

## 機器系組み込み型ソフトウェア開発支援システム

5S-7

岩淵洋一、菊地一成、渡辺浩康、浅野俊昭  
 キヤノン(株) 情報システム研究所

## 1. はじめに

機器系組み込み型のソフト開発には、並列性や、リアルタイム性といった制御ソフト固有の問題があり、従来のデータ処理中心のソフト開発技法を直接導入することは困難である。このため、我々は本領域の仕様記述に適した形式的図式仕様記述法であるSRグラフを考案し、それに基づくプログラムの骨格(スケルトン)の自動生成、動的検証を行う開発環境を提案してきた[1]。今回、この構想に基づく開発支援システム、SRGシステム、を計算機上に実現したのでその概要、特に本システムにおけるソフト開発について報告する。

## 2. SRGシステムの構成

システムの構成を図1に示す。システムは、

- SRグラフにより仕様を定義するための入力系(グラフィックエディタ)
- 対象機器の動作を網羅的に検証する検証系
- 同期関係に基づきプログラムの骨格(C言語)を生成する生成系

の三つの部分からなる。

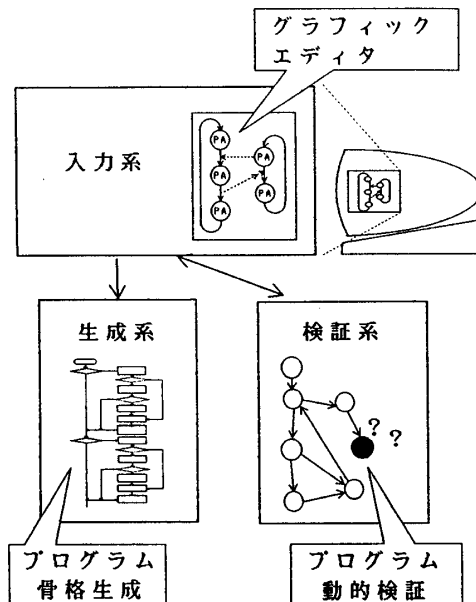


図1 システムの構成

また、今回構築したシステムの動作環境は以下の通りである。

マシン：HP-9000/300 シリーズ EWS  
 システム：HP-UX, X-Window with mouse  
 言語：C

## 3. SRGシステムにおけるソフト開発

従来の制御ソフトの開発では、本来直接には関係のない、(1) 実機内の各ユニットが協調動作するための同期関係に関する記述と、(2) 各ユニット固有の制御手続きの詳細に関する記述とを分けて設計することが難しく、このことが開発や保守を困難なものとしていた。

本システムではSRグラフによりこれらの記述を分離して定義できるため、設計を

PHASE 1： 同期関係の定義

PHASE 2： 制御手続きの記述

の2段階に分けて行うことができる。また、特にPHASE 1では、本領域に固有の問題でもあり、開発を困難とする主たる要因ともなっている同期関係の定義について計算機による集中的支援を行っており、生産性の向上を期待することができる。

## 3.1 PHASE 1

制御ソフトの開発を難しくしている同期部については、以下に示す支援を行う。

## 3.1.1 マウスによるグラフィカルな仕様定義

図2はグラフィックエディタを用いた仕様定義の様子を示している。

ここで用いるSRグラフは設計対象機器との対応関係がよく、プロセス間の同期関係を明示的に定義可能な形式的図式仕様記述法であり、グラフィックエディタ中では以下の四つのシンボルで表わされる。

① PA ( Primitive Action )

○は、システム内の各ユニットがある時点ですべき詳細手続きを示しており、PAと呼ばれる。PAはその手続きの実行中であることを他のユニットに知らせるためのフラグ(後述)を出す機能を有する。また、実際の手続き内容であるPAの中身についてはPHASE 2で記述する。

② 遷移

↓は、PAの実行順序のつながりを示すもので遷移と呼ばれる。↓で接続された一連のPA群は各ユニットの制御シーケンスを構成する。遷移はそれが発生したことを他のユニットに知らせるためのトリガ(後述)を出す機能を有する。

③ フラグ

—□□—は、制御シーケンス間の同期関係を記述するためのものでフラグと呼ばれる。フラグは、あるPA実行中に、他のユニット内の特定の遷移に対し遷移の許可を与えるのに用いる。

④ トリガ

—◇◇—は、制御シーケンス間の同期関係を記述するためのものでトリガと呼ばれる。トリガは、ある遷移の発生と同期して他ユニット内の特定の遷移を強制的に引き起こすのに用いる。

今回のシステムでは、各ユニットの一連の動作を示すシーケンス毎にウィンドウを割り当てており、対象機器との対応関係が視覚的に分かり易くなっている。また、SRグラフの編集作業はすべてポップアップメニューからマウスでコマンドを選択することで行うことができる。

3.1.2 同期関係の計算機による網羅的検証

図3は、図2で定義した仕様の検証例である。ここではSRグラフで定義した、個々のユニット毎の動きと、それらの間の同期関係を基にシステムとしての振舞いを予測する。検証結果は、対象機器のシステムとしての初期状態から始まる状態遷移として示され、到達し得る全ての状態が木構造図を用いて網羅的に表示される。プログラマはこの検証結果を分析し、同期関係に関する仕様が意図したように定義されているかどうかを、実際のコーディングを待つ事なく事前に評価・修正することができる。

3.1.3 ソースコードの自動生成

SRグラフで定義した仕様から、プログラムの骨格(C言語)が自動生成される。

3.2 PHASE 2

PHASE 2では、PHASE 1で得られたプログラムの骨格に各ユニット内の個々の手続きを埋め込むことでプログラムを完成させる。既にPHASE 1で、各ユニット内のPAの実行順序及びそれらの間の同期関係の定義を終えており、また、制御手続きの詳細であるPAの中身はシステム全体の動きとは独立しているため、ここでの記述については、

- ① 他のユニットとの同期関係を考えることなくシーケンシャルに記述
- ② ユニット固有の制御要素として再利用、部品化といったことが可能となる。

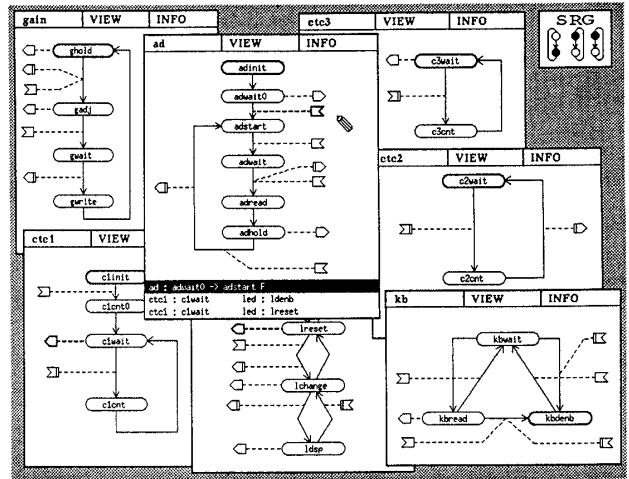


図2 仕様定義画面

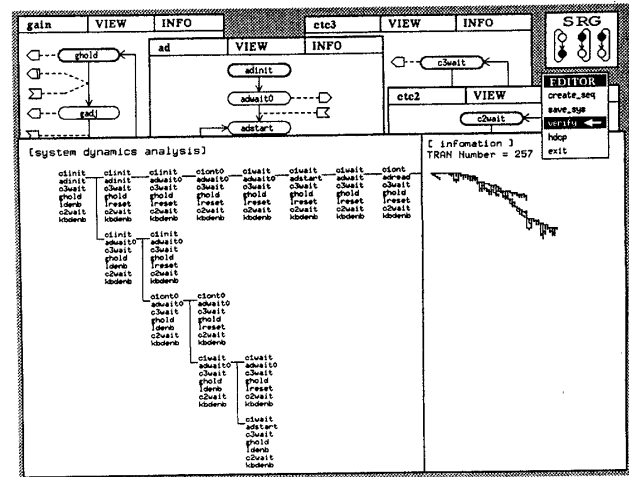


図3 仕様検証画面

ちなみに、図2で示した仕様を基に作成したプログラムでは、PHASE 1で自動生成したコード量と、その後PHASE 2で書き込んだコード量の比は約3 : 1であった。

4. おわりに

形式的図式仕様記述法SRグラフに基づく機器組込みソフトウェア開発支援システムを構築した。本システムでは設計を(1)同期関係の定義、(2)制御手続きの記述の2段階に分け、計算機による支援を行うことで、本領域におけるソフトウェアの生産性の向上を目指している。

今後は、ユーザインターフェイス部の充実、検証機能の強化を計り、実用的なシステムに拡張していく予定である。

[参考文献]

[1] 菊地 他：組込み型ソフトウェア開発環境、情報処理学会第37回全国大会(3M-10)