様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２８ 年　 ５ 月 ３１ 日現在

機関番号：３２６６５
研究種目：挑戦的萌芽研究
研究期間：2013～2015
課題番号：２５５４００２７

研究課題名（和文）異常が発生しても実行を継続できる柔らかなソフトウェアの研究

研究課題名（英文）Research on the Flexible Software Architecture for Degradable Execution on Runtime Failures

研究代表者
杉山　安洋（SUGIYAMA, Yasuhiro）

日本大学・工学部・教授

研究者番号：７０２４６８４１

交付決定額（研究期間全体）：（直接経費）　　2,000,000円

研究成果の概要（和文）：システムの一部の部品に異常が発生した場合でも，システム全体の異常終了を防ぎ，システムの機能を最大限維持しながら実行を継続できる柔らかなソフトウェアシステムの実現に向けて研究を行った．オブジェクトの仮想化に基づく柔らかなソフトウェアのモデルを構築し，不良の発生したソフトウェアを縮退実行するために必要な，不良隔離技術と不良範囲診断技術を確立した．さらに，Java言語で記述された既存システムを縮退実行させるための実行環境の試作を行ない，これまでに開発してきた技術の有効性の検証を行った．

研究成果の概要（英文）：A research on the flexible software architecture in order to realize the degradable execution on runtime failures has been conducted. A flexible software model for degradable execution based on the object virtualization has been built. A defect isolation mechanism and a defect impact analysis mechanism to detect objects that have relation to the defective objects have been designed. A runtime environment that allows degradable execution of exiting software systems written in Java has been built and evaluated.

研究分野：ソフトウェア工学

キーワード：ソフトウェア　高可用性　縮退実行

１．研究開始当初の背景

社会のインフラとなるような情報システムは，一時的な停止であっても社会生活に大きな影響を及ぼす．最近でも，鉄道や航空会社のシステムがダウンし，社会生活に大きな影響を与えたことは記憶に新しい．そのため，システムに不良が発見された場合でも，異常停止することなく実行が継続でき，その間に不良部分の修復ができることが望ましい．これまでも，ハードウェアが故障してもソフトウェアの実行を継続できる高可用性ソフトウェアの研究や，正常に動作しているソフトウェアの機能変更が柔軟に行えるソフトウェア発展や自己適応システムの研究などが行われてきた．研究代表者も高可用性ミドルウェアや実行時のソフトウェア発展機構機構を持つ Java 仮想マシンを研究してきた．しかし，どの研究も不具合が発生したソフトウェアの実行を継続することを目的とはしていなかった．

２．研究の目的

ソフトウェアシステムは多くの部品の組み合わせで構成される．システムの一部の部品に異常が発生した場合，図１に示すように，異常の発生した部品は使わずに，正常な部品のみで実行が継続できれば，そのシステムの機能を部分的にではあるが継続して提供でき，使用者の利便性の低下を押さえることができる．このように，すべての部品を使うのではなく，一部の部品のみで実行を継続することを縮退実行と呼ぶ．

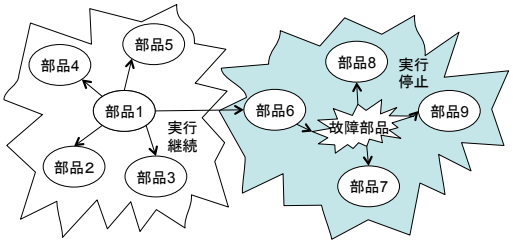

図 1：縮退実行

本研究では，想定外の異常が発生した場合でも，異常部品とその影響を受ける部品を切り離し，正常部品のみでシステムを縮退実行させ，その間に不良部品を修正し，再度システム全体の機能を復元することが可能な柔軟なソフトウェアを実現する．しかも，対象とするソフトウェア自身には，縮退実行機能を個別に組み込むことなく実現できる手法の実現を目指す．

３．研究の方法

(1)　柔らかなソフトウェアモデルの構築

まず，システムの一部の部品に異常が発生した場合でも，システム全体の異常終了を防ぎ，システムの機能を最大限に維持しつつ実行を継続することができる，柔らかなソフトウェアのモデルを構築する．

これまでのソフトウェア工学では，工業的な手法でソフトウェアを開発することによって，ソフトウェアの品質や開発効率の向上に貢献してきた．いわば，ソフトウェアを固いハードウェアと同じ手法で開発しようという考え方である．しかし，近年になって，ソフトウェアに柔軟性がなく，ソフトウェアの修正や機能変更などが難しい問題が多く指摘されるようになった．

縮退実行についても同様で，現状のモデルでは，システムの一部の部品だけを用いて実行可能なソフトウェアを構築することは難しかった．縮退実行のためには，不良部品の切り離しや，一時的な代替部品への切り替え，修復の完了した部品の再接続などが必要で，これらはすべて部品間の柔軟な接続性を必要とする．一度接続してしまうと接続替えができないモデルだと，これらの機能は実現できない．

例えば，現在主流のオブジェクト指向開発法では，ソフトウェアを異なる機能を持つ部品であるオブジェクトに分割し，それらを協調実行させる．オブジェクトは，関連した機能の集合体として完結した構造を持ち，異なるオブジェクト間の接続インタフェースも厳密に定義されている．厳密な構造とインタフェースは，ソフトウェア開発コストの低減には効果があるが，その構成要素のひとつでも機能しなくなると，関連する機能も実行を継続することができなくなる．そのため，システムに異常が発生した場合には，ソフトウェアの実行を継続することは，難しかった．

そこで，本研究では，部品をより緩やかに結合してソフトウェアを構成する，柔らかなソフトウェアモデルを構築することにより，縮退実行可能なソフトウェアの構成法を実現する．柔らかなソフトウェアモデルの特長は，システム中の部品の結合度が低いことと，部品の入れ替えが容易であることの２点にある．

(2)　柔らかなソフトウェアモデル実現のための基本機能の解明

次に，柔らかなソフトウェアモデルをもとに開発されたソフトウェアを縮退実行させるために必要な実行時の基本メカニズムの解明を行う．

①　不良隔離機構

システム中の一部の部品に異常が発生した場合でも，システム全体を異常終了させないためには，発生した不良を隔離する必要がある．隔離には以下の２つのケースが考えられる．

通常は，システム中の一部の部品に想定外の異常が発生すると，それがシステム全体へ伝播し，異常停止へとつながる．従って，異常の発生を検知し，その伝播を必要最小限の範囲に押さえておき，その異常によりシステム全体が異常停止することを防ぐ機構が必

要となる．
　また，不良の発生した部品を使用しないでシステムの実行を継続するためには，その部品を使用する必要が発生するような処理の実行を抑制する必要がある．例えば，ユーザの選択によって実行が開始される場合は，ユーザがそのような機能を選択できないようにする必要がある．

②　不良範囲診断技術
　一部の部品に不良が発生した場合に，その使用を停止するわけであるが，問題のある部品だけを使用停止するだけでは十分ではなく，その部品に関連する他の部品の使用も取りやめる必要がでてくる．そのため，使用を停止した部品と共に実行を停止する部品の範囲を決定できる機構が必要となる．

③　不良の可視化技術
　システム中の一部の部品で不良が発生した際には，システム中のどの部分で不良が発生したのか，あるいは，その影響部分がどこなのかという情報を可視化できると，縮退実行の実施や，不良の修正に有効である．そのため，システム中の部品の状況を可視化する機構を検討する．

４．研究成果
(1)　オブジェクトの仮想化による柔らかなソフトウェアモデル
　本研究における柔らかなソフトウェアモデルの実現方針は，オブジェクトの仮想化である．現代のソフトウェアは数多くのオブジェクトと呼ばれる部品を組み合わせることにより実現されている．通常のシステムでは，これらのオブジェクトが直接的に相互接続し，協調しながら必要な処理を実行する．これが，一部の部品の異常がシステム全体の停止を招いたり，縮退実行をさせることが難しい原因であった．直接的に接続してしまうため，接続先の変更や，接続済みのオブジェクトの書き換えなどを行うことが大変難しくなっていた．

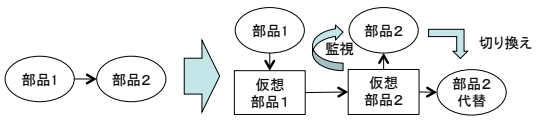

図 2：オブジェクトの仮想化

　本研究で実現したオブジェクトの仮想化とは，システム中のオブジェクトを直接的に相互接続するのではなく，間接的に接続する．これまでのモデルでは図２の左側に示すように，部品であるオブジェクトが直接接続している．仮想化においては，図２の右側のように，オブジェクトごとに，そのオブジェクトの外部インタフェースとなる仮想オブジェクトを生成する．オブジェクトへのアクセスは，その仮想オブジェクトを経由して行われ，オブジェクトは，他のオブジェクトから直接アクセスされることはない．
　これにより，次に示す３つの効果が得られた．まず，異常の発生したオブジェクトをシステム全体から切り離すことが可能となった．オブジェクトへの処理の依頼は，仮想オブジェクト経由で行われるため，仮想オブジェクトが，その処理の委譲の可否を判断できるためである．
　また，外部インタフェースを変更することなく，内部処理が異なるオブジェクトとの入れ替えが可能となった．また，元のオブジェクトには無かった機能や能力を必要に応じて追加することもできるようになった．これにより，異常の発生したオブジェクトを代替部品と交換することもできるようになった．これも，仮想オブジェクトにより，処理の委譲先を自由に変えられるようになったためである．
　さらに，オブジェクトの内部の状態の変化や処理の状況を継続的に観察することにより，オブジェクトの健全性を検証することができるようになった．

(2)　Java 言語を対象とした柔らかなソフトウェアモデル実現のための基本機能の解明
①　不良隔離手法
　一般のソフトウェアは，不良が一部の部品で発生すると，その影響がシステム全体に及び異常終了してしまう．異常終了を防ぐためには，不良の発生を検出するだけでなく，その影響をシステムの他の部位に及ぼさないように隔離する仕組みが必要である．
　本研究では，仮想化されたオブジェクトを個別に独立して実行することにより，異常の発生の検出と，発生した異常を隔離する手法を確立した．あるオブジェクトで発生した異常は，そのオブジェクトの仮想オブジェクトが処理する方式である．
　また，異常の発生したオブジェクトを使用しないようにする方式としても，仮想オブジェクトを活用した．使用する側のオブジェクトの仮想オブジェクトに，不良オブジェクトの情報を伝えておき，不良オブジェクトを呼び出すような処理を行わないようにした．

②　不良範囲診断技術
　システムの一部の部品に不良が発生した場合には，不良部品のみならず，その不良が影響を及ぼす範囲を特定する必要がある．しかも，健全な部品をできる限り不良範囲に含めないことが重要である．本研究では，次の２通りの不良範囲診断技術を開発した．
　第一は，システム中の部品の静的な依存関係を求める手法である．ソースコードを解析し，依存関係をグラフ化することにより，特定の部品に依存する部品の範囲を検出する．
　第二は，システム中の部品の動的な依存関

係を求める手法である．システム中の部品の依存関係は，実行中のシステムの状態や，その使用状況などによって，日々刻々と変化する．例えば，同一の機能を利用する場合であっても，利用するユーザごとに使用する部品が異なることもあれば，利用時刻によって異なった部品が使用される可能性がある．さらには，実行してみなければ判定できないような依存関係も存在する．このため，実行中のオブジェクトの内部状態を考慮したプログラムの仮想実行手法を考案し，さらに，仮想実行の結果を用いた不良の波及解析手法を考案した．

静的な依存解析手法は，実行に先立って解析を行うことができるため，実行時のオーバヘッドが少ないという利点はある．しかし，実行時の情報が無いと，不良部品の影響範囲を必要以上に大きく見積もってしまうという欠点がある．そこで，静的解析で確定できる範囲までは静的解析で解析しておき，確定できない部分については動的依存解析を行う手法を確立した．

③　Java仮想マシン中のオブジェクトの可視化手法

システムの実行を開始すると，その状況に応じて，部品であるクラスがメモリ中にロードされ，そのオブジェクトが生成されて実行される．従って，ある時点で，どのようなクラスがメモリ中にロードされているか，また，どのようなオブジェクトが生成されて使われているのかを把握することは重要である．

そこで，Java 仮想マシンにクラスのロード状況やオブジェクトの生成状況を把握できる仕組みを組込み，その情報をもとにオブジェクトを可視化できる手法を考案した．

(3)　検証

Java 言語で記述された既存システムを縮退実行させるための実行環境の試作を行ない，これまでに開発してきた技術の有効性の検証を行った．

一般の実行環境では，実行中のシステムに異常が発生すると，その異常に対する例外処理が予め実行中のシステム自身に組み込まれていない限り，システム全体が異常終了してしまう．本研究では，想定外の異常が発生しても，その異常を検知し，その異常を切り離し，その異常に影響を受けない部分のみで，システムの縮退実行を行える実行環境を試作した．縮退実行環境の実現方式は，任意のシステムの縮退実行を可能とする汎用的な実行環境を開発するのではなく，異常が発生した場合でも縮退実行させたいシステムを解析し，その解析結果にもとづいて縮退実行を可能とする実行環境を生成する方式とした．

縮退実行対象のシステムを解析することにより，当該システム中のオブジェクトを仮想化し，異常発生の検出と不良の隔離を可能とした．生成する仮想オブジェクトには，オブジェクトの切り替え機能や，監視機能を実装した．

さらに，解析結果から縮退実行対象のシステムの状態マシンを生成し，不良発生時に実オブジェクトの内部状態にもとづいて状態マシンを仮想実行することにより，不良の影響範囲の診断を可能とした．以上により，実行を抑止する範囲を絞り込んで縮退実行を行うことが可能となった．

検証結果としては，仮想化による実行時のオーバヘッドや，仮想実行による副作用が検出されたが，全体としてほぼ予想した結果が得られた．

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）

〔雑誌論文〕（計６件）

① 小野寺駿一，名倉正剛，杉山安洋，動的情報に基づくソフトウェア縮退実行機構の提案，レクチャノート／ソフトウェア学，査読有，vol.41，pp.31-40，2015

② 杉山安洋，無形労働としてのソフトウェア開発に関する一考察，レクチャノート／ソフトウェア学，査読無，vol.40，pp.247-248，2015

③ 高橋克幸，杉山安洋，インクリメンタル開発のための Java クラスの簡易実行ツール，レクチャノート／ソフトウェア学，査読有，vol.40，pp.129-134，2014

④ 渡部聡，杉山安洋，不具合の発生したソフトウェアの実行を継続する一手法の提案，電子情報通信学会技術研究報告，査読無，113(160)，pp.25-30，2013.

〔学会発表〕（計７件）

① 高橋克幸，杉山安洋，Java クラスを手軽にテストするための簡易実行ツールの開発，情報処理学会東北支部研究会，2014 年 3 月 12 日，山形大学工学部

② 渡部聡，杉山安洋，ソフトウェアの縮退実行の実現に向けたプログラム依存グラフの考案，第 56 回日本大学工学部学術研究報告会，2013 年 12 月 14 日，日本大学工学部

６．研究組織

(1)研究代表者

杉山　安洋　（SUGIYAMA, Yasuhiro）

日本大学・工学部・教授

研究者番号：70246841