対象となるのはオブジェクト指向言語でいうところのオブジェクトや関数型言語の関数などである。よって、用法はオブジェクトであれば具体的なメッセージ式であり、また、関数であれば具体的な引数が入った関数呼び出しである。この分析の結果、表現モデルの具体的な知識表現形式は、部品化の対象となるSMの性質を反映したオントロジーに基づく属性要素記述群(記述プリミティブ)を中心に定義される。そして、記述プリミティブは既存のSMの部品知識の記述やSMの実装の基礎に利用される。

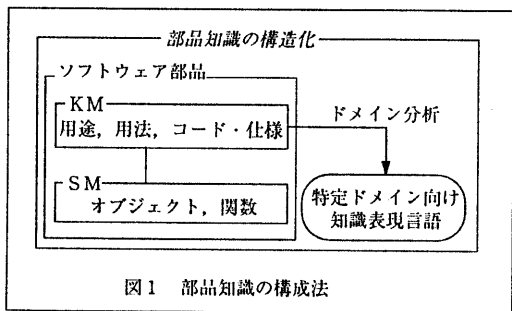


図1 部品知識の構成法

2.3 部品知識の利用モデル

部品知識の利用モデルは通常利用／再利用の考え方に基づいた部品知識の構成手段を与える(図2)。利用モデルでは、表現モデルに従って記述された複数のソフトウェアモジュール(SM)の部品知識が論理部品(LP)、物理部品(PP)、関係(R)と呼ばれる形式に変換され、LPとPPは関係(R)によって結合される。表現モデルから利用モデルへの変換は次のステップで行われる。

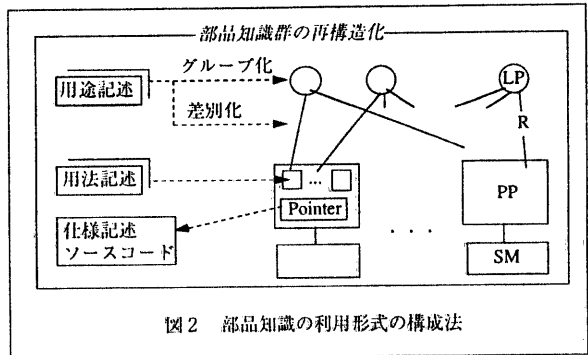


図2 部品知識の利用形式の構成法

(1) 論理部品の導出：用途に共通性がある

SMをグルーピングする。或るいは、予め実装時に想定したカテゴリに属するSMに着目する。この時、同一グループ内の各SMの用途に共通する属性プリミティブの記述セットをLPとする。

(2) 関係の導出：同一グループ／カテゴリのSMの部品知識の記述から共通用途記述LPを除いたものをRの用途に関する属性プリミティブの記述セットとする。更に、LPとPPを関係付けるリンクに関する記述を追加する。

(3) 物理部品の導出：用法とコード・仕様へのポインタとSMを利用するための用途別の用法の記述をPPとする。用途別の用法は、図2で示されるようにLPURの用途に対応する。

上述した変換の結果、複数のソフトウェアモジュール群から利用者の目的(uv)に応じて適切な用途(dv)を持つソフトウェアモジュールの選択が可能となり、選択されたソフトウェアモジュールをその用途(dv)で利用するための用法(dv)が抽出できる。

3. インタフェースソフトウェア部品の再利用

3.1 ドメイン分析

ここでは、視覚的部品と呼ばれるインタフェース設計で利用される様々なソフトウェアモジュールのひとつのクラスを対象とした枠組みについて述べる。

視覚的部品は、以下の3つの機能を有するソフトウェアモジュールとして定義される。

- (1) インタフェース表現において1つ以上の視覚的構成要素の生成・表示を行う機能、
- (2) 利用者や他のソフトウェアモジュール(アプリケーションソフトウェアモジュールや他の視覚的部品など)との相互作用や通信を解釈・処理する機能、
- (3) 他のソフトウェアモジュールの起動、或はそれらのモジュールの応答を受信する機能。

視覚的部品は、インタフェースソフトウェアの実行時に表示機能を実行することによりインタフェース表現の一部もしくは全体を利用者に提示する。更に、インタフェース表現に対する利用者の操作の解釈、及びアプリケーションソフトウェアモジュールや他の視覚的部品との通信を行う。

視覚的部品の部品知識は、利用者の視点(uv)に関係する実行時視点と設計者の視点(dv)に関係するコーディング時視点によって獲得・整理される。実行時視点では、視覚的部品がインタフェースソフトウェアの部品として稼働する際の構造に基づいて、各部品の持つ用途に関する知識が整理される。一方、コーディング時視点では、インタフェースソフトウェアに視覚的部品を組み込む際に必要とされる用法、コード・仕様に関する知識が整理される。視覚的部品の場合、用途は利用者の動作、部品の表現、これらに関連する効果から既存のマニュアルや設計者を知識源として分析を行う。同様に、用法では、部品を組み込む際のプロトコルのカテゴリを生成、結合、配置から分析を行う。

3. 2 部品知識の表現形式

[1] 表現モデルに基づいた表現形式

図3は、3.1で述べたドメイン分析による視覚的部品の部品知識の表現形式である。同図において下線を付した非終端項は、図4に示す記述プリミティブが選択的に割り当てられるものを示す。

【部品知識表現形式の基本構造】実行時視点に基づく<用途>の記述、及びコーディング時視点に基づく<用法>と<コード・仕様>の記述からなる。

【用途知識の記述】視覚的部品の利用目的を、インタフェース表現に対する利用者の行為とその効果によって表現する。利用者の行為は、その行為の前提として存在する視覚的部品の部品表現に対する有意なく利用者動作>の記述からなる。一方、<効果>の記述は、利用者の動作単位に応じた視覚的部品の挙動とその意味を表現する。

【部品表現の記述】視覚的部品が利用者に対して提示するインタフェース表現の論理構造とその出現/消滅の形態を宣言的に表現する。ここで、<論理的視覚構成>の記述では、表象の記述属性である<観点>や<対象クラス>などを指定することで、表象の表現レベルやその意図/目的などが表現される。

【利用者動作の記述】インタフェース表現に対して可能な利用者の動作が、利用者の知覚と運動に関する記述プリミティブ群を組み合わせることによって表現される。なお、この記述では、アイコンボタンのクリックによるポップアップメニューの提示のように、操作対象と認識対象が異なるインタラクションも表現できる。

【効果の記述】視覚的部品の機能を、利用者に対する応答動作や他のソフトウェアモジュールに対する動作の記述プリミティブを指定することによって表現する。

【用法知識の記述】視覚的部品の本体であるソフトウェアモジュールをインタフェースソフトウェアの部分要素機能として利用する際の部品の使い方に関する知識を記述する。ここでは、視覚的部品のソフトウェアモジュールの起動手順、他のソフトウェアモジュールとの接続手順、或はインタフェース表現要素の表示位置の指定手順などが、3種類の情報交換形式(生成/結合/配置プロトコル)によって表現される。

【コード・仕様知識の記述】視覚的部品の詳細設計情報を与えるもので、部品本体のソフトウェアモジュールの設計仕様ファイルやソースコードファイルへのポインタによって指定される。

```
<部品知識表現モデル> ::= <用途> <用法> <コード・仕様>
<用途> ::= <部品表現> <利用者動作> <効果> { <利用者動作> <効果> } *
<部品表現> ::= <出現形式> <論理的視覚構成> <消滅形式>
  <出現形式> ::= <出現タイプ> <位置> { <方向> <形態> }
  <論理的視覚構成> ::= <論理的視覚要素> <表象> { <論理的視覚要素> <表象> } * { <要素間制約> } *
    <論理的視覚要素> ::= <論理的視覚要素識別子> <要素名> *
    <表象> ::= <観点> <対象クラス> <メディア> <レベル> <変化>
    <要素間制約> ::= <論理的視覚要素識別子> <論理的配置関係> <論理的視覚要素識別子>
  <消滅形式> ::= <消滅タイプ> { <方向> <形態> }
<利用者動作> ::= <論理的視覚要素識別子> <知覚系動作> |
  { <論理的視覚要素識別子> } <物理的視覚要素> <運動系動作>
  { <論理的視覚要素識別子> } <知覚系動作>
<効果> ::= { <対モジュール部品動作> } { <対利用者部品動作> } <意味> |
  <対モジュール部品動作> ::= <動作タイプ1> <動作対象クラス1>
  <対利用者部品動作> ::= <動作タイプ2> <動作対象クラス2>
  <意味> ::= { <運動系意味> } <運動系対象> | { <知覚系意味> } <知覚系対象>
<用法> ::= <生成プロトコル> <結合プロトコル> <配置プロトコル>
  <生成プロトコル> ::= <プロトコル名> <パラメータ値> { <プロトコル名> <パラメータ値> } *
  <結合プロトコル> ::= <プロトコル名> <パラメータ値> { <プロトコル名> <パラメータ値> } *
  <配置プロトコル> ::= <プロトコル名> <パラメータ値> { <プロトコル名> <パラメータ値> } *
<コード・仕様> ::= <設計仕様ドキュメントへのポインタ> *
  <設計仕様ドキュメントへのポインタ> ::= <ソースコードへのポインタ> | <データ構造へのポインタ> |
  <モジュール構造へのポインタ>
```

図4 視覚的部品の部品知識表現モデルの記述形式

位置			方向	形態	論理的配置関係
TopLeft	BottomCenter	On	CenterToOutside	Mosaic	Forward
TopCenter	BottomRight	Left	LeftToRight	Spot	Backward
TopRight	TopSide	Down	OutsideToCenter	Light	Top
CenterLeft	BottomSide	Right	RightToLeft	Melt	Left
CenterCenter	LeftSide	Top	Random		Bottom
CenterRight	RightSide	MouseCursorPoint			Right
BottomLeft	FullSide	Fixed			Inside

要素						
Abbreviate	Board	Cycle	Handle	Multiple	Polygon	Simple
Accept	Border	Dark	Icon	Number	Pop	Slider
Alarm	Box	Dial	Image	Object	Pull	Slot
Alert	Button	Display	Indicator	Oval	Push	Space
Alphabet	Cable	Down	Item	Palette	Radio	Spot
Alphabet	Canvas	Exclusive	Knob	Pane	Rectangle	State
Anchor	Caret	Execute	Label	Panel	Region	Switch
Animate	Cell	Field	Left	Part	Result	Text
Apply	Character	Fix	Light	Pattern	Right	Title
Arc	Choose	Flash	Line	Pattern	Rim	Toggle
Area	Circle	Form	Lock	Pin	Ring	Underline
Arrange	Clip	Function	Margin	Plus	Scale	Up
Bar	Close	Glyph	Mark	Point	Scroller	Very
Black	Cross	Gray	Minus	Point	Select	White
Blank	Cursor	Gauge	Move	Pointer	Sheet	Window

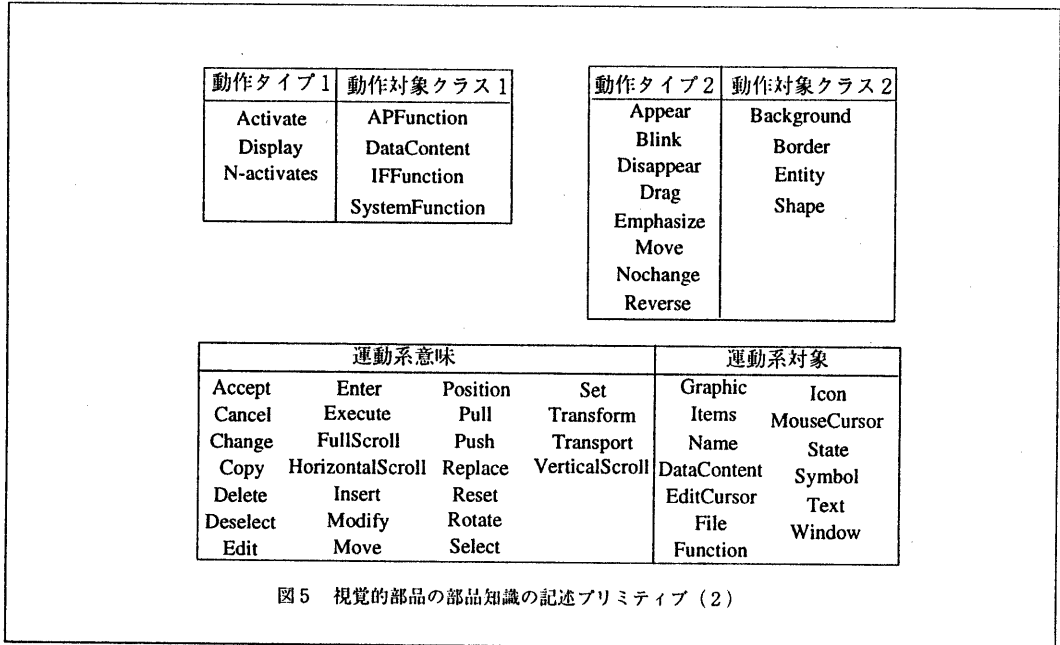
観点	対象クラス	メディア	レベル	変化
Association Key	Action	Character	Abstract	Reverse
	Data	Image	Concrete	Replace
	Event	Composite		Remove
	Function			Add
	Object			Move
	State			Modify
	Structure			Fix

出現タイプ	消滅タイプ
PopUp	PopOut
FadeIn	FadeOut
Resident	Resident

知覚系動作	物理的視覚要素	運動系動作
View	Mouse	Press
	MouseButton	Release
	Keyboard	Move
	CtlKey	Type
	...	Click
	"A"Key	Drag
	...	DoubleClick

知覚系意味	知覚系対象
Prompt	DataContent
Caution	OperableState
Confirm	ProcessState
Remind	UserOperation
Report	VisibleRate

図5 視覚的部品の部品知識の記述プリミティブ(1)



[2] 利用モデルに基づいた表現形式

視覚的部品の部品知識表現形式に基づいて表現された視覚的部品の部品知識の記述は、2.2で述べた利用モデルの要素である論理部品 (LP) と物理部品 (PP), 及びそれらの関係記述 (R) に変換される。このとき、視覚的部品のLPの導出で設定される共通用途として、利用者が期待する視覚的部品の表現情報、利用者の動作情報、視覚的部品に期待する効果情報が用いられる。

3.3 再利用と知識獲得のプロセス

種々のLP/PP/Rとソフトウェアモジュール群により構造化され管理される部品知識を利用して、利用者の目的を満足するソフトウェアモジュール (SM) がソフトウェアシステムの設計プロセスで利用される。これは視覚的部品に関する部品知識の表現/利用モデルに基づいた記述形式において通常利用、あるいは再利用の枠組みで行われる。部品の再利用は知識獲得による通常利用への縮退となる。いま、利用者の目的UDが部品知識の記述プリミティブを用いて与えられるとすると部品の利用手順は次のようになる。

- (1) LP候補の抽出：LP記述がUD記述に含まれるLPの集合を抽出し、これをLPの候補集合とする。
- (2) PP候補の抽出：LPの候補集合が空集合のときは停止する。LPの候補集合から要素を1つ抽出する。このとき、LPの候補集合から抽出した候補を除いたものを再び候補集合とする。抽出した候補をノードとし、そのノードに接続するRの記述がUD記述に含まれるRの指示する用法を持つPPからPPの候補集合を設定する。
- (3) SMの検索：PPの候補集合から要素を1つ抽出する (ただし、PPの候補集合が空集合のときは(2)に戻る)。このとき、抽出した候補を除いた集合をあらたにPPの候補集合とする。抽出したPPが指示するソフトウェアモジュールSMとRが指示する用法に対して(4)を試みる。
- (4) SMの組込：検索されたソフトウェアモジュールの仕様が設計目標のソフトウェアシステムの仕様

に適合すれば、それを目標システムに組み込み、利用者の目的UDに対する利用プロセスを終了する。適合しない場合は(3)に戻る。

再利用は、上記(2)でLPの候補集合が空集合のときに行われ、LP空間とPP空間に対する修正プロセスを経て通常利用に縮退する。前述した各再利用パターンにおけるソフトウェアモジュールの再利用と、通常利用への縮退は以下の操作により行われる(ただし、再利用が不可能な場合は、この利用者の目的UDを満足するソフトウェアモジュールを新規に設計することになる)。

<再利用1の場合>既存のソフトウェアモジュールに対する利用経験やその仕様から利用者の目的UDを満足する既存のソフトウェアモジュールの既知の用法を検索し、そのソフトウェアモジュールに対する新たな用途に対応した用法として、検索されたソフトウェアモジュールの既知の用法を利用する。通常利用の形式にするには、LPの候補集合に属する最適なLPを選択する、あるいは新たな用途に対応したLPを新規に定義することにより、LPを設定する。そして、このLPと検索された用法に対応するPPを関係Rで結合する。この操作によって、次回からは同じUDに対しては通常利用プロセスが適用できる。

<再利用2の場合>既存のソフトウェアモジュールに対し、利用者の目的UDを満足する新規の用法を作成し、新たな用途に対応した新規の用法として利用する。ここでは、再利用1の場合と同様にLPを設定し、更に新規の用法に対するPPを定義し、設定されたLPとPPを関係Rで結合することにより、通常利用の形式となる。

<再利用3の場合>利用者の目的UDを満足するよう既存のソフトウェアモジュールの実装知識を加工し、新規のソフトウェアモジュールを作成して利用する。新規のソフトウェアモジュールの登録は、再利用1や再利用2の場合と同様にLPが設定され、次に新規のソフトウェアモジュールの用法に対するPPが定義され、これらLPとPPを関係Rによって結合することにより行われる。

4. おわりに

ソフトウェア部品の知識の利用/再利用のための部品化モデルについてドメイン分析・モデリングの観点から述べた。また、本モデルを視覚的部品と呼ばれるインタフェースソフトウェア部品に適用し、その知識表現形式と再利用における知識獲得について述べた。一方、本モデルは、知識型設計方法論⁴⁾における一つの応用例とみることでもできよう。その場合、部品知識の表現モデルと利用モデルが知識モデルに対応する。そして、表現モデルから利用モデルへの部品知識の表現形式の変換が知識モデル間のマッピングに対応する。今後は視覚的部品の知識ベース化の実験研究に基づいてモデルをリファインメントしてゆく予定である。

【参考文献】

- 1) 田村, 伊藤, 杵嶋, "ドメイン分析・モデリング技術の現状と課題", 情処学会誌, Vol.35, No.10, pp.952-961, Oct. 1994.
- 2) 岩根, 城風, 木下, "インタフェースソフトウェアの部品化とその知識表現モデル", 信学技報(人工知能と知識処理) A193-26(1993-07)。
- 3) 岩根, 木下, "インタフェースソフトウェアの部品化と再利用の一手法", 信学技報(人工知能と知識処理) A194-21(1994-07)。
- 4) 木下, 岩根, 菅原, 白鳥, "知識型設計方法論に基づくインタフェース設計の形式化と設計支援システムの構成", 情処学会論文誌, Vol.31, No.6, pp.906-915, 1990.