自動販売機制御ソフトウェアの再開発
～アスペクト指向を用いた組込みソフトウェアの実行前検査結果表示ツールの開発～
―検査プロセスを考慮して―

M2008MM026 岡本侑久
指導教員：野呂昌満

1　はじめに

組込みソフトウェアは，一般的に並行に動作する機器を制御しており，システムの挙動が複雑で予期せぬ不具合が含まれる可能性がある．予期せぬ不具合は発見が遅れるほど，修正にコストや工数がかかる．

早期に予期せぬ不具合を発見する開発プロセスとして，本研究室では，E-AoSAS++に基づく開発プロセスを提案している [3]．E-AoSAS++とは，組込みソフトウェアのためのアスペクト指向ソフトウェアアーキテクチャスタイルで，E-AoSAS++で提案している開発プロセスでは，アーキテクチャ記述に対して実行前検査をおこなう．実行前検査は，挙動の把握が困難な並行処理システムに対して網羅的に検査する事が可能で，予期せぬ不具合を発見する事が可能である [4, 6]．実行前検査では，CSP 記述と実行前検査ツールである FRR を用いてフェーリア[1]を発見する [2]．FDR は，フェーリアを検出した際にフェーリア発生までのイベント系列を反例として出力する．

開発者は，反例からアーキテクチャ記述上に存在するフォールトを特定し修正する．現在，FDR で発見したフェーリアに対応する，アーキテクチャ記述上に存在するフォールトを特定する作業は，開発者の経験や知識に基づき行なわれている，フェーリア発生までのイベント系列を目で 1 つずつ追い，システムの挙動を考えながらアーキテクチャ記述上のフォールトを特定している．

イベントの発生系列からシステムの挙動を把握する事は大規模システムほど困難である．また，フェーリアの原因となるフォールトは様々なので，他視点からフォールトを特定して行く必要があり，フォールトの発見に工数やコストがかかる．上記の問題を解決するために，実行前検査結果表示ツールを開発し，フェーリア発生までのシステムの挙動を表示し，フォールトの特定を支援する機能を提供する．

本研究の目的は，Product Line Software Engineering (以下，PLSE) の概念に基づき，実行前検査結果表示ツールを開発対象として，アスペクト指向アーキテクチャ，MVC アーキテクチャを中心とした開発で再利用性，柔軟性の高いソフトウェア開発をおこなう事である．ドメインの特徴から構築したアーキテクチャに基づき開発した実行前検査結果表示ツールにおいて，表示画面追加時や，他の実行前検査ツールへの適用可能性を考察し，適用したアーキテクチャの妥当性を示す．

[1]一般的なフェーリアは，実行時における挙動と仕様との不一致を指すが，ここでは，実行前検査で検出できる，予測される実行時の挙動と，仕様との不一致を指す．

本研究は OJL として，富士電機リテイルシステムズ株式会社と連携し，自動販売機制御ソフトウェアの再開発をテーマに実行前検査結果表示ツールの開発をおこなった．

2　開発対象と開発プロセス

2.1　実行前検査結果表示ツールの概要

本研究で開発する実行前検査結果表示ツールでは，フェーリア発生までのイベントの系列，各 CSTM 間のイベントの送受信を表示する．表示した情報に対して，特定の情報のみ表示するフィルタリング機能や各 CSTM の状態やキューの状態を表示する機能を提供する．実行前検査結果表示ツールのメイン画面イメージを図 1 に示す．

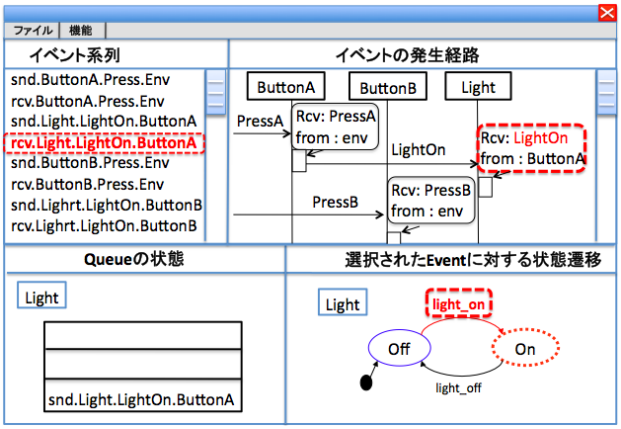

図 1　実行前検査結果表示ツールの画面イメージ

本ツールの機能の特徴として，以下の 3 つの点があげられる．入力となるデータが対応する実行前検査言語ごとに異なり多種多様である．内部で保持するデータに対する処理が追加，変更される可能性が高い．出力となる画面表示はテキスト表示や，シーケンス表示，状態遷移表示，キューの状態表示など多岐にわたる．

上記の特徴に対して柔軟性や再利用性，拡張性を考慮したアーキテクチャ設計をする必要がある．次章より，実際におこなっている検査プロセスより，データに対する処理にどのような処理が必要であるか整理し，表示系としてどのような出力を表示する必要があるかについての整理をおこなう．整理した情報を基に，入出力やデータに対する処理のアーキテクチャ設計，詳細設計をおこなう．

2.2　開発プロセス

開発するツールは，GUI を用い，対象となるデータや画面表示が変更されやすい特徴を持つので，MVC アーキテクチャ，アスペクト指向ソフトウェアアーキテクチャ

を中心としたPLSEの概念に基づく開発が妥当であると考えた．ツールの特徴である，対象となるデータに対する処理や追加の変更が多く，アスペクト指向の観点から再利用性を考慮したアーキテクチャが先行OJLにおいて構築されている．本研究では，先行OJLで構築された核資産であるデータに対する処理の変更，追加が多いツールに対するアーキテクチャ構築の考え方や，再利用性を考慮した開発プロセス，先行OJLで構築したアーキテクチャの共通部分を再利用した開発を行なう [5, 7].

3 検査プロセスの分析

実行前検査ツールFDRで検査可能な検査項目から，アーキテクチャ記述上に現れるフォールトを発見するためのプロセスを分析する．FDRで検査可能なフェーリアを整理し，各フェーリアからフォールトを特定するプロセスにおいて必要な情報や機能について記述する．

3.1 検査項目

E-AoSAS++の開発プロセスの実行前検査において，FDRで検査するフェーリアは次の４つである．

- 安全性
- 活性
- デッドロックフリー
- ライブロックフリー

安全性の検査は，開発者が期待するイベント系列をシステムが満たす事を検査する．期待するイベント系列と違うイベント系列が生じた場合，安全性を満たさない．活性の検査は，安全性を満たした上で，仕様が期待するイベント系列が生じる事を検査する．期待するイベントが発生しない場合，活性を満たさない．デッドロックフリーは，システムが意図しない停止をしない事を検査し，ライブロックフリーはシステムの処理中にイベントの無限ループが存在しない事を検査する．

3.2 反例からフォールトを特定する過程

FDRの反例であるイベント系列から，アーキテクチャ記述上に現れるフォールトを特定するプロセスにはどのようなプロセスがあるか記述する．

- 安全性の反例からフォールトを特定
  安全性の検査に対する反例からは，どのような理由で想定外のイベントが出現したか特定する必要がある．

1. 想定外のイベントが出現した場所
   開発者が意図しないイベントが発生したSTMやイベント，アクションを確認する．
2. 状態の確認
   想定外のイベント発生までの各STMの状態や，キューの状態を確認する．
3. 反例のイベント系列をステップ実行
   STM間のイベントの送受信をワンステップずつ確認する．想定外のイベント発生までのSTM間の関係が確認できる．
4. 送信－受信ペアの確認
   E-AoSAS++において，キューにイベントは溜まるので送信と受信のタイミングが異なる．送受信のペアに着目した検査が必要である．
5. フィルタリング
   対象の範囲をせばめる際に特定のSTMやイベント，アクションに着目して，イベントの送受信や状態の遷移を確認する．
6. 次ステップの候補
   次に処理可能なイベントやアクションを表示する事で反例に至らない場合との比較から原因となる可能性のある場所を推測可能である．

- 活性の反例からフォールトを特定
  活性の検査に対する反例からは，どのような理由でイベント系列で停止したのかを確認する必要がある．活性の検査は，安全性の検査とほぼ同様のプロセスで問題を発見していくと考えられるが，活性の検査を満たさない場合，本来発生すべきイベントが発生していないので，本来発生すべきイベントを表示する機能が必要である．
- デッドロックフリーの反例からフォールトを特定
  デッドロックの反例からは，開発者が意図するイベントが発生しないので活性と同様の機能で検査が可能である．
- ライブロックフリーの反例からフォールトを特定
  ライブロックの反例からは，無限ループがなぜ発生したのか特定する．ライブロックの反例からフォールトを特定する際は，無限ループのイベント系列を抽出することでフォールト箇所の発見が支援できる．

4 ソフトウェアアーキテクチャの設計

本研究の開発対象である実行前検査結果表示ツールの特徴として次の点が上げられる．

- 表示する情報が多種多様
  - 要求によって必要な画面表示が変わるので画面表示の追加，変更に対する柔軟性，再利用性を保証する必要がある．
  - 表示する情報に応じて，必要なデータが変更される可能性がある．
  - GUIを用いた分かりやすいインタフェースを提供している．
- フォールト箇所を特定するプロセスの支援
  - 開発者が知りたい情報を表示する機能を提供するので，追加，変更が容易な必要がある．
  - 実現する機能は複数の機能を同時に使うので，同時に実行可能な設計にする必要がある．

上記のドメインの特徴と，実現する機能からアーキテクチャ設計をおこなう．

4.1 機能

実行前検査結果表示ツールの機能として，次の機能を実現する．

- イベント送受信の時系列表示，ステップ実行
- 特定の情報をフィルタリングして表示
- STM の状態，キュー の状態を表示
- 機能の組み合わせ

イベントの送受信の時系列表示は，反例における STM のイベント送受信の関係を視覚的に分かり易い形で表現する機能であり，ステップ実行で 1 つずつ処理を進める．フィルタリング表示は，特定の情報のみを表示する．状態表示は，反例の中のあるイベント発生時における各 STM の状態やキューの状態を表示する．上記の機能を組み合わせた処理が可能なツールのアーキテクチャを設計する．

4.2 アーキテクチャ設計

実行前検査結果表示ツールのデータと表示画面が変更，追加される可能性が高いという特性から，MVC アーキテクチャに基づいた開発を行なう．先行 OJL で構築されたアーキテクチャを基に，可変部分である View と GUI と Model の一部を変更した．検査結果表示ツールのアーキテクチャの概要図 2 に示す． 検査結果表示ツールの表

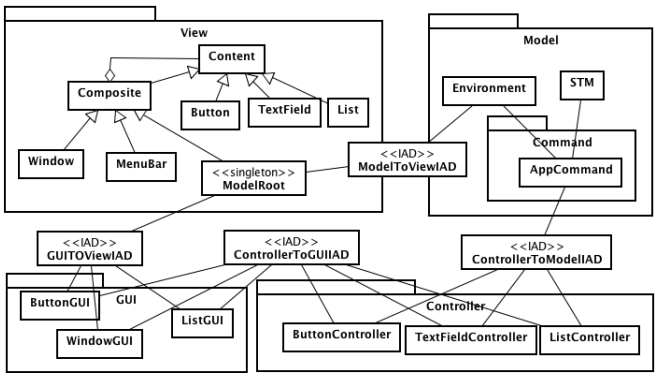

図 2 検査結果表示ツールのアーキテクチャの概要

示において，GUI アスペクトは，ユーザの入力に対するインタフェースを表し，View アスペクトはツールの見た目に関する処理を表す．Controller アスペクトは．ツールの入力を管理する処理を表し，Model アスペクトは，データを保持するアスペクトを表す．Model 内の STM は，E-AoSAS++のアーキテクチャ記述の情報を保持し，Environment では，STM の現在の状態や，キューがどのような状態にあるかの情報を保持する．各アスペクトは，アスペクト間記述でつながっており，各アスペクトは独立している．アスペクトはそれぞれ個別に実現されているので互いに独立して修正，追加が可能である．

複合的な処理に関する部分は，先行 OJL で構築したアーキテクチャに用いられた Pipes and Filters アーキテクチャの技術を再利用し，処理の結合，交換に柔軟に対応できるようにし，複数の機能を処理ステップの集合と捉えて実現する．

本アーキテクチャに基づく設計では，単機能の View を組み合わせる事で，他視点からの表示が可能になる．今後の追加や変更が容易な，ソフトウェアが設計でき，ソフトウェアの柔軟性が確保される．また，GUI を用いることで，ユーザインタフェースの違いを吸収する事が可能であり，GUI に対する変更も独立して行なう事が出来る．

4.3 モデル変換による自動生成

E-AoSAS++ に基づくアーキテクチャ記述の PIM から，実行前検査結果表示ツールの自動生成をおこなう．実行前検査結果表示ツールに必要な情報は，STM の初期状態，状態，遷移，イベント，アクションの情報である．これらの情報は E-AoSAS++に基づく PIM の抽象構文木が全て保持している．本研究室で提案している E-AoSAS++ の PIM からモデル変換でプログラムコードを出力する自動生成の手法を用いる事で，PIM のモデルからシミュレート用のプログラムコードの大部分は自動生成可能である．実現するツールはワンステップずつ実行するので，並行処理に関する処理と，キューに関する処理に関しては，実行前検査結果表示ツール用の変換規則に基づきプログラムコードを生成する．

4.4 実現

実現したツールでは，特定の STM やイベントにフィルターをかける機能，イベント系列の特定のイベント発生時の STM の状態を表示する機能を実現した．また，実現した複数の機能を同時に実行する事が可能である．実現した機能であるモデルが保持するデータに対する処理を図 3 に示す．

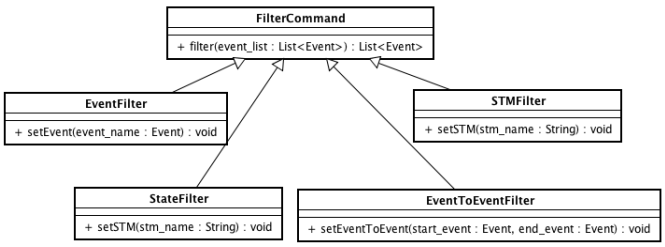

図 3 データに対する処理

図 3 は，本ツールの特徴である次の点を考慮し，pipes and filter アーキテクチャとコマンドパターンを用いた設計をおこなった．本ツールの機能の特徴として，他視点から情報に処理を与える事が考えられる．CSTM 単体として考える場合の視点として，状態を表示する場合やイベントを表示する場合，キューを表示する場合，アクションを表示する場合などが考えられる．複数の CSTM を考える場合の視点として CSTM 間のイベント送受信関係や，次に送信可能なイベントと対応する CSTM 群などの情報を表示する事が考えられる．このような様々な視点を複合的に考慮した表示が機能要求としてあげられる．様々な視点から表示にフィルター処理をかける場合に，Filiter の追加，変更，複合的な組み合わせが容易という点を考慮し Pipes and Filters アーキテクチャを用いてツールの開発をおこなった．また，処理の追加，変更が容易なコマンドパターンを用いて，機能を実現した．

5 考察

表示画面追加時の拡張性についての考察と，FDR と似た実行前検査ツールの検査結果を表示する際の柔軟性について考察をおこなう．

5.1 表示ツールの拡張性の考察

表示画面に新しい情報を表示する場合を考える．新しい表示画面を追加する際の変更箇所を図 4 に示す．

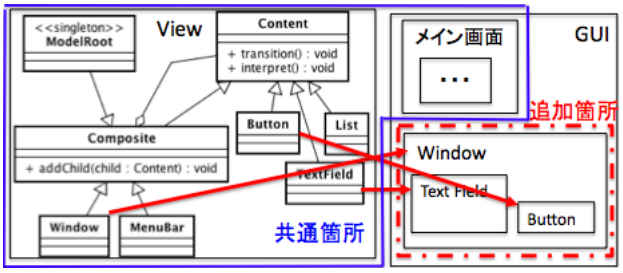

図 4 表示追加時の変更箇所

本研究で提案したアーキテクチャは，内部の処理と表示画面はアスペクトとして分離されており，新しく表示画面を追加した場合のインタフェースは共通であるので，内部の処理を変えずに表示画面を追加する事が可能である．画面に表示する内容が変更または追加された際に，GUI に表示するコンポーネントがすでに View で定義されている場合，GUI を追加する際に必要な変更は GUI アスペクト内の表示に関係する部分だけの修正でよいので，拡張性が高い設計になっていると言える．

5.2 他の実行前検査ツールの結果表示に対する柔軟性

他の実行前検査ツールへの適用可能性を考察する．SPIN など他の実行前検査ツールの検査結果を入力として同様の情報を表示する事が可能か考察する．SPIN とは，FDR と同様に実行前検査用のツールである．SPIN ではプロメラ記述で記述した入力を与える事で，フェーリア発生までの状態の遷移を出力する．FDR の出力と SPIN の出力の対応関係を図 5 に示す．

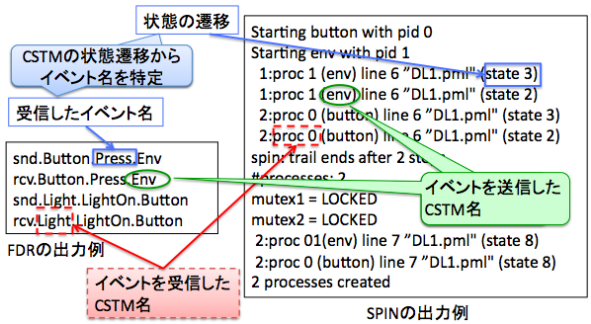

図 5 FDR と SPIN のイベント系列の対応

SPIN も FDR の出力同様，CSTM のイベントによる状態遷移の軌跡の情報を取得可能である．出力ファイルから遷移を起こした CSTM と状態の変化から，イベント名や，CSTM 状態の変化，キューの状態を保持する事が可能である．イベントの意味と，ワンステップ実行の通信の意味の変更が変わる．本研究で開発するツールでは，上記の 2 つの意味の解析方法を strategy パターンを用いて各ツールごとに変更する．strategy パターンを使用する事で，各出力ファイルに対して解析パターンをそれぞれ追加する事で，ツールの内部や，画面表示の View や，ユーザインタフェースの GUI を変更する事なく，他の実行前検査ツールに対応する事が可能である．新しく違うイベント系列を出力するような実行前検査ツールの結果を表示する際も，strategy パターンの振る舞いに関する実装を増やし，切り替える事で対応可能である．

6 ベース技術・メタ技術

本研究で利用したベース技術は，アスペクト指向ソフトウェアアーキテクチャ，MVC アーキテクチャ，Pipes and Filters アーキテクチャ，デザインパターン，実行前検査である．

メタ技術は，PLSE である．PLSE の概念に基づくアーキテクチャを中心とした開発プロセスのもと開発をおこない，先行 OJL で構築したアーキテクチャを再利用，カスタマイズし，本ツールの特性に特化したアーキテクチャを構築した．

7 おわりに

本研究では，フォールト特定プロセスの整理と，実行前検査結果表示ツールの開発を行なった．追加，変更の可能性が高い，View や GUI の独立性の高いアーキテクチャを構築し，考察することで，アーキテクチャの妥当性が証明でき，再利用性，柔軟性の高いツールを実現する事ができた．今後の課題として，フェーリア発生までの状態遷移をシミュレートする機能の実現があげられる．

参考文献

[1] E. Gamma, J. Vissides, R. Helm, and R. Johnson, Design Patterns Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.

[2] Formal System (Europe) Limited, "FDR," http://www.fsel.com/index.html/ , 2007.

[3] Noro, M, Sawada, A, Hachisu, Y and Banno, M, "E-AoSAS++ and its Software Development Environment," Proceedings of the 14th Asia-Pacific Software Engineering Conference (APSEC 2007), pp.206-213, 2007.

[4] 張漢明，蜂巣吉成，沢田篤史，野呂昌満，" アスペクト指向ソフトウェアアーキテクチャの振る舞い検証に関する考察，" ソフトウェア工学の基礎ワークショップ XVI，pp.267-274，2009.

[5] 加藤大地，"自動販売機制御ソフトウェアの再開発 ～自動販売機制御ソフトウェアのログ解析支援ツールの設計と実現～，" 南山大学大学院 数理情報研究科 2008 年度 修士論文要旨集，pp.114-117，March 2009.

[6] 加藤大地， 蜂巣吉成， 沢田篤史， 野呂昌満，" E-AoSAS++に基づく開発支援環境 - 実行前検査ツールの提案 - ，" 情報処理学会研究報告，Vol.2009，No.31，pp.121-128.

[7] 小島守道，"自動販売機制御ソフトウェアの再開発 ～自動販売機制御ソフトウェアのログ解析支援ツールのアーキテクチャ設計～，" 南山大学大学院 数理情報研究科 2008 年度 修士論文要旨集，pp.118-121，March 2009.