マルチメディア通信と分散処理 41 － 5
（１９８９. ３. １３）

分散処理システムの機能要求定義

菅原 研次[+] 福島 学[+] 木下 哲男[++] 白鳥 則郎[+++]
[+]千葉工業大学電子工学科 [++]沖電気工業㈱ [+++]東北大学電気通信研究所

分散処理システムの設計を支援するための知識型設計支援システムを構築中である。 本システムの目的は設計過程に使われる様々な経験的知識を計算機処理可能な形式で定義し、それを利用することにより分散処理システム設計の効果的な支援を行うことである。 本稿では利用者要求を定義するために必要な機能要求を記述するための知識モデルである論理機能モデルを提案する。 論理機能モデルは機能の階層的記述が可能な再帰的グラフ形式であるユニバーサルグラフ形式で記述されている。 論理機能モデルは、要求定義支援システムにおける機能要求を記述する言語に使われる。

Requirements Definition of Logical Functions of Distributed Processing Systems

Kenji Sugawara[+] Manabu Fukushima[+] Tetsuo Kinoshita[++] Norio Shiratori[+++]
[+]Chiba Institute of Technology [++]OKI Electric Industry Co., Ltd. [+++]Tohoku University
2-17-1, Tudanuma Narashino-shi, Chiba 275 JAPAN

We propose a knowledge model to represent logical functions of information processing systems, which is a problem oriented knowledge of support system for defining user's requirements for computer communication systems. The knowledge model, which is called logical function model, is modeled as a set of recursive graph forms which consist of information nodes and functional nodes. So structured decomposition of the logical function is described in a well defined form.

1．はじめに

多数のコンピュータシステムを通信回線で結合して、エンドユーザの多種多様な利用者要求を分散処理システムとして実現するコンピュータコミュニケーションシステムは高度情報処理システムを構築するための基盤技術として、その重要性を増している。　分散処理システムは多数のソフトウエアシステム、ハードウエアシステムから構成され、これらの複雑な組合せにより、利用者要求に対応した情報処理システムの複合体を実現しなければならない。　また、分散処理システムのような大規模システムは多数の設計エンジニアによりいくつかの設計段階を経て開発が行われる。従って、分散処理システムの設計開発過程で個々の利用者の要求の一部が失われ、開発されたシステムが利用者にとって使いづらいものになる可能性がある。　もう一つの問題は、システムが大規模化するに従い、その開発に関わる設計者の負担が大きくなることである。　一般に分散処理システムの設計にはフィールドエンジニアや通信システムエンジニアなどのような様々な技術分野における豊富な経験に基づいた設計知識が必要であるが、熟練した設計者の不足により効果的なシステム開発が行われなくなる可能性がある。

分散処理システムの開発におけるこれらの問題を解決するために我々は、知識型設計方法論を提唱した。[1][2]　この方法論は以下の２点を主な目的としている。

(1)熟練した設計者の設計知識を利用した、効果的な設計支援システムを実現する。

(2)利用者要求を分散処理システム上で忠実に実現する。

(1)の知識型設計支援システムを実現するためには設計対象の分散処理システムにおける知識を計算機処理可能な形式で定義されなければならない。

定義すべき知識には、問題依存型知識と領域依存型知識の２種類に分けられる。　問題依存型知識とは要求仕様や設計仕様に対応する知識であり、領域依存型知識とは前記の仕様を作成するための熟練した技術者の知識に相当する。　我々の知識型設計方法論では、設計過程は問題依存型知識を領域依存型知識により変換していく過程としてとらえている。　このような枠組みを用いるより、設計に使われる知識を計算機処理可能な形式で取り込むことができ効果的な設計支援システムが実現可能になる。

次に(2)の問題である利用者要求を忠実に実現するためには、設計の最初の段階で利用者の要求を正確に定義しなければならない。　この要求仕様定義技術として様々な手法が提案されている。しかしながら、そのような要求定義のプロセスは、人間の設計者の作業により実現される方法がほとんどであり、計算機処理を前提にした定義技法は少ない。　特に利用者要求の獲得技術については機械化されてはいなく、全く設計者の人的作業に負っている。　利用者の人数が少ないシステムでは、これでも十分正確な利用者要求定義が可能であるが、分散処理システムのような大規模で多種多数の利用者が含まれるシステムは要求獲得、分析、定義の過程を計算機処理することが必要である。　現在我々は分散処理システムの知識型設計支援システムの開発を目的としているが、本稿では要求定義段階の要求分析の支援システムについて論じている。　特にシステムで使われている問題依存型知識である機能要求を記述するためのモデルを中心に論じている。

2．分散処理システムの設計要求定義

2．1．分散処理システムの知識型設計方法論

分散処理システムを開発するための効果的設計方法論として、我々は以下の特徴を持つ知識型設計方法論を提案した。[1][2]

(1)知識指向

利用者要求、設計仕様などの問題依存型知識と、設計者の設計知識等の領域依存型知識の知識モデルを定義し、設計の過程を問題依存型知識と領域依存型知識を用いて変換していく過程ととらえる。　この考えに基づいて分散処理システムの設計者の設計作業を支援する知識型設計支援環境を開発する。

(2)利用者要求指向

利用者要求を明確に定義し、以降の設計過程に反映させることにより、利用者要求の実現を計る。

以上の考え方に基づく、分散処理システムの知識型設計過程を図１に示す。　設計過程は、要求定義、概念設計、詳細設計の３つの段階に分けられる。　図１には各設計段階における問題依存型知識と設計担当者及びその支援知識（領域依存型知識）が記述されている。　要求定義フェーズでは、利用者及び開発管理者からの要求を獲得し、それを分析することにより利用者要求仕様とシステム制約を生成し、次の概念設計フェーズに渡す。利用者要求仕様は、利用者視点仮想マシン（ＵＶＭ）としてモデル化される。　概念設計フェーズでは、利用者要求とシステム制約より、利用者要求を実現する分散処理システムの論理設計を行う。

概念設計仕様の記述のための知識モデルは設計者視点仮想マシン（ＤＶＭ）としてモデル化されている。　ＤＶＭはリソースノード、ユーザノー

ドとそれを結合する論理通信機能から構成され、ＤＶＭ上でＵＶＭに記述された利用者要求が分散的に実現される。　概念設計フェーズでは、概念設計仕様を次の段階の詳細設計フェーズ（リソース系、ユーザインタフェース系、コミュニケーションネットワーク系のサブシステムから構成されている）に渡すため、仕様の分割を行う。

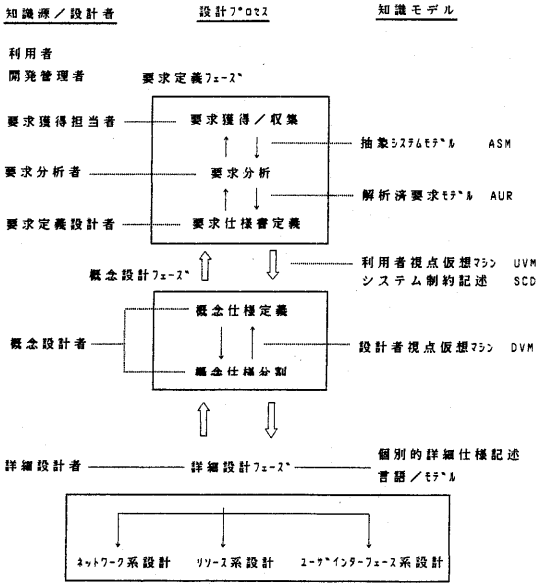

図１　分散処理システムの知識型設計プロセス

２．２．要求定義フェーズ

本節では図１に示した分散処理システムの設計過程の要求定義フェーズについて述べる。

要求定義フェーズの設計過程を図２に示す。要求定義フェーズでは２種類の要求が定義される。

一つは利用者（エンドユーザ）の分散処理システムに対する機能要求である。　もう一つは各機能に対する性能要求や開発管理者により要求される開発条件などのシステム制約である。　概念設計フェーズでは利用者の機能要求はトップダウンデザインの情報として、システム制約はボトムアップデザインの情報として使われる。

要求獲得プロセスでは利用者や開発管理者の要求をインタビューにより獲得する。　要求獲得の方式は知識工学技術における知識獲得の方式を適用している。　獲得された利用者の要求は、抽象システムモデル（ＡＳＭ）という論理機能記述のための初期記述形式で知識ベースに蓄えられる。

ＡＳＭ記述（ＡＳＭＤ）は多数の利用者の種々雑多な要求をそのまま記述したものであるから、ＡＳＭＤの間では重複や矛盾などが生じる。

従って、ＡＳＭＤの集合を分析し、整理・統合などを行い、利用者集団の要求を満足する最適な機能要求集合を構成することが必要である。　要求分析プロセスでは、要求分析技術者が要求分析支援システムと協調して、膨大な初期要求記述集合ＡＳＭＫＢを調査分析する。　その結果の最適機能要求集合は分析済要求知識ベース（ＡＵＲＫＢ）に蓄えられる。　ＡＵＲ記述（ＡＵＲＤ）はＡＳＭＤと同じ、機能要求モデルにより記述されている。

要求定義フェーズにおける利用者要求の最終仕様は利用者視点仮想マシンモデル（ＵＶＭ）により記述される。　ＵＶＭは図３に示すようにプ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２層構造を持つ仮想マシンである。　ＡＵＲＫＢに含まれる全ての利用者要求（ＡＵＲＤ）に対応して一つのＭＦが定義されている。　ＭＦはいくつかのＰＦの組合せにより構成されている。　ＰＦは概念設計におけるアプリケーションリソースに対応する機能要求仕様であり、ＭＦはユーザインタフェースとアプリケーションリソースを組み合わせて構成される分散アプリケーションへの仕様になっている。

以上が要求定義フェーズの概要であるが、本稿では初期要求ＡＳＭＫＢを分析済要求ＡＵＲＫＢに変換する要求分析プロセスについて次の２、３節で述べる。　要求定義プロセス及びＡＵＲＫＢからＵＶＭを定義する要求定義プロセスについては別稿で述べる予定である。

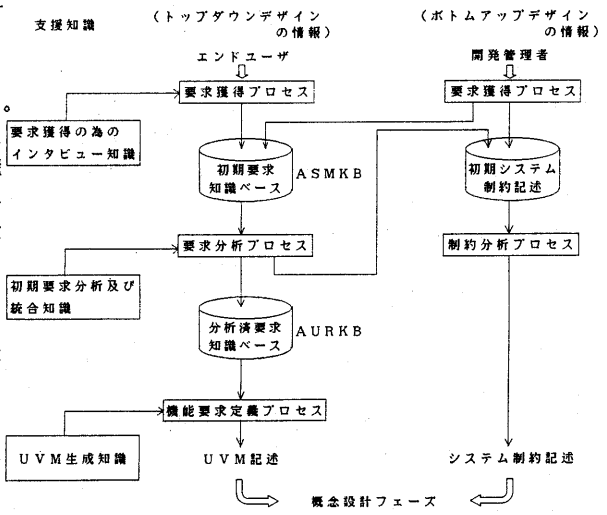

図２　要求定義フェーズ

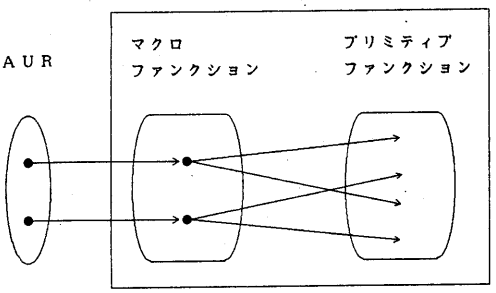

図３　利用者視点仮想マシン（ＵＶＭ）

２．３．要求分析プロセス

初期要求知識ベース（ＡＳＭＫＢ）に蓄えられている初期要求記述（ＡＳＭＤ）は、要求を記述した利用者／開発管理者が通常使用する業務用語か日常用語で表現されている。　すなわち、分散処理システムの一般利用者は、計算機システムの知識あるいは用語に精通していないため、厳密な情報処理技術用語で自らの要求を記述することは困難で、通常使用している業務用語で要求を記述したがるであろう。　インタビュー方式による要求獲得システムでは、このような利用者にとって表現し易い記述形式を許している。

更に分散処理システムの利用者は様々な業務にわたっていることが考えられ、情報処理システムとして実現するときは、同じタスクであっても、その機能要求表現が異なっている可能性がある。

従って、要求分析プロセスでは、図４に示すように利用者の要求表現を記述した初期要求記述を、情報処理技術用語で記述された正規化要求記述に変換することが必要になる。　すなわち、正規化操作を経て初期要求知識ベースは正規化要求知識ベースに変換される。　正規化要求記述にすることにより、要求間の関係、例えば同値性、包含性などが明確になる。

次に正規化された機能要求記述（ＲＲＤ）の間の関係に基づいてＲＲＤの結合、包含などの統合操作を行う。　この操作の過程で要求間の一貫性、正当性、完全性のチェックが行われる。　統合操作を経て、正規化要求知識ベースは分析済要求知識ベースに変換される。　一般に要求記述の集合は膨大であり、要求分析者が単独でその処理を行うことは負担が大きく、また効率も悪い。　本システムでは、要求分析プロセスにおける分析操作は要求分析技術者と分析支援エキスパートシステムとの協調作業により行われる。

正規化操作を支援する知識は、基本要求記述（Primitive Requirements）と正規化変換ルールからなる。　基本要求記述（ＰＲ）は初期要求を正規化要求に変換するための辞書に相当する知識があり、正規化変換ルールはこの辞書を参照しながら初期要求の変換を行う。

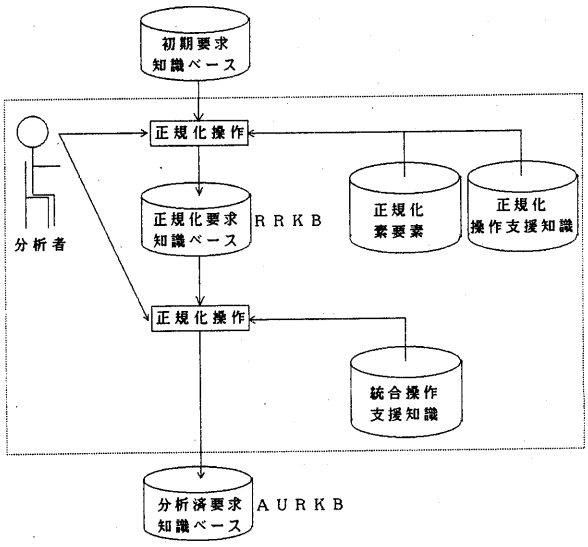

図４　要求分析プロセス

３．機能要求の記述

３．１．論理機能モデル

論理機能モデルとは、情報処理システムの機能要求を抽象的に記述するために定義されたモデルである。　論理機能モデルには次の条件が必要となる。

(1)機能が処理する対象の情報の構造を記述できる
(2)機能が処理する対象の情報の意味を記述できる
(3)情報の流れや変換過程の記述ができる
(4)高次の複雑な機能の表現までが一貫した形式で記述できる
(5)機能の階層的表現が記述できる
(6)計算機処理可能な形式で記述できる

要求仕様を定義するための方法論としてはSREM[3]、PSL/PSA[4]、SA[5]、UGF[6]等がある。　SREMにおけるR-net[7]はリアルタイムシステムの要求記述方法論で、一貫性、正当性等のチェックが計算機処理できる点で優れている。　ところが、情報の意味記述や階層的表現に難点がある。　またPSL/PSAは情報の構造や意味記述に優れているが、情報の変換過程の記述に問題がある。　SAは情報の構造や意味、変換過程、階層構造の記述が可能

であるが、計算機処理に難点がある。　UGFは各種の要求記述言語を統一的に扱うためのグラフ表現によるメタ言語として定義されている。　UGFは情報構造や階層構造の記述に優れており、形式的な言語であるから計算機処理も可能である。　従ってUGFはデータモデルの記述などには適した言語であるが、情報の変換過程や情報の意味記述に対しては、UGFを応用する対象領域でそれを定義しなくてはならない。

本稿では分散処理システムに対する機能要求を定義する記述モデルとして基本的にUGF記述形式を用いる。

機能記述（ＦＤ）は次の３項のタプルで記述される。

$$FD = < I, O, F >$$

但し、Ｉは入力情報、Ｏは出力情報、Ｆは処理概念を表す。　図５はＦＤをグラフ表現したものである。　機能の意味は入力情報Ｉと出力情報Ｏにより記述される。　入力情報と出力情報は情報集合Ｉｎｆの要素として記述される。

$$Inf = AI \cup IG$$

ＡＩは素情報（Atomic Information）の集合であり、本論理機能モデルではこれ以上分解されない情報を表す。　すなわち、任意の情報は３．３節で説明する構成法によって素情報ＡＩを組み合わせることにより生成される。　但し素情報ＡＩは空集合φを含むものとする。　ＩＧはユニバーサルグラフ[6]の集合であり、再帰的構造を持って有向グラフである。　本稿ではこれを情報グラフ（Information Graph）と呼ぶ。　情報グラフ（ＩＧ）は節集合N_Iと弧集合A_Iからなるグラフ集合である。

$$IG = < N_I, A_I >$$

節集合N_Iは情報節集合ＩＮと関係節集合ＲＮに直和分解される。

$$N = IN + RN \quad （直和）$$

情報節集合ＩＮは情報集合Ｉｎｆの部分集合である。　これにより情報グラフは再帰的に定義され情報の階層的表現が可能になる。　関係節集合ＲＮは分散処理システムにおける情報間の関係を表す。　弧集合A_Iの要素ａは

$$a = (n_1, n_2, l_I)$$

と表され、$n_1 \in N_I$、$n_2 \in N_I$であり、弧はn_1からn_2に向かう有向辺として表される。　l_Iは情報ラベル集合L_Iの要素であり、情報間の関係の補完情報となる。　l_Iはφであってもかまわない。

情報記述ＦＤにおける入力及び出力の情報は上記の情報グラフで表現される。　数値集合の情報に対応する情報グラフの例を図６に示す。

次に論理機能の集合Ｆｕｎｃを次のように定義する。

$$Func = AF \cup FG$$

ＡＦは素機能（Atomic Function）の集合であり、本論理機能モデルではこれ以上分解されない論理機能を表す。　すなわち、任意の機能はこれから説明する構成法によって素機能（ＡＦ）を組み合わせることにより得られる。　素機能ＡＦは図５に示すような、２つの情報節と１つの機能節からなる単純なグラフである。　情報節は情報グラフが埋め込まれていてもかまわないが、機能説はこれ以上の分解は許されない。　ＦＧもユニバーサルグラフの集合である。　本稿ではこれを論理機能グラフ（Function Graph）と呼ぶ。　論理機能グラフは節集合N_Fと弧集合A_Fからなる。

$$FG = < N_F, A_F >$$

節集合N_Fは情報節集合ＩＮと機能節集合ＦＮに直和分解される。

$$N_F = IN + FN$$

情報節集合ＩＮは情報集合Ｉｎｆの部分集合であり、機能節集合ＦＮは機能集合Ｆｕｎｃの部分集合である。　弧集合A_Fの要素ｂは

$$b = (m_1, m_2, l_F)$$

と表され、$m_1 \in N_F$、$m_2 \in N_F$である。　l_Fは機能ラベル集合L_Fの要素であり、機能間の関係の補完情報となる。　l_Fはφであってもよい。　図７に例として加法平均算出機能記述FD_1＝＜数値集合，加法平均値，加法平均算出＞の機能グラフを示す。

図５　機能記述

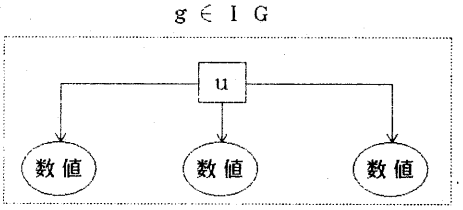

図６　ｎ個の数値集合の情報グラフ

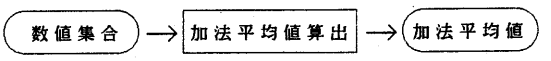

図７　加法平均値算出機能ＦＤの機能グラフ

３．２．論理機能モデルによる意味記述

論理機能モデルは情報処理システムの機能を記述するための抽象的言語と考えられる。　２．３節で述べた要求分析プロセスにおける初期要求記述ＡＳＭＤと分析済要求記述ＡＵＲＤは論理機能記述ＦＤを使って記述される。　情報グラフと機能グラフはその要求機能の構造、意味、関係を記述している。　初期要求記述と分析済要求記述の違いは、素情報ＡＩと素機能ＡＦの記述の深さの違いによる。　論理機能記述における意味の表現は、概念依存理論（ＣＤ理論）[8]と関係している。

ＣＤ理論は自然言語による文章の意味を理解するための方法に関する理論であるが、それによると文章の意味は１４個の基本的概念の依存関係で表現されるとしている。　ＣＤ理論においては文章の意味構造は依存関係のグラフ形式で記述されている。　ＣＤ理論は日常的な文章理解を対象としているが、日常的文章を極めて低レベルな基本概念（primitive）に分割することは、大規模なグラフとなって非効率であり、不可能な場合もある。

これは日常的文章が表現している意味世界とprimitiveが表現している意味の世界の間の意味表現のギャップが大きいことに起因するであろう。ところが、我々が意味記述の対象としている情報処理システムの世界は日常的文章の世界よりはだいぶ制限されており、ＣＤ理論で問題となったprimitiveとの間のギャップがより小さいと考えられる。　現在我々は初期要求記述と、分析済要求記述における<primitive>について検討を行っている。

これらの<primitive>は情報集合Ｉｎｆの素情報ＡＩ及び機能集合Ｆｕｎｃの素機能ＡＦとして利用される。　これらの素概念より情報や機能を構成的に表現する方法について次節で述べる。

３．３．論理機能グラフの操作

論理機能モデルにおいては、機能の意味や制御構造及び情報構造が再帰的グラフ表現により表されている。　このことより大局的構造から詳細構造までの階層構造における任意のレベルの記述が可能になっている。　同レベルにおける機能間の関係は、グラフの接続構造により表現できる。本節では機能グラフ間の結合や埋め込みなどのグラフ操作について述べる。　その前に次の記法を定める。

［記法］

ℓ ：Ｆ	→	Ｉｎｆ	機能Ｆの入力情報を与える関数
σ ：Ｆ	→	Ｉｎｆ	機能Ｆの出力情報を与える関数
D_I：ＩＮ	→	ＩＧ	情報グラフにおける情報節に対応する情報グラフを与える関数（情報詳細化関数）
D_F：ＦＮ	→	ＦＧ	機能グラフにおける機能節に対応する機能グラフを与える関数（機能詳細化関数）

［グラフ操作］

(1)直列合成

機能F_1と機能F_2に次の条件が成立するとする

$\sigma(F_1) \supset \ell(F_2)$

このときF_1とF_2は直列接続可能であるという。

図８に表すようにF_1のグラフ表現をg_1、F_2のグラフ表現をg_2とすると、g_1とg_2の直列接続によりグラフg_3が構成される。　いま、機能Ｆが次の条件を満足しているとする

$\ell(F) = \ell(F_1)$

$\sigma(F) = \sigma(F_2)$

$\sigma(F_1) \supset \ell(F_2)$

このとき機能ＦはF_1とF_2から直列合成されるといい

$F = F_1 \cdot F_2$

と表現する。（図８のg_4）　また、機能グラフg'_3はg_4の機能Ｆに直列的に埋め込まれているという。　即ち$D_F(F) = g'_3$。

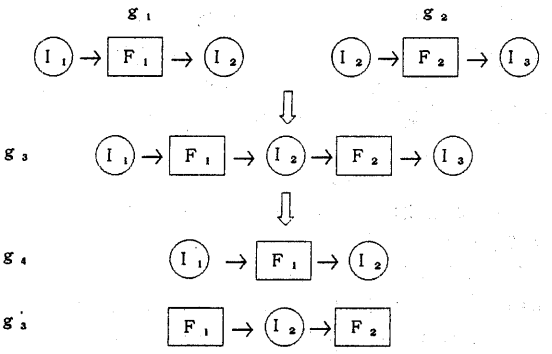

図８　直列合成

(2)並列合成

機能F_1とF_2に次の条件が成立するとする

$\mathcal{I}(F_1)=I_1$

$\mathcal{O}(F_1)=I_2$

$\mathcal{I}(F_2)=I_3$

$\mathcal{O}(F_2)=I_4$

更に次の条件を満足する情報Ｉ、Ｏが存在するとする。

・グラフD_1（Ｉ）の情報節にI_1、I_3が含まれる

・グラフD_1（Ｏ）の情報節にI_2、I_4が含まれる

このときF_1とF_2が並列接続可能であるといい

$F=F_1 \oplus F_2$

と表現する。（図９）　また機能グラフg_3はg_4の機能Ｆに並列的に埋め込まれているという。

これらの操作を用いると、素情報ＡＩと素機能ＡＦから情報システムにおける抽象的な機能記述を構成することができる。　論理機能モデルを用いれば、初期要求記述や分析済要求記述が計算機処理可能な形式で記述される。　初期要求、正規化要求、分析済要求に対する素情報ＡＩと素機能ＡＦを決定し、各変換プロセスにおける変換知識を記述することが今後の問題である。

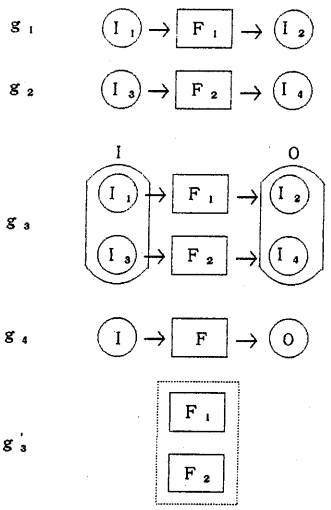

図９　並列合成

４．おわりに

我々は分散処理システムの知識型設計支援システムの開発を進めている。　そのための知識型設計方法論については別稿[1]で述べた。　現在我々は本方法論に基づく要求定義段階の設計支援システムを開発中である。　本稿では要求定義段階の問題依存型知識である論理機能モデルについて述べた。　論理機能モデルは、入力情報と出力情報により機能の意味を記述し、機能グラフの接続関係や埋め込みにより制御構造が記述される。　情報も同様にグラフ構造により情報の持つ意味やデータ構造を記述できる。　本論理機能モデルはユニバーサルグラフ形式に基づいておりその再帰性により段階的詳細化の記述が可能になっている。

本論理機能モデルを使って利用者から獲得される初期要求記述ＡＳＭと分析された結果の要求記述ＡＵＲが形式的に記述される。

参考文献

[1]T.Kinoshita,K.Sugawara,N,Shiratori,"Knowledge-Based Design Support System for Computer Communication System",IEEE JSAC,Vol.6,No.5,1988

[2]木下哲男、菅原研次、白鳥則郎、「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」、マルチメディアと分散処理研究会資料、３７－９、１９８８

[3]M.W.Alford,"A requirements engineering methodology for real time processing requirements",IEEE Tr.SE,Vol.SE-3,No.1,pp.60-69,Jan.1977

[4]D.Teichroew and E.A.Hershey,"PSL/PSA:A computer-aided technique for structured documentation and analysis of information processing systems",IEEE Trans. on Software Engineering,Vol.SE-3,No.1,Jan.1977,pp.41-48

[5]D.T.Ross,"Structured Analysis:A language for communinating ideas",IEEE Trans. on Software Engineering,Vol.SE-3,No.1,Jan.1977,pp.16-34

[6]国井等、「ユニバーサル・グラフ形式（ＵＧＦ）とデータ・モデルへの応用」、ソフトウエア工学、要求仕様技術、共立出版、１９７８

[7]M.W.Alford and I.F.Burns,"R-Nets:A graph model for real time software requirements",MRI Conf. Soft. Eng.,N.Y.,Apr.1976

[8]Schank,R.Conceptual dependency:A theory of natural language understanding,Congnitive Psychology,1974,3(4)