The University of Osaka
Institutional Knowledge Archive

Title	Construction of Multiversion Software and Its Reliability Evaluation
Author(s)	島, 和之
Citation	大阪大学, 1997, 博士論文
Version Type	
URL	https://hdl.handle.net/11094/40381
rights	
Note	著者からインターネット公開の許諾が得られていないため、論文の要旨のみを公開しています。全文のご利用をご希望の場合は、大阪大学の博士論文についてをご参照ください。

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

【 80 】

氏　　名	島（しま）　和（かず）　之（ゆき）
博士の専攻分野の名称	博　士（工　学）
学位記番号	第　12828　号
学位授与年月日	平成9年2月20日
学位授与の要件	学位規則第4条第2項該当
学位論文名	Construction of Multiversion Software and Its Reliability Evaluation （マルチバージョンソフトウェアの構成手法とその信頼性評価）
論文審査委員	（主査） 教授　井上　克郎 （副査） 教授　都倉　信樹　教授　菊野　亨　講師　楠本　真二 奈良先端科学技術大学院大学教授　鳥居　宏次

論文内容の要旨

コンピュータシステムの応用分野が広がり，その役割が重要になればなるほど，その故障が社会に及ぼす影響が大きくなってくる。例えば，鉄道・原子炉・航空機・医療機器などの制御システムに故障が起きると，人命に危害の及ぶ可能性がある。また，航空会社の座席予約システム，証券会社の契約システム，製造業の生産ライン制御システムといった企業の業務に関わるコンピュータシステムがダウンすれば，その企業は多大な損害を被る。このようなシステムでは，システムの耐故障性（システムの一部に欠陥があったとしても，システムが故障しにくい性質）が重要になってくる。

ソフトウェアの耐故障性を実現する方法の1つとして，マルチバージョンプログラミングがある。マルチバージョンプログラミングでは，複数のバージョン（同じ要求仕様に従って複数のチームが独立に開発したプログラム）とドライバ（バージョンを制御するプログラム）を用意する。ドライバはシステム外部からの入力を各バージョンに与え，それらの結果の多数決を行い，最大多数となった結果をシステム外部に出力する。このため，少数のバージョンが誤った結果を出しても，多数のバージョンが出した正しい結果がシステム外部に出力されるので，外部からはシステムが故障していないように見える。

マルチバージョンプログラミングによって開発されたソフトウェアをマルチバージョンソフトウェアと呼ぶ。バージョンの数を増やすことによってマルチバージョンソフトウェアの信頼性は向上するが，開発コストも増大する。バージョンの数を増やさずに，マルチバージョンソフトウェアの信頼性を向上させる方法としてコミュニティエラーリカバリ（Community Error Recovery, CER）が提案されている。コミュニティエラーリカバリでは，チェックポイントにおいてバージョンの中間結果を集め，多数決を行う。多数決で，少数派となったバージョンの中間結果を多数派のバージョンの中間結果に置き換える。バージョンの誤った中間結果が正しい中間結果に修正されるので，バージョンの信頼性が向上するとともにマルチバージョンソフトウェアの信頼性も向上する。コミュニティエラーリカバリを用いたマルチバージョンソフトウェアでは，チェックポイントの数の増加にともなってソフトウェアの信頼性は向上すると考えられる。しかし，チェックポイントにおいて誤りが起こり得ると仮定すると，コミュニティエラーリカバ

リはマルチバージョンソフトウェアの信頼性を逆に悪化させることもあるという指摘がなされている。

本論文では，バージョンの数を増やさずにマルチバージョンソフトウェアの信頼性を向上させる方法としてソフトウェア増殖法 (Software Breeding) を提案する。この方法では，バージョンは共通のモジュール仕様に従って独立に開発された複数のモジュールから構成されるものとする。コミュニティエラーリカバリでは中間結果を置き換えることに対して，ソフトウェア増殖法ではモジュールを置き換える。ソフトウェア増殖法では，バージョン中にチェックポイントを挿入しないため，チェックポイントによる故障を考慮する必要がない。

また，本論文では，ソフトウェア増殖法のリアルタイムシステムへの適用方法を提案する。提案する方法では，Nバージョンプログラミングとソフトウェア増殖法の両方を用いる。これにより，入力から出力までの時間はNバージョンプログラミングと同等になる。提案する方法を用いた場合のソフトウェアの信頼性を評価するために故障シミュレーションを行った。その結果，Nバージョンプログラミングのみを用いた場合に比べて故障する頻度が少なくなることが確認された。

論文審査の結果の要旨

本論文は，ソフトウェアフォールトトレランスの一つのアプローチであるマルチバージョンソフトウェアに着目し，開発するバージョンの数を増やさずにマルチバージョンソフトウェアの信頼性を向上させるための方法として，ソフトウェア増殖法を提案し，提案した方法を用いたソフトウェアの信頼性を評価した研究をまとめたものである。

バージョンの数を増やさずに，マルチバージョンソフトウェアの信頼性を向上させる方法としてコミュニティエラーリカバリ（Community Error Recovery, CER）が提案されている。コミュニティエラーリカバリでは，バージョン中に複数のチェックポイントを設定し，チェックポイントにおいてバージョンの誤りを検出し修復する。しかし，チェックポイントの故障によって，コミュニティエラーリカバリは信頼性を悪化させることもある。

ソフトウェア増殖法では，共通のモジュール仕様に従って開発された複数のバージョンとそれらのバージョンを制御するドライバを用意する。ドライバはバージョンの故障を検出すると，欠陥を含むモジュールを特定し，それを他のモジュールに置き換える。コミュニティエラーリカバリでは中間結果を置き換えることに対して，ソフトウェア増殖法ではモジュールを置き換える。ソフトウェア増殖法では，バージョン中にチェックポイントを挿入しないため，チェックポイント故障が起こらない。ソフトウェア増殖法を用いたマルチバージョンソフトウェアの信頼性をモデル化し，コミュニティエラーリカバリを用いたマルチバージョンソフトウェアとの比較を行った結果，ソフトウェア増殖法はコミュニティエラーリカバリよりも高い信頼性を達成できることが示されている。

また，本論文では，ソフトウェア増殖法のリアルタイムシステムへの適用方法を提案している。その方法では，Nバージョンプログラミングとソフトウェア増殖法の両方を用いる。これにより，入力から出力までの時間はNバージョンプログラミングと同等になる。Nバージョンプログラミングのみを用いたソフトウェアと提案している方法を用いたソフトウェアの故障をシミュレーションし，提案している方法を用いることによって減少する故障の割合を示している。

以上の研究成果は，マルチバージョンソフトウェアの高信頼化技術の発展に貢献しており，本論文は博士（工学）論文として価値あるものと認める。