形式手法による組み込みソフトウェア開発

第5回 ステート・マシンによる仕様の分割とデッドロックの排除

藤倉 俊幸

第3回（2012年8月号）から続いているカップラーメン・タイマの設計の続きを行います．前回は形式手法を使用することにより「タイマを起動する前に設定時間を30秒延長できてしまう」というバグを設計段階で発見することができました．引き続き，設計にバグがないかを見ていくことにしましょう．（筆者）

1．仕様をステート・マシンで表現し，分割する

表1に前回から扱っているカップラーメン・タイマの要求リストを再掲します．この中でSpec05とSpec06について解説します．

● Spec05（タイマが動いている間）とSpec06（設定時間が経過後）の仕様

Spec05とSpec06はどちらもLED4の点滅に関する仕様です．Spec01の時と違い，「タイマが動いている間は～」と「設定時間が経過すると～15秒間～」のように条件が付いています．「15秒間～」の所は振る舞いとして記述できます．「タイマが動いている間は～」と「設定時間が経過すると～」は条件としてTimeLeftと¬TimeLeftで記述できます[注1]．この二つは排他的なので，直接モデル化しても問題はない程度の複雑さです．

TimeLeftの定義からSpec05とSpec06の関係をモデル化すると**図1**のようになります．**図1**のSpec05状態とSpec06状態の中に，第3回で作成したSpec01型の動作モデルが入ります．この二つの状態は，UMLの状態図の階層化状態と同じです．状態の内部は後回しにして状態間の関係について考えてみます．

最初，すなわち電源を入れた直後は，add30は行われていないのでSpec06から始まるように**図1**では描いてあります．実際にSpec06が有効なのはexpireしてから15秒間と考えると，状態名としてSpec06というのは不適当かもしれませんが，ここではSpec06の中に含めることにします．つまり電源を入れた直後とexpireして15秒以上経過した状態をSpec06状態の内部状態とします．Spec06状態の時にadd30が実行されるとSpec05状態に遷移します．Spec06状

表1　カップラーメン・タイマの要求リスト

要求IDのReqは要求を，SpecはReqを満たすための仕様を示す．仕様を実装すれば要求を満たすことができる

要求ID	内　容	実　装	形式化
Req01	タイマを起動してから，設定した時間が経過したことを知らせる	—	
Req02	設定時間は30秒刻みで設定できる	—	
Req03	LED1で電源がONであること（プログラムが動いていること）を表示する	—	
Req04	LED4でタイマが動いていることを表示する	—	
Spec01	電源がONの間（プログラムが動いている間），LED1の点灯・消灯を1秒間隔で繰り返す	実装済み	済み（第3回）
Spec02	SW1がONになると設定時間を30秒にして，タイマを起動する	実装済み	
Spec04	タイマが動いている間にSW8がONになると，ONの間LED2を点灯させ，設定時間を30秒延長する	実装済み	
Spec05	タイマが動いている間は，10秒間隔で，LED4を2回点滅させる．LED4の2回点滅は，点灯・消灯を0.25秒間隔で2回繰り返すことで行う	実装済み	
Spec06	設定時間が経過すると，LED4を15秒間点滅させた後，消灯する．LED4の点滅は，点灯・消灯を0.25秒間隔で繰り返すことで行う	実装済み	
Req05（Spec07）	LED2でSW8が押されたことを表示する	—	済み（第2回）
Spec09	SW8で設定時間を30秒延長する	実装済み	

注1：¬は否定．