様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成２４年５月１５日現在

機関番号：１７１０２
研究種目：若手研究（Ｂ）
研究期間：2009～2011
課題番号：２１７０００３５
研究課題名（和文）　プロダクトラインアプローチによる異常処理系の再利用性向上に関する研究
研究課題名（英文）　A Study on Improving Reusability of Exceptional Operations with Software Product Line Engineering
研究代表者
中西　恒夫（NAKANISHI TSUNEO）
九州大学・大学院システム情報科学研究院・准教授
研究者番号：70311785

研究成果の概要（和文）：
（200 字程度，難解な用語は使わない。）
ソフトウェアの異常処理系は，実用システムではソフトウェア全体の 3 分の 2 を超えるほどの分量となることも珍しくなく，効率的な再利用が望まれる。本研究では，FMEA，HAZOP 等の既存の信頼性／安全性解析手法を応用し，開発工程の初期から異常処理系の機能と再利用性を洗い出し，一製品ではなく，一連の製品群の開発において効率的にソフトウェア再利用を実現するプロダクトライン開発プロセスを提案した。

研究成果の概要（英文）：
Exceptional operations should be developed with efficient reuse of existing software artifacts, since they occupy a large part of the whole in the scale. This work proposed a software product line development process that efficiently reuses the artifacts for exceptional operations. The process identifies necessary exceptional operations and their reusability not for one product but for an expected set of products.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
２００９年度	1,100,000	330,000	1,430,000
２０１０年度	1,200,000	360,000	1,560,000
２０１１年度	900,000	270,000	1,170,000
年度			
年度			
総　計	3,200,000	960,000	4,160,000

研究分野：　総合領域
科研費の分科・細目：　情報学・ソフトウェア
キーワード：　ソフトウェア工学，ソフトウェアプロダクトライン

１．研究開始当初の背景

プロダクトライン開発方法論は，プロダクトラインに属する製品間の共通性と相違性を明確に分離し，プロダクトラインのソフトウェアアーキテクチャを定め，アーキテクチャにあわせたコンポーネント開発を進めることで，ソフトウェア資産の再利用性の向上を図るソフトウェア開発のパラダイムである。研究開発当初，我が国は産業界においてプロダクトライン開発方法論に対する関心が高まっていたが，米国ならびに欧州ではすでに十数年にわたって研究が続けられていた。

既存のプロダクトライン開発方法論は異常処理系に関する特別の考慮はなされていなかった。しかしながら，現実に使用されているソフトウェアでは，その 3 分の 2 以上を異常処理系が占めていることは珍しくなく，

異常処理系の再利用を図らずにソフトウェアの十分な再利用をなし得ることはできない。

一方で，ソフトウェアの異常処理系に関してはすでに多くの研究がなされており，たとえば FTA，FMEA，HAZOP といったシステム全体，あるいはメカやハードウェアにおける異常処理系分析手法をソフトウェアに応用したり， N-Versioning などソフトウェアアーキテクチャレベルでの冗長化や処理方式の冗長化によってフォールトトレランスを実現したり，あるいは各種プログラミング言語における例外処理構文の導入が行われたりしていた。しかし，これら異常処理系に関する方法論や方式をプロダクトライン開発方法論のパラダイムに組み込む研究については，世界的に見ても，十分に実施されてはいなかった。

２．研究の目的

上述の背景を鑑みて，本研究は研究開始当初，以下の事柄を明らかにすることを目的に据えた。

(1) 既存の異常処理系分析・設計・実装に関する方法論あるいは方式のプロダクトライン拡張
(2) 異常処理系に関する製品間相違性の実現
(3) 異常処理系に関するトレーサビリティ実現手段

３．研究の方法

(1) プロダクトライン開発において第一に重要視される開発工程は，プロダクトラインに属する製品間の共通性と相違性の分析である。ゆえに，本課題では，共通性／相違性分析，すなわちフィーチャモデリングにおいて異常処理系に特有の事柄があるかどうかを明らかにすべく，まずは正常処理系のフィーチャモデリングのガイドラインを定め，続けて異常処理系のフィーチャモデリングのガイドラインの策定を試みた。成果は次節(5)に述べる通りである。

(2) 既存の異常処理系の処理方式を網羅的に調査し，それらのプロダクトライン拡張，すなわちプロダクトライン開発方法論のパラダイムとの統合を図った。複数の処理方式に対して研究を行ったが，プロダクトライン開発方法論のパラダイムとの統合を図れたものは同一仕様異実装のコンポーネントと多数決処理を用いてシステムのフォールトトレランスの確保を図る N-Versioning のみであった。成果は次節(4)に述べる通りである。

(3) プロダクトライン開発方法論の本質は上流からの再利用の作り込みであると捉え，プロダクトライン開発の上流工程での信頼性／安全性分析に手法を重要視し，それらのプロダクトライン拡張を図った。FMEA，HAZOP のプロダクトライン拡張を図り，製品群全体を対象とするソフトウェア資産の開発活動であるドメインエンジニアリング工程の要求・仕様といった上流工程に FMEA，HAZOP を組み込むことを試みた。

この試みが突破口となり，前節で述べた研究目的の大部分を満足する，体系的な異常処理系を考慮したプロダクトライン開発プロセスを確立することが可能となった。成果は次節(1)～(3)に述べる通りである。

４．研究成果

(1) システムの信頼性・安全性を向上する分析手法として，①システムの構成要素の故障モードとその影響を分析・評価し，それらへの対策を検討する FMEA（Failure Modes and Effects Analysis）や，②システムの諸属性値の大幅な逸脱から異常状態を想起し対策を検討する HAZOP（Hazard and Operability Analysis）が知られ，これまでさまざまな産業分野で応用されている。

機械製品等の FMEA では，システムのコンポーネントの故障モードを検討することが多い。本研究では，（コンポーネントに分割される以前の）要求，より具体的にはユースケースの各ステップに関する故障モードを洗い出し，その影響を分析し，機能面や工程面での対策を講じることで，システムの信頼性や安全性の面での改善を図るプロセスを提案した（学会発表②）。

同プロセスでは，機能面での対策について製品間の共通性／相違性分析を施し，プロダクトライン開発に備える。

故障モードの洗い出しには HAZOP の考え方を応用する。ユースケースの各ステップにおいて，そのステップで取り扱われるデータや振舞いに関する諸属性に対して，ガイドワードを適用し，それら諸量の想定値からの大幅な逸脱から異常状態を想起することで故障モードを洗い出す。

提案プロセスの考え方は，ドメインエンジニアリングの要求・仕様から実装に至る工程で適用できるものである。しかし，要求・仕様といった開発初期段階での適用は，プロダクトラインの将来製品における異常処理系に関する機能を明らかにし，ひいてはそれら機能の実現に関わる設計や実装，テストのコア資産化を促す。結果，異常処理系の拡張によるソフトウェアアーキテクチャの劣化防止や異常処理系の再利用性の向上が期待される。

(2) (1)の提案プロセスの考え方を設計，ならびに実装段階で応用し，C++や Java の例外処理構文（try-catch-finally 構文）で用いる例外クラスを設計する方法論を提案した（学会発表③）。

これら言語の例外処理構文は，一定の指針に基づいて効果的に用いれば，異常系の可読性や再利用性の向上に大きく寄与する。しかし，上流で例外処理の仕様が十分に記述されていないがために，異常処理系の設計が実装者任せとなり，しばしば場当たりに例外処理構文が用いられる。処理すべき例外処理の無視，例外の自身の呼出元への無責任な転送，過剰な数の例外クラスの定義が行われ，本来システムの信頼性向上に寄与すべき異常処理系が欠陥の温床になっている。

提案方法論では，各メンバ関数の正常処理仕様記述の各処理ステップについて研究成果(1)の方法論を用いて故障モードを導出し，FMEA を実施する。FMEA で検討された対策に対して，共通性／相違性分析を実施し，実行時対策相違性モデルを記述する。この実行時対策相違性モデルは，製品間の相違性のみならず，製品の実行時の振舞いの相違性をも表現するようにフィーチャモデルを拡張したものである。同モデルを用いて，例外処理の実行時の振舞いの違い，例外処理で必要とされるデータとその違い，さらにはそれらの製品間での違いを整理し，try-catch-finally 例外処理構文で使用される例外クラスを設計する。

(3) (1)，(2)で述べた開発プロセスの各工程における FMEA や HAZOP の結果，開発中の製品，あるいは次期製品において，システムの信頼性・安全性を向上するべく，機能の追加や変更が必要となる。こうした機能の追加や変更が計画にあがっていなかった場合，あるいは（スクラッチからではなく）既存製品群からのプロダクトライン開発の導入を行っている場合は，プロダクトラインの拡張，すなわちコア資産の拡張が必要となる。こうしたプロダクトラインの計画外の機能の拡張や変更に対応するべく，派生開発方法論 XDDP : eXtreme Derivative Development Process を導入し，さらに XDDP からのプロダクトライン開発への漸次的移行プロセスを提案した。(学会発表①。但し，信頼性・安全性に関する機能強化の文脈は含めず，一般化された漸次的移行プロセスの論文として発表している。)

プロダクトライン開発方法論はしばしば導入障壁の高さが指摘されている。プロダクトライン開発方法論は，既存の開発プロセスの変更（そもそもプロセスに基づいた開発が行われていない場合は開発プロセスの導入）や開発組織の変更，ソフトウェアアーキテクチャの確立など導入前に求められることが多い。また，製品群全体のことを考えてコア資産の開発を行うドメインエンジニアリングのコストは大きくなりがちであり，アプリケーションエンジニアリングにおいてこれらのドメインエンジニアリングのコストを回収する期間を耐えられるどうかが大きな問題となる。そのためプロダクトライン開発方法論の導入を躊躇している企業も少なくない。一方，そもそもソフトウェアアーキテクチャの概念を持たず，コア資産の蓄積や保守を行わない XDDP は，派生製品の迅速な開発というプロダクトライン開発と共通の目的を有しつつも，その導入障壁は低く，産業界での実践例も多く発表されている。但し，ソフトウェアアーキテクチャの概念を持たない XDDP は，度重なる機能の追加・変更によるソフトウェア構造の劣化を防ぎ得る方法論ではなく，長期的に見ればプロダクトライン開発方法論のほうが QCD の改善を図れるものと考えられる。

提案プロセスは，相当数の派生製品を開発しつつもプロダクトライン開発方法論への移行に及び腰の企業に対して，導入障壁の低い XDDP の導入を進め，その後，プロダクトライン開発方法論のパラダイムを導入，すなわちコア資産の蓄積・保守やソフトウェアアーキテクチャの確立を行い，プロダクトライン開発への移行を促す効果が期待される。

本課題から発展したこの課題はさらなる研究が必要であり，平成 24 年度科学研究費（基盤 C)（『プロダクトライン開発の派生開発からの導入と適用範囲拡大に関する研究』，研究代表者：中西恒夫，課題番号：24500042）の助成を受けて，継続して研究する予定である。

(4) 既存の異常処理系の設計・実装方法論として，セーフティクリティカルアプリケーションで用いられる N-Versioning に着目し，そのプロダクトライン拡張を行った。N-Versioning は同一仕様異実装のコンポーネントを並行に動作させ，多数決処理等により，たとえコンポーネントに欠陥があってもコンポーネントが仕様通りの挙動を示すようにする，すなわちコンポーネントの信頼性を保つ手法である。N-Versioning を行う場合，同一仕様異実装のコンポーネントの多様性をどう管理するかが問題となるが，本研究では製品間相違性とコンポーネント間多様性をフィーチャ概念により統一的に記述する手法，ならびに同手法を前提としたコア資産管理手法を提案した（学会発表⑥)。

この手法は成果(1)と組み合わせて活用されるべきものである。FMEA では故障モードの影響分析をした後，それらへの対策を検討する。コンポーネント開発時につくり混まれた欠陥（＝故障モード）に対する対策のひとつとしてプロダクトライン拡張された N-Versioning を導入し，プロダクトライン開発の枠組み内で多様性のある同一仕様異実装のコンポーネントを開発する。

(5) プロダクトライン開発の分野においては，製品間の相違性を記述するのに，フィー

チャモデリングがデファクトスタンダード的に用いられている。しかし，フィーチャモデルはどうとでもそれらしいものを記述できるため，どのようにフィーチャモデリングを行えばよいのかモデリング実施者を混乱させている。異常処理系の製品間相違性を論じる以前にフィーチャモデリングのあり方を論じる必要がある。本研究では，ソフトウェアアーキテクチャの事前設計を目的とする，フィーチャモデリングのガイドラインとアンチパターンを定義するとともに（学会発表⑦⑧），大規模なフィーチャモデルを理解しやすく描画するフィーチャモデル描画アルゴリズムを開発した（学会発表⑨）。

その後，異常処理系を考慮したフィーチャモデリングのガイドラインを構築する計画であったが，多くの異常処理は既存のフィーチャモデリングの枠組みでモデリング可能であり，せいぜい機能とその故障時の対策となっている異常処理機能とを紐づける程度で実用上十分であり，発表に値するほどの新規性のある成果は得られなかった。

(6) アドホックネットワークのルーティングプロトコルである AODV とその派生プロトコル，DSR とその派生プロトコルをそれぞれプロダクトラインに見立てて，公開文献に基づいてこれらプロトコルファミリの異常処理系のモデリングを試みた。これらプロトコルの公開資料は，正常処理系についてはよく記述されているものの，異常処理系については記述が曖昧であり，実装依存となっているケースが多く，満足な質と量の文献を得ることができず，プロトコルの調査に想定以上の時間がかかった。期待通りに研究を進めることができず，正常系のモデリングをやっとできた程度で十分な成果を得ることはできなかった。(学会発表⑤)

(7) 製品出荷後においても信頼性・安全性改善を図って改版を行うケースに対応すべく，出荷後のソフトウェア更新手法を網羅的に調査し，その実装を行った（学会発表④）。

大規模なソフトウェアの更新にも耐えられるよう，すでに携帯電話やテレビにおいて行われている放送によるバイナリ差分更新方式（現有ソフトウェアに対する新ソフトウェアのバイナリ差分を放送する）を採用し，セーフティクリティカルソフトウェアである車載ソフトウェアでの利用を想定して，ソフトウェア更新手法の設計と実装を行った。具体的にはバイナリ差分更新ツールである bsdiff／bsupdate を，メモリが潤沢にない環境でも利用できるよう，現有ソフトウェアのバイナリを直接編集するインプレース型更新手法を提案した。提案手法は，フラッシュメモリを備えたシステムでの利用を前提とし，アップデート中での電源を喪失してもアップデートを再開できるようなアルゴリズムとしている。

商業レベルでの実用にも耐えるアルゴリズムを開発できたが，決してプロダクトライン開発の特性を活かしたアルゴリズムではなく，本課題から浮いた位置づけの研究成果となった。

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔学会発表〕（計 9 件）

① Tsuneo Nakanishi, Claes Jæger-Hansen, Hans-Werner Griepentrog, “Evolutional Development of Controlling Software for Agricultural Vehicles and Robots,” 3rd Int. Conf. on Machine Control and Guidance, 2012 年 3 月 28 日, ドイツ Stuttgart.

② 中西 恒夫, 久住 憲嗣, 福田 晃,「FMEA を援用した例外処理設計手法」, 情報処理学会ソフトウェア工学研究会, 2012 年 3 月 15 日, 東京都千代田区.

③ 中西 恒夫, 久住 憲嗣, 福田 晃,「ソフトウェア FMEA の一手法とプロダクトライン開発におけるその利用」, 電子情報通信学会ソフトウェアサイエンス研究会, 2011 年 3 月 13 日, 沖縄県那覇市.

④ 施 欣漢, 中西 恒夫, 久住 憲嗣, 福田 晃,「放送による車載機器向けソフトウェア差分更新方式」, 情報処理学会組込みシステム研究会, 2011 年 3 月 11 日, (震災のため開催中止, 文献は公開）.

⑤ Yang Cao, Tsuneo Nakanishi, Kenji Hisazumi, Akira Fukuda, “Organizing Ad Hoc Network Routing Protocols with Product Line Engineering,” 情報処理学会組込みシステム研究会, 2011 年 3 月 11 日, (震災のため開催中止, 文献は公開）.

⑥ 中西 恒夫, 久住 憲嗣, 福田 晃,「セーフティクリティカルシステムのプロダクトラインにおける N バージョンプログラミングの導入」, 電子情報通信学会ソフトウェアサイエンス研究会, 2010 年 3 月 8 日, 鹿児島県鹿児島市.

⑦ 中西 恒夫, 久住 憲嗣, 福田 晃,「ソフトウェアアーキテクチャ事前設計を目的とするフィーチャモデリングのアンチパターン」, 組込みシステムシンポジウム 2009 (ESS2009), 2009 年 10 月 23 日, 東京都渋谷区.

⑧ 中西 恒夫, 久住 憲嗣, 福田 晃,「ソフトウェアアーキテクチャ事前設計を目的とするフィーチャモデルのガイドラインとアンチパターン」, 電子情報通信学会ソフトウェアサイエンス研究会, 2009 年

8 月 7 日，北海道北見市.

⑨ 蓮尾 雅昭，中西 恒夫，田頭 茂明，荒川 豊，久住 憲嗣，福田 晃，「フィーチャモデル自動描画アルゴリズムの提案と実装」，電子情報通信学会ソフトウェアサイエンス研究会，2009 年 5 月 14 日，秋田県秋田市.

６．研究組織

(1)研究代表者

中西 恒夫（NAKANISHI TSUNEO）

九州大学・大学院システム情報科学研究院・准教授

研究者番号：70311785

(2)研究分担者

なし

(3)連携研究者

なし