

Ada* 並列プログラムのための 4K-8 グラフィック表示を用いた実行再演ツール

白木原 敏雄, 程 京徳, 荒木 啓二郎
九州大学

1. はじめに

プログラミング言語 Ada¹⁾は, 実時間処理プログラミングに適用できる並列言語である. Adaでは, タスクと呼ばれる並列処理単位が, 様々な形で通信(ランデブー)を行う. また, タスクは実行中に動的に割り付けられ, それらの間には, 特殊な依存関係が存在する.

我々は, Adaプログラムの実行中における, タスクのこれらの振舞いをモニタするために, 事象駆動型実行モニタ EDEN²⁾を開発した. EDENの開発にあたっては以下のような基本的方針を設定した.

- Adaプログラムの並列事象にのみ注目する.
- 支援環境の要求仕様³⁾にしたがって, Adaのソースコードレベルで実現する.
- プログラムの動作に対するモニタリングの影響を小さくするために, 実行中に起こる事象の間に成り立つ半順序を保存する.

EDENの機能を大別すると, 次のようになる.

- ある瞬間のタスクの振舞いを報告する.
- プログラム実行中のタスクの振舞い(タスキング振舞い)の履歴を報告, 保存する.
- タスク間の通信デッドロックを検出する.

EDENはこれらの機能により, 様々な情報をユーザに提供することができる. Adaプログラムのテスト・デバッグを行う際には, EDENが提供する情報に基づいて, プログラムのタスキング振舞いを解析する事が必要である. しかし, 履歴情報はテキストの形で提供され, 更にその量が莫大なため, 読みにくく, また理解しにくい.

今回, 実現したAdaプログラム実行再演ツールは, EDENが提供する履歴情報を利用して, プログラム実行中のタスクの振舞いをグラフィック表示を用いて視覚的に理解しやすく再現するツールである. 本稿では, このツールの概要を紹介し, その有用性を示す.

2. 実行再演ツール

Adaプログラム実行再演ツールは, EDENが提供する履歴を利用して, 実行の様子を視覚的に再現するものである. Data General社の ECLIPSE MV/10000のAda処理系上のEDENのもとで実際に実行させたAdaプログラムの実行の履歴をPC-9801に渡して, PC-9801上でそのAdaプログラムの実行の様子を再演させる. PC上の実行再演ツールはC言語で実現している.

EDENからPC-9801に渡される履歴はすべてのタスクで起こった事象が, 時間的な順序で並んでいるものである. 履歴ファイルの例を図1に示す.

*Task_name	*Time(sec)	*Event or State	*Entry or Subprogram
.....			
* T2	* 1.1992187500	* EXECUTION_START	
* T2	* 1.1992187500	* WORKING_FOR_INTERNAL_AFFAIRS	
* T1	* 1.1992187500	* ACTIVATION_START	
* T1	* 1.1992187500	* ACTIVATING	
* T1	* 1.1992187500	* ACTIVATION_COMPLETION	
* T1	* 1.1992187500	* EXECUTION_WAITING	
* T1	* 1.1992187500	* EXECUTION_START	
* T1	* 1.1992187500	* WORKING_FOR_INTERNAL_AFFAIRS	
* T3	* 1.3007812500	* EXECUTION_START	
* T3	* 1.3007812500	* WORKING_FOR_INTERNAL_AFFAIRS	
* T1	* 1.3007812500	* ENTRY_CALL	* T2 . E2
* T1	* 1.3007812500	* ENTRY_CALLING	* T2 . E2
* MAIN_TASK	* 1.3007812500	* BLOCK_EXECUTION_START	* CEC
* MAIN_TASK	* 1.3007812500	* WORKING_FOR_INTERNAL_AFFAIRS	
* MAIN_TASK	* 1.3007812500	* BLOCK_EXECUTION_COMPLETION	* CEC
* MAIN_TASK	* 1.3007812500	* BLOCK_COMPLETED	* CEC
* T3	* 1.4003906250	* ACCEPTANCE	* T3 . E3
* T3	* 1.4003906250	* ACCEPTING	* T3 . E3
* T2	* 1.4003906250	* ENTRY_CALL	* T3 . E3
* T2	* 1.4003906250	* ENTRY_CALLING	* T3 . E3
* T3	* 1.5996093750	* RENDEZVOUS_START	* T3 . E3
* T3	* 1.5996093750	* WORKING_FOR_INTERNAL_AFFAIRS	
* T3	* 1.5996093750	* ENTRY_CALL	* T1 . E1
* T3	* 1.5996093750	* ENTRY_CALLING	* T1 . E1

図1 時間順序による履歴ファイルの一部

タスクの状態⁴⁾の変化は, 次のように図示する.

- 1) タスクが確立され実行を開始するまで, または,

* Adaはアメリカ合衆国政府(Ada Joint Program Office)の登録商標である.

実行が終了して消滅するまでの状態の変化

タスクを表す平行四辺形をタスクが確立されると表示し、消滅すると消す。タスクの状態の変化は平行四辺形の色の变化で表す。

2) タスク間通信の状態の変化

あるタスクが別のタスクのエントリを呼び出した場合、相手のタスクを表す平行四辺形との間に線を引く。他方、エントリへの呼び出しを受け付ける場合、自分の平行四辺形の中に、受付窓口となるエントリを表す小さい平行四辺形を描く。通信待ち、通信中の区別は、色で行う。

3) 手続き呼び出しにより、手続きに制御が移った場合の状態の変化

タスクを表す平行四辺形を、長方形に書き換える。状態の変化は、色の变化で表す。

実行再演ツールの実行中の様子を図2に示す(但し、色は示していない)。

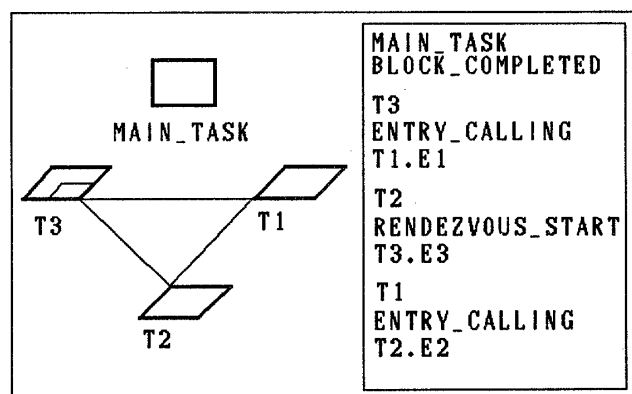


図2 実行再演中の表示例

左側の図式的な部分は、タスクの状態を上述の方法に従って表示したものである。右側のウィンドウはタスクの状態をテキストで表している。

実行再演ツールの現在の主な機能は次の通りである。

- 1) タスクの状態を(1ないし10ステップ単位で)時間的に前後に変化させる。
- 2) 特定の事象を指定することにより、実行再演の中断点(ブレイクポイント)を設定できる。
- 3) ブレイクポイントかプログラムの開始または終了の状態まで時間的に前後にスキップする。

3. 考察

プログラムの構成やデータの流れは、図式的に表現

すると理解しやすい。⁵⁾ 実行再演ツールでは、プログラム実行の履歴によりタスク間の通信を視覚的に表しているの、実行中のタスクの振舞いが設計時の構想と一致しているかを確認する際に便利である。⁶⁾ 実行再演ツールで使用したタスク等の図形表現は、文献5を参考にした。

本ツールの機能をより充実させて、Adaプログラムのテスト・デバッグに更に有効な情報を、理解しやすい形で表示させることが今後の課題である。例えば、

1) 特定のタスクの集合を指定することにより、その集合のみを観察できるようにする。

2) Adaのタスクでは、同時に複数のエントリで呼び出しを待つことが出来るが(選択待機)、現在それに関する情報は提供していない。そこで、選択待機に関する情報を提供する。

のようなことが考えられる。

また、実行再演ツールでは現在のところ、タスク間の依存関係を表現していない。Adaでは、タスク間の依存関係によりデッドロックが起きる可能性があることなどから、依存関係の情報は重要な意味を持っている。そこで、タスキング振舞いの中で起こる事象の間に成り立つ半順序関係を表すツールを実現し、依存関係をも分かりやすい形で提供することを計画している。今後、EDENの利用に関する研究をさらに進めて行くことにより、並列プログラムのテスト・デバッグの方法論の確立を目指す。

参考文献

- 1) 情報処理振興事業協会編: bit別冊 最新Ada基準文法書, 共立出版, 1984年8月
- 2) Cheng, J., Araki, K., and Ushijima, K.: Event-Driven Execution monitor for Ada Tasking Programs, Proc. COMPSAC87, pp381-388, 1987.
- 3) United State Department of Defence: "STONEMAN", Requirements for Ada Programming Support Environment, Feb. 1980.
- 4) Cheng, J., Ushijima, K.: Modeling the Ada Tasking Using Extended Petri Nets, the Memoirs of the Faculty of Engineering, Kyushu University, Vol. 48, No. 1, pp17-30, Mar. 1988.
- 5) Buhr, R. J. A.: System Design with Ada, Prentice-Hall, INC., 1984.
- 6) Nichols, K. M. and Edmark, J. T.: Modeling Multicomputer Systems with PARET, Computer, Vol. 21, No. 5, May 1988.