ソフトウェア工学 114－7
（1997．5．23）

設計知識の自動獲得と自動設計を行なう知的CASEツール

陳 慧† 勇 宏幸† 堤 永保‡ 河野 善彌†

†埼玉大学工学部情報システム工学科
〒338 埼玉県浦和市下大久保255
‡日立中部ソフトウェア（株）研究開発部

あらまし ソフトウェアの自動設計には，知識等の大きな技術課題が多数ある．筆者らは人の設計に倣うエキスパートシステムによるソフトウェアの設計自動化を研究している．本論文は知的*CASE*ツール，ソフトウェアの設計を自動的に行なう汎用的なシステムについて述べる．システムの基本的な考え方は，設計者が*CASE*ツールを利用して設計図面を書きながら設計知識を自動的に獲得し設計知識ベースを構築し，獲得した設計知識を再利用して設計情報の展開を自動的に行わせることである．これらにより設計に使い易いものとなり，全体システムの設計コストも大幅に減少させることができる．

An Intelligent CASE Tool of Acquiring Design Knowledge and Design Automatically

Hui Chen† Hiroyuki Isami† Nagayasu Tsutsumi‡ Zenya Koono†

†Department of Information and Computer Science,
Faculty of Engineering, Saitama University
255 Shimo-okubo, Urawa 338, Saitama, Japan
‡Research & Development Div., Hitachi Chubu Software, Ltd.

Abstract This paper reports on an experimental Intelligent CASE (Computer Aided Software Engineering) tool featuring automatic software design as well as automatic design knowledge acquisition. The system is composed of a conventional CASE tool and expert systems. The main idea is that human designers use a CASE tool for an initial design of a software system and, designers' design knowledge is automatically extracted upon designers' command and stored in an expert system. Reusing thus acquired design knowledge by the expert system, a similar software design can be performed automatically. As this system saves many hours required for design, it is a much effective tool than a conventional CASE tool. As a result, the intelligent CASE becomes a convenient tool for software designers, which saves software development costs substantially.

1 はじめに

ソフトウエアの自動設計はソフトウエア開発者の夢と言われ，ソフトウエア工学と知識工学の両面から各種の研究が多年なされている [8]. ソフトウェアの自動設計には，知識等の大きな技術課題が多数ある．筆者らは人の設計に倣うエキスパートシステムによるソフトウェアの設計自動化を研究している [9, 10]. これは部分毎の自動化をボトムアップに実現し，徐々に高度化して自動化率を高める長期構想に基いている．本論文は，この構想に基づき，最下位レベルの作図作業を自動化するもので，

1. 汎用的な CASE ツールを使う
2. 設計知識を自動的に獲得する
3. 設計知識を再利用して自動設計を行なう

ことにより，設計作業の下層の作図の自動化を行なわせるものである [4, 5, 14].

2 自動設計の基本構想

本研究のソフトウエア自動設計の基本構想 [10, 11, 12] は次のとおりである.

1. 設計とは，ある概念を階層的に展開する事を繰り返し，次第に具体化し詳細化し，最後に対応するデータ演算処理のソースコードに変換することである.
2. エキスパートから設計知識を取り出して，人の設計知識体系を得る．人の設計をシミュレートする考え方で，ボトムアップにエキスパートシステムを再構築し，自動設計を行なわせる.
3. 対象とするエキスパートとして高い成熟度のソフトウェア開発組織をとる.
4. エキスパートの知識体系の基本モデルとして階層的な工程をとる.
5. エキスパートの作業方法として，細かいステップ毎に文書図面を作る事を利用する.

3 知識獲得とその再現

3.1 原理

知識獲得は系統的方法がなく，知識エンジニア方式が従来の中心的技術である．本研究では，まづモデル化により系統的な設計知識の系統的な獲得を可能とし，更に今回自動獲得を可能とした．次にソフトウェア工学的により，エキスパートシステムの系統的構築を可能とした．これにより設計の再現すなわち自動設計が系統的にできる [6].

前提として，高成熟度ソフトウェア開発組織を対象エキスパートにとり，これをモデル化して図 1のような階層的な設計工程すなわち，系統的な知識体系をとる．系統的な工程の最下位レベルの作業は図面作成作業である.

図 1[12, 13] の左の階層的な工程の下流の設計について，その下層の作図工程を取り上げる．フローチャート類の設計作業を小さな進行毎に区切り，それぞれ図面を残せば，図 1左下のように設計情報は階層的に展開する．図のシンボルに併記する自然言語による説明には，統一し標準化した用語を用いる．作業規約に従って正確に図面を作りさえすれば，相隣る図面間の差から「設計ルール」を獲得できる．これらの単位的展開「設計ルール」を繋ぎあわせて 1 枚の図面を構成するための，より一般的な設計知識も図面から再現できる．設計ルールと図面は設計の双対な知識の体系になる.

上位階層では個別作業を抽象化した結果について作用する汎用的な知識が顕著なので，細分化し特定した作業毎に「如何なる場合に如何にすべきか？」を指定して熟練者から回答を得て，工程毎の知識を再現できる．これら両方法を併用すると，設計の知識は系統的に獲得し，再現できる．図 1の右はこれを示す．この方法によれば設計知識を，確実に，高い再現性と高い信頼度で，系統的に，誰でも，獲得できる [11, 12, 13].

知識体系のモデルに従った階層的工程を枠組とし，各工程の設計知識をエキスパートユニッ

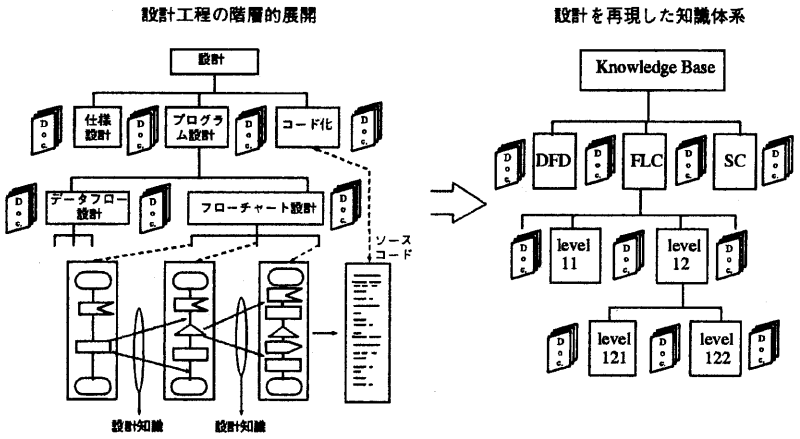

図 1: 設計工程と再現した知識体系の階層構成

ト化（ここでは，工程毎の小形エキスパートシステムをエキスパートユニットと名付ける．）して組込み全体で一つのエキスパートシステムを構成し，設計作業を自動化させる [2, 3]．漸進的に研究を進めるので，まづ，作業量が多く作業が簡単な下位階層からエキスパートユニットを実現して枠組に作り込み，これを順次繰返して疑似度を向上させ，設計の自動化率を向上させる．

3.2 自動設計システム

図 2[11] は自動設計システムの最下層の部分の原理図である．上の部分は初回の人による左から右へと 2 段階の設計の詳細化過程を示している．この各段階の展開関係を単位的な設計知識と考え「設計ルール」と名付け，これを抽出してエキスパートシステムの知識ベースに蓄積する．エキスパートシステムにより設計知識を再利用し，設計を自動的に行なわせる．当初の入力を与えると，当初の最終出力と同じ出力が得られる．設計知識が足りる場合は自動的に設計でき，不足の場合は当該部分を人が設計し，その新しい設計知識を追加する．すなわち，本研究では，システムを使いながら知識ベースの内容の蓄積を行なう．この方式は長期間にわたり継続的に機能拡張や変更を重ねるシステムへの適用を意図している．この方式では設計経験を積めば，高い自動化率を達成できることが見込める．

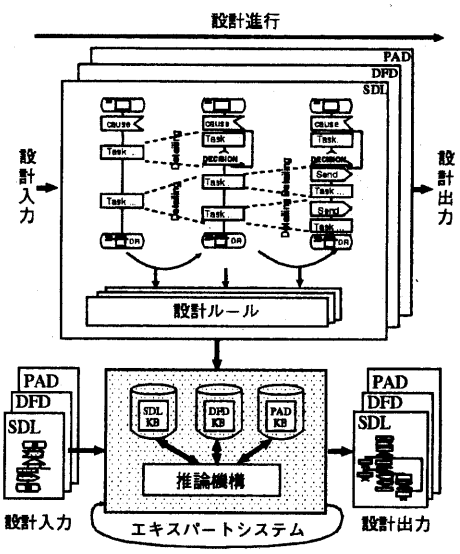

図 2: 自動設計システム構成

3.3 エキスパートの特性

図 3[1] は，横軸に設計経験累計数をとり，縦軸はそれまでに使用した設計ルールの累計数を両対数尺度で示す．プロットの傾向線は直線に

なり，習熟効果[7]の存在を示す．図3のように第1次(フローチャートレベル)方式，(構造化チャートPADの使用，設計ルール群のブロック化等で)大幅に改良した方式のいずれも習熟効果を示す．これは両直線尺度でプロットすると初めは急激に増え，次第に増加が緩やかになる傾向を示し，設計毎の新しい知識の追加の割合が低下する事を示している．これは人の場合に「設計経験を積むにつれて作業が楽になる」ことと同じであろう．この方式は使用経験を積むにつれて自動化率が向上していく．

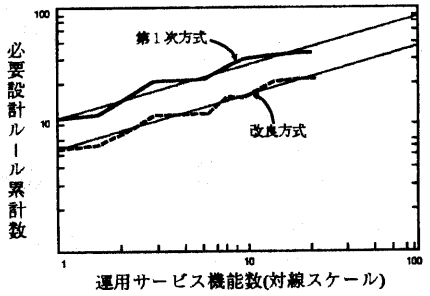

図 3: 運用サービス機能数と設計ルールの関係

3.4 今回の改善

これまでの知識獲得の構想は，初回の母体の設計を全部人手で行い，設計の完了後に，設計知識を人間が抽出する考え方であった．これにはいくつかの欠点がある．

- 初回の設計では多くの試行を繰り返すから，かなりの手間がかかっている．
- 設計知識の抽出は全面的に人手に依存しているので，更に手間がかかる．

この問題を解決するため，次のようにする．

- 初回の設計から自動設計を併用する．
- CASE ツールを使用して設計作業を行ない，設計知識の獲得を自動化する．

4 知的 CASE システムの構成

4.1 システムの基本構想

従来構想より進歩させて初回の母体の設計から設計知識を再利用する．設計者は，設計する度に，設計知識の自動抽出を指示するのみで抽出を行なわせる．基本的な考え方[4, 5]は次のとおりである．

- システム開発の初回設計からCASE ツールを用い，設計文書図面を作成させる．
- 2枚のデータフロー図やフローチャートからでなく，1枚の図面から設計知識を取り出し，獲得を容易化する．1図面内の設計はより小さいステップで単機能的階層展開するから，より基礎的汎用的な知識となる．
- 書いた設計図面から設計ルールを自動的に抽出し，蓄積する．蓄積した設計ルールを再利用し，設計を自動的に行なう．
- 自動設計にあたってはいくつかの候補を表示し，その中から選択させる．
- 設計者は選択した設計知識の候補を必要に応じて修正や拡張を行なえる．この修正した設計結果についても設計知識を自動抽出し，整理して蓄積する．
- 設計文書図面の生成支援だけでなく，CASE ツールのソースコード自動変換の機能を用いてプログラムを自動生成させる．
- システム開発の初回の設計では，書いたり消したりが多い．この修正過程中に前項の蓄積知識を再利用して自動設計を行なわせ，初回の設計をより容易にする．

4.2 システムの構成

この知的CASE システムの主要な構成と機能は図4に示したとおりで，PAD CASE ツール本体，設計知識の自動抽出，設計知識の蓄積と設計知識の再利用の各機能である．

1. 設計：図4の上部に示すように設計者はCASE ツールの作図機能を用いてシステム，サブシステム，デパートメント，詳細論理など論理レベルで段階的にシステムの設計図面を作成し，修正する．設計は全て階層的展開であり，全階程で同様な手続き

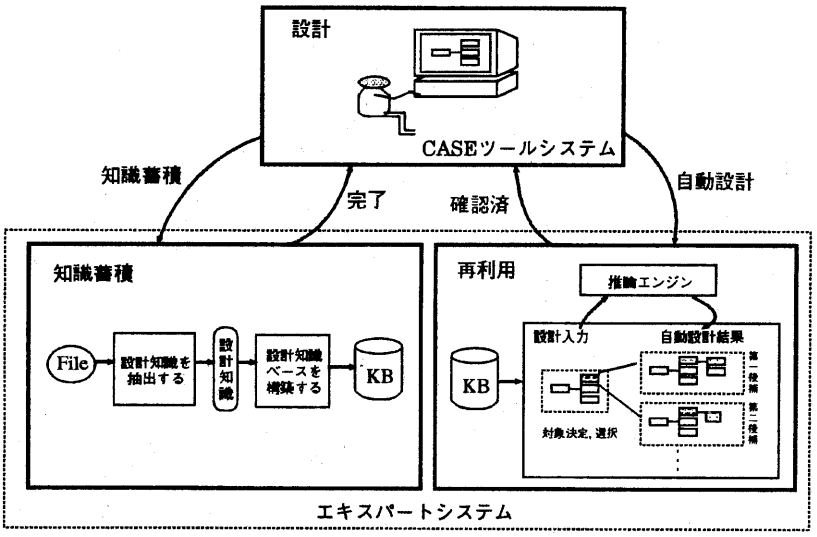

図 4: 知的 CASE ツールシステム構成

を繰り返し，最後はモジュール，関数，各単位機能に至るまで同一方法で詳細化できる．

2. 知識蓄積：図4の下左部に示すように，設計者は小さな段階毎の設計を終える度に，知識蓄積指令をエキスパートシステムに送って，知識抽出エンジンを働かせる．ある設計の前後の展開関係を単位的な設計知識と考え「設計ルール」と名付けて自動的に抽出する．抽出した設計知識を知識ベースに蓄積する．

3. 自動設計：図4の下右部に示すように，設計者は設計情報と自動設計要求指令をエキスパートシステムに送って，推論エンジンを働かせる．設計情報により既存知識ベースから設計知識を検索し，設計情報から対応する幾つかの小さな段階毎の設計結果(第一候補，第二候補・・・)を求め，出力する．設計者はこれらの結果から最も要求に合うものを選ぶ．選んだ設計候補については，人による修正や拡張を行う事ができ，この修正された結果についても設計知識を自動抽出し，整理して蓄積する．

4. 詳細化設計が終ると，CASE ツールのコード変換機能を利用して，詳細な設計ドキュメントに基づいてプログラムのソースコードを自動生成させる．

5 設計知識の獲得と蓄積

5.1 構造化チャート PAD

このシステムは既存の PAD CASE ツールを利用する．この章は PAD について説明する．

PAD (Problem Analysis Diagram) は構造化チャートの 1 種である．図5の上部に小さい間隔毎に設計文書を作っていく流れを示す．

PAD では，プログラムの全体を構成する制御の流れを図のように表す．プログラムは左端の垂直な基本線を基に始め，上から下へ制御が流れる．それぞれの機能は基本線を基に，左から右の横方向に，第 1 列，第 2 列と展開されていく．PAD ではある設計段階の入力から出力への変換過程は図に示すように PAD シンボルの展開で表されるので，この単位的変換を設計ルールとする．

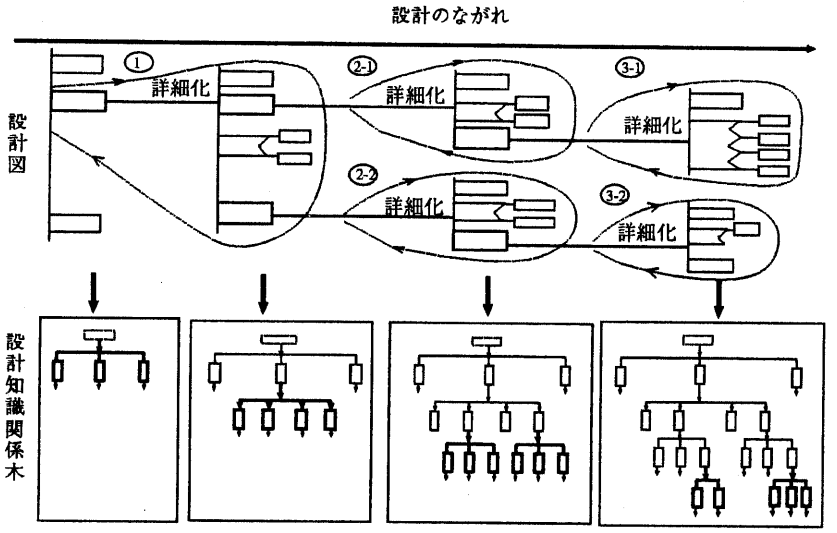

図 5: 設計知識の獲得と蓄積

PAD の各シンボルは，各設計段階での自然言語による概念を記入する．全体として，PAD 図は構造的なソースコードの表記形態とよく一致している．そこで，最終段階では，このシンボル中にソースコードを記入しておき，ほぼ機械的に変換してソースコードリストを得る事ができる．

筆者等のこれまでの方式では，設計の前と後の 2 枚のフローチャート等の図面を用いていた．これは図面にまたがるので相互の対応関係が見にくい．構造化チャート PAD を使うと，(1) 設計の小さな進行は同じ図面の中での進行になり，(2) より小さいステップで設計ルールを取りだす事ができて柔軟性が増し，(3) 異なる 2 枚の図面を使うより見やすくなり，(4) 同一図面上で設計知識の獲得を自動的に行なうことができ，(5) 自動コード変換などプログラム化に適合している．

5.2 知識の自動獲得と蓄積

図 5に示す例について説明する．図 5の上 (1) について，図の矢印のように一つのシンボル (親)(設計入力パターン) から出発して図形情報をたどり，それを展開した一番上の子のシンボル群（設計出力パターン）を経由して出発点に戻る．これは一つの設計ルールにあたるから，図をたどりながらシンボル毎の図形情報と設計情報を取り出す．取り出した情報を解析すれば，常に単位的な展開結果を設計ルールとして自動的に獲得できる．出発点を展開の親として波面状に 1 の次に 2-1, 2-2, ... と設計ルールの獲得を進める方式の例を示している．

設計情報は図 5のように階層展開になるので，下に示す知識関係木は順次大きくなり，クリスマスツリー状の知識ベースになる．設計が進むにつれてクリスマスツリーを下りながら，新たな知識を知識ベースにどんどん加えて行く．その最後の段階では，自然言語で表記した概念をソースコードに変換した結果を蓄積する．

本システムにおける設計知識ベースは「シンボルデータ」，「シンボル対応関係」，「図面情報」と呼ばれる 3 種類のデータから構成される．設計仕様中のシンボル毎に関する情報は「シンボルデータ」テーブルとして保存し，シンボルの展開関係，すなわち設計ルールは「シンボル対応関係」テーブルとして保存される．

各シンボルに関する図面情報は「図面情報」テーブルとして保存される．シンボル対応関係は図5の下のように木構造を用いて記述され，この方式により，多くの情報を記録する事ができ，各種の検索も高速が容易に行なえる．

設計では，書いたり消したり，修正することが多くある．設計情報（図面）を修正変更したら，知識情報の一貫性のため，設計知識ベースをも修正変更する．設計の知識を再利用するため，修正後のものを次の候補として知識ベースに追加登録していく．

このようにして，全く知識のない場合にも設計の試行過程の設計情報を蓄えて，初回から自動化の効果を挙げたいと考えている．

6 自動設計

設計者は出発点となるＰＡＤ図（１以上のシンボルを含む）を書き，自動設計を開始させる．設計情報と自動設計要求指令がエキスパートシステムに送られ，推論エンジンを働かせる．設計情報により既存知識ベースから設計知識を検索する．もし設計の要求に対応する設計ルールを見つけた場合は，図面情報をもつ設計ルールをデータベースから読み出し，この部分の設計を自動的に詳細化する．図6の上の図はこれを示す．

もし設計の要求に対応する設計結果が見つからない場合には，「知識なし」表示が出力される．この場合には人が新しく詳細化してPADを作り，必要なら設計知識ベースに蓄える．設計知識が沢山収集すればするほど，自動設計率は高くなる．

初回の設計は全体を見ながら段階的に進めるので，部分的に自動抽出し，自動設計を併用する．また既存システムの変更，拡張にあたっては，ある指定点から作業できることが必要である．また将来は完全な自動設計を行わせたい．このような各種の運転制御を行わせるために始点，停止点，除外点，レベルを設定し，自動抽出，自動設計モードにおいてシステムをきめ細かに制御できる．

6.1 候補の表示，選択

自動設計した結果が不適当な場合がありえる．この時には，ＣＡＳＥツール機能により変更や追加が自由に行える．この修正された結果についても設計知識を自動抽出し，蓄積する．

一つの親に応じて複数の設計ルールが対応する場合がある．この場合には，複数の設計ルールから適当なものを選んで設計する．図6の上図の中央は複数の設計ルールがある場合の例で，シンボルのハッチにより複数候補の存在が表示され，知識ベースの第１候補が自動的に選択されて自動設計する．これが不適当，あるいは他の設計ルールを見たい場合には，設計者は親の設計情報と候補の表示要求指令をエキスパートシステムに送って，推論エンジンを働かせる．設計情報により既存知識ベースからいくつかの候補を検索し，検索された候補を図の下に示すように候補表示 Window に出力する．設計者はこの候補の中から一つを選択すると，設計図面の PAD 図上でこの部分の設計は自動的に入替えられる．自動設計の効率を高くするため，選択された候補番号は知識ベースに登録され，よく使われる候補を第一候補として知識ベースに登録する．

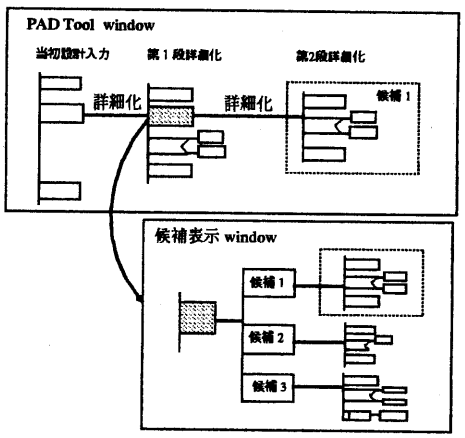

図 6: 自動設計と候補の表示，選択

7 まとめ

この論文は，まず，ソフトウェア設計を対象として，系統的に設計知識を獲得と再現するについて述べた．この方式は下記を前提とする．

- 対象エキスパートの作業区分である階層的設計工程を対象知識のモデルとする．
- 階層的工程による細分化を利用し，設計文書を用いて系統的に知識体系を獲得する．

次は知的CASEツールシステムについて述べた．このシステムは以下の効果を挙げられる．

- 設計図面を作成しながら，設計知識を自動的に獲得し，蓄積する．
- 蓄積した設計知識を再利用し，自動設計が可能になる．
- 汎用的で事前準備の負担が少ない．

現在のシステムは基本モデルで，高度化し使い易くするには，多くの課題がある．まだ，現在の作図より上位の各種の設計知識を研究し，ここに報告したオープンエンドな成長により順次高度化し自動化率を向上させることと考えている．

謝辞

本研究のCASEツールとエキスパートシステムのファイル連動については文部省の科学研究費基盤研究B-1．課題番号０７５５８０４３によるものである．御協力頂いた日立中部ソフトウェア株式会社の関係各位に感謝します．またエキスパートシステムについては日立製作所コンピュータ事業本部のご援助を頂いた．ここに記して謝意を表します．

参考文献

[1] Chen, H., Far, B. H. and Koono, Z. : Software Creation: Reuse of Design Knowledge of Switching Software, in Proc. International Conference on Communication Technology, China, pp. 63-66 (1994).

[2] 陳 慧，町田 経一，河野 善彌: ソフトウェアクリエーション: 設計における系統的なエキスパートシステム構築の方式の研究, 1995年度人工知能学会大会, pp. 431-434.

[3] Chen, H., Machida, K., Far, B. H. and Koono, Z. : Software Creation : A Systematic Construction Method of Expert Systems Used for Design, 3th World Congress on Expert System' 96, Korea, pp. 577-584 (1996).

[4] 陳 慧, Far, B. H. , 河野 善彌: ソフトウェアクリエーション: 知的CASEツールの方式について, 1996年度人工知能学会大会, pp. 273-276.

[5] 陳 慧, ベルーズ・H・ファー, 堤 永保, 河野 善彌: "ソフトウェアクリエーション: Intelligent CASEツールの試み，" 電子情報通信学会 信学技報, KBSE 96-26 ～ 32, pp. 31-36, 1997.

[6] 陳 慧 , Behrouz H. Far, 河野 善彌, "ソフトウェア自動設計における系統的なエキスパートシステムの構築, ー設計工程からの設計知識の獲得と再現ー", 人工知能学会誌 Vol 12, No 4, 1997年7月 掲載予定.

[7] Hancock, W. M. , Bayer, F. H. : Ch2. Learning curve, Handbook of Industrial Engineering, John Wiley & Sons (1982)

[8] 編著 原田実: 自動プログラミングハンドブック, オーム社 (1989).

[9] Koono, Z. , Baba, T. , Yabuuchi, T. : Software creation: A trial for switching software, Proc. of the 1992 Joint Conf. on Communnication Network, Switching and Satellite Communnication, pp. 261-265 (1992).

[10] Koono, Z. , Far, B. H. , Baba, T. , Yamasaki, A. , Ohmori, M. and Hatae, K. : Software Creation : Towards Automatic Software Design By Simulating Human Designers, The 5th International Conference on Software Engineering and Knowledge Engineering, pp. 327-331 (1993).

[11] 河野善弥, B. H. Far, 杉本 崇: 設計知識の系統的な獲得, 1994年度人工知能学会全国大会, pp. 443-446 (1994).

[12] Koono, Z. , Far, B. H. , Sugimoto, T. , Ohmori, M. and Chen, H. : A Systematic Approach for Design Knowledge Acquisition from Documents, 3rd Japanese Knowledge Acquisition for Knowledge-Based System Workshop JKAW '94, pp. 253-265.

[13] Koono, Z. , Chen, H. and Far, B. H. : Systematic Knowledge Acquisition for Expert Systems Used For Design, 3th World Congress on Expert System' 96, Korea, pp. 847-855.

[14] 河野善弥, "CASE と Intelligent CASE の方向付けについて", 電子情報通信学会 信学技報, KBSE 96-26 ～ 32,pp. 23-30, 1997.