

## 分散環境におけるソフトウェアの自動設計のアプローチ

陳 慧\*, 間明田 忠\*\*

(2003年12月24日受付, 2004年1月23日改訂)

### An Approach of Automatic Software Design in Distributed Environment

HUI CHEN\* and TADASHI MAMYOUDA\*\*

**Synopsis:** Integrated Intelligent CASE Tools have being developed, that feature automatic software design as well as automatic design knowledge acquisition. This integrated environment may be applicable from top-level system architecture design, data flow diagram design down to flow chart design and coding. The integrated environment is only used by stand-alone way so they are low on sharing the design knowledge. In order to improve software development efficiency and design quality, it is getting indispensable to share and reuse the design knowledge and design information among related organizations. This paper reports on a new approach of automatic software design in distributed environment. The construction method of the knowledge base is discussed and the experimental system consists of three layers Client, Server and Knowledge Base. By using CORBA and Java, the development costs of this system may be decreased substantially.

**Key words:** distributed environment, knowledge base, automatic design, CASE Tools

#### 1. は じ め に

ソフトウェアの膨張はとめどもなく続き、人手によるソフトウェアの開発では限界がある。そのため、ソフトウェア開発の自動化が求められている[6]。筆者等の方針は初めから完全な自動設計システムを目指すのではなく、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。これはCASEツールとエキスパートシステムの組み合わせにより、省力効果を高める事を可能にする方法である。最初開発された知的CASE (Intelligent Computer Aided Software Engineering: ICASE) システム[1]は、下流設計の構造化チャート (Problem Analysis Diagram, PAD [7]) を用いて設計図面から設計知識を自動的に獲得し、それらを知識ベースに蓄える。獲得した知識を再利用することによってプログラミングフェーズを自動化する。単純化した方式ではあるが、経験を重ねると自動化率は高まる。この成功に基づき、設計の最下流から上流に遡って、上流設計の機能構造図及びデータフロー図へ

---

\* 国士舘大学21世紀アジア学部, 情報科学センター

School of Asia 21, Center for Information Science, Kokushikan University

\*\* 富士通サポート&サービス㈱

Fujitsu Support and Service INC.

拡大し、統合的な知的 CASE ツール[2]が開発された。これにより、上流設計からコーディングまで適用できる標準化された設計プロセスを自動化することができる。これらの知的システムは商用描画ツールとそれぞれのエキスパートシステムから構成される。しかし、統合知的 CASE ツールはスタンドアロン動作が基本のため、限られた人、場所でしか使うことが出来ない。ソフトウェアの開発規模は、増大の一途を辿っており、数メガステップというシステムも珍しくなくなってきた。従って、1プロジェクトで数百人のというソフトウェア開発プロジェクトが存在している。数百人のというソフトウェア開発者を同じ場所のみで確保することはほとんど不可能であり、地理的な分散、分業化が進むのは必然的状况である。また、ソフトウェア開発効率と設計品質の向上を図るには、関連する組織間での設計技術情報の幅広い共有化と相互活用が必須となってきた[3][11]。

一方、インターネット技術の普及がネットワークを通して、物理的に離れた場所での作業や情報の共有を可能にしてくれた。つまり、ソフトウェア開発に必要な設計知識、ノウハウなどの設計情報を組織の枠を超えて相互に活用することが可能になった。本研究の目的は統合知的 CASE ツールの自動設計技術を発展させ、分散環境での開発に適用する方式を確立し試作することである。分散環境に発展するアプローチとして、分散環境での問題点を分析し、解決の為に分散環境の調査を行う。自動設計用に適した分散方式の設定を行い、開発には既製の部品技術を利用し最大限開発の手間を減らし標準化された技術を用いる。

この論文は、2章で分散環境における新しい課題を検討し、3章で分散化階層的な設計知識モデルの方法を説明する。分散環境のソフトウェア自動設計システムのアーキテクチャや機能について4章で述べる。5章は今後の課題とまとめである。

## 2. 分散環境の自動設計システムの検討

上述のように分散環境での開発は様々な利点があるが、分散化に伴って単独動作では発生しなかったさまざまな問題が発生する。例えば、操作性の良いプログラムの開発を行うと開発量が膨大化する。通信が十分に高速でない場合、また、ネットワーク化を前提にした十分な設計がされていない場合パフォーマンスが低下する。分散化して作るために設計の同期化が困難になり、設計情報のインテグリティを保つ強力な制御が必要になる。また、セキュリティの確保が新しい課題となる。分散環境におけるソフトウェア自動設計の統合知的 CASE ツールについて以下の問題を検討する。

### 1. 実用的な速度

自動設計とは、人の手間を省き短時間での開発を可能にすることを目標とするため、実用的な速度ではないと無意味なものになる。本研究対象の自動化方式は、知識ベースへの頻繁なア

アクセスが基本であり、通信がボトルネックになる可能性がある。自動設計時間は人手による設計の場合より短くならねばならない。統合知的 CASE ツールでの自動設計方式は一つのシンボル展開するたびに知識ベースへアクセスする。知識ベースへの頻繁なアクセスが発生し、知識ベースに蓄積したシンボル数に比例して設計時間が増加している。分散環境におけるデータベースと通信とかわる時間について以下いくつかの方法を使って調査を行った[5]。

- (1) ftp の場合はデータサイズに関わらずオーバーヘッドが発生する。その原因はユーザ認証などの通信前の処理から生じるものである。
- (2) 通信とのまったく関係が無い Oracle データベースの場合は、データサイズの増加にともない通信時間が増加している。
- (3) 比較的オーバーヘッドの少ない http を使う場合は、これもデータサイズの増加にともない通信時間が増加している。

本システムの自動設計にかかる時間＝往復通信時間＋データ検索時間＋自動処理時間となり、これは人手による設計よりも小さくならなくてはならない。ネットワークを介した場合にかかる自動設計はスタンドアロンによる知的 CASE ツールの最低数倍の時間がかかることがわかった。つまり通信のオーバーヘッドが大きいため頻繁なやりとりを行う方式は不向きである。

## 2. 知識ベースの共有方式

今まで同じ設計図面、設計知識（ルール）のみをデータベース化したが、分散環境への移行に伴う設計情報を共有するため、様々な設計知識や設計情報、また、組織、ユーザや製品情報なども必要である。データの増大に伴いさまざまな設計資源が入り混じり、再利用が困難になる。

## 3. 分散型のシステム構成

現在の知的 CASE ツールはスタンドアロンで動作する。CASE ツールとネットワークをつなぐためのインターフェースが必要になる。

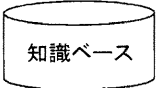
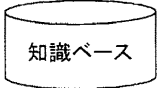

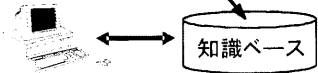
## 4. セキュリティ

設計情報共有のためのセキュリティの問題である。誰が作った設計資源でも登録でき、他人の作った設計資源を書き換えることができた場合、再利用する設計資源の品質が低下する。そのためアクセス制限が必要になる。

# 3. 知識ベースのモデル

知的 CASE ツールにおける自動設計システムは設計を経験するにつれて、新たな階層展開が新しい知識になる。また、自動設計では、階層展開の度に対応する知識が必要になるため、自動設計システムと知識ベースの間は、情報が頻繁に往来する。知識ベースの共有方式は以下

表 1 知識ベースの方式比較表

	方式 1	方式 2
サーバ		
ネットワーク		
クライアント		
長所	一元管理が容易	処理が高速
短所	通信時間がネック	一元管理が困難

の表 1 に示しているようにいくつか方式が考えられる。

(1) サーバのみにデータベースを置く

サーバ側のデータベースにすべての設計情報をおき、ネットワークを通して、必要な設計情報をサーバ側のデータベースに問い合わせ、設計情報を取得する。この方式はデータの一元管理が容易であるが、毎回データのやり取り時には通信を行うため、通信速度が問題になる。

(2) サーバ側、クライアント側にデータベースを置く

サーバ側のデータベースからクライアントのデータベースに必要な設計情報のみをダウンロードして使用する。この場合はデータの整合性などの管理が煩雑になり、一元管理が困難である。必要な設計情報のみをクライアントのデータベースにダウンロードし、クライアントのデータベースを使っているため、速度に関しての問題が解決できる。すなわち、自動設計する時に、頻繁なデータのやり取りを快速にすることができる。

本システムの自動化方式では知識ベースへの頻繁なアクセスが発生するため、応答速度が最も重要である。上述方式(2)は適当である。従って、本システムは方式(2)の知識ベース共用分割方式をとる。データの整合性の問題はサーバデータベースの読み取り、書き込みにアクセス制限をかけることによりデータの信頼性を向上させる。

本システムの設計情報は階層展開になるので、知識関係は順次におおきくなり、ツリー状の知識ベースになる。設計が進むにつれて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いく。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。本システムの知識ベースはドメイン毎に分かれており、各ドメインは用語やブ

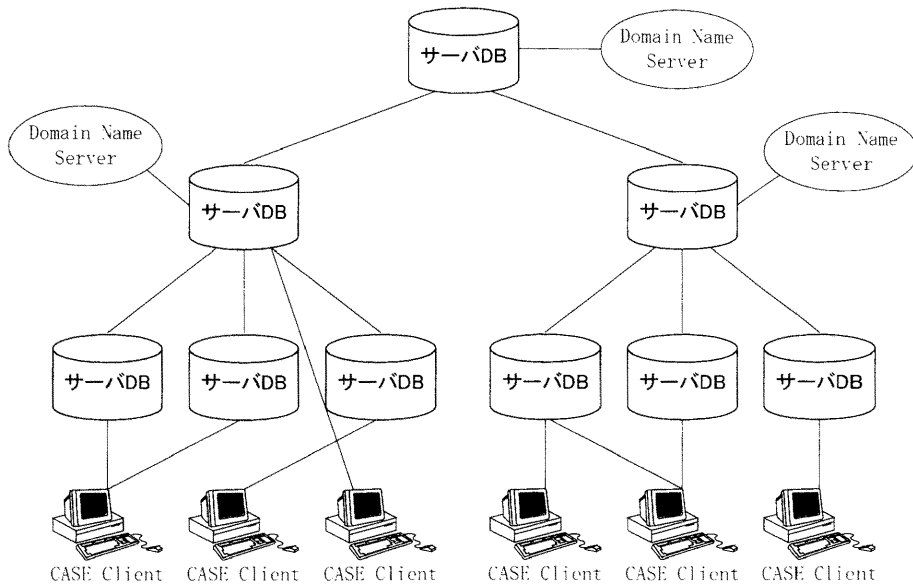


図1 複数データベースサーバによる知識ベースのモデル

プログラム条件の揃う範囲に選ぶ。ソフトウェアの構成方法によるが、設計対象の階層的なプログラム構成に階層的な知識ベースの構成が対応することになる。

本システムは図1に示すように複数データベースサーバによる階層型のデータベース方式をとる。この方式によりドメインの階層化ができ、それぞれの参照には Domain Name Server を用いて管理する。このデータベースシステムはネットワークのデータベースホスト名、データベース名とどのようなドメイン知識をもつものを変換する機能を持ち、ユーザの要求に応じて実際のデータベース名を検索する。ドメインごとにデータベースを持っているので管理がしやすく、目的の知識を検索しやすい。複数のデータベースに平行してアクセスできるので検索時間が短縮できる。この方法は再利用できるデータをローカルに持っているため、データベースの負荷を分散し、ネットワークの負荷が低い、他のクライアントの情報を全て参照できるという利点を持つ。サーバ側のデータベースの負荷が減少する。この方法は Domain Name Server とドメインモデルを考える必要がある。

#### 4. システムのアーキテクチャ

幅広く各種のプログラムに対応できる汎用的なシステムを目指して分散的な統合的 CASE ツールシステムの開発の試みをするため、実現上の具体的方針は下記のとおりとした。

- 手軽に使用できて便利であり、設計効率向上に役立つ。

- 各種の CASE ツールと容易に統合できる。
- 各機能を相互に独立化し、機能拡張・変更・保守が容易に行なえる。
- 実行速度を向上させる。
- ユーザに対する親和性を重視する。

#### 4.1 システムの概要

オブジェクトは、もともと分解可能なものであるため、拡張性の高い3階層クライアント/サーバシステムを提供することは非常に運用性、汎用性などの点で利点が多い。本システムのアーキテクチャは、CASE ツールの動作する PC クライアント、サーバ、データベースの3層 C/S 方式[4] [8]をとる。この C/S 構成が図2に示されている。

中間層のサーバオブジェクトはクライアントと相互作用するものである。また、複数のデータベースからその永続的な状態を取出すことが出来る。サーバオブジェクトは異なるデータベースを統合することが出来る。クライアントはドメインの実体に対応するオブジェクトと相互作用すればよく、第3階層に存在する煩雑な機能（各種データベースなど）と関わる必要が無い。サーバオブジェクトはこれらのものを隠すことが出来る。サーバオブジェクトはローカルなデータベースから取出したデータへのその後のサクセス速度を向上させるために、データをキャッシュすることも出来るし、オブジェクトの状態を保持できるため、第3層のデータベースを直接更新することも出来る。クライアントは通常の ORB (Object Request Broker)

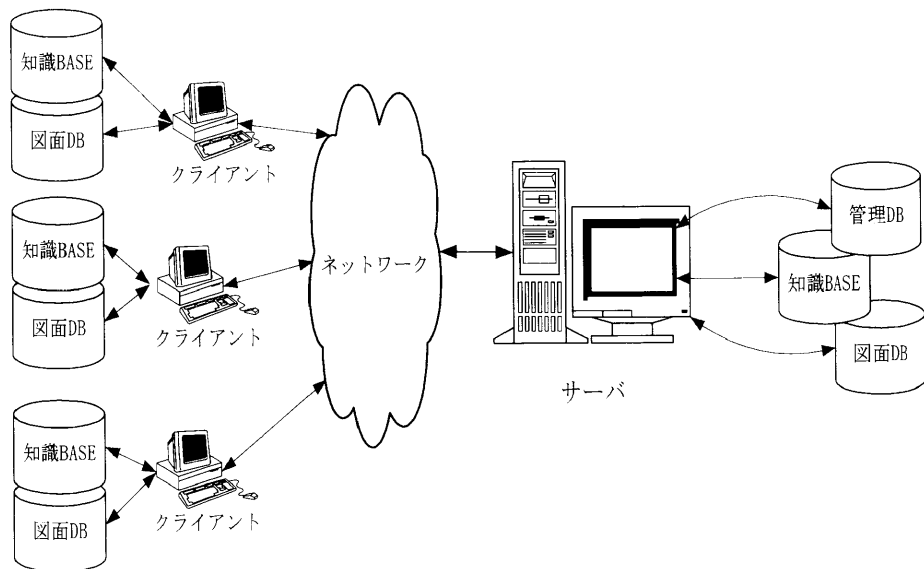


図2 システムのアーキテクチャ

[8]を介して中間層のサーバオブジェクトと対話することになる。

サーバモデルによって、ワークステーションのレベルから、汎用機レベルまでの拡張性を持ち合わせ、将来の業務拡張までを見通した上最適な機種を選択を可能にする。また、サブシステムごとにサーバを分散させる事も可能なため、現場の運用形態に合った柔軟なシステム導入を可能にする。役割などによりサーバを振り分けることにより業務を分散させる事が可能であり、又サーバとクライアント間でも処理の分担が明確にされているため、サーバ及びクライアントコンピュータの性能を十分に発揮させることができる[9] [10]。

## 4.2 システムの機能

本システムでは図3に示すように設計情報管理、設計情報登録、設計情報検索の3つの機能から構成される。設計者から設計情報登録要求がある場合、クライアント側の画面からユーザ認証など行い通信を確立し、管理用の情報を付加したものと共にサーバ側のデータベースに登録する。クライアントからの設計情報検索要求がきた場合、要求に適する情報をクライアントにダウンロードし、それをを用いて自動設計を行う。管理情報はサーバで管理するという方式をとる。データベース検索/登録以外をクライアントで処理するためサーバの負荷が軽減できる。

1. 設計情報管理は図面管理、ユーザ管理、組織管理、商品管理、バージョン管理、設計ルール管理に分けられる。プロジェクト、設計情報ごとにセキュリティ、バージョン情報など

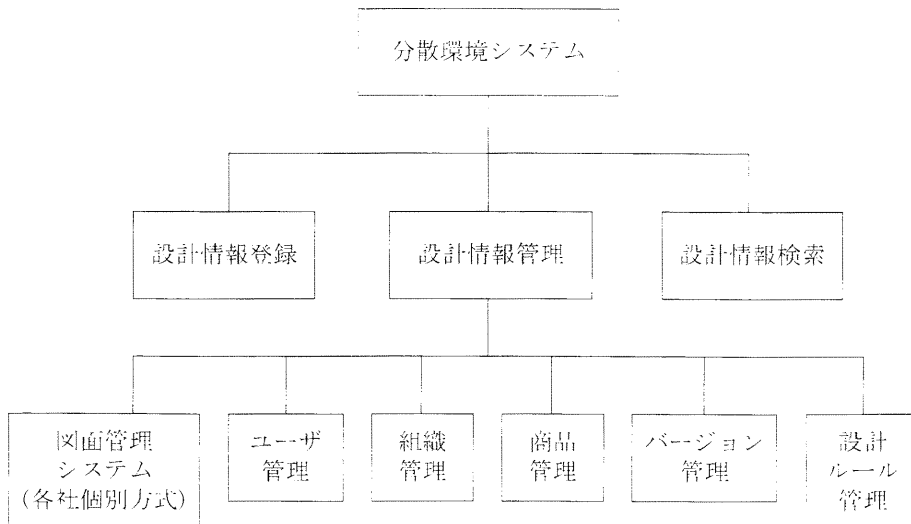


図3 システムの機能概略

を持つ。図面管理システムは各 CASE ツールの方式を使用する。設計情報登録、検索のとき、セキュリティを保証するため、ユーザやグループなどの認証を行う。プロジェクト管理部は、あるユーザあるいはグループはそのプロジェクトに参加しているかどうかを管理する。

2. 設計情報登録では、ユーザが登録したい情報を入力し、サーバに登録する。設計情報登録時には、ユーザにアクセス制限をかけ、他人の設計情報を書き換えられないようにする。また、再利用のため上記機能により情報を付加する。

3. 設計情報検索では、ユーザの入力に基づいた情報を検索し、検索結果を表示する。ユーザはその情報を必要と判断した場合、ダウンロードし設計に役立てることが出来る。設計情報検索時には登録時に付加された情報によって類似機能を持つ設計資源を絞りこむことにより自動設計のための設計情報を再利用することができる。

4. CASE ツール個々の情報はそれぞれをひとつのオブジェクトとみなし管理する。

## 5. ま と め

本研究では情報の共有、知識共存のため知的 CASE ツールの自動設計技術を発達させ、分散環境での開発に適応する方式を確立することを目的に研究を行ってきた。

分散環境では、単独で発生しなかったさまざまな問題が発生した。第1に速度低下の問題があったが、これに対しては知識ベースの共用方式に知識分割方式を採用した。第2に知識ベースの共有の問題があった。これに対してはサーバからクライアントにダウンロード時、クライアントからの設計情報のサーバへの登録にアクセス制限を設けた。

分散環境に適応するための方式として本システムでは CASE ツールの動作する PC クライアント、サーバ、データベースの3層 C/S 方式を採用した。設計者から設計情報登録要求がある場合、クライアント側の画面を経由してユーザ認証など行い通信を確立する。次に管理用の情報を付加しサーバ側のデータベースに登録する。クライアントからの設計情報検索要求がされるとき、要求に応じた必要な情報をクライアントにダウンロードし、クライアントはこれを用いて自動設計を行う。これにより、最も頻繁な自動設計システムと設計知識ベース間のアクセスとクライアントの中に限定させる。管理情報はサーバで管理する方式をとる。ドメイン毎で設計知識ベース検索/登録以外をクライアントで処理するため、サーバの負荷は軽減する。C/S 方式を採用したためクライアント側で複雑なデータベース処理を行わずにすむ。負荷を分散し汎用性の高いシステム構成といえるだろう。今後の課題としてはプロトタイプの完成及びシステムの評価を行うことである。



謝 辞

本研究は有益な御助言をいただいた元埼玉大学教授河野善彌教授およびカルガリー大学助教授 B. H. Far 博士に深く感謝します。

参 考 文 献

- [ 1 ] Chen, H., Tsutsumi, N., Takano, H., and Koono, Z., “Software Creation: An Intelligent CASE Tool Featuring Automatic Design for Structured Programming”, The Journal of Institute of Electronics, Information and Communication Engineers, Vol. E81-D, No. 12, pp. 1439-1449, 1998.
- [ 2 ] Chen, H., Takano, H., Abolhassani, H., Koono, Z., “Software Creation: an Integrated Environment for Automatic Software”, Proceedings of the Fifth World Conference on Design and Process, Texas, U.S.A, (in conference CD-ROM), 2000.
- [ 3 ] Kawaguchi, S. et al., “Development and Applications of a High-Quality Software Design Support System”, Fifth European Conference on Software Quality, Session 3A Paper No. 1, 1996.
- [ 4 ] 倉貫義人, “ビジュアルプログラミング環境 Pad Tree for C の構築”, 命館大学情報学科, 1997.
- [ 5 ] 間明田忠, 陳慧, B. H. Far, 河野善彌, “ソフトウェア自動設計のための分散環境の試作”, 2000 年度情報処理学会全国大会, 2000.
- [ 6 ] Martin, J., “Information Engineering”, Prentice-Hall, 1989.
- [ 7 ] 二村良彦, “プログラム技法: PAD による構造化プログラミング”, オーム社, 1984.
- [ 8 ] ロバート・オーファリ, ダンハーキー, “Java & CORBA C/S プログラミング第2版上, 下”, 日経 BP 社, 1999.
- [ 9 ] トマス・J・モーブレイ/ラファエル・C・マルポー, 大谷真監訳, “CORBA Design Pattern—アーキテクチャからプログラミングまで—”, IDG コミュニケーションズ, 1998.
- [10] 成田雅彦, 保西義孝, 勝亦章善, 島村政義, 島村真己子, 杉能康明, “CORBA と Java 分散オブジェクト技術”, ソフト・リサーチ・センター, 1997.
- [11] 山本憲一, 熊澤克久, 石川博之, 川口 進, 白石辰巳, “イントラネットを活用した広域分散協調ソフトウェア開発環境の構築”, 電子情報通信学会信学技報 SSE98-219, IN98-191, pp. 45-50, 1999.