

統合ソフトウェア開発環境“ソフトウェアCAD”

4E-11

—分析・チェック機能と計算機言語生成機能—

吉岡明彦, 藪田和夫, 村田芳和, 宮崎比呂志, 松原泰昭, 渡辺伸一, 村上憲稔
(富士通株式会社 ソフトウェア開発企画本部)

1. 概要

ソフトウェアCADで作成・編集されたドキュメント情報は、全てドキュメントデータベースに格納される。ソフトウェアCADでは、このドキュメントデータベースへの問い合わせによって、ソフトウェア開発作業に必要な分析・チェックの作業を行う。また、ドキュメントデータベースに格納された設計情報に対して、「生成ルール」を定義することにより、ソースプログラム・JCL・各種定義体などの計算機言語を生成する機能を実現している。これにより、特定の技法や工程に依存せず、ソフトウェア開発作業全般にわたって、設計情報の「分析・チェック」と計算機言語への「変換」という人間作業を支援する。

2. 「分析・チェック」機能

設計ドキュメントに記述された情報は、全てドキュメントデータベースに格納されている。このドキュメントデータベースは、リレーショナルデータベースであり、簡単な問い合わせ言語によって、ソフトウェア開発過程において設計者と管理者が逐次必要とする情報を検索できる。

開発過程において設計者や管理者が必要とする情報には例えば以下の様なものがある。

・設計情報

「全てのデータ項目を列挙したい」
「データ項目 SHAIN-CODE を参照するプログラムは？」
「プログラムXXが出力するデータを入力するプログラムは？」

・管理情報

「今ある全てのドキュメントの一覧がほしい」
「プログラムXXの詳細仕様を作成した担当者は？」
「担当者高橋が一番最近に作成したドキュメントより新しいドキュメントは？」

これらの問い合わせは、リレーショナルデータベースの検索言語によって簡単に実現することができる。

また、情報の検索という手段によって、単に一覧情報の取得→分析という作業のみならず、開発ドキュメント内容の一貫性 (consistency) や完備性 (completeness) のチェック作業も行うことができる点が重要である。

・一貫性のチェックの例

「同名のデータに異なる属性が定義されていないか」

・完備性のチェックの例

「入力も出力も定義されていないプログラムはないか」

「目次にあつて、まだ作成されていないドキュメントはないか」

3. 計算機言語生成機能

ソフトウェアの設計情報が、リレーショナル型の構造で格納されていることにより、明確で、均一なモデルの上で、設計情報の加工を行うことができるというメリットがある。プロトタイプシステムでは、ドキュメントデータベースの中に格納されている設計情報と、それを元に作り出すべき計算機言語の文法を定義した「生成ルール」をもとに、計算機言語を生成する機能を実現している。この機能は、「生成ルール」を入力するとその定義に従い、ドキュメントデータベースの内容を参照し、テキストの形式に整形する変換プログラムを出力するというかたちで実現されている (図1)。

表記法は、予めソフトウェアCADのテンプレート・フォーム登録機能によって登録しておき、その表記法によって記述されたドキュメントから、どのような言語を生成したいのかが、この「生成ルール」を使って記述される。これにより、図・表をベースにしたドキュメントから、言語を自動生成するツールを構築することができる。

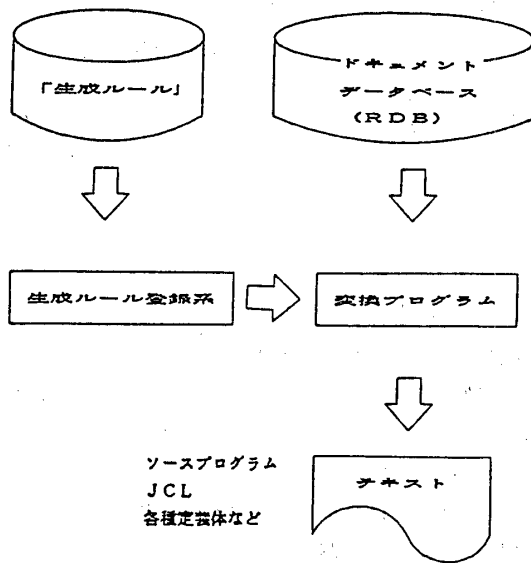


図1. 計算機言語生成機能の構成図

現在は、プロトタイプシステムを使って、

- ・プロセスフロー図 ⇒ ジョブ制御言語
- ・状態遷移図 ⇒ C, COBOL, FORTRAN 等のプログラミング言語
- ・バックマン図 ⇒ DBスキーマ定義言語

などの計算機言語の生成を行っている。

こうした汎用のツールで、図・表表現からテキスト表現への変換を行うツールそのものを自動生成することによって、次のようなメリットがもたらされる。

●計算機処理目的以外のテキストでも生成できる

図・表表現からテキスト表現への変換ツールが容易に構築できることから、計算機言語だけでなく、人間が読み、分析するための一覧表・クロスリファレンス等を生成させることもでき、開発過程における人間作業を支援することができる。

●生成された「自動生成ツール」の保守性

従来の「自動生成ツール」には、生成対象言語が変わった場合には、その対象言語の仕様を追隨して変更または作り直さなければならないという宿命を負っている。ソフトウェアCAD上では、「自動生成ツール」そのもの

【参考文献】

- [1] 藪田他 「ソフトウェア開発のライフサイクル全体を支援するツールのフレームワーク」
情報処理学会第32回全国大会
- [2] 村上他 「統合ソフトウェア開発環境“ソフトウェアCAD”～ソフトウェア開発の自動化・機械化を目指して～」
情報処理学会第33回全国大会
- [3] 藪田他 「統合ソフトウェア開発環境“ソフトウェアCAD”～ドキュメント編集機能～」
情報処理学会第33回全国大会

のを定義する「生成ルール」の一部を変更することで、生成対象言語の変化に対応できる。

4. 生成ルールとそのしくみ

「生成ルール」は、文脈自由文法を拡張したものであり、対象言語の文法定義に、関係演算の操作等を持ち込んだ。これは通常の意味での文脈自由文法の定義。

$$G = \{V_n, V_t, P, s\}$$

V_n : 非終端記号の集合

V_t : 終端記号の集合

P : 生成規則の集合

s : スタートシンボル ($s \in V_n$)

に対して、

- (1) P の各規則には、関係演算操作を付加する
- (2) V_t の要素としてリレーショナルデータベースの任意のドメインと、必要に応じて、計算機言語の中に現れる定型情報を割り当てる。

という拡張を行ったものである。この文法、及びそれを処理する生成ルール登録系は、ソフトウェアCADのリレーショナルデータベースの形式に依存する部分は少ない。従って、その意味でこの計算機言語生成機能は、リレーショナルデータベースを入力とする独立なツールとしても位置づけられる。

5. おわりに

ソフトウェアCADの分析・チェック機能では、現在、リレーショナルデータベースの検索言語をそのまま利用しているが、将来は、図形インタフェースを用いて、より簡単に、種々の検索が表現できるマン=マシン=インタフェースを実現したいと考えている。

また、計算機言語生成機能については、

- (1) 生成ルール登録系により出力される変換プログラムの性能の向上
- (2) 生成ルール記述の容易化
- (3) 本機能の適用実績を積み上げ「生成ルール」を蓄積などを行い、実用化へ向けての研究・開発を行っていく。