2P-4 フールプルーフ化を考慮した教育用ロボット言語の開発

長沢伸也、大瀧　厚
明治大学

1．はじめに

ソフトウェア生産における品質管理が最近提唱されているが、手法的には確立されていない。また、工業製品やその作業を対象として、信頼性・安全性解析手法であるＦＴＡ・ＦＭＥＡ[1]の適用や作業のフールプルーフ化[2]の研究が多くみられるようになってきているが、ソフトウェアの生産過程に適用している例はみられない。

そこで、本研究では、これらをソフトウェアへ適用することを試み、フールプルーフ機能を充実させたソフトウェアの開発を行った例として、教育用ロボット動作制御用言語ＡＬＡＲＭ(A Language for Arm Robot Motion)の開発について述べる[3]。

2．ＦＭＥＡ・ＦＴＡのソフトウェアへの適用

(1)ＦＭＥＡ(Failure Mode and Effect Analysis)の適用

ＦＭＥＡは、設計されたシステムのすべての構成品目について、使用中の潜在的な故障モードを仮定し、この故障がシステムの任務達成に及ぼす影響を検討し、信頼性上の弱点を指摘し、故障の未然防止を図る手法である。

ＦＭＥＡの実施手順に従ってＡＬＡＲＭの設計上の弱点の解析を行った。この結果、信頼性ブロック図と表１に示すＦＭＥＡチャートが得られ「行番号直接入力」等４つの故障モードが挙げられた。また、その影響や頻度から「致命的」他の３つの故障等級に分類される。

(2)ＦＴＡ(Fault Tree Analysis) の適用

ＦＴＡは、トップ事象と基本事象間を論理的に連結した図であるＦＴ図により、トップ事象の発生経路を知り、致命的事象を定め、システムを改善する手法である。

ＦＴＡの実施手順に従ってＡＬＡＲＭの欠陥の発生構造の解析を行った。「誤操作発生」をトップ事象に選定し、その発生構造について、ＦＭＥＡチャートの各故障モード毎の推定原因との因果関係をＡＮＤゲートやＯＲゲートを用いて解析した結果、図１に示すＦＴ図が得られた。基本事象としては「コマンドを忘れる」等の人為的不適切な行為や過失であるので、その対策方法として、以下のようにフールプルーフの考え方を適用する。

3．ソフトウェア作業へのフールプルーフ化の適用

フールプルーフとは「人為的不適切な行為や過失等がおこってもアイテムの信頼性・安全性を保持するような設計又は状態」と JISで定義されている。この考え方を全ての誤操作や故障モードに対して適用する。

(1)フールプルーフ・レベルの設定

まず「人為的不適切」の程度を検討する必要がある。

①ユーザー層に関するフールプルーフ・レベル

ソフトウェアに関しては経験や使用目的によるユーザーの層別が一般に困難であることが多いが、ＡＬＡＲＭではＢＡＳＩＣ言語の初歩的知識を有する者を対象とする。

②システム処理内容に関するフールプルーフ・レベル

処理の重要度が高いものに関してはフールプルーフを

表1　ALARMのFMEAチャート

サブシステム	モジュール	故障モード	原　因	影響		評価		故障等級
				ブロック	システム	影響度	頻度	
テキスト編集	行挿入	行番号直接入力	専用編集コマンドを忘れる BASIC のエディタと混同する	機能不全	遅れ発生	小	多	軽微
	行削除	行番号直接入力	専用編集コマンドを忘れる BASIC のエディタと混同する	機能不全	遅れ発生	小	多	軽微
	行修正	行番号直接入力	専用編集コマンドを忘れる BASIC のエディタと混同する	機能不全	遅れ発生	小	多	軽微
		矢印キー使用	専用編集コマンドを忘れる BASIC のエディタと混同する	テキスト消失	機能停止	大	多	致命的
	行番号管理	行番号狂い	挿入・削除を続けて行う リストを確認しない 行番号の変化を考慮しない	テキスト消失	遅れ発生	大	中	重大
テキスト実行	動作命令出力	ストップキー使用	ロボットが意図しない動作をする 専用停止コマンドを忘れる STOPキーを動作停止と思い込む	テキスト消失	機能停止	大	多	致命的

Developement of Educational-use Robot Language with Fool Proofs
Shin'ya NAGASAWA, Atsushi OOTAKI
MEIJI University

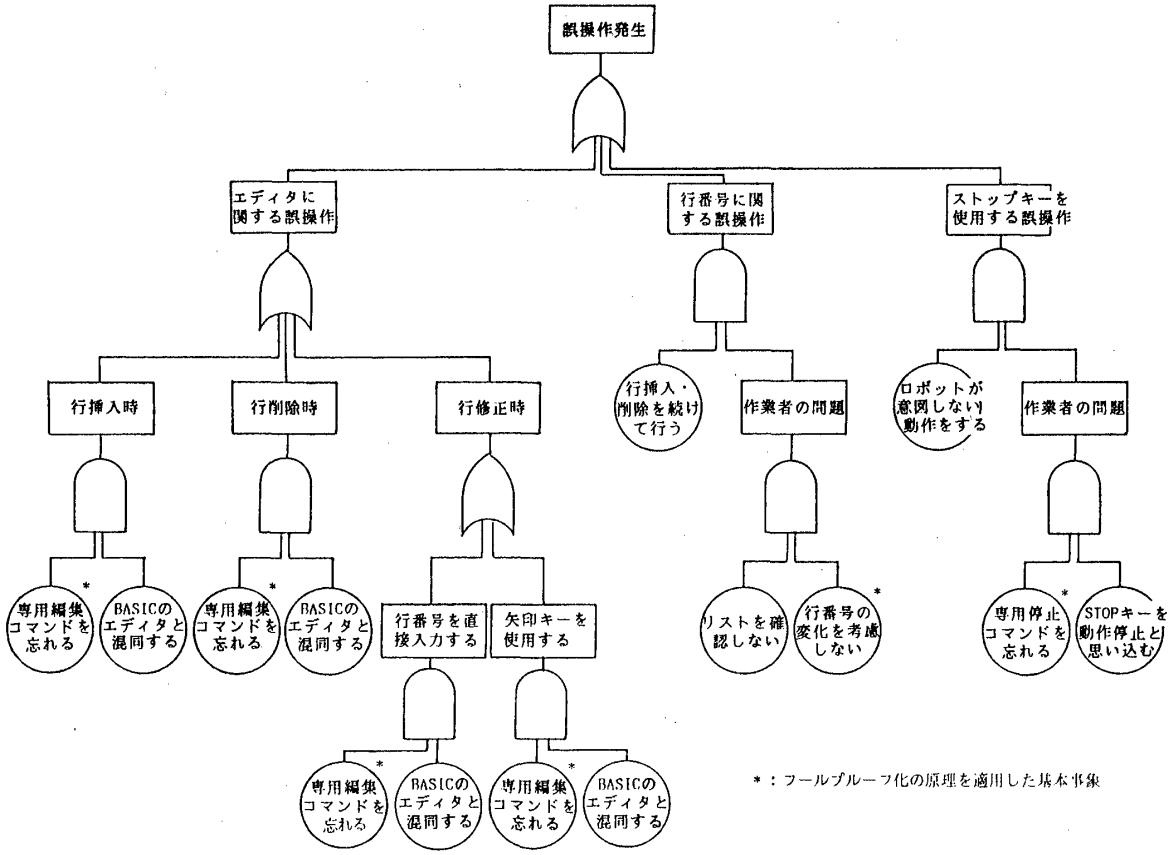

図1　ALARMのFT図

最大限に実施する必要がある。ここでは、表1のFMEAチャート中の故障等級に応じてレベルを決定する。

(2)フールプルーフ化の原理[2]の適用

表2の5項目の原理をソフトウェアに適用する。

例えば、テキスト編集時に専用編集コマンドを用いない為に生ずる誤操作に対しては、行番号のみによりテキスト編集を行うように設計変更し、専用編集コマンドを用いる必要性をなくしてしまうという「排除」の原理を適用する。

表2　フールプルーフ化の原理[2]

原　理	説　明
(1)排除	作業が必要とされる要因（あるいは作業の禁止事項が生じる要因）を取り除き、作業に必要な記憶・知覚・判断・動作等の機能を不要にする。
(2)代替化	作業者に要求される記憶・知覚・判断・動作等の機能を、より確実な何等かの方法により代替化する。
(3)容易化	作業に必要な記憶・知覚・判断・動作等の機能を作業者の行いやすいものにしてミスを低減させる。
(4)異常検出	作業ミスが発生しても、引き続く作業系列の中でそれに起因する標準状態からのずれが検出され是正されるようにする。
(5)影響緩和	作業ミスの影響をその波及過程で緩和吸収することを目的とし、作業を並列化する、あるいは緩衝物や保護を設ける。

4．まとめ

本研究では、信頼性・安全性解析手法であるFMEA・FTAや作業のフールプルーフ化を、以下のようにソフトウェアへ適用することを試みた。

(1)FMEAを適用することにより、ソフトウェアの設計上の弱点を解析することができた。

(2)FTAを適用することにより、ソフトウェアに内在する欠陥の発生構造をモデル化し、解析することができた。

(3)FMEAによる故障等級に基づき、ソフトウェアのフールプルーフ・レベルを設定した。

(4)「フールプルーフ化の原理」に基づき、ソフトウェアのフールプルーフ化を実現した。

以上の結果、フールプルーフ機能を充実させた教育用ロボット動作制御用言語の開発を行うことができた。

参考文献

1)鈴木順二郎・牧野鉄治・石坂茂樹：FMEA・FTA実施法、日科技連出版（1982）

2)中條武志・久米　均：作業のフールプルーフ化に関する研究－フールプルーフ化の原理－、品質、15（1984）

3)長沢伸也・大瀧　厚等：教育用ロボット言語の開発(1)、(2)、明治大学工学部研究報告、47,51（1984,1986）