

組み込みソフト向け Hichart 開発環境における動作仕様検査

荻原 秀人[†] 後藤 隆彰[†] 夜久 竹夫^{**} 土田 賢省[‡]

[†]東洋大学大学院 ^{**}日本大学 [‡]東洋大学

1 はじめに

組み込みソフトウェアの大規模化・複雑化は益々進み、その開発およびテストの効率的支援が強く望まれており、様々な試みがなされている。その中でも、動作仕様を開発の上流工程で検証することが重要になってきている。近年では、上流工程での検証に関して、モデル検査の手法の適用が注目されている。一方、我々は長年にわたり、図式プログラム Hichart に基づく様々な処理系や開発支援ツールの研究開発を行ってきた[1, 2]。

これらの背景から、我々は Hichart 組み込みソフトウェア開発環境にモデル検査の手法を取り入れ、動作仕様の検査を支援する機能を追加し、システムの拡充を図った。本稿では、今回考案したモデル検査手法に基づく動作仕様の検査方法を中心に説明する。また、実装と実行例についても述べる。

2 システム概要

本研究では、レゴ社の「教育用レゴマインドストーム RCX」を対象とした既存の組み込みソフト向け Hichart 開発環境[3] を拡張する。そのシステム構成を図 1 に示す。

Hichart 開発環境では、プログラムを図式で表示し、図式上での編集操作が可能である。図 1 において、NtoH では、JavaCC を用いて、マインドストーム 用言語で記述された NQC ソースコードから Hichart 内部データへの変換処理を行う。また、HtoN では、同様に Hichart 内部データから NQC ソースコードへの変換処理を行う。

生成された NQC ソースコードは、マインドストーム用の統合開発環境である BricxCC によりコンパイルされ、RCX へ転送される。

3 SPIN を用いた動作仕様の検査

本研究では、Hichart 図式に基づくソフトウェア開発環境において、プログラムを実機で実行する前に、モデル検査ツール SPIN[4] を用いて、与えられ

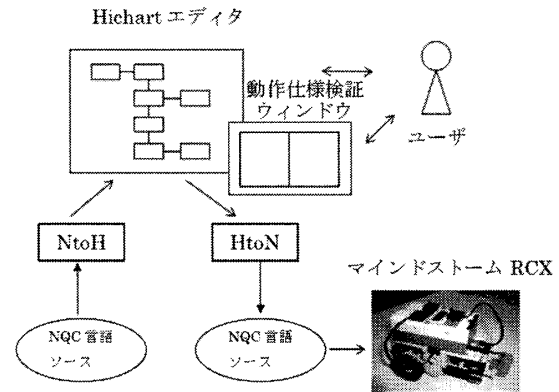


図 1 システム構成

た動作仕様を満たしているかを論理的に検査する方法を提案する。

我々は、既存のシステムに、次の 2 つの機能を追加した。1 つは、エディタで作成された、あるいは NQC ソースコードから自動変換された Hichart 図に対して、与えられた条件の動作仕様を満たすかを検査するための Promela 記述に自動変換する機能である。もう 1 つは、与えられた仕様を満たさない場合には、SPIN において、その該当箇所と推測される場所に対して、その箇所を Hichart グラフィカルエディタ上で反映する機能である。

具体例として、「前進と左折を行い、タッチセンサーが検知したとき進路を変更する」を仕様とした場合について説明する。この仕様に対する動作仕様とは、例えば、「移動体が移動中にタッチセンサーに触れたとき、必ず旋回する」などである。これは、ある観点から見たときに動作が満たすべき要件の検査内容と捉えることもできる。本研究では、作成されたプログラムが動作仕様を満たすか否かを、SPIN を用いて実機での実行前に検査する。

図 2 は前述の仕様に対する NQC ソースコードの一部とその部分に対応する自動変換された Promela のソースコードである。

4 実行例

前節の仕様を例題として、実行手順を説明する。処理の流れは、以下の通りである。

- (1) Hichart エディタで NQC ソースコードを読み込む。
- (2) 検証したい性質（表明）をノードに埋め込む。
- (3) その性質の検証が可能となるような Promela に変換する。

Behavioral Verification in the Hichart Development Environment for Embedded Software

[†] Shuto OGIWARA(gz0700154@toyonet.toyo.ac.jp)

[†] Takaaki GOTO(dz0410023@toyonet.toyo.ac.jp)

^{**} Takeo YAKU(yaku@cs.chs.nihon-u.ac.jp)

[‡] Kensei TSUCHIDA(kensei@toyonet.toyo.ac.jp)

Toyo University Graduate School ([†])

Nihon University (^{**})

Toyo University ([‡])

^{†, ‡} 2100, Kujirai, Kawagoe-shi, Saitama, 350-8585, Japan

^{**} 3-25-40, Sakurajosui, Setagaya-ku, Tokyo, 156-8550, Japan

- (4) Promela から pan.c を生成, コンパイルし実行する.
- (5) エラーがない場合は trail ファイルを生成せずに終了し, エラーがある場合は trail ファイルを生成する.
- (6) trail ファイルを解析する.
- (7) 動作仕様検証ウィンドウを終了する. trail ファイルの解析を行った場合は, Hichart エディタ上に結果がフィードバックされる.

上記で(4)と(5)は SPIN の機能を直接利用しているが, それ以外の処理は本研究で独自に開発したものである.

<pre>task move_square(){ while(true){ OnFwd(OUT_A + OUT_C) Wait(100); OnRev(OUT_C); Wait(85); } }</pre>	<pre>proctype move_square() { do state = OnFwd; state = Wait; state = OnRev; state = Wait; od }</pre>
---	---

図2 NQC (左) と Promela (右) のソースコード

図3は Hichart エディタ上で表明の埋め込み操作を行っている画面である. SPIN で検査を行い, 仕様を満たしていない結果が得られた場合に, Hichart エディタ上で原因と考えられるタスクを強調表示する機能を実装した. 図4は trail ファイルを解析した結果の一部である.

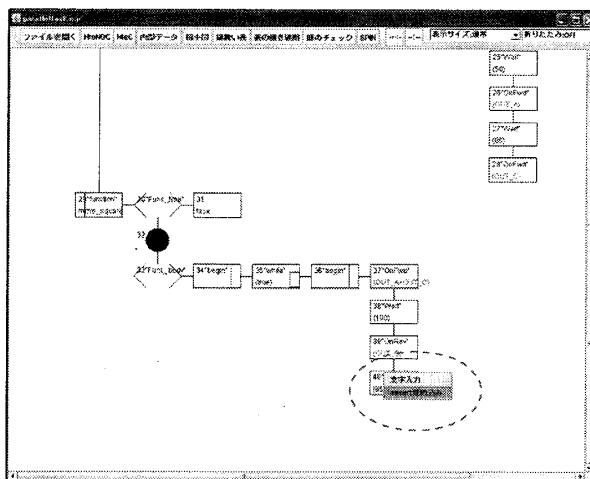


図3 Hichart エディタ上での表明の埋め込み

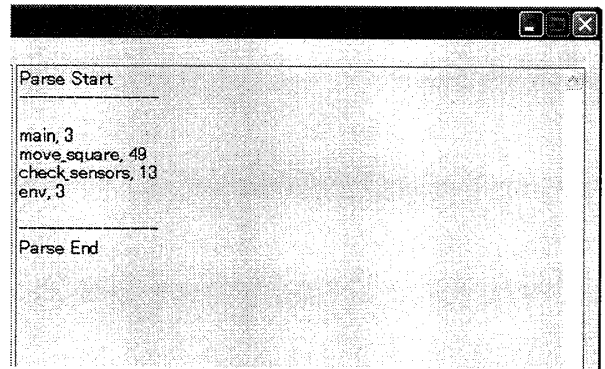


図4 trail ファイルの解析結果

図5は Hichart エディタ上にフィードバックされた画面の一部である. Hichart エディタに結果をフィードバックすることで, 視覚的に動作仕様を満たしていない箇所を探し, 組み込みソフトにおける開発の効率的支援につながる可能性を確認できた.

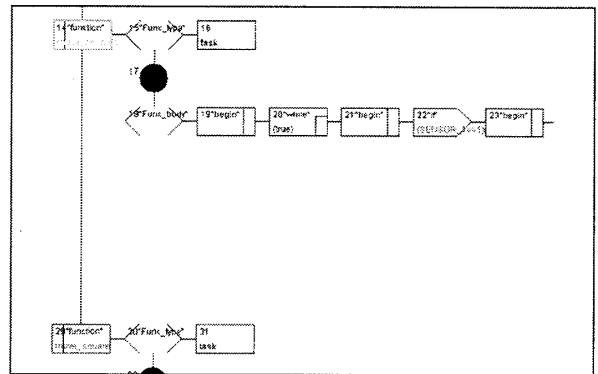


図5 フィードバックされた画面の一部

5 まとめ

Hichart 組み込みソフトウェア開発環境に対してモデル検査の手法を取り入れるために, NQC ソースコードから動作仕様検査のための Promela に変換する機能を追加し, Hichart 上に結果を反映させる機能の実装を行った. 今後は, より柔軟で適切なモデリングが出来るよう改善を重ねたい. さらに trail の解析方法についても強化していく予定である.

参考文献

- [1] T.Yaku Y.Adachi, K.Tsuchida, et al., A visual programming environment based on graph grammars and tidy graph drawing, Proc. of the ICSE '98, pp. 74-79, 1998.
- [2] Takaaki GOTO, Kenji RUISE, Takeo YAKU, Kensei TSUCHIDA, Visual software development environment based on graph grammars, IEICE TRANSACTIONS on Information and Systems, (to appear).
- [3] 大徳雄太, 後藤隆彰, 切島忠昭, 夜久竹夫, 土田賢省. 組み込みソフトウェア向け Hichart 処理系の開発, 電子情報通信学会 2008 年総合大会講演論文集, D-3-3, p.30, (2008).
- [4] 吉岡信和, 青木利晃, 田原康之. SPIN による設計 モデル検証. 近代科学社, 2008.