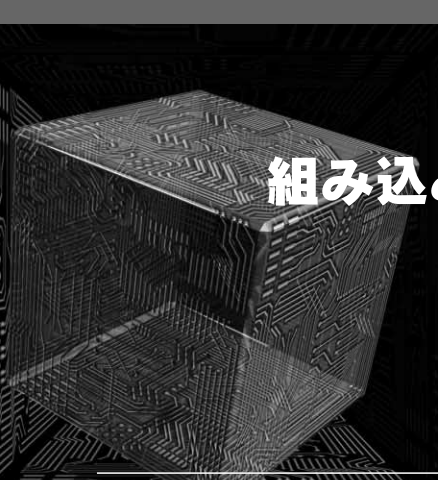

組み込みプログラミング・ノウハウ入門（第32回）

LTSAをタスクのテストに使う

――タスク動作モデルの作成（3）

藤倉 俊幸

1 マルチタスク環境でのデバッグの難しさ

今回は，タスク・モデルのテストへの利用方法について説明する．マルチタスク環境では，微妙なタイミングによってタスクの動作が変わり，同一の入力を与えても同じ結果を得られない場合がある．この結果，バグの再現性がなくなるのでテストが難しいと言われている．

微妙なタイミングの最大の原因は，RTOSが利用するインターバル・クロックである．普通は，10ms程度が利用されるが，10msというのはタスクの実行速度と比較するとかなり長い時間である．たとえば，本連載第30回（2006年3月号）の例（**図1**と**図2**）を再度使用すると，タイマ割り込みがいつ入るかによって動作パターンが変わる（A1，A2，B1，B2）ことがわかる．つまり，同一の入力を与えても，この場合は動作パターンは変わるということだ．そのため，マルチタスク・システムのテストは難しいと言われる．最近はテスト関係の本がいくつも出版されているが，並行実行による競合条件下でのテスト法や結果の評価法については，あまり詳しく扱われていない．

タイミングによって動作パターンは変わるけれども，タスク・モデルを作っておけばどのように変わるかは事前にわかる．どのように変わるかさえわかってしまえば，対策も立てられる．「微妙なタイミング」などという，言い訳がましいことをいう必要もなくなる．

RTOSテンプレートを入れたタスク・モデルの実行パターンは，実はそれほど多くない．したがって，問題の起こりそうなところをある程度絞り込むことが可能になる．つまり，計算の順序に依存する部分を特定して，そこがどのような順序で実行されるかもわかるので，その順序に依存しないプログラムにするのである．たいていは，共有変数の更新順序や，データ鮮度の問題になる．これらの実装の方法は興味のあるテーマだが，今回はタスク・モデルの続きとして，タスク・モデルをテストに利用する方法について説明する．

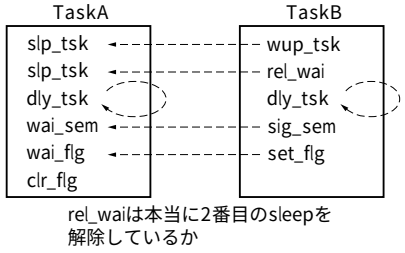

図1
dly_tskの例

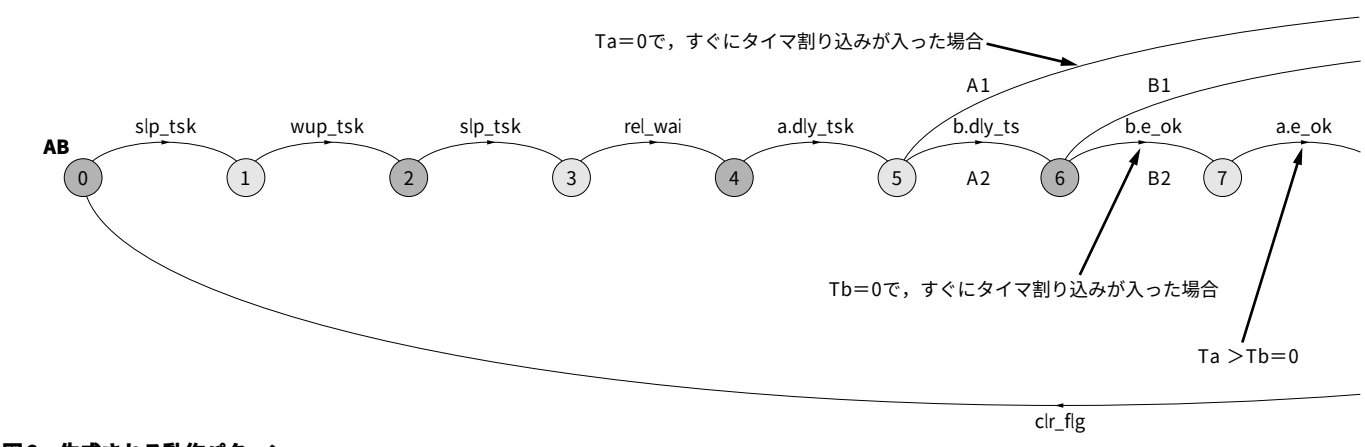

図2　生成される動作パターン

KEYWORD ――ソフトウェア・テスト，LTSA，モデル検査ツール，V字型開発モデル

2 タスク・モデルをテストに利用する

● LTSA の使える範囲

LTSA を利用して，動作モデルの作り方とタスク・モデルの作り方について説明してきた．その結果，検証済みのタスク仕様を作れるようになった．次のステップはタスク実装になる．実装は，ソース・コードのコーディング，コンパイル，デバッグの繰り返しになるので，LTSA のようなモデル検査ツールは使えなくなる．SPIN などのもう少し詳細なモデルを記述できるツールであれば，もう少しソース・コードに近いところまでがんばることはできるが，少なくとも LTSA は使えなくなると思ってよい．つまり，安全性検査やデッドロック，プログレス検査をここまでやってきたのに，最後の実装のところで見放されてしまうことになる．それでは，片手落ちというものである．しかし，また後で復活する．

LTSA は，ローカル・プロセス内に閉じた変数しか使えない．無理にくふうして変数を使うようなモデルを作ると状態数が爆発してしまう．プログラムの仕様が詳細になり，プログラム全体を記述しようとすれば，LTSA に限らずほとんどすべてのモデル検査ツールで状態数爆発が起こる．状態数爆発は突然やってくるので，そろそろ危ないと思ったら作業をむだにしないために，それ以上の詳細化は行わないほうが良い．この引き加減が重要である．目安としては，現状の PC 環境や LTSA のバージョンでは状態数が 2^{20} あたりから危険領域に入る．

その先は，問題を分割するなどのくふうが必要になる．具体的には，モデルをコンポーネント化して，内部動作と外部動作に分けて，コンポーネント内のモデル検査と全体のモデル検査に分割する．全体のモデル検査の際には，内部動作は考慮しなくて良いので状態数を減らせるようになる．このようにして状態数を 2^{20} に抑えて設計していくのである．つまり，アーキテクチャの設計と検証ツールとして使用するということである．システム全体を見たければアーキテクチャ・レベルで使用する．コンポーネント内であればもう少し詳細化できるが，ソース・コードの検証には限界がある．しかし，宇宙船が大気圏に再突入する際にいったん，地球との交信が途絶えるが，また交信が再開するように，実装が終了してテストやデバッグの段階になると再び利用できるようになる．この利用の仕方は，形式手法とテストの融合という新しい展開である．

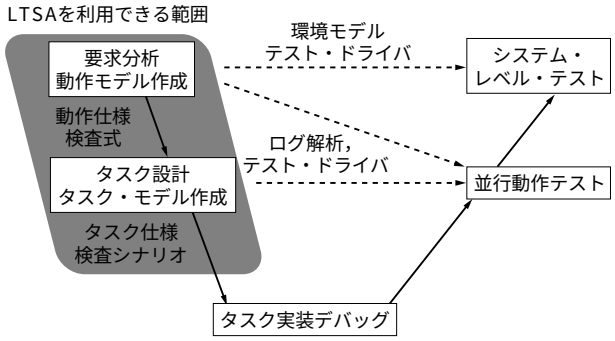

図3　LTSA 利用領域

● 実装後に LTSA を利用する：テストとデバッグ

図3 に LTSA を利用する領域を V 型開発モデルにマップして示した．最近は，V 字の内側にテスト計画とかテストを入れた W 型開発モデルもあるらしいが，これから説明するテスト法は，作り込みの段階で作成したモデルをテスト目的に応じて利用する方法なので W 型といったほうが，新しいことは何でも知っているんだぜ，というような感じが出て良いかもしれない．何とか型開発プロセスということばもやたらと増えてきているが，たとえば Y 字型，この記事の中でも後で，m 型，逆さ v 型，繰り返し X 型，二階建て m 型などを発明してしまった．個々の開発現場の実情に合わせると，このほかにも千差万別の