

(2) UML を拡張した出力記法

検出した不具合の出力には、入力記法と同様に UML シーケンス図を用いる。さらに、本記法ではシーケンス図を拡張し、オブジェクトの状態を同一図上に表記する。オブジェクト間通信に加え、オブジェクトの状態を明示することで、オブジェクトの状態が原因となる不具合を判別することが可能となるためである。拡張記法により不具合の判別が可能となる例を次節に示す。

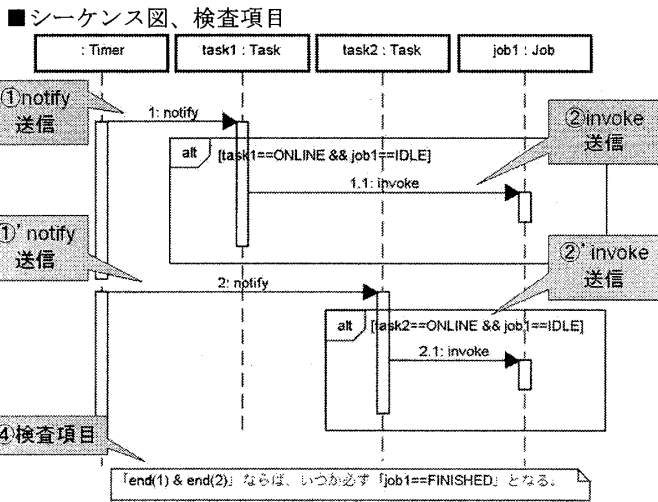
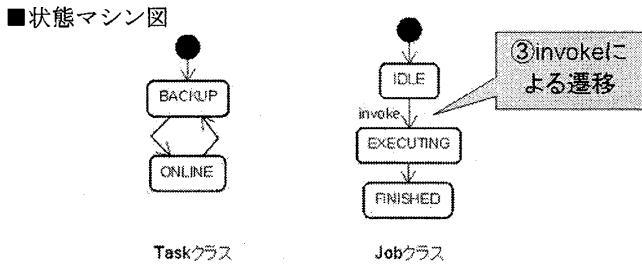


図2 入力モデルと検査項目

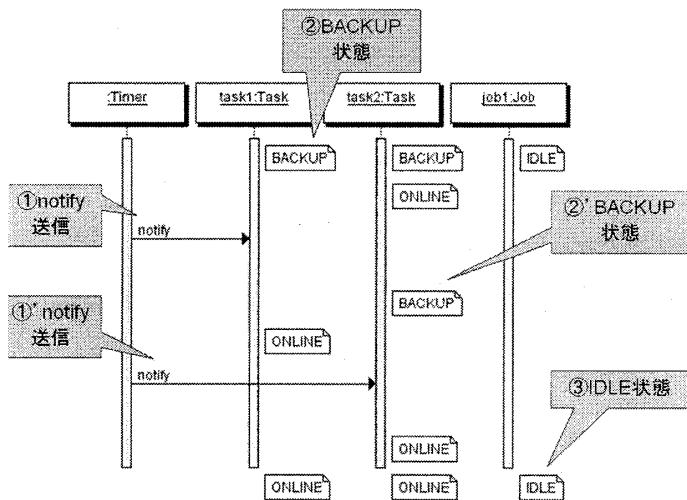


図3 検査結果

3. タイマーによるジョブ起動システムへの適用

3.1 入力記述

本ツールの効果を確認するために、開発済システムから動作仕様を抜粋・単純化して適用した。

検証対象は、「分散システムにおけるタイマーによるジョブ起動」の仕様である(図2)。システムの構成要素は Timer、task1、task2、job1 である。①Timer が task1 と task2 に notify を送信すると、②各 task は ONLINE 状態であれば job1 に invoke を送信する。③job1 は、invoke を受け付けると EXECUTING 状態に遷移する。④検査項目は「各 task2 が notify を受信し実行終了した後ならば、いつか必ず job1 は FINISHED になる」ことである。

なお、検査項目は UML シーケンス図のノートに記載されており、自然言語の定型文と括弧内の条件式を組み合わせで記述される。

3.2 検査結果

本ツールの出力した反例の一つを図3に示す。この出力は、①Timer が notify を送信した際、②task1 と task2 がいずれも BACKUP であるため、job1 に invoke を送信せず、③job1 の内部状態が IDLE のまま終了する不具合を示している。本記法によって、状態に起因する不具合の内容と原因を判別することが可能となったことが分かる。

3.3 ツールの評価

本ツールの記法によって、プログラムを用いずに入力モデル・検査項目を記述し、モデル検査を実行することが出来た。また、拡張記法により、オブジェクトの状態が原因となる不具合を判別することが出来た。これらの結果から、固有の記法を用いる従来ツールに比べ、導入の容易な機能を実現できたと考えられる。

4. おわりに

本稿では、UML を用いた記法によりオブジェクト間通信をモデル化し、モデル検査を行うツール DESIMOC について報告した。

現在、我々はいくつかのシステムを適用し本ツールの評価を進めている。現段階では、オルタナティブとアサーションを除く複合フラグメントが未定義であるなど、入出力の記法に改善すべき点が残る。多様なシステムのモデル化に取り組み、不足する記述要素を追加することが今後の課題となる。

参考文献

[1]中島 震: SPIN モデル検査, 近代科学社, 2008.
 [2]篠崎 孝一, 太田 弘, 早水 公二, 星野 光勇, 今村 哲典, 吉田 雅昭: モデル検査の実用化課題と支援ソフトウェアの開発, 第三回システム検証の科学技術シンポジウム, 2006.
 [3]OMG Unified Modeling Language™ (OMG UML), Superstructure, Version 2.2, OMG formal/2009-02-02, Object Management Group, 2009.
 [4]Gerard J. Holzmann, "The Model Checker SPIN", IEEE Trans. Soft. Engin., Vol.23, No.5, pp.279-295, 1997.