

応用層プロトコルの知識型設計支援システムについて

菅沼拓夫[†] 渡辺博之[†] 菅原研次[†] 木下哲男^{††} 白鳥則郎^{†††}

[†]千葉工業大学情報工学科

〒275 習志野市津田沼 2-17-1

^{††}沖電気工業(株)総合システム研究所

〒108 東京都港区芝浦 4-11-22

^{†††}東北大学電気通信研究所

〒980 仙台市青葉区片平 2-1-1

あらまし

情報ネットワークに対する利用者要求の高度化、多様化により、分散応用プロセスが必要とする通信ネットワークの通信サービスに対する要求が多様化している。これまでのシステム指向の情報ネットワーク開発アプローチではこの要求に対応できず、応用指向の開発アプローチ技術が注目されている。このアプローチで分散応用システムを実現するためには、応用層プロトコルの拡張により応用プロセスの通信サービス要求に対応する方法が一般的である。本研究では多様化、複雑化する応用層プロトコルの設計を効率的に支援するための方法論として、知識型設計方法論を採用し、これに基づき支援システムの設計を行った。

和文キーワード プロトコル設計、知識型設計方法論、設計支援システム

Knowledge-based Design Support System for Application Layer Protocol

Takuo Suganuma[†] Hiroyuki Watanabe[†] Kenji Sugawara[†] Tetsuo Kinoshita^{††} Norio Shiratori^{†††}

[†] Dept. of Computer Science, Chiba Institute of Technology
2-17-1, Tsudanuma, Narashino 275

^{††} System Laboratories, OKI Electric Industry Co., Ltd.
4-11-22, Shibaura, Minato-ku, Tokyo 108

^{†††} Research Institute of Electrical Communication, Tohoku University
2-1-1, Katahira, Aoba-ku, Sendai 980

Abstract

Requirement for communication services of communication networks for distributed application processes becomes diverse as user's requirement for information network systems becomes advanced and diverse. Conventional system-oriented approach to developing information networks cannot satisfy the requirement, and application-oriented approach to developing them is getting attention. This approach to realizing distributed application systems makes application protocol larger and more complicated.

This paper discuss the knowledge-based methodology and design support system based on it in order to develop diverse kind of and complicated application layer protocol.

英文 key words Protocol Design, Knowledge-based Design Methodology, Design Support System

1. はじめに

情報技術及び通信技術の発展に伴い、コンピュータネットワークシステムと、それにより利用可能となる分散アプリケーションに対する要求の高度化と多様化が進んでいる。従来の分散アプリケーションシステムは、コンピュータネットワークシステムやデータベースなどの、既存のシステムにより提供される標準的サービスから、その制約の範囲で最もアプリケーションの要求を満たすことのできる通信サービスプリミティブを選択し利用するという、システム指向のアプローチで開発が行われてきた。このアプローチでは利用者指向のアプリケーションシステムに対する全ての要求を実現するためには不十分であり、利用者にとって使いやすいシステムを実現するという重要な設計要件が満たされてはこなかった。

しかしながら、近年の情報システムの社会への浸透に伴い、情報システムの能力の評価として「使い易さ」が重要視され、従って利用者指向応用システム

(UAS:Use-oriented Application System)を実現する技術の開発が重要な課題となっている。UASとは利用目的や利用スキルの異なる様々な利用者の要求に対処可能な応用システムであるが、この機能を大規模システムであるコンピュータネットワーク上の分散処理システムとして実現しようとすると、システム内に存在する様々な制約により困難な作業となる。

これは、大規模システムを構成する要素が持つ制約が複雑に絡み合うため、設計プロセスにおいてそれらを管理し、最適解を発見する作業が人間の設計者にとって困難だからである。

これを解決するためには分散応用システムを構成する要素の制約をなるべくゆるめることにより、全体の制約を緩和し、利用者指向の設計を実現することが望ましい。筆者らは、この方針に基づき、分散応用システムを構成する主要要素の一つである通信サブシステムの制約を緩和する技術として、応用指向プロトコル(AOP:Application Oriented Protocol)を提案した[菅沼93]。これは応用層プロトコルの部品化と再利用技術を確認することにより、多様な応用プロセスの通信サービス要求を充足する通信サービス機能を実現することを目的としたものである。

AOPを開発するためには、多様な要求の分析、部品管理と再利用など、複雑な設計プロセスが必要であり、このような複雑な作業を支援する有効な方法として知識型設計方法論(KDM)が提案されてきた[Kino88][Shir92]。本稿ではAOPの概念設計プロセスの知識型支援システムについて述べる。

2. 応用層プロトコルモデルとその設計プロセス

2.1 応用指向プロトコル

応用指向プロトコルの概念図を図1に示す。通信ネッ

トワークは通信サービスを行なうシステムの論理的記述であり、例えばOSI参照モデルでは第6層(プレゼンテーション層)以下が通信ネットワークに相当する。応用指向プロトコル層は通信ネットワークと応用プロセス群の中間に位置する層である。応用プロセスAPはその通信に関する活動が標準的であれば通信ネットワークが標準的に提供する機能のみを利用して分散処理システムを構成できるが、標準から外れた通信サービスを必要とするときは応用指向プロトコル層にそのサービス機能を実現しなければならない。

従来、その必要性に応じて様々な応用指向プロトコルが開発されてきている。例えばOSI参照モデルにおいてはMHS/MOTIS, RDA, FTAM, JTM[滝沢93]などがそれであり、特定業務向きアプリケーションサービス要素(ASE)と呼ばれている。これらのサービス要素のうち特に共通に利用できるものは共通サービス要素(CASE)と呼ばれている。ASEおよびCASEは再利用性が高いサービス要素であり、これらを有効に利用することにより、今後利用者要求に応じて開発しなければならない応用指向プロトコルの開発効率を向上させることができる。

すなわち応用指向プロトコルという概念を導入することにより、利用者指向設計という観点から、標準化体系に縛られることなく、しかも標準化体系を構成部品として利用できるという利点を積極的に生かしながら、利用者要求に即した応用システムを効率的に開発できる。

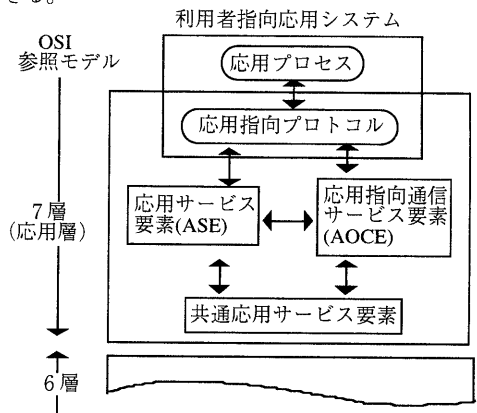


図1 応用層プロトコル

2.2 応用層プロトコルの設計プロセス

応用層プロトコルの設計プロセスを図2に示す。応用層プロトコルの設計プロセスは、筆者らが提案しているコンピュータコミュニケーションシステムの設計プロセスの一部を構成している[Kino88]。この設計方法論は、応用指向の設計アプローチを採用しており、利用者要求定義から始まるトップダウン設計によりサブシステムの詳細設計を進めて行き、利用者要求は分割

されたサブシステムの各設計プロセスで保存されている[Suga92]。

多様な利用者の要求は、図2の応用システムの要求仕様として定義され、これらの要求は、ネットワークの概念設計で統合され、次に情報資源サブシステム、ユーザーインタフェースサブシステムと共に通信サブシステムへ分散され、通信サブシステムでも、ハードウェア仕様、通信ネットワーク管理仕様などと併にプロトコルに対する要求仕様が生成される。

図2では、ネットワークアーキテクチャとしてOSIが選択されていることを仮定している。この条件より、ネットワークアーキテクチャの要求仕様から、応用層プロトコル要求定義プロセスを経て、応用層プロトコルの要求仕様書が生成される。この要求仕様書は、応用層プロトコル設計プロセスを経て、プロトコルの内部記述である応用層プロトコル設計仕様書に変換される。この設計仕様書を基に、プロトコルを実行するプログラムを作成するが、中間段階として、プログラムシステムを構成するプログラムモジュールの設計仕様書が作成される。このプログラム設計仕様書に基づいてプログラマはコーディングを行い各モジュールのソースコードが作成され、単体テスト、統合テストを経て運用システムが作成される。

実際には、各プロセスにはレビューやデバッグのプロセスが付随しているが、ここでは省略してある。

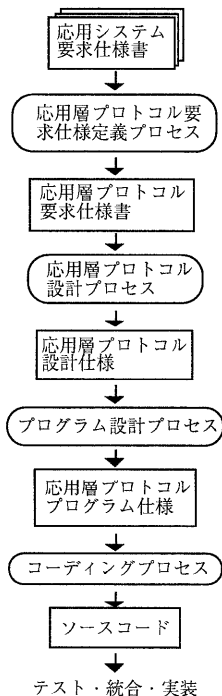


図2 応用層プロトコルの設計プロセス

3. 応用層プロトコルの知識型設計プロセス

3.1 知識型設計方法論の枠組み

一般にシステム設計プロセスは多くの仕様書で表現される設計ステージと、それらの仕様書の変換・詳細化のタスクとしての設計サブプロセスから構成される。知識工学の立場から設計タスクを問題解決プロセスとしてとらえ、設計のために必要な知識を定式化し、これを利用して設計を効率的に行うためのアプローチとして、知識型設計方法論(KDM)が提案されてきた[Kino88]。

KDMの基本的枠組みは、以下の通りである。

- (1) 要求仕様定義から運用・保守のライフサイクル全体に渡る大規模平行設計プロセスを系統的に支援する。
- (2) 設計プロセスを問題領域知識の間の知識に基づく変換プロセスの系列としてとらえる。
- (3) 仕様書は問題領域知識のインスタンスであり、問題領域知識は様々な背景知識によりそのセマンティクスが支持される。
- (4) 設計知識は要求仕様書と設計仕様書の間の知識変換事例として獲得される。
- (5) 設計プロセス間の協調処理はマルチエージェント間の協調アクティビティでモデル化され支援される。

3.2 知識型設計プロセスモデル

KDMの枠組みに基づく設計プロセスモデル(DPM)を図3に示す。設計プロセスモデルは、設計の要求仕様の知識表現である上流知識モデル、設計結果の仕様の知識表現である下流知識モデル、及び設計プロセスの知識表現である設計知識から構成される。設計プロセスは、設計知識の利用によって行われる上流知識モデルから下流知識モデルへのマッピングとしてとらえられる。設計プロセス全体はこの設計プロセスモデルの連続として定式化される。

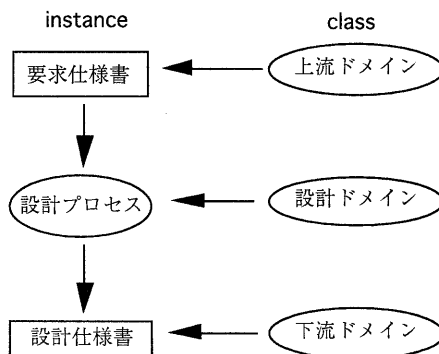


図3 設計プロセスモデルDPM

この設計プロセスモデルDPMは、上流ドメイン（問題領域知識）、下流ドメイン（解領域知識）、設計ドメインによって設計プロセスが構成されるという枠組みのみを提供する、最も抽象的な設計プロセスの表現であり、実際に実行される設計プロセスにおいてはより具体的な各ドメインの知識が利用される。具体化された設計プロセスをより一般化、抽象化し、その設計プロセスの抽象概念をモデル化して、分類、階層構造化することにより、類似した設計プロセスを定義する際に上位概念を継承するサブクラスとしてとらえることが可能となる。

3.3 ドメイン分析

ドメイン分析とは、ある対象ドメインの中に存在する多くの「対象」の中からその共通した性質を抜き出し、その共通の性質をドメイン特有の言語を用いて記述した、ドメインモデルを構成することである [Diaz87]。ドメインモデルはその対象ドメインの中の全ての有意な「活動」と「対象」を明示するべきである。ドメイン分析はソフトウェア工学における、部品再利用をおこなうための一手法として開発された [Neigh84]。すなわちあるアプリケーションの分野の中の様々な対象世界の性質を分析し、そのドメインで共通に使われている言葉や概念を抽出しモデル化することによって、部品再利用に有効な情報を抜き出すという試みであった。しかしその方法論は知識工学での知識獲得、モデリングにおいても有効であるとされている [Diaz91]。

ドメイン分析の方法としては、まず対象ドメインの中で共通に使われている「対象」と「操作」を抽出し、それら自身を分析、それらの関係を抽出した後、分類を行ない、最後に関連「対象」や機能のルールを記述したドメイン言語を定義するのが一般的に行われている手法である。

KDMにおいても、ある設計ステージの設計仕様を知識モデルとして表現するときに、その設計仕様の示す知識を余すことなく明確に表現する必要がある。知識モデルが十分に設計仕様の事象を表現することができるよう定義されていれば、マッピングは知識モデル間の単純な変換作業として表現する事が可能である。従って、知識モデルのモデリングを行う際にはドメイン分析を十分行う必要がある。ソフトウェア工学でのドメイン分析は、再利用情報の獲得が目的であるが、ここでは設計活動すなわち知識のマッピングに必要な設計知識の獲得が主な目的となる。

3.4 応用層プロトコル設計支援

図2における応用層プロトコル設計プロセスに、図3の知識型の設計プロセスモデルを適用すると、図4のプロセスモデルが得られる。要求仕様ドメインおよび設

計仕様ドメインは、各仕様を記述するための様々な抽象的オブジェクトが定義されている。オブジェクトはフレームモデルにより記述され、お互いに関連付けられ、構造化されたクラスフレームの集合から構成されている。クラスフレームの内容は、OSI応用層のプロトコルを特徴化する様々な概念を属性名として持っている。

設計プロセスに与えられる要求仕様書は、このクラスのインスタンスであり、対応するクラスフレームの各属性の属性値が具体的に定められている。設計プロセスは、要求仕様書のクラスと、各属性値より、その段階的詳細化である設計仕様のクラスフレームとその属性値を決定する。このクラスフレームの選択と属性値の決定アルゴリズムを設計知識と呼ぶ。

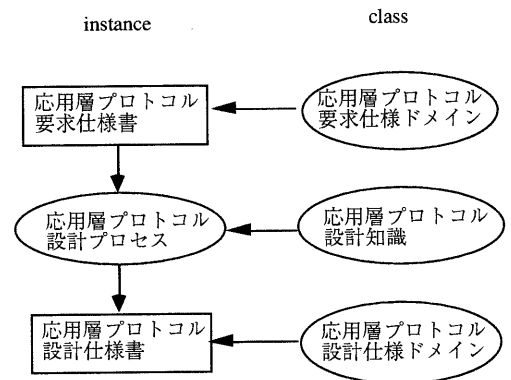


図4 応用層プロトコルの知識型設計プロセスモデル

4. 応用層プロトコル設計の要求仕様ドメインと設計仕様ドメイン

4.1 階層プロトコルのモデル

KDMに基づいた応用層プロトコル設計に関する知識モデルのモデリングを行うために、ドメイン分析を行うが、ドメイン分析を直接応用層プロトコル設計に適用するのではなく、通信アーキテクチャに依存しないドメインモデルを生成するために上位の概念である階層プロトコル設計に適用する。応用層プロトコルのドメイン分析は階層プロトコル設計に関するドメイン分析の特殊化によって行う。階層プロトコルの設計対象概念を図5に示す。プロトコルはいくつかの階層より構成され、その一つのプロトコル層が設計の対象である。これを対象層とよび、これに対して上位層、下位層とよぶことにする。対象層のプロトコルは互いに通信を行なう抽象機械であるプロトコルマシンPMのあい

だの相互作用として定義される[滝沢93]。上位層の抽象のプロセスであるプロトコルマシンユーザPMUの組がひき起こす相互作用を実現するための論理的通信手段として、PMUはPMが提供する通信サービスを利用する。PMはその通信サービスを実現するためにその下位層の論理的通信路LCC(Low)が提供する通信サービスを利用する。

この関係は任意の階層で成立する概念である。対象層のプロトコル設計とは上位層のPMUがPMに対して要求する通信サービスプリミティブを、下位層の提供する通信サービスプリミティブを利用して構成する問題である。

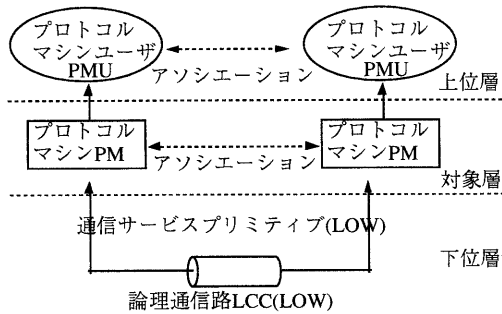


図5 階層プロトコル設計の解ドメインの概念モデル(S-DOMAIN/LP)

4.2 応用層要求仕様ドメインモデル

図4の設計プロセスの要求仕様の記述モデルを図6に示す。これは図5の階層プロトコルモデルの対象層である、応用層プロトコルに対する上位層であるアプリケーションプロセスの、応用層プロトコル階層に対する通信サービス要求を記述するためのモデルである。

アプリケーションプロセスAP同士が行なう論理的通信単位LCUは図7に示すように3つのタイプに分類される [西沢91]。ただし通信行為を要求するAPをアクティブAPとよび、その対象APをパッシブAPとよぶことにする。

(a) 確認型 LCU(C)

確認型の論理的通信単位LCU(C)はアクティブAP(A-AP)の要求(Request)がパッシブAP(P-AP)に対して指示(Indication)として受信され、その応答(Response)がLCC(AP)を通して返送され、A-APに確認(Confirm)として受信される型の通信行為である。

(b) 非確認型

非確認型の論理的通信単位LCU(U)はA-APの要求(Req)をLCC(AOP)を通してP-APに指示(Ind)として送信し、確認は必要としない通信行為である。

(c) 提供者起動型

通信システムエラーなどにより下位層の通信サービス提供者が通信サービス利用者に通知を行なう通信行

為である。

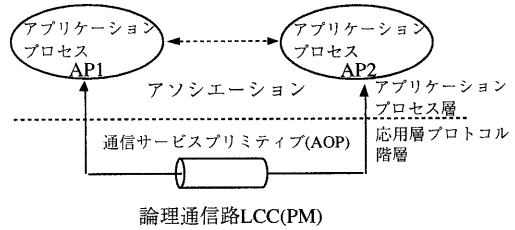


図6 応用層プロトコルの要求仕様モデル

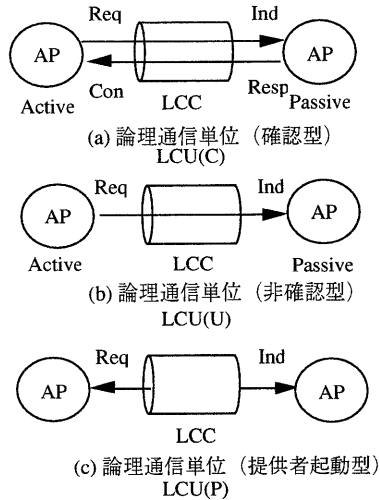


図7 アプリケーションプロセスの論理的通信単位

アプリケーションプロセスAPはある分散タスクを実現するために対象層の通信サービスLCC(PM)を利用する。分散タスクは図8に示す分散トランザクションプロセスグラフ(TPGグラフ) tpgにより記述される。

$tpg = \langle Node, Arc \rangle$

ただし

$Node = IP[A] \cup TP[A] \cup LCC \cup TP[P] \cup IP[P]$

であり

LCC: 論理通信単位lcuの集合

TP[A], TP[P]: アクティブ、パッシブの分散トランザクションプロセスの集合

IP[A], IP[P]: 外部プロセスとのインタラクション事象の集合

である。

ただし

$|IP[A]| + |IP[P]| \leq 2$

かつ

IP[A], IP[P]のノードの入次数も出次数も1以下

である。

一つのTPGグラフtpgはPMUが行なう分散トランザクションの論理的単位を表し、TPGグラフの集合とそれらの関係によりPMUの動作仕様が表される。

PMUの対象層の通信サービスに対する要求は図9に示されるように論理的通信単位lcu (トランザクションプロセスのペア) の集合で表される。

$$REQ = \{ lcu \mid lcu = \langle lcu_name, tp[A], tp[P] \rangle \}$$

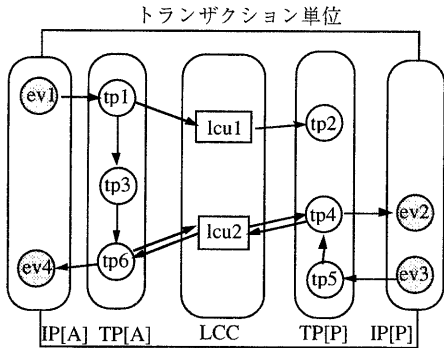


図8 分散トランザクションプロセスグラフTPG

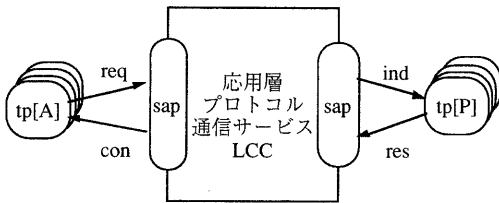


図9 プロトコルマシンユーザの論理通信単位 (通信サービス要求仕様)

4.3 設計仕様ドメイン

設計仕様は上位層のアプリケーションプロセスのすべての通信サービス要求に対して、それを実現する応用層プロトコルの内部構造を明示した仕様である。設計仕様のモデルは図8のTPGのモデルを継承している。対象層の設計仕様モデルの概念図を図10に示す。LCC [PM]のサービスアクセスポイントsapではイベントとして"require"、"indicate"、"response"、"confirm"が発生する。これによりプロトコルマシンのプロセス系列が活性化され、tp[A]とtp[P]のトランザクションプロセスの組により要求された通信サービスが実現される。この過程で、下位層により提供される通信サービスプリミティブLCC[LOW]と対象層のプロトコルマシンPMで利用可能なプロトコル部品Parts[PM]が呼び出される。

対象プロトコル層通信サービス設計仕様

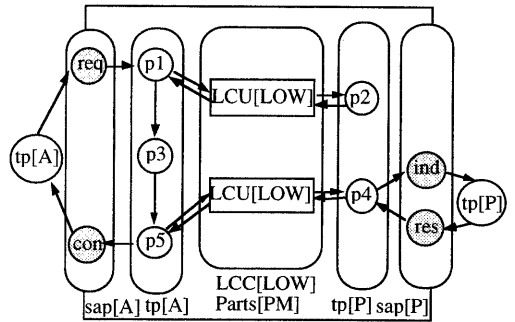


図10 対象層のトランザクションプロセスグラフTPG

4.4 設計知識

実際の要求仕様書は図8のようなTPGグラフと設計に関する制約をフレーム形式で記述したものであり、設計仕様書は要求仕様書で要求された分散トランザクションプロセスを設計に関する制約のもとで実現するために必要なサービス要素の時系列と各ノードの振舞いをフレーム形式で記述したものである。設計知識は要求仕様フレームから設計仕様フレームを生成するためのプロダクションルールの集合である。

5. 応用層プロトコル設計の流れ

図11に本システムにおける応用層プロトコル(ALP)設計の大まかな流れと、設計の際に使用される知識との関係を示す。

設計の際に用いられる知識ベースは大別してプロダクションルールの集合により構成されるもの (ALP要求仕様定義知識ベース、ALP設計仕様抽出知識ベース、ALP設計知識ベース) とクラスフレームにより構成されるもの (ALP要求仕様知識ベース、ALP設計仕様知識ベース) の2通りに分けられる。基本的に設計は、プロダクションルールの知識ベースから具象化された知識によって、クラスフレームの知識ベースからある特定のインスタンスフレームを導出し、更にその未確定の属性を決定することにより進められる。

まず応用システムの要求仕様からALP要求仕様の導出が行なわれる。ALP要求仕様定義知識ベースより具象化された要求仕様定義知識によって、ALP要求仕様知識ベースよりALP要求仕様となる適当なインスタンスフレームが導出される。フレーム選択に必要な情報が不足して要求仕様定義知識により属性の確定が行えない場合は、設計者に対して情報の補充を要求し、未確定の属性値を埋めながらフレームの絞り込みを行なう。

次に導出されたALP要求仕様からALP初期設計仕様が抽出される。このプロセスにはALP設計仕様抽出知

識ベースから具象化された設計仕様抽出知識が用いられ、ALP要求仕様に応じてALP設計仕様知識ベースより適当なインスタンスフレームが導出される。この設計のベースとなるインスタンスフレームがALP初期設計仕様であり、まだこの段階ではフレームを抽出するのみで、未確定の属性値はそのままである。また、主にいくつかの候補フレームを仮説として提示し最終決定を仰ぐことによって、先ほどと同様設計者とのインタラクションを行なう。

最後に前のプロセスで提示されたALP初期設計仕様とALP要求仕様から、ALP設計仕様が導出される。このプロセスにはALP設計知識ベースより具象化された設計知識が使用される。ALP要求仕様のクラスと各属性値に応じてALP初期設計仕様内の未確定スロットに順次値を割当てることにより設計が進められる。ここでも不足情報の補充や仮説に対する決定などの設計者とのインタラクションが行なわれるが、最も複雑な作業の部分であり、設計活動のトレースや使用された知識のreviewなども行なわれる。

なお、設計の要求仕様、設計結果、及び設計過程での中間的な各仕様は、設計者に対し、人間が理解しやすい表現形式で提示される。

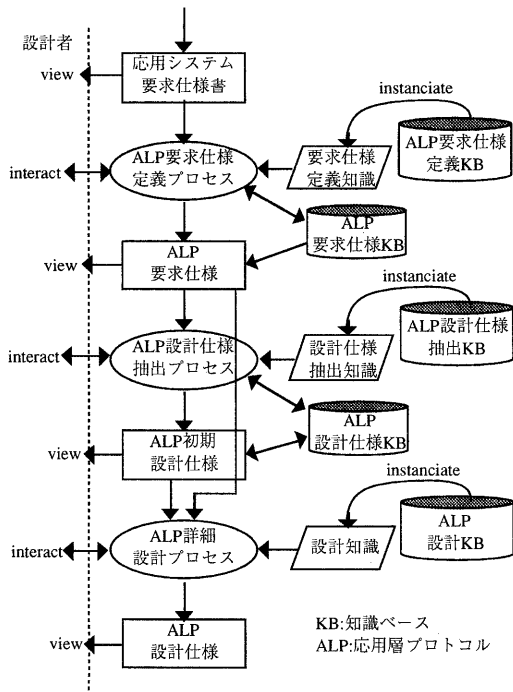


図11 応用層プロトコル設計の流れ

6. 応用層プロトコルの知識型設計支援システムの枠組み

図12に応用層プロトコルの知識型設計支援システムの構成図を示す。

応用層プロトコルの設計は、主に設計システムの構成要素である3種類の知識ベース (ALP要求仕様知識ベース、ALP設計仕様知識ベース、ALP設計知識ベース)、設計機構、及び設計者インタフェースの相互作用によって行なわれる。

ALP要求仕様知識ベース及びALP設計仕様知識ベースには、設計の際に用いられる要求仕様と設計仕様を抽象化、階層化したクラスが、フレーム形式で保存されている。設計時にはこのクラスフレームより、その未確定スロットを埋める形でインスタンスフレームが生成され、それが各々の内部形式仕様として設計に用いられる。

設計機構はこの支援システムの中心となる構成要素であり、各知識ベース及び設計者と相互作用を起こしながら、要求仕様から設計仕様への変換を行なう部分である。主に設計知識ベースより必要な知識群を選択し、その知識を用いて推論を行なう推論機構と、知識や情報の補充や意志決定の要求を人間設計者に対して行なう機能を有する。

人間設計者は設計者インタフェースを通じて設計活動の監視、知識や情報の補充、設計機構によって提示された仮説に対する最終決定などを行なう。知識型の設計においては計算機内部での処理がどのような形で進められているかを設計者が把握することは、設計が失敗した場合の後処理や知識の修正、拡張の面で重要である。本システムは設計の際に使用された知識や推論結果を人間設計者がトレースできる機能を持つ。また設計活動の背景となっている各知識がどのように構成されているかを把握するための機能を持つ。これは人間設計者と計算機内部の知識との一貫性を取るために重要である。更にトランスレータを介して計算機内部の各仕様の表現形式から変換された、人間設計者が理解できる形での表現形式での表示を行なう機能を有する。

図13に本支援システムのソフトウェア構成図を示す。本システムはUNIXワークステーション上に構成される。ユーザーインタフェースと設計機構の一部はObjectworks¥Smalltalk Release4.1を用いて実現し、知識ベース及び設計機構内の推論機構としてはKBMSを用いる。

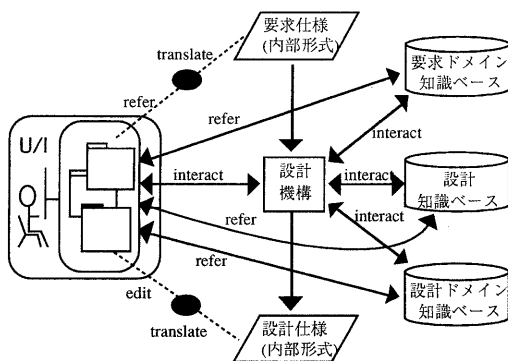


図12 設計支援システムの構成図

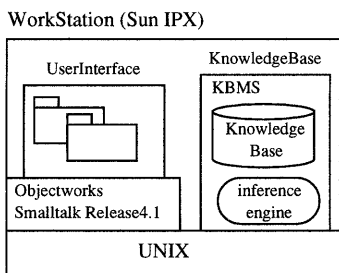


図13 支援システムのソフトウェア構成

7. おわりに

情報ネットワークに対する利用者要求の高度化、多様化に伴い、分散応用プロセスが必要とする通信サービスに対する要求が多様化し、これに対処するための応用層プロトコルが複雑化、大規模化の傾向を強めている。従って応用層プロトコルの効率的な開発支援システムを提供することは、利用者指向の情報ネットワークを構築するために重要である。

本研究では、知識型設計方法論に基づいて応用層プロトコルの設計プロセスの定式化を行った。このモデルに基づいて応用層プロトコルの要求仕様ドメイン、設計仕様のドメインの定義を行った。今後知識の実装を行い、知識型支援システムの実現と評価を進めて行く。

参考文献

- [菅沼93]菅沼、木下、菅原、白鳥: "応用指向プロトコルの知識型設計方法について", 情処学会SIGDPS研究会資料、1993年7月
- [Kino88]Kinoshita T., Sugawara K., Shiratori N.:

"Knowledge-Based Design Support System for Computer Communication System", IEEE Journal SAC, Vol.6, No.5, pp.850-861, 1988

[Shir92]Shiratori N., Takahashi K., Sugawara K., Kinoshita T.: "Using Artificial Intelligence in Communication System Design", IEEE Software Mag., Vol.9, No.1, pp.38-46, 1992

[滝沢93] 滝沢,中村: "OSIプロトコル技術解説", ソフトリサーチ・センター, 1993

[Suga92]Sugawara K., Kinoshita T., Shiratori N.: "Knowledge-based Design Methodology with Application for Requirements Specification and Definition of Communication Protocols", Proc. 5th JC-CNSS & KITE, pp.303-306, 1992

[Diaz87] R. Prieto-Diaz: "Domain Analysis for Reusability", Proc. of COMPSAC '87, pp.23-29, 1987

[Diaz91] R. Prieto-Diaz, G. Arango: "Domain Analysis and Software Systems Modeling", IEEE, 1991

[Neigh84] J. M. Neighbors: "The Draco Approach to Constructing Software from Reusable Components", IEEE Trans. on Software Eng., Vol.10, No.5, pp.564-574, 1984

[西沢91] 西沢,長谷川,中井訳: "実践的OSI論", トッパン, 1991