TR-282

ソフトウェア概念設計に於ける知的設計支援

市古　喬男

May, 1988

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

ソフトウェア概念設計に於ける知的設計支援

An Experimental Expert System for Software Conceptual Design

市古 孝男

Takao ICHIKO

(財) 新世代コンピュータ技術開発機構

Institute for New Generation Computer Technology

Keywords:computer aided design, expert systems, conceptual design

Summary----------

This paper discusses current design issues and expert system technologies introduction to its design method in large scale size engineering system, especially from the viewpoint of software conceptual design. Some simulation results applied in an experimental expert system for conceptual design which was developed in the research for reducing software data design overhead found as a typical design issue, are shown as an application example, including next forthcoming directions of intelligent design aids research based on highly parallel network concept (CSP based) for efficient process communication.

はじめに

本研究は、最近益々必要性が高まりつつある大規模エンジニアリングシステムの構築に際し、種々のニーズに対応すべく高度化するソフトウェアの概念設計にエキスパートシステム技術を導入することにより、従来の設計支援技術を知的設計支援技術に高めようとするものである。特に、人工知能システムの為の先進的な知的枠組みだけでなく、大規模エンジニアリングの実現に向けて高度の並列性を効果的に導入し得ることを配慮しつつ、マンマシン・インタフェースでの知的役割分担を明確にする人工知能システム化技術の観点から先進的なツール技術とプロセス通信概念及び超高集積化技術にもとづく汎用性のネットワーク・システム設計技術とを併せた新たな設計手法としての枠組みを指向している。

人のもっとも知的な作業の一つといわれる設計等の作業に関し、人とコンピュータ（以下、ディジタル式電子計算機を主にコンピュータと略称）との係わりは、研究開発史的に見ても比較的古くかつ複雑で、必ずしもその実体を適切に把握することは容易でない。コンピュータは、一つの利用形態としてその誕生当初、人が主として必ずしも複雑ではないが膨大な計算量を消化し得る傍らの道具立てとして、解析・設計作業等の理工学上の領域を中心に注目され使用されていた。然し、この段階ではコンピュータは高速・大容量計算に重宝な道具ではあるが、その本質は計算機能を主体とした算盤・計算尺等に相当するカルキュレータ（計算器）の域を脱するものではなかった。元来、航空工学に於ける弾道計算にあって、高速・大容量の数値計算を主としたコンピュータが必要とされた経緯を考察すれば、それも充分に首肯し得る。而も、計算上の高速・大容量性というコンピュータ利用の側面は、昨今のスーパーコンピュータにあっても見られる如く、現在に於いてなおその需要は縮少されぬばかりか益々拡大の一途を辿りつつあるとも云える。この間、たとえばコンピュータ開発初期の第１世代機と最近の１チップVLSIコンピュータとの比較を引合いに出すまでもなく、コンピュータの装置規格・高速性・消費電力等のハードウェア及び設計概念・制御機能・プログラム言語／サイズ等のソフトウェアに於ける主設計要目での技術的改良・革新には目を見張る程である。然し、この利用形態に関しては、たとえば技術的に原用の微分解析器（DDA ）等に見る母線方式からプログラム内蔵方式（Stored Program Control）へ画期的な飛躍を経たものの、人が設計関連作業のうち計算機能を切出し主として前述のような計算上の高速・大容量性の側面を利用していたに過ぎず、設計等の実現環境として人とコンピュータが一体化された捉え方に注目されはじめたのは比較的最近のことである。この点、人がコンピュータを単なる計算能力をもつ道具とだけ捉えるのでなく、効果的なコンピュータ支援にもとづく協調作業という発想から新しい概念（Computer Aided Design ）が提唱され、人の設計作業とコンピュータとの係わりが広くかつ明確に意識されたのは、今から20年程前で、その後コンピュータグラフィックスなる名称も定着し、関連の設計技術領域等を中心に注目を浴びてきた。それまでのコード化された数字・記号列を介してコンピュータとのインタフェースを実現していた多くの設計者（／操作者）にとって人の感覚に直接訴え得る判り易いコンピュータとの通信手段の獲得は、設計作業（／操作）等に於ける人との係わり及び利用形態を捉え直す貴重な機会となり便利なコンピュータに対する期待は少なからぬものがあった。それ以前と違い視覚化された図形・文字情報の取扱いを実現できるコンピュータグラフィックスペースの設計支援研究は効果的なマンマシンインタフェースの実現を目指してその後多くの研究開発努力が投入されつづけた。然し、コンピュータグラフィックスなる魔法の箱の裏側に組込まれた仕かけの仰々しさに言及する厳しい有識者の指摘を俟つまでもなく、単なる図形入出力機器の位置からたとえばCAD/E 、CAM/T を経て昨今のEWS の重要な要素として脚光を浴びつつある現在まで、関連のソフトウェアをはじめとする環境構築の難しさもあってコンピュータ利用による設計支援（／操作）環境の実現として顕著な地歩を固めてきたものは夫程多くはない。確かに、一方でシリコンブレインのような研究方向も必ずしも定まらぬかにみえる現在、設計作業等に於けるマンマシンインタフェース本来のあり方を探り、効果的な設計支援（／操作）を実現するには、技術的な進歩の速度に比し時日の経過は決して充分とは云えぬ面がある。設計支援（／操作）に於けるマンマシンインタフェースの実現研究は、設計概念・方式・ツール・利用技術等の関連領域が多く、将来の技術動向も踏まえそのあり方を的確に策定する必要がある。大規模システムの実現では、設計（／操作）対象の把握・コンピュータ内外での設計（／操作）モデルの生成・設計（／操作）プロセスの確立と効果的な図式記号化等の表現形式・支援ツール・人の理解／認知モデル・知識情報処理を中心とした大規模システム実現への利用技術等が特に重視される。

１．概要

本知的設計支援研究では、知識情報処理技術を具体的に駆使し、大規模エンジニアリングシステムの開発を容易に行えるよう支援機能に於ける基礎確立を目標としている。特に、大規模エンジニアリングシステムの構築では２．でも言及するが、システムの大規模化・複雑化に伴い、ソフトウェアの設計開発が困難になり、誤りのないプログラムを短期間で作成する手法の開発が大きな課題になっている。

本研究は、上記の設計開発手法に於ける広義のソフトウェアを含めた技術的課題を対象に、知識情報処理技術を中心とした利用可能技術を導入・適用し、知的設計支援の基礎確立に資することとした。即ち、知識情報処理技術を高度化するソフトウェアの開発手法に効果的に導入すること、そして実用化に向けその基盤を固めていくことを目的に、ソフトウェア危機と指摘されつつ未だに確たる解を見出し得ないソフトウェア開発手法の観点から知的設計支援の問題点を明らかにし、研究開発上の知見を得ると共に、基礎的な手法確立の為の確認及び実験を行った。

２．設計問題

本研究の設計手法と知識情報処理技術に於ける技術的課題の分析結果でも示され得る[1,5]が、大規模エンジニアリングシステムの構築では情報・知識の量が膨大になること、同時にその媒体も含め情報・知識が質的にも高くなりそれに伴ってデータ・機能等の特性が益々捉え難くなること、処理速度・応答時間をはじめシステムの設計要目が格段に厳しくなり、設計（操作）等でシステム全貌の把握が難しくなること、ソフトウェアをはじめとしたシステムの移植性・互換性・整合性等が従来以上に重視されること、そして設計（操作）等に於ける自動化・半自動化の的確なトレードオフの選択が構築上必須となること、等々のように挙げられる。従って、上記諸問題の解の一つを見出すべく大規模システムの構築に適した高度の知識情報処理技術が効果的な設計技術と共に要請される。

設計（含、機械設計）に於いて人とコンピュータがはじめて意識的に係わりをもったコンピュータ支援設計CADの提唱以来少なからぬ時日が経過した[1]が、最近では知識情報処理技術の導入と共にコンピュータと設計者を一体としてとり込んだ協調的な設計環境を実現しようとする意欲的な研究開発例が見出される[2]。然し乍ら、最近注目されるEWSでも見られるように設計作業を中心とするコンピュータによる支援は、設計者にとって必ずしもはかばかしい効果を生みだしているとはいい難い。これは、本来的に設計作業が人の創造性に深く根ざしたものであり、その要求が高度かつ複雑でそのものの全貌を的確に把握し難いこと、設計の為のコンピュータをベースとした利用技術の整合性が図り難く基盤技術の確立が容易でないこと、特に実用化を指向したエキスパートシステム（Expert Systems : ES）等の知識情報処理技術を導入・適用しようとする場合知識そのものの表現・獲得の難しさと相俟って利用技術に見合う設計概念・手法等の切出しが複雑で而も従来のCAD以上に設計環境のシステム化を図り難いこと等々を挙げることができる。研究開発の豊富な適用例の数に比して、これらが設計に於ける知識情報処理技術の実用化を著しく阻んでいる主要な原因として指摘できる。

本知的設計支援研究では、このような状況を踏まえて、設計支援機能に於ける基礎的な手法確立の為以下にのべる研究開発を行った。

3. 知的設計支援の為の小規模実験システム

(1) ソフトウェア開発手法

設計支援システムに於けるソフトウェアの重要性は、情報処理の高度化・多様化に伴い益々高まり、システム構築に於けるハードウェアとの比重逆転を促す状況が著しく顕著になりつつある。かつて工芸から確たる科学技術を目指して提唱された構造化プログラミング、ソフトウェア生産性・設計効率向上等を指向したモジュール化構造の概念、そして昨今、ソフトウェア再生産を可能とし真の工業化を促進しようとするソフトウェア部品化のアプローチ等最近のソフトウェア工学の主要な動きを見直しても、その研究開発の困難さが充分窺える。本来プログラム内蔵方式に端を発し問題解決の方法論を中心として発達してきたソフトウェアは、情報処理の有意性・重要性が広く浸透してくるに及び、様々の課題を内包しつつ益々実体の把握し難い巨大な構造を秘めつつあるとも云える。

勿論、高い記述性を要求される肥大化した（高級）ソフトウェア言語の改良・改革、実行母胎となるハードウェアシステムの並列性を主とした高性能化・高機能化、製造技術の画期的な努力により結実しつつある超LSI等のデバイス要素技術の効果的適用、ソフトウェアCADのように必ずしもシステム化が図られているとは云い難いが地道な努力のつづけられている開発支援ツール技術等々のそれを取り巻く周囲の技術状況も大きく変貌を遂げつつある。さきのソフトウェア工学に於ける種々の努力も、その時点での利用可能技術と密接に関連し合い結果として徐々にではあるが奏功しつつあるものの、ソフトウェア技術の研究開発は総じて極めて多様で奥行きが深い。的確な課題分析・設定と幅広い研究開発が要請される所以である。本知的設計支援機能の研究開発でも、この立場を踏まえ知識情報処理技術を指向する[3,4]体系的な知的支援の実現の為前記ソフトウェアの技術的課題を解決すべく努力している。

(2) 実験システムと技術的課題

これまでの記述で示されたように、ソフトウェア工学上の問題と従来の種々のアプローチ及びその成果の比較分析につづき本知的設計支援研究では、知識情報処理に於ける利用技術実現可能性の分析を、さらに設計支援の立場からの見直し（Table 1 設計支援に於ける現利用技術）を行い、その体系的な位置づけに従い以下のような研究開発を進めた。

Table 1　設計支援に於ける現利用技術

知的設計支援システム		
理解の程度	浅い	表層的知識（経験則の羅列等）にたよりすぎる。設計に有効な定量的推論の導入化
扱える領域	狭い	宣言的知識、経験則は豊富でも問題解決能力に乏しい。より原理的な知識から（浅い）知識の生成機能が重要。不完全な情報からの推論、信念の翻意に弱い。非単調論理の導入及び知識の高い整合性の保持指向。 （設計上の）常識の不備。 多様な知識表現の関連を有効に取扱う。
自己成長性	無い	学習能力・帰納能力の段階的装備。 知識の系統的収集の確立化。
処理速度	遅い	並列処理の導入と並列化・分散化へ効果的対応。 人の情報処理様式の模倣可能性（類推、直観的判断他）

知識表現の枠組み	
フレーム型	（設計上の）関連事項を構造体として一つの枠組みに記述。 他のobjectとの共通項もそのobject固有の事項もobjectをkey として把握。
階層型	分類、上位概念、下位概念、抽象と具象、規範とモデル等の関係を親子として記述。 全体像を記述する為の一つの方法ではあるが、意味ネットのような多様な関係の記述を必要とする場合に不充分。 継承性の利用：共通項の記述の節約、階層関係にあるものの論理的必然としての共通項の記述。
概念依存型（スクリプト型）	典型例により関連項を記述。類推常識的判断を容易にする。
データ駆動型	データの変化に伴う影響を局所的に記述。あいまい知識の整合性指向。
ルール	経験則等を断片的に記述。それらの集合の論理的関係により手続きをダイナミックに記述生成可。 概念レベルでの記述ができる（特に変数の使用）。
手続き型	限界と可能性

推論の枠組み
知識データベースの宣言的事実からの推論
ルールによる推論
前向き推論
後向き推論
両方向推論
定性推論
深い知識からの推論
（View Point等による）複数の推論過程の並列処理
（Blackboard型）知識群の協調　　等々

設計ではシステムの入出力要求が与えられたとき、これを実現する為構成要素を組合せかつ各構成要素を順次詳細化することを設計作業と捉えることができるが、本実験システムではソフトウェア開発手法に於ける部品化の概念を効果的に生かした知的設計支援機能の基礎を確立する為、上位の概念設計を主対象に知識情報処理技術を導入した効果的な設計支援の一実現例を試作・開発した。

一般に、設計では目的システムの仕様記述からはじまり出来上り状態の記述までに亘り、要素部品／部品間の関係構造の記述・動作のダイナミックな記述・実現／製造に関する記述等が要求される。一方、設計者は断片的な記述をすることが多く、従って初期設計段階から設計案は矛盾・重複・不足等を含んでいるのが通常である。而もこれまでの記述からも明らかなように、設計自体の全貌を把握することが困難であり、設計者はその設計案の不足を補い整合性をとり全体として一貫性のとれた設計目標に見合うものとしていくことが重要となる。本知的設計支援研究は、この立場から設計プロセスに於ける設計の見通しを良くし設計に於ける基本戦略の方針決定と、設計対象となる目的システムの構造的な捉え方を効果的に支援することを目指すものである[5]。

(3) 実験システム概要

知的設計支援の為の本実験システムでは、前項にも示したような設計プロセス上目的システムの機能要求を概念的に記述できるようにし、設計者の意図した設計案を生成し、かつ検証できることを目標とした：

i) 機能要求
ii) 設計案生成
iii)検証

設計プロセス上、i)で実現された概念的記述は、記述の形式化を通してii)，iii) のプロセスで有効に働くよう記述の構造を設定した。これにより要求獲得の為の効果的なマンマシンインタフェースを実現する。ii) では一般に設計の方法論自体が未確立で、多様な推論・類推等を駆使し演繹的には決まらない所での判断が必要とされ得る。従って、設計プロセス上要求の構造化を図り副次的機能に帰着させることが有効となる。他に、設計方法のルール化、機能ライブラリィ概念の導入が重視される。特に、典型例として取扱うデータの変数化により拡大解釈可能な方法を探り適用範囲・制約条件を設計ルールでチェックできることを目指した。

さらに本実験システムで、機能の記述をルールの集合として表現すれば、ルールの起動によりシミュレーションの結果が得られるということで、概念レベルで機能を記述できることになる。iii)では検証すべき内容が確定すれば、この部分では演繹的推論とシミュレーションが有効に働くよう意図されている。

設計者は、前期のi)～iii)が本実験システム（ESベース）の入出力データを介して実現できる。いわゆるESは本来特定の分野に固有な専門的知識を利用して専門家の代行をしたり、専門的知識の論理的帰結を示して専門家支援をしたりするものを指すが、本実験システムでは必ずしも専門的といえぬ知識の活用・戦略的推論を行うも

のを含め得る。本実験システム構築の詳細は後述されるが、知識表現としては扱おうとしている対象世界の全体像の記述が必要で、多様な断片的知識（対象物も多様、表現内容とその表現形式も多様）とそれらの結合として対象世界を記述することになるので種々の枠組みによる複合型（ハイブリッド）にならざるを得ない。

(4) 実験システム詳細

(a) 設計支援システムとしての役割

本実験システムは、次の設計手法による設計を支援する：

目的システムを実現する為の概念レベルでの設計を主対象とする。

概念レベルでの設計は、一般に

・データ構造とその関係（データ設計）

・機能の規定とそれらの関係（機能設計）

より成る。

i) データ設計

目的システムを記述する為に必要な情報を実体化する。

機能を実現するという観点から、示唆される情報を実体化する。データ間の関係を記述する。

ii) 機能設計

機能の規定としてinput , outputを指定する。機能のfacilityの把握からinput , output（制約付）の組を決める（機能を主体にした観点）。

データの性質からデータ間の関係としてinput , outputの組を決める（データを主体にした観点）：目的指向的（後向き）／駆動方式（前向き）／データ順序の制約条件。

ここで下降型（Top-Down方式）の分解及び上昇型（Bottom-Up 方式）による統合が可能であるので、概念レベルの設計段階でも又それ以降の段階での詳細化設計も指向し得る（モジュール仕様・詳細仕様化）。

以上のような設計手法により、本実験システムでは局所的に設計されたものの集合の整合性をとり構造化することにより、設計の全体像を埋め込むことが可能となる。

(b) 設計案の獲得と解析

(3) でも言及したように、設計者は概念レベルの設計案を本実験システムにより獲得できその設計案の解析を効果的に進めることができる（利用者固有の検証も構築可能）。

それは以下のように実現される。

i) 設計案の獲得

利用者によりデータ記述が与えられると、データ概念に属するものをデータのサブクラス又はメンバーのユニットとして登録する（各データを構成する項目をスロットとして記述）。

データ間でIs-a, instance-of の関係にあるものに親子関係をつける。

一方、機能記述についても、機能概念に属するものを機能のサブクラス又はメンバーのユニットとして登録する（このとき機能概念として用意されている属性名／block , input , outputがスロットとして継承される）。各機能を規定するものとして次のスロット値を与える：

block 値　マクロ機能名。

input 値　入力データの概念的記述による指定。

output値　出力データの概念的記述による指定。

入出力データの概念的記述（それにより意味するものを概念的データと呼ぶ）として次のものが解釈可能である：

概念的データ名

データ記述で与えたユニット名、又は（equivalent　データ名　概念的データ）によって直言されたデータ名によって指定。

(pair　概念的データ　seq 指定コード)

seq[6,7]指定コードによりその概念的データが受け渡される順序の指定を行う（その概念的データとは、擬似等価の関係（quasi-equivalent）にあるとする）。

(specify 概念的データの項目　項目の値)

項目の値によって分類された概念的データの特定例（その概念的データのIs-a又はinstance-of）を指定する（その概念的データとは、同一クラス内で等価（in-class equivalent ）の関係にあるとする）。

(part　概念的データ　部分項目名)

部分項目名によってその概念的データの部分を指定する（その概念的データとは、擬似等価の関係にあるとする）。

なお、上記でseq 以外はpar である。

ii) 設計案の解析

利用者より与えられた設計案の解析の為に機能の結合関係を（論理的な）整合性の観点から検証する。

機能に関するinput , outputから結合関係にある機能を検出することにより設計案の全体像を示す。

link　　：同一（又は等価な）概念的データの受渡しによる結合。

in-class link ：データ記述に於けるis-a, instance-of の関係又は概念的記述に於ける同一クラス内等価関係にある概念的データの受渡しによる結合。

quasi-link ：擬似等価の関係（Ex. part-of ）にある概念的データの受渡しによる結合。

なお、block 化された機能の全体としての入出力データを検出することも可能である。block 内機能の結合関係にないinput , outputを検出することにより、block 全体の整合性のチェック又は統合化されたblock （マクロ機能）をその全体としてのinput , outputによって規定する。後出のEntry , Exit等の例もこの点効果的である。

(5) 実験システムによる応用・評価

これまで、ソフトウェア開発に於ける知的設計支援機能として知識情報処理技術を駆使し得る基礎技術の現状分析と諸検討を踏まえて小規模実験システムを切出しその効果的な実現研究をのべてきたが、ここでは前項までの本実験システム構築の趣旨に従いソフトウェア開発手法に於ける典型的な設計課題に具体的に適用した結果を論ずることとする。

本実験システムの応用対象としてとり上げたのは、オンライン在庫管理に関する設計課題で、これは過日ソフトウェア設計に関する情報処理学会シンポジウムで話題となった共通の設計課題である[5,8] 。このオンライン在庫管理は、注文処理・データベース管理・入庫管理・出力処理等からなり、階層性・並列性の高い比較的高度の情報処理システムとして捉えられ、順次設計の基本戦略を設計者の思想に従い設計案として実現していく本実験システムに恰好の応用例である。

前記(4) に沿い、(3) のi)～iii)等を順次くり返しつつ得られた結果を次のFig.1-4 に掲げる。特に、Fig.1 はソフトウェア概念設計に於ける機能分解／統合化の支援過程で、ソフトウェア概念構造の部分及び全体の整合性が図れる本実験システムの効果的な利用のしかたを示す例である。設計者のinput , outputは、uservariable unitのスロット値として示され（Fig.2 ）、設計案として出力結果が前述のlink・in-class link ・quasi-link等他に分けて示される（Fig.4 ）。これにより、従来のソフトウェア開発手法に於けるデータ設計等の大きな利用負担を負うことなく、設計目的に見合う概念構造にもとづいて効果的に自己の設計案を構築していくことが可能となる。

本研究では、いわゆるESの第２世代として最先端技術とされる知的枠組みを用いているが、標準機能だけでなく非標準的な（Unstructured）構造も多く導入せざるを得ず（Ex. Unstructured Fact, Unstructured Description for Rule Conditions, etc. ）、又ルール数の制限等からルール表現も通常の場合に比して深さをはじめ幾多の工夫が加えられた。

(3) の後部で言及したが、必ずしも専門的でない常識等の導入をはじめ高度の論理的な推論を行うという観点から設計支援のような知的システムを捉える場合、現状のESでは処理速度の低さの他理解の程度が浅く扱える領域の幅が狭くかつ自己成長能力がない等の欠点を指摘できる。本実験システムでは、前記を克服すべく特に知識表現・高次推論の枠組みを高めること及びソフトウェア部品化の概念を向上すること等を重視している。

むすび

現在、診断向け等に比し困難視されている設計向けエキスパートシステムについて、試作・開発が盛んになりつつあるが、そこでは知識の質・量ともに広大な設計のどの領域に焦点があてられ実用的にどの側面が解決されようとしているかが重視される。この場合、人とコンピュータの通信形態を規定するマンマシンインタフェースの知的化では、オブジェクト指向の有効性と併せて現実の設計に於けるたとえば知識の形式化から知識の表現・獲得・蓄積さらに学習機能等に亘る実用性を見究めて、設計知識を有効活用しつつ設計の全貌(Design Coverage)を把握し易くする知的設計支援技術の追求される必要がある。試行錯誤問題として、たとえば知的木探索手法なる捉え方を重視する向きもあり注目されるが、設計問題に於ける設計知識・手法等の構造化の可否をはじめ人の高度な知的活動を妨げずにコンピュータ間との整合性を効果的に維持していける知的化が切に望まれる。設計の危機が関連技術領域から叫ばれて久しいが、たとえば極度に創造的な設計問題から著しくルーチンワーク的なものまでの類型化にあっても実用的な設計向けエキスパートシステムの試作・開発では、人とコンピュータとの知的役割分担を相補的な観点から見出していく努力が要請される。本研究では、周囲の技術動向を比較検討しつつ[9] 、現実の設計問題に立ち返り実用性を重視し設計知識の有効活用を図り得るよう人とコンピュータの役割分担を位置づけている。そこでは、ソフトウェア概念設計に於ける知的設計支援として開発済みの実験システムにより得られた概念構造にもとづいて現実の１チップVLSIコンピュータ[6,7] をベースにした複数のノードから成る汎用性の並列化ネットワークシステム上でその応用システムとしての実現可能性を確認している。今後は、その総合的な設計手法としてのシミュレーション系をベースにその枠組みを活かして性能・機能の両側面からも効果的に並列性を導入しつつ高度のill-structuredな設計問題への適用拡張性をさらに図っていくこと、及び昨今の研究開発の内でもたとえば設計理解のレベルに合せた説明能力の向上が期待される説明ベースの学習機能（Ex. EBL ）につきその導入・利用可能性を具体的に検討していくこと等々が上記の趣旨からも例示される。

◇参考文献◇

[1] 穂坂　衛、コンピュータグラフィックス（産業図書、東京1974）、315.
[2] D.C. Brown, B.Chandrasekaran, Knowledge and control for a mechanical design expert system, IEEE Expert, 92-100 (1986).
[3] 渕　一博、武井欣二、第五世代コンピュータ前期計画の成果、電子通信学会、Vol.69, No.6, 556-567 (1986)他.
[4] 第五世代コンピュータに関するシンポジウム　昭和61年度成果概要報告　黒住恭司、他.
[5] T.Ichiko, VLSI oriented computer science study for the FGCS, IEEE Int'l conf. on VLSI and computers, 74-77 (1987).
[6] C.A.R. Hoare, Communicating sequential processes, Comm. of ACM, Vol.21, No.8, 666-677 (1978).
[7] D.May, R.Shepherd, Occam and transputer, IFIP Concurrent languages in distributed systems, 19-33 (1985).
[8] 山崎利治、共通問題によるプログラム設計技法解説（その２）、情報処理、Vol.25, No.11, 1219 (1984).
[9] Y. Nagai, H. Kubono, Y. Iwashita, -PROTON- Expert system tool on the PSI machine, ICOT TR-0226, 1-20 (1986) (published at Avignon'87).

謝辞
本研究に関しては、渕　一博所長、黒住恭司次長をはじめとした（財）新世代コンピュータ技術開発機構等の関係各位に感謝する。特に広義のソフトウェアまで含め竹内（憲）氏、PROTONにつき永井（保）氏にも深く謝意を表する。

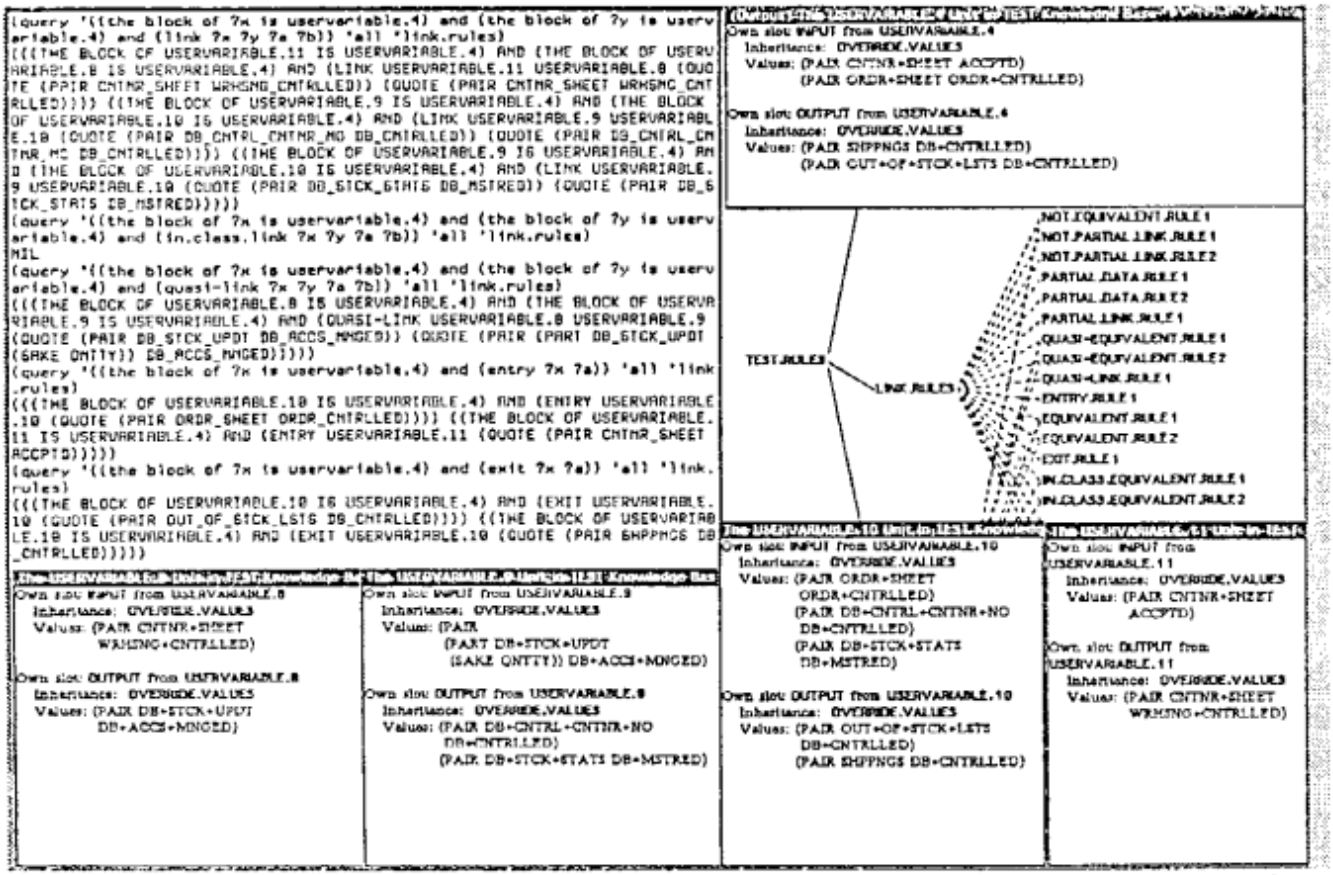

Fig.1 An application result in a specified block.

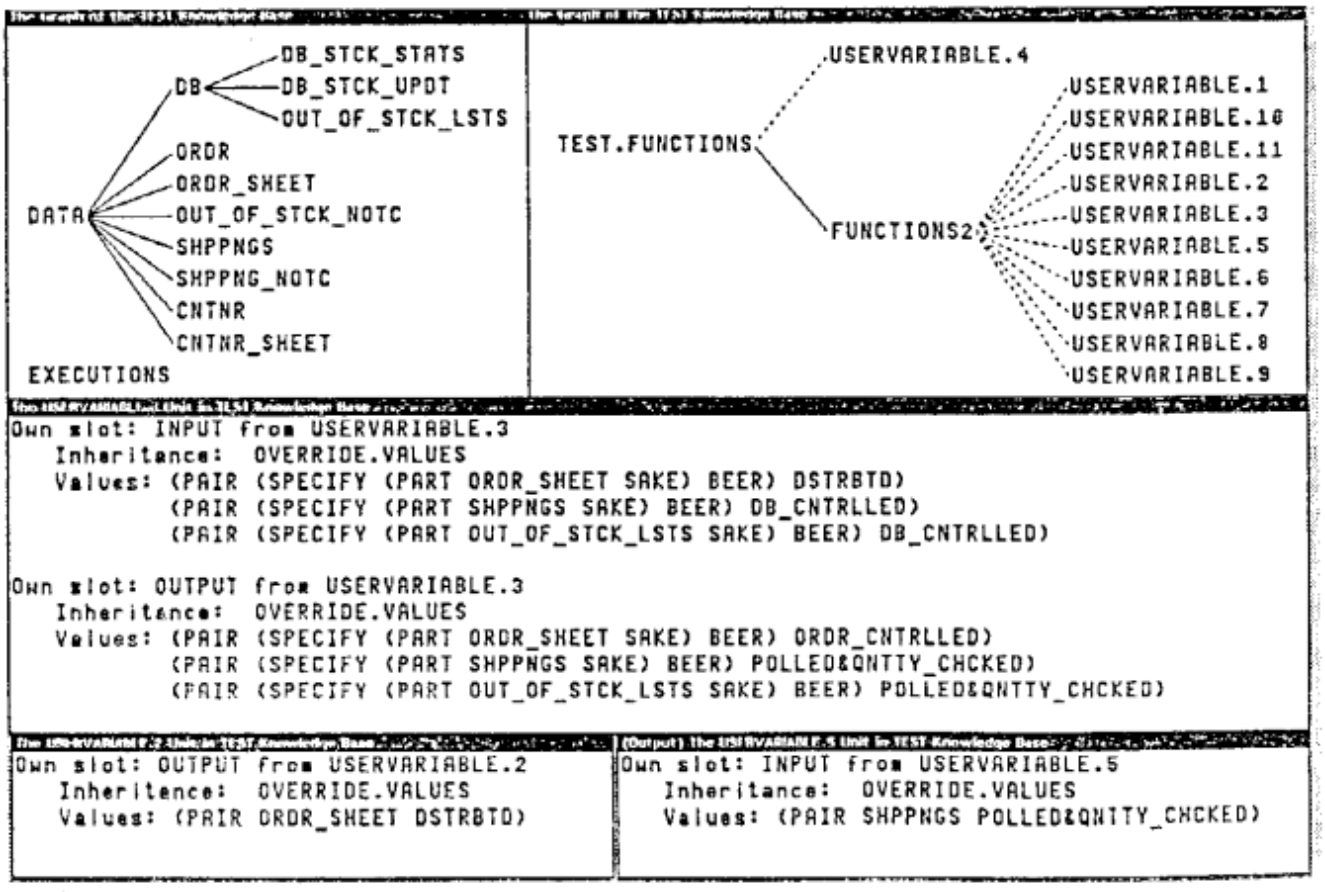

Fig.2 Conceptual data description and its hierarchical representation.

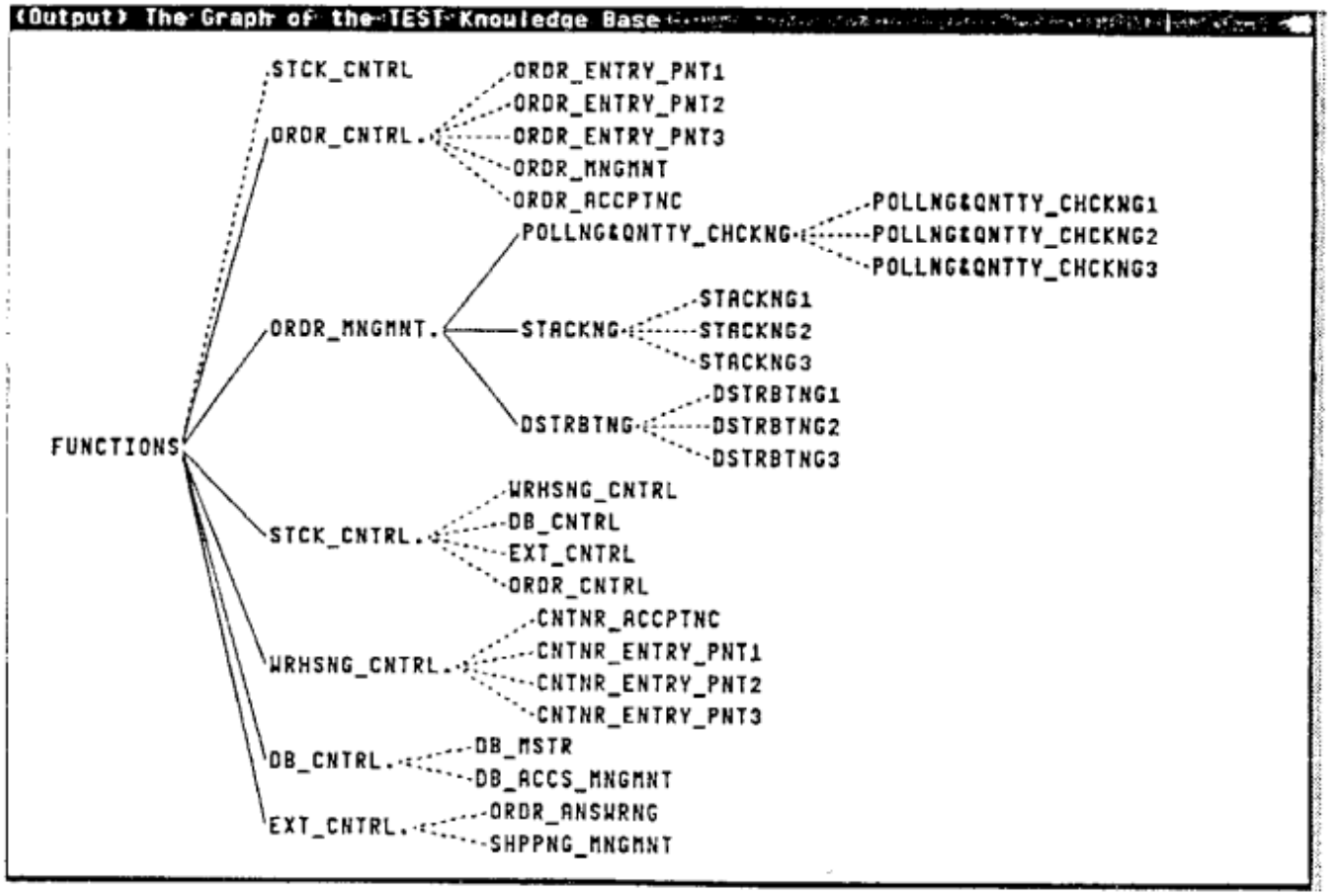

Fig.3 Hierarchical representation of functions.

```
(Output) The RESULTS Slot of the EXECUTIONS Unit
Own slot: RESULTS from EXECUTIONS
   Inheritance:  OVERRIDE.VALUES
   Values: ((LINK USERVARIABLE.9 USERVARIABLE.10 (QUOTE (PAIR DB_STCK_STATS DB_MSTRED))
             (QUOTE (PAIR DB_STCK_STATS DB_MSTRED)))
            (LINK USERVARIABLE.9 USERVARIABLE.10
             (QUOTE (PAIR DB_CNTRL_CNTNR_NO DB_CNTRLLED))
             (QUOTE (PAIR DB_CNTRL_CNTNR_NO DB_CNTRLLED)))
            (LINK USERVARIABLE.7 USERVARIABLE.11 (QUOTE (PAIR CNTNR_SHEET ACCPTD))
             (QUOTE (PAIR CNTNR_SHEET ACCPTD)))
            (LINK USERVARIABLE.11 USERVARIABLE.8
             (QUOTE (PAIR CNTNR_SHEET WRHSNG_CNTRLLED))
             (QUOTE (PAIR CNTNR_SHEET WRHSNG_CNTRLLED)))
            (LINK USERVARIABLE.1 USERVARIABLE.2 (QUOTE (PAIR ORDR_SHEET ACCPTD))
             (QUOTE (PAIR ORDR_SHEET ACCPTD))))
           ((IN.CLASS.LINK USERVARIABLE.3 USERVARIABLE.6
             (QUOTE
              (PAIR (SPECIFY (PART OUT_OF_STCK_LSTS SAKE) BEER) POLLED&QNTTY_CHCKED))
             (QUOTE (PAIR OUT_OF_STCK_LSTS POLLED&QNTTY_CHCKED)))
            (IN.CLASS.LINK USERVARIABLE.3 USERVARIABLE.6
             (QUOTE (PAIR (SPECIFY (PART SHPPNGS SAKE) BEER) POLLED&QNTTY_CHCKED))
             (QUOTE (PAIR SHPPNGS POLLED&QNTTY_CHCKED)))
            (IN.CLASS.LINK USERVARIABLE.3 USERVARIABLE.5
             (QUOTE (PAIR (SPECIFY (PART SHPPNGS SAKE) BEER) POLLED&QNTTY_CHCKED))
             (QUOTE (PAIR SHPPNGS POLLED&QNTTY_CHCKED)))
            (IN.CLASS.LINK USERVARIABLE.3 USERVARIABLE.10
             (QUOTE (PAIR (SPECIFY (PART ORDR_SHEET SAKE) BEER) ORDR_CNTRLLED))
             (QUOTE (PAIR ORDR_SHEET ORDR_CNTRLLED)))
            (IN.CLASS.LINK USERVARIABLE.2 USERVARIABLE.3
             (QUOTE (PAIR ORDR_SHEET DSTRBTD))
             (QUOTE (PAIR (SPECIFY (PART ORDR_SHEET SAKE) BEER) DSTRBTD)))
            (IN.CLASS.LINK USERVARIABLE.10 USERVARIABLE.3
             (QUOTE (PAIR SHPPNGS DB_CNTRLLED))
```

Fig.4 Final application results mapped as links (1/2).

```
(Output:The RESULTS Stock of the EXECUTIONS Unit
        ((IN.CLASS.LINK USERVARIABLE.2 USERVARIABLE.6
          (QUOTE
           (PAIR (SPECIFY (PART OUT_OF_STCK_LSTS SAKE) BEER) POLLED&QNTTY_CHCKED))
          (QUOTE (PAIR OUT_OF_STCK_LSTS POLLED&QNTTY_CHCKED)))
         (IN.CLASS.LINK USERVARIABLE.3 USERVARIABLE.6
          (QUOTE (PAIR (SPECIFY (PART SHPPNGS SAKE) BEER) POLLED&QNTTY_CHCKED))
          (QUOTE (PAIR SHPPNGS POLLED&QNTTY_CHCKED)))
         (IN.CLASS.LINK USERVARIABLE.3 USERVARIABLE.5
          (QUOTE (PAIR (SPECIFY (PART SHPPNGS SAKE) BEER) POLLED&QNTTY_CHCKED))
          (QUOTE (PAIR SHPPNGS POLLED&QNTTY_CHCKED)))
         (IN.CLASS.LINK USERVARIABLE.3 USERVARIABLE.10
          (QUOTE (PAIR (SPECIFY (PART ORDR_SHEET SAKE) BEER) ORDR_CNTRLLED))
          (QUOTE (PAIR ORDR_SHEET ORDR_CNTRLLED)))
         (IN.CLASS.LINK USERVARIABLE.2 USERVARIABLE.3
          (QUOTE (PAIR ORDR_SHEET DSTRBTD))
          (QUOTE (PAIR (SPECIFY (PART ORDR_SHEET SAKE) BEER) DSTRBTD)))
         (IN.CLASS.LINK USERVARIABLE.10 USERVARIABLE.3
          (QUOTE (PAIR SHPPNGS DB_CNTRLLED))
          (QUOTE (PAIR (SPECIFY (PART SHPPNGS SAKE) BEER) DB_CNTRLLED)))
         (IN.CLASS.LINK USERVARIABLE.10 USERVARIABLE.3
          (QUOTE (PAIR OUT_OF_STCK_LSTS DB_CNTRLLED))
          (QUOTE (PAIR (SPECIFY (PART OUT_OF_STCK_LSTS SAKE) BEER) DB_CNTRLLED))))
        ((QUASI-LINK USERVARIABLE.8 USERVARIABLE.9
          (QUOTE (PAIR DB_STCK_UPDT DB_ACCS_MNGED))
          (QUOTE (PAIR (PART DB_STCK_UPDT (SAKE QNTTY)) DB_ACCS_MNGED))))
        ((ENTRY USERVARIABLE.7 CNTNR_ENTRY3) (ENTRY USERVARIABLE.7 CNTNR_ENTRY2)
         (ENTRY USERVARIABLE.7 CNTNR_ENTRY1) (ENTRY USERVARIABLE.1 ORDR_ENTRY3)
         (ENTRY USERVARIABLE.1 ORDR_ENTRY2) (ENTRY USERVARIABLE.1 ORDR_ENTRY1))
        ((EXIT USERVARIABLE.6 OUT_OF_STCK_NOTC) (EXIT USERVARIABLE.6 ANSWERS)
         (EXIT USERVARIABLE.5 SHPPNG_NOTC))
```

Fig.4 Final application results mapped as links (2/2).