

# 通信ソフトウェアCASE環境ICAROS: 状態図ジェネレータ

6H-3

米津恵美子 青山幹雄  
富士通株式会社

## 1. まえがき

リアルタイムシステムの設計を支援する分散CASE環境であるICAROS(Integrated Computer-Aided environment for cOperative Software development)を開発している[1]。

本稿ではICAROSの一システムである状態図ジェネレータについて述べる。既存のソースからソフトウェア開発の設計書(状態遷移図)を生成するリバースエンジニアリング[2]を支援している。

## 2. 開発の背景

### 2.1 ニーズ

#### (1)ソースと設計書の一致

ソフトウェアの開発保守において、ソースプログラムの変更が設計書に反映しないことがある。そのギャップを埋めることにより、ソフトウェアの保守及び設計情報の活用を効率的に行える。

#### (2)既存資産の流用とICAROSへの移行

膨大なソフトウェア資産を新しい開発環境であるICAROSに移行するための橋渡しが必要である。既存のソースプログラムに対する状態図を再度、新規に作成するのは効率的でない。従って、既存のソースプログラムをICAROSが支援する状態図に自動的に変換する必要がある。

#### (3)試験の高度化

通信ソフトウェアの品質に対する要求は厳しく、試験効率の向上が重要な課題となっている。このため従来よりソースプログラムレベルで行われている試験を設計書(状態図)レベルで行うことができれば、システムの挙動がわかりやすくなり試験効率は向上する。

### 2.2 シーズ

#### (1)一人一台のワークステーション(WS)

ソフトウェア開発では、一人一台のWS環境を導入している。このWSの処理能力とグラフィカル・ユーザ・インタフェース(GUI)を活用すれば視覚に訴える設計情報を提供できる。

#### (2)ツールの活用

UNIXが提供しているツールであるYACC(Yet Another Compiler Compiler)やLEX(LEXical analyzer)を活用する事により構文解析部を効率的に開発することができる。

## 3. アプローチ

ソースプログラムから設計書(状態図)を自動生成する状態図ジェネレータを次のようなアプローチに基づき開発した。

### 3.1 解決のアプローチ

#### (1)設計情報の構造

状態図生成に必要な情報はICAROS共通のテキスト形式中間言語表現で定義している[3]。この中間言語表現を論理構成情報と図形表現情報に分離した。論理構成情報は状態遷移論理を定義する情報であり、高水準言語で記述されたソースプログラムから抽出できる。図形表現情報は、状態図中の状態やシンボルの配置などを規定し、生成時にヒューリスティックな判断が必要である。この情報は、設計者が簡単に指定できるようなGUIを提供することにより、対話的に設計者が設定することとした。

#### (2)階層的な論理構成情報の構造

階層的ソフトウェア構造を表現するために、論理構成情報は、さらに、状態遷移レベルと各状態遷移内の詳細な処理(タスク)レベルとに階層化した。

#### (3)段階的な変換方法

中間言語表現を生成するために次のような段階的変換方法をとった。

①ソースプログラムから論理構成情報を生成する。

②論理構成情報と図形表現情報から中間言語表現を生成する。

#### (4)エディタとの連携

ジェネレータは、状態図エディタも支援しているICAROS共通の中間言語表現を生成する[3]。中間言語表現を画面に表示するには状態図エディタを用いる。

### 3.2 開発のアプローチ

状態図ジェネレータの開発については、まずプロトタイプを作成し機能の実現性を明確にした後、本格版の開発を行うという手法をとった。

4. 実現

4.1 システム構成

状態図ジェネレータのシステム構成を図-1に示す。状態図ジェネレータはWS上で動作する。ホスト上に格納されているソースプログラムをWSに転送して変換を行う。

4.2 機能概要

状態図ジェネレータは以下に示す機能により構成される。

(1)ジェネレータ本体

ソースプログラムを解析し論理構成情報を生成する。

(2)フォーマット

ウィンドウを表示し、設計者に状態図のレイアウトなどの図形表現情報を入力させ、それを中間言語ビルダに渡す。

(3)中間言語ビルダ

上述の(1)と(2)によって生成された図形表現情報と論理構成情報から、状態図の中間言語表現を生成する。

4.3 処理概要

処理概要を図-2に示す。

(1)LEXで作成した字句解析ルーチンはソースプログラム中の予約語やパラメータ等のトークンを抽出する。

(2)ソースプログラムの論理構造の解析をYACCで行う。YACCの内部で(1)の字句解析ルーチン呼び出ししながら構文解析を行い状態遷移を表現する論理構成情報を生成する。

(3)X-ウィンドウ上でユーザにレイアウトなどの図形表現情報の入力を促す。標準的レイアウトはテンプレートとして用意している。

(4)(3)で生成された図形表現情報と(2)で生成された論理構成情報をもとに、状態図における様々な図形と座標情報からなる物理的情報—状態図エディタがサポートする中間言語表現—を生成する。

5. 評価

プロトタイプを開発し、実際のソースプログラムから状態遷移レベルでの状態図を生成し、状態図エディタとの連携が可能となった。大規模交換ソフトウェアの開発保守へ適用を予定している。

6. 今後の課題

開発済のプロトタイプを発展させて本格版状態図ジェネレータを開発する予定である。これにより状態遷移レベルからタスクや各端末までの階層化された状態遷移図が自動生成可能となる。

7. まとめ

ソースプログラムから状態遷移図などの図形設計情報の生成は、ソフトウェア開発保守において重要な役割を果たすと考えている。今後開発する、本格版状態図ジェネレータは操作性が良く、設計者の作業の効率を促進するものにしていきたい。

8. 参考文献

[1] 青山 功 "通信ソフトウェアCASE環境ICAROS: 構想" 情報処理学会第41回全国大会, 1990年9月.  
 [2] E. Chikofsky et al., "Reverse Engineering and Design Recovery", IEEE Software, 1990年1月, pp. 13-17.  
 [3] 蛭原 功 "通信ソフトウェアCASE環境ICAROS: 状態図エディタ" 情報処理学会第41回全国大会, 1990年9月.

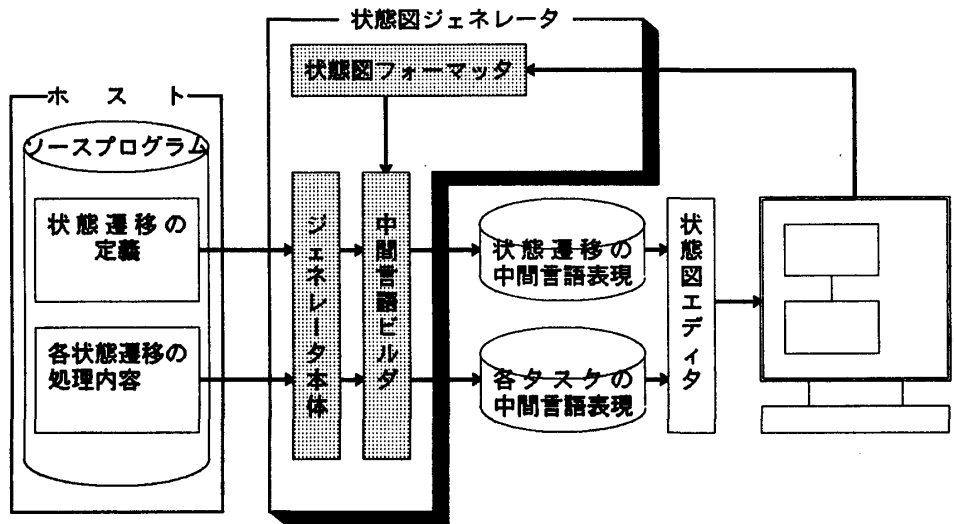


図-1 状態図ジェネレータのシステム構成

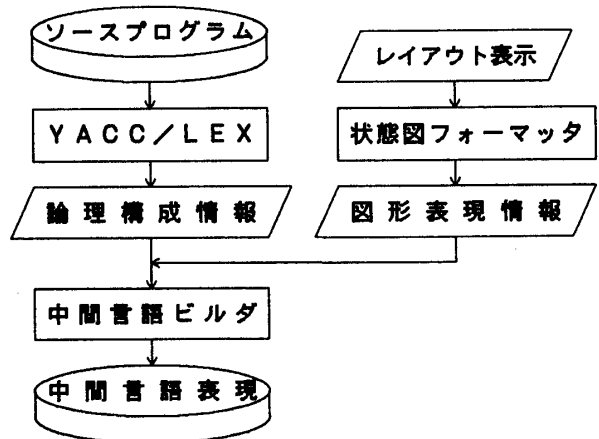


図-2 状態図ジェネレータの処理方式