

52B-04 ソフトウェア自動設計のための分散環境の試作

間明田 忠, 陳 慧, 河野 善彌, B. H. Far

埼玉大学工学部情報システム工学科

mamyou@cit.ics.saitama-u.ac.jp

1. はじめに

ソフトウェアの需要は定常的に増えており、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。我々のプロジェクトでは、中流～下流設計を自動化する知的 CASE ツールが開発されている。その自動化方式^[1]は、(1) 設計者が CASE ツールを利用して、設計図面(ドキュメント)を作る、(2) 設計図面から設計知識を自動的に獲得し、知識ベースに蓄積する、(3) 蓄積した知識を再利用して自動設計を行う、ことである。現在の知的 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、場所しかその利点を享受することが出来ない。そ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が本研究の目的である。

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、解決の為の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おこない、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。

2. 問題点と解決方法

2.1 問題点

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。

(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より小さくならねばならない。これを実測した結果、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。

(2) 知識ベースの共有方式

知識ベースの共有方式は以下のようないくつかの方式が考えられる。

a. サーバのみにデータベース(DB)

サーバ側 DB にすべての設計情報をおき、毎回通信を行う場合、データの一元管理が容易なため、管理は簡単だが、通信速度が問題になる。

b. サーバ側、クライアント側に DB

サーバ DB からクライアント DB に必要な情報のみダウンロードして使用する場合、データの整合性等、管理が煩雑になるが、速度に関しての問題が解決できる。

2.2 解決方法

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 b は適当であるが、データの整合性の問題はサーバ DB の読み取り、書き込み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。従って、b の知識ベース共用分割方式をとる。

3. システム構成

本システムは、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 方式をとる。本システムの構成を図 1 に示す。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 DB 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

An Experimental Distributed Environment
for Automatic Software Design

Tadashi Mamyouda

Dept. of Information and Computer Sciences,
Saitama University

255 Shimo-okubo, Urawa, 338-8570, Japan

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。

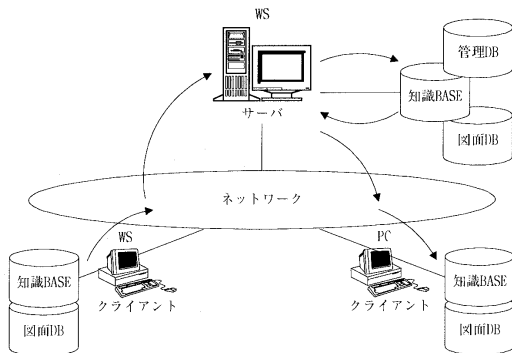


図1 全体構成

3.1 機能構成

本システムでは図2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つに成立つ。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。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。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できる。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。図面管理システムは各CASE ツールの方式を流用する。

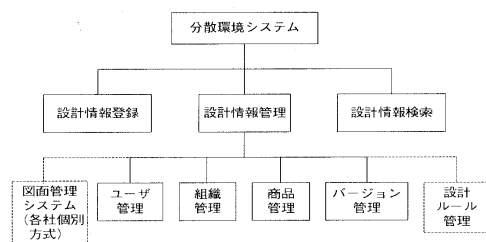


図2 機能構成

3.2 実現手段

本システムでは、データベースには RDB であり OODB の機能をもつ、ドキュメントが豊

富であるなどの理由から PostgreSQL^[2]を使用する。通信には、TCP/IP 上で通信でき、拡張性に優れるなどの理由で CORBA^[3]を使用する。実際のプロトコルには IIOP を使う。開発言語には開発の容易さ、CORBA との親和性などから Java を用いる。図3にシステムのサンプルを示す。左のウィンドウはユーザ認証、グループ認証、認証用データベースの選択など機能を提供する。右のウィンドウは検索条件を受け付けサーバに要求を出す。真中のウィンドウは選択された CASE ツールである。

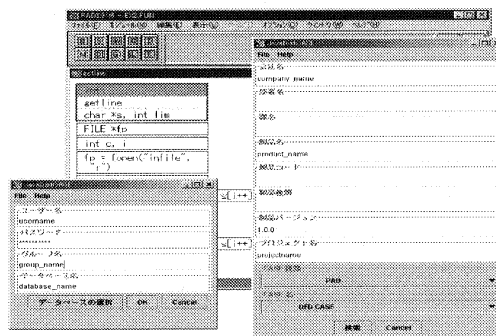


図3 システムサンプル

4 まとめ

知識ベースの共用方式について実用速度を重視し共用分割方式をとった。また、設計情報の検索、登録方式を決め、機能構成、データ構成等決定した。今後の課題としてはプロトタイプ of the 完成及びシステムの評価を行うことである。

5 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] 石井達夫, "PC UNIX ユーザのための PostgreSQL 完全攻略ガイド", 技術評論社.
- [3] ロバート・オーファリ, ダン・ハーキー / 著, 並河英二, 水野貴之, 池浦規之 / 役, "Java & CORBA C/S プログラミング第2版", 日経BP社.