新連載

組み込みソフトへの数理的アプローチ

第1回 時間仕様のあつかい ――ガス・バーナー問題の例

藤倉 俊幸

前号までの「組み込みプログラミング・ノウハウ入門」では，LTSAを使って組み込みシステムの仕様の中に存在する，処理順序に関与している部分を中心にあつかってきた．今回から連載タイトルを一新するとともに，順序だけでなく時間を直接あつかっていくことにする．

時間をモデルに入れると，LTSAではできなかったことがいろいろとできるようになる．たとえばLTSAでは，シングルクリック2回とダブルクリックを区別することができない．組み込みシステムでは時間のあつかいは避けて通れないので，LTSAだけでは不自由な部分があるのだ．もちろんLTSAでも，カウンタのような形で時間をモデルに取り込むことはできるが，ツールとしては直接サポートしていない．したがって，モデリング・テクニックとしての時間のあつかいの前に，仕様としての時間について考えておく必要がある．

1 仕様モデルに時間を入れる

● UPPAALは時間をモデルに盛り込めるツール

仕様モデルに時間を取り込んだ例として，**図1**の照明器具の例をあげる．この照明器具は，ボタン一つで電源のOn/Offと明るさの調整を行うような仕様となっている．

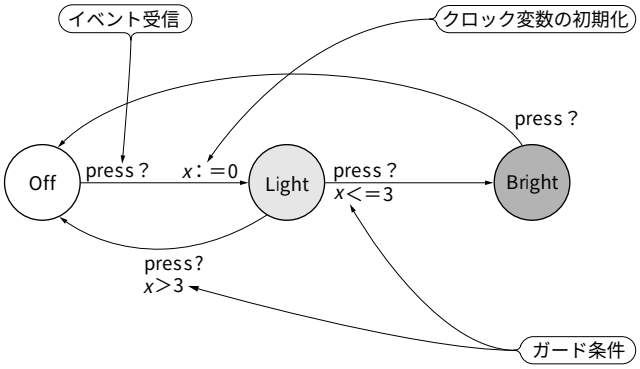

図1 時間オートマトンによるダブルクリック表現

この図はUPPAAL[注1]（ウパールと読む）時間オートマトン・ツールを利用している．オートマトンはステート・マシンのことなので，UPPAALは時間＋ステート・マシンをベースとしたツールである．

図1のステート・マシンは，OffとLight，Brightの三つの状態があるところまではLTSAと同じである．しかし，そのほかにLTSAにはなかった表記法が使用されている．

状態間の遷移はpress?というイベントで起こる．“？”はpressというイベントの受信を意味する．一般に，送信は“！”で，受信は“？”が使われる．**図1**に受信イベントしかないのは，このステート・マシンとは別のボタンの動きを表現したステート・マシンがあり，そのステート・マシンにpress!が定義されているという設定になっているからである．

● モデルに時間を盛り込んだ例

ここで，重要なのはxという変数である．この変数はクロック変数と呼ばれ，RTOSのインターバル・タイマのようなものである．クロック変数を使えば，経過時間を表現することができる．例では，Off状態からLight状態に遷移する際に，x:=0によってリセットされ，以後タイマ割り込みごとに+1される．Light状態からBright状態と，Light状態からOff状態への遷移のところに付いているx <=3 とx > 3は，それぞれの遷移に対するガード条件になっている．このガード条件があるので，このモデルは，「Light状態になってから，3時間単位以内に再度ボタンを押すと明るくなり，それ以上経ってからボタンを押すと消える」という動作仕様を表現している．

モデルに時間を入れるというのはこのようにすることである．時間の入れ方として一番わかりやすいのは，ここで示した時間オートマトンだ．

このようにUPPAALの利点は，LTSAと違ってイベントの送信と受信を区別するなど，LTSAより設計や実装に近いモデルを作成できることだ．だからといって詳しいモデルを作りす

注1：UPPAAL本体はhttp://www.uppaal.com/からダウンロードできる．また，UPPAALチュートリアルを翻訳して，以下のURLからダウンロードできるようになっている．
http://www.it.uu.se/research/group/darts/uppaal/documentation.shtml

ぎると，状態数が多くなって，たいていは状態爆発を起こして有用な結果を得る前に，使用を諦めることになる．そこで，設計や実装に近いところにいく前に，もっと上流の要求における時間のあつかい方から，時間の世界に入ろう．

2 サンプル要求仕様の記述例

● ガス・バーナーの例

要求の記述にはいろいろなレベルや形態がある．時間が重要な要素として含まれるガス・バーナーの点火に関する要求記述を例として取り上げる（**図2**）．

● ガス・バーナーの物語文的要求記述

ガスコンロに点火する際には，点火用の火花を飛ばす前に，まずガスを流さなければならない．点火が遅れたりすると，余計にガスを流してしまい，臭いし，危険である．

点火するまでのガスが流れる音は，ちょっと緊迫感がある．この緊迫感に耐え切れなくなるとつい手の力が抜けてしまう．そして心が落ち着いてからリトライする．しかし，何度もリトライを繰り返すと，ポリネシアのファイヤ・ダンスに匹敵する大きな炎がいきなり出てもっと怖い思いをする．そこで，ボタン一発で自動点火できるようにしたい．

● リスト・アップ記述とその分析

まずは要求仕様について考えてみよう．関連する要求項目を上記の文章から抽出すると，以下のようになる．

- 点火する前にガスを流さなければならないこと
- 点火が遅れないこと
- 余計にガスを流さないこと
- 臭くないこと
- 危険でないこと
- ガスの流れる音がしないこと
- いきなり大きな炎が出ないこと
- 怖くないこと

これらの中には，依存関係があったり，ソフトウェアと直接関係しないものがあるので，整理する必要がある．

たとえば，天然ガスなどはもともと無臭だが，ガス漏れに気づけるように，わざと腐臭をつけている．これは安全のためであり，「危険でないこと」を実現する手段になっている．したがって，「臭くないこと」という要求仕様から設計仕様を作り出すためには，放出するガスの濃度が臭いと感じるしきい値以下になるようにしなければならない．このことは，「余計にガスを流さない」という定量的ではない要求に対して，定量性を与えることになる．

「ガスの流れる音がしないこと」は，ガスに臭いを付けているのと同様に危険の有無に関する表現だが，ソフトウェアとはあまり関係ないのでパスする．大きな炎については，やはりガス濃度の問題であるが，点火を失敗してリトライする間隔と，前回の失敗で放出されたガスが拡散する時間との関係で決める必要がある．

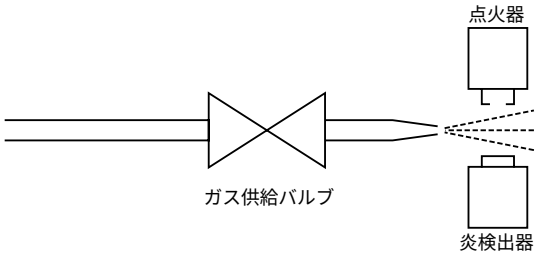

図2　ガス・バーナーの構造

ここまで考察すると，ようやく長さをもった「時間」の糸口にたどり着く．もとの文書には，長さをもった時間に関する記述は，表面には表れないという点に注意する必要がある．

● リスト・アップ記述

一般に，時間の仕様にたどり着くには，文章を読んで常識で判断したり，物理系の数理モデルを作ったりしなければならない．数理モデルは，時間を含んだ微分方程式などで表現されるので，それぞれのアプリケーション・ドメインの専門家の仕事である．ソフトウェア担当者は，専門家の出す時間仕様を設計仕様に落とすことが仕事になる．しかし，この境界はあいまいである．また，常識で判断する部分を明確に説明することは，時間が基本的な概念であるだけに，意外と難しい．

このガス・バーナー問題は，1990年頃発表された論文[(1)]が最初で，1993年のIEEEの論文[(2)]やDuration Calculus[(3)]という本に載っている．発表のたびに条件が少しずつ変わっているが，だいたいここで述べたことと同一である．

時間関連の要求項目は，

A：火がついていない状態で，ガス・バルブを開けている期間には上限を設定する必要がある

B：点火に失敗した場合は，ガス・バルブを閉じてから，リトライするまでの時間間隔には下限を設定する必要がある

の二つに集約される．参考文献(3)では，次のCのような定量的要求仕様になっている．

C：1分間以上の任意の期間におけるリーク時間の割合は，経過時間の1/20を越えてはならない

ここでの「リーク」は，ガスが流れているのに火がついていないガス・バーナーの状態のことである．AとBからCの記述の間には，ギャップがあるが，これは物理モデルを介在させた結果であり，ガスの専門家にしかわからない．

● 形式化記述

Cの文章にはよくわからないところがあるので，形式化したほうが良い．まず，「任意の期間におけるリーク時間の割合は，経過時間の1/20を越えてはならない」の部分を考える．任意の期間は，その期間のスタート時刻をt_s，終了時刻をt_eとすれば，$(t_e - t_s)$となる．リーク時間*LeakTime*を，この時間の20分の1以下にしなければならないということである．つまり，