

コンピュータコミュニケーションシステムの知識型設計支援について

木下哲男 沖電気工業(株)総合システム研究所
菅原研次 千葉工業大学 電子工学科
白鳥則郎 東北大学 電気通信研究所

コンピュータコミュニケーションシステムのトータルな設計を支援するアプローチの一つとして、本稿では、知識型の設計方法論を提案すると共に、それに基づいた知識型の設計支援システムの枠組みについて述べる。新たに提案される知識型設計方法論は、利用者の要求に従って設計が進行するアプローチを採用しており、利用者からの種々の要求に基づき、利用者視点仮想マシン(UVM)及び設計者視点仮想マシン(DVM)によって段階的に形式的な仕様が定義される。これらの仮想マシンは、設計仕様を表現するための知識モデルとなっている。また、この設計方法論に基づいて、コンピュータコミュニケーションシステムのための知識型設計支援システム(KDSS)の枠組みを提案する。KDSSは、知識型設計方法論で定義される設計フェーズに対応した多数の設計エキスパートシステムから構成されている。

ON THE KNOWLEDGE-BASED DESIGN SUPPORT FOR COMPUTER COMMUNICATION SYSTEMS

Tetsuo Kinoshita OKI Electric Industry Co., Ltd.
Kenji Sugaware Chiba Institute of Technology
Norio Shiratori Tohoku University

11-22, Shibaura 4-Chome, Minato-ku, Tokyo 108, Japan

This paper proposes a knowledge-based design methodology for computer communication system and a framework of a knowledge-based design support system. In this new design methodology, according to the end user's requirements, formal specifications are defined as User's view Virtual Machine (UVM) and Designer's view Virtual Machine (DVM). These virtual machines are constructed as knowledge models for the specifications. Due to the design methodology, a knowledge-based design support system for computer communication systems, called KDSS, is constructed. This design support system consists of many design expert systems corresponding to the design phases given by the methodology.

からより細かな分類が必要となるが、現段階ではこの二種類に大別するに止めておく。

一方、これらの知識がCCS設計といふと、この二つの設計の以下を捉えるか否か、あるいは設計の過程でどのように使われるかなど、その特徴として以下の二つが次の一級とされる。本設計方法論では、 CCS設計サブプロセスの集合によって、 CCS設計の全構成部分は、 いわば式化する。即ち、 CCS設計サブプロセスの集合によって、 CCS設計を表す二つの知識モデル（マッピング）が、 像定義（マッピング）される。図1は前記1)の問題依存型知識モデルは、 一般的な形態設計論では、 各段階で扱う設計対象に応じて、

(A) 各設計ステージを知識モデルにマッピングする。
(B) 各設計サブプロセスを、 像定義（マッピング）する。
とし、 提えられる。この二つは、 各設計プロセスに適用すると図2が得られる。

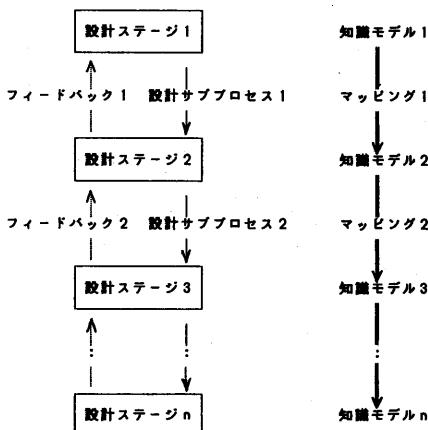


図1 一般的な知識型設計プロセス

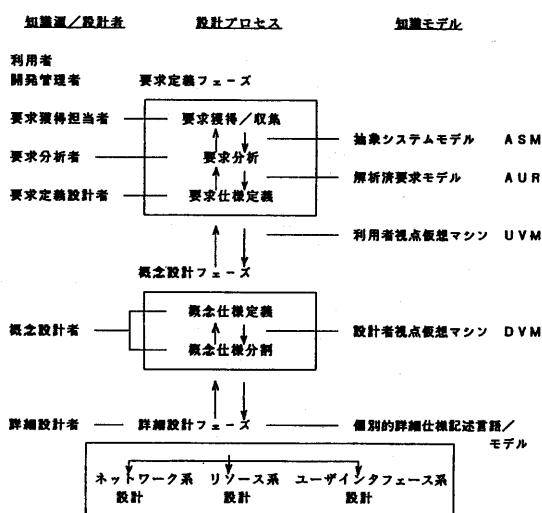


図2 CCSの知識型設計プロセス

2.2 要求定義フェーズ

要求定義フェーズでは、 命題的設計は、 それを実現するための、 理想化された構造を規定する。これは、 各種の構造要素の組合せや、 具体的な仕様を定めることである。また、 これらは、 通常、 その命題的設計の範囲内に含まれる。そこで、 その命題的設計が、 その実現方法によっては、 具体的仕様を規定する。

(1) 要求抽出： 設計者によって、 利用者からの要求事項を抽出する。これは、 命題的設計の範囲外に含まれる。また、 その命題的設計が、 その実現方法によっては、 具体的仕様を規定する。

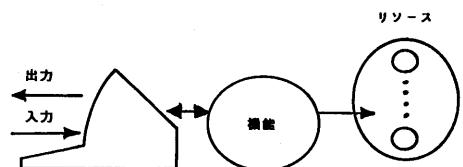


図3 抽象システムモデル(ASM)

1) 直接入力モード：エンドユーザ向き
 2) システム設計者モード：設計者向き

の二つのモードが設定されている。1)のモードはエンドユーザが直接自己の要求は直接表現するためには、要求指向型の設計法を実現するためには、本方論では、抽象システムモデル (Abstract System Model; ASM) と呼ばれる知識モデル (図3) を定義し、それに従って要求を投入して行く。ASMの記述形式であるASMDは、

$ASMD = \langle INPUT, OUTPUT, FUNCTION, RESOURCE, USER, RELATION \rangle$

という6項目組で定義され、エンドユーザの視点から見たシステムの抽象的な機能を与える基準となっている。エンドユーザは、ASMの構造に従って、何をえども（INPUT）、どのような機能（FUNCTION）とどのような資源（RESOURCE）を用いて、どんな結果を得る（OUTPUT）、という風に自己の要求を記述してゆく。詳細な定義は省略するが、一つの要求に混入する曖昧性を極力省くため、こうした比較的単純なモデルが用いられる。要求の一部として与えられる設計制約条件等も同様のモデルで表現されることになるが、本稿では割愛する。

(2) 要求分析ステージ

初期の要求 (ASMDの集合) は種々雑多な要求の集合体であり、システムの要求仕様を得るためにには、これらはより整理され形式化された要求として再定義される必要がある。ASMD集合上では幾つかの記述変換操作が定義され、これで整理された要求の結果が解析済要求モデル (Analyzed User's Requirement; AUR) という知識モデルにより表現される。その記述形式AURDは、

$AURD = \langle S_INPUT, S_OUTPUT, FUNCTION, REQ \rangle$

という4項目組で与えられる。AURDは、機能的に共通したASMDを統合した形式を持ち、当該機能(FUNCTION)のもとで入出力が統合(S_INPUT, S_OUTPUT)され、整理・統合された要求の構造は、要求グラフ (REQ) として表現される。この段階で、初期の種々の要求群が抽象・統合化されることになる。

(3) 要求仕様定義ステージ

AURの集合に基づいて、最終的にユーザから見た要求仕様を定義するのが本ステージである。要求仕様定義のための知識モデルとして利用者視点仮想マシン (User's view Virtual Machine; UVM) が利用される。即ち、UVMは、

$UVM = \langle \{PF\}, \{MF\} \rangle$

として、アリミティブファンクション PF とマクロファンクション MF の集合の組によって定義される。PF及びMFは、

$PF = \langle Command, Response, SideEffect \rangle$

$MF = \langle MacroFunctionNode, MacroFunctionArc \rangle$

で与えられる。PFは、AURとして形式化された個々の利用者要求を実現する

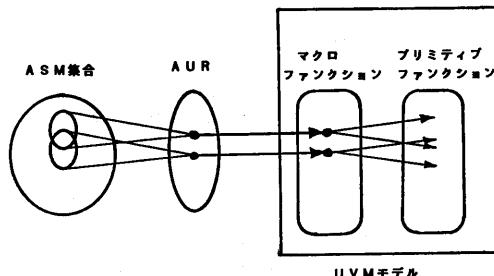


図4 利用者視点仮想マシン (UVM)

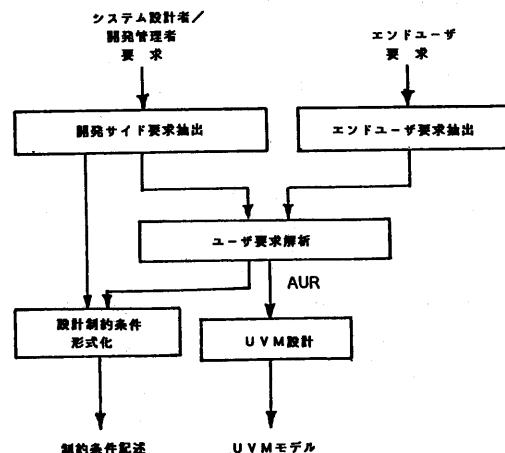


図5 要求定義フェーズの設計フロー

（領域依存型知識）を表すノードが、このための抽象化された機能モデルである。UVMは、この機能モデルを用いて、各機能の構成要素を定義する。これらの構成要素は、UVMによって構造化され、機能の実現を可能にする。

このUVMによる機能モデルは、各機能の構成要素を明確に定義する。UVMは、各機能の構成要素を明確に定義する。UVMは、各機能の構成要素を明確に定義する。UVMは、各機能の構成要素を明確に定義する。

2.3 概念設計フェーズ

利用者の種々の要求とその機能がモデル化される。そこでは、以下で定義する知識モデルUVFに基づいて、目標のCCCSの概念設計から成る。そのうえで、以下のフェーズは、二つに分類される。

(1) UVMを用いて、機能を構成する構成要素を明確に定義する。UVMは、各機能の構成要素を明確に定義する。UVMは、各機能の構成要素を明確に定義する。UVMは、各機能の構成要素を明確に定義する。

DVM = < {NODE}, {RELS} >

ノード記述は、定義する機能を対象とする。ノード記述は、定義する機能を対象とする。ノード記述は、定義する機能を対象とする。ノード記述は、定義する機能を対象とする。

RELS記述は、機能の構成要素を明確に定義する。RELS記述は、機能の構成要素を明確に定義する。RELS記述は、機能の構成要素を明確に定義する。RELS記述は、機能の構成要素を明確に定義する。

UVF記述は、機能の構成要素を明確に定義する。UVF記述は、機能の構成要素を明確に定義する。UVF記述は、機能の構成要素を明確に定義する。UVF記述は、機能の構成要素を明確に定義する。

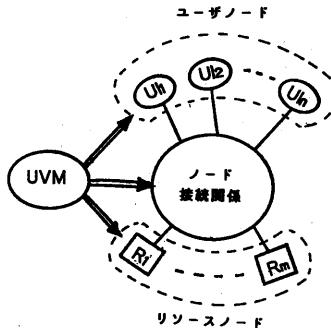


図6 設計者視点仮想マシン (DVM)

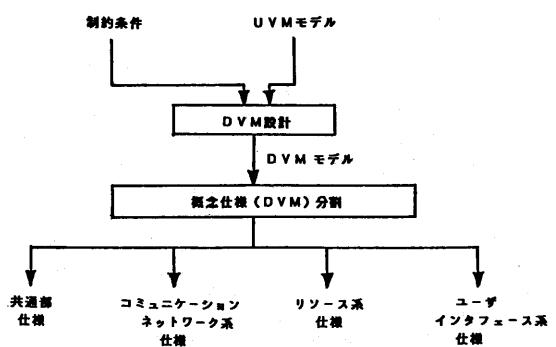


図7 概念設計フェーズの設計フロー

表1 設計用KSの例

コミュニケーションネットワーク系設計
トポジット設計
プロトコル仕様設計
プロトコル検証
通信ソフトウエア設計／生成
プロトコル変換
コンフォーマンス試験設計
性能評価
etc.

リソース系設計
新規リソース設計
リソース割り当て／配置設計
コマンドインタフェース設計
通信系インタフェース設計
etc.

ユーザインタフェース系設計
ユーザコマンドシステム設計
ユーザ要求解析システム設計
オンラインヘルプ機能設計
ユーザモデル設計
etc.

ムは、各知識設計者等と各工種の設計者等との間で、DDESは、CoDESが管理する協調型設計ESに接続される。B/Bは、DDESの設計ESとCoDESの設計ESとの間で、設計取りに利用される。B/Bは、DDESの設計ESとCoDESの設計ESとの間で、設計取りに利用される。

一方、詳細設計系、リソース系、データベース系、バーチャルペーパークス、バーチャルペーパークスの各知識設計者等と各工種の設計者等との間で、DDESは、CoDESが管理する協調型設計ESに接続される。B/Bは、DDESの設計ESとCoDESの設計ESとの間で、設計取りに利用される。

詳細設計系、リソース系、データベース系、バーチャルペーパークスの各知識設計者等と各工種の設計者等との間で、DDESは、CoDESが管理する協調型設計ESに接続される。B/Bは、DDESの設計ESとCoDESの設計ESとの間で、設計取りに利用される。

- (1) KDSSの構造は、CCSのためのパラメータ化された知識モデルである。
- (2) KDSSは、多數の設計原則知識とベーシックな知識を有する。
- (3) KDSSでは、詳細設計レベルでの柔軟性の高い型ラクティアを提供し、設計の一部は（半）自動的に行わせる。
- (4) KDSSでは、インターフェースシステムを提供し、設計の一部は（半）自動的に行わせる。

3.2 KDSSにおける知識の表現形式について

本稿で定義したCCS設計のための各種の知識モデルは、利用者などから与えられる要求、或いは制約条件などの問題依存型知識を表現・記述する枠組みであ

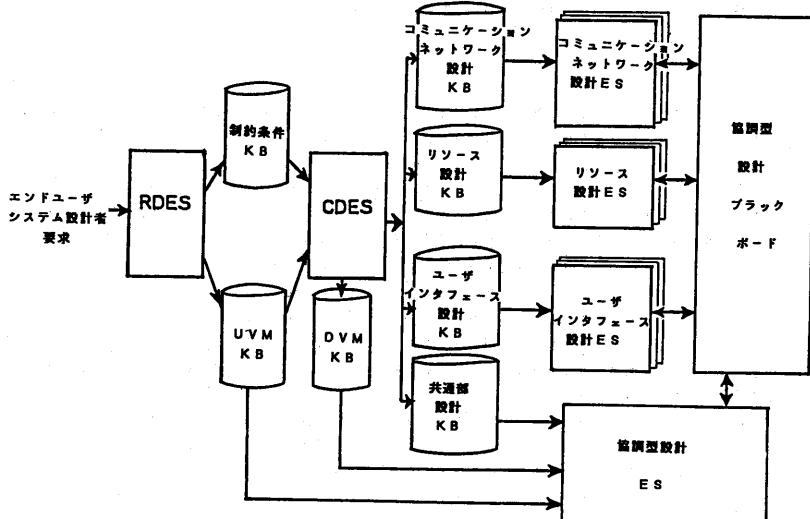


図8 KDSSの全体構成

