

論文 / 著書情報
Article / Book Information

題目(和文)	異種システムを共存させるゲートウェイを用いたシステムの継続的移行に関する研究
Title(English)	
著者(和文)	豊内順一
Author(English)	Junichi Toyouchi
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第11115号, 授与年月日:2019年3月26日, 学位の種別:課程博士, 審査員:出口 弘,山村 雅幸,三宅 美博,小野 功,石井 秀明,吉川 厚,寺野 隆雄
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第11115号, Conferred date:2019/3/26, Degree Type:Course doctor, Examiner:,,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

異種システムを共存させるゲートウェイを用いたシステムの継続的移行に関する研究

総合理工学研究科

知能システム科学専攻

豊内 順一

概要

企業を取りまく経営環境や市場が刻々と変化する状況では、企業が保有する情報制御システムは、変化へのより柔軟な対応と高いロバストネスが求められる。例えば、グローバルでの激しい競争にさらされている製造業の場合には、少品種大量生産から多品種少量生産への移行を余儀なくされている。生産管理システムが扱う生産品の種類や生産量は短い周期で変動し、頻繁な装置替えやテストが発生するが、それらに伴って起きる、生産設備の部分的な停止や故障などの異常状況の工程全体への影響は最小限に抑えることが期待される。

1980年代まで、生産管理システムはトップダウンに制御を行う、集中型システムが主流であった。集中型のメリットは、システム全体の状態を把握することで、製造計画の立案や各装置へのタスク割振りの最適化を図ることが可能な点にある。しかし、システムの大規模化に伴い、システム全体の情報収集が困難になり、またシステムの一部に停止や故障が発生する確率も上がる。このため、システムの階層化や機能的な分割が行われるようになり、徐々にシステムの分散化が行われるようになった。さらに、現場の管理装置（エッジ）の高機能化によって、各官吏装置がローカルな情報を収集し、タスク割振りを自律的に行う方式が有効となってきた。この方式は計算負荷の分散に加え、故障の際にシステム停止を局所化でき、ロバストネスの向上も期待できる。

本研究では、情報制御システムのアーキテクチャ、およびシステム内の機器・装置やソフトウェアの構成と接続関係の変更などを、システムを全停止することなく、現場の技術者が柔軟かつ容易に行える方式を提案する。現行システムから新システムへと移行させる作業は、従来は広範囲かつ長期間のシステム停止を要し、システム設計と開発の専門家であるシステムエンジニア（SE）による、綿密な移行計画書の作成やリハーサルなどの事前準備が必須だった。また、実際の移行作業や移行後のテストもSE中心に進められ、システムを利用するユーザのみで行うことは困難であった。本研究では、システムの移行やテストなどの作業を、現場の状況に応じて任意のタイミングで行えることを「柔軟」、システムを利用する現場のユーザのみで、SEや事前の準備に頼らず行えることを「容易」とそれぞれ定義する。

「柔軟」かつ「容易」なシステム移行とテストを実現するために、異種アーキテクチャのシステムの共存と連携、システム間連携テスト、およびシステム機能評価を行う技術を開発し、実システムへの適用と評価を行った。具体的には製造業の生産管理システムを事例とし、集中型の生産管理システムの持つ機能の分散型システムでの実現、および集中型から分散型へのシステム移行とテストを実施し、評価を行った。

本研究の実施にあたり、集中型の生産管理システムの機能、特に資源割り当て機能の分散型システムでの実現方式を設計・実装したのち、次の3ステップで行った。

まず、異種システムを共存させるために、従来のシステム間連携ゲートウェイを拡張することで、集中型と自律分散型のシステムを共存、稼働させる方式を設計した。この方式を実装し、生産管理システムで適用可能なこと確認した。

続いて、共存状態における部分的、継続的な移行と統合テストを行うため、集中型から自律分散型に段階的に移行する手順を設計した。また、既存のオンラインテスト方式を拡張することで、新規部分の機能、および既存部分との連携をオンライン状態でテスト出来る方式を設計した。この手順とテスト方式を上記の生産管理システムに適用することで、サブシステム単位での段階的な移行と、異種共存システムのオンライン連携テストとを実現した。

最後に、システムの機能を検証するテストの基準とテスト用データの容易な作成を行うために、機能検証のためのシステムのモデルと検査したい条件の定義手順、及びテストケースの生成フレームワークを設計した。提案した手順に従い、生産ライン管理システムのモデル化と検査条件の定義を行い、フレームワークを用いて満たすべきテストケースを生成した。また、モデル仕様記述言語によって、生産ライン管理システムとその振舞い、およびテストケースに対応する検査条件を記述し、現場の技術者が現場の状況を観察してパラメータを設定すれば、モデル検査ツールを用いて検査条件を満たすテストデータの自動生成を可能とする方式を設計し、実装した。

本研究の結果、以下の3点を結論とする。

異種アーキテクチャを持つシステムへの移行、特に集中型から自律分散型に移行する際に、(1)集中型システムの機能を自律分散型システムで実現すると同時に、障害や変更へのロバストネスを高めることが、可能である。(2)システム連携ゲートウェイを用いた異種システム共存状態を構築し、その状態でオンラインテストを行うことで、システム全体を停止することなく、段階的かつ継続的なシステム移行が可能である。(3)システムのモデルと検査したい条件の定義手順、及びテストケースの生成フレームワークを用いることで、満たすべきテストケースを生成し、モデル検査ツールを用いて検査条件を満たす、テストデータの生成が容易に行える手順を得た。

本研究の貢献は、SEによる事前の膨大な準備を要し、長期の工場休止時期のみに実施できる移行手順ではなく、現場の技術者が平常時の工場稼働状態で、システムの段階的な移行とオンライン状態での連携テストを行い、またシステム機能の検査を行うテストデータの作成も容易に行える継続的なシステム移行のための手順と方式の体系を提案し、その有効性を確認したことにある。さらに、システム移行とテストに要する総時間の短縮、影響範囲の限定化を確認した。

目次

第1章 序論	1
1. 1 はじめに.....	1
1. 2 背景.....	1
1. 3 目的と構成.....	3
第2章 情報制御システムの移行に関する課題と研究の位置づけ	10
2. 1 情報制御システムの動向とシステム移行の課題.....	10
2. 1. 1 情報制御システムとは.....	10
2. 1. 2 情報制御システムへのニーズ	10
2. 2 本研究で取り扱う範囲と課題.....	12
2. 3 先行研究の調査.....	14
2. 3. 1 自律分散システム.....	14
2. 3. 2 本研究で対象とする分野の先行研究.....	17
第3章 異なるアーキテクチャを持つシステムの共存	26
3. 1 はじめに.....	26
3. 2 提案方式.....	30
3. 3 適用結果.....	32
3. 4 考察.....	32
3. 5 おわりに.....	33
第4章 共存システムの継続的移行と統合テスト	35
4. 1 はじめに.....	35
4. 2 提案方式.....	35
4. 3 適用結果.....	37
4. 4 考察.....	38
4. 5 おわりに.....	38

第5章 システム機能を検証するテスト基準、及びテスト用データの容易な作成....	40
5. 1 はじめに.....	40
5. 2 提案方式.....	40
5. 3 適用結果.....	41
5. 4 考察.....	44
5. 5 おわりに.....	44
第6章 結論と課題.....	45
6. 1 結論.....	45
6. 2 今後の課題.....	46
謝辞	47
文献	48
研究業績一覧.....	56

第1章 序論

1.1 はじめに

本章では研究の概要を述べて、研究の対象範囲を明確化するとともに、論文の構成を説明する。

情報制御システムを更新する場合、既存のシステムを廃して、全く新しいシステムを構築して置き換えるケースは多くない。既存システムの機能や構成要素を可能な限り維持しつつ、新システムへと再構築するケースがほとんどである。既存システムから、新システムへと移行するための長期にわたる過程において、特に図 1.1 に示すように移行作業とテストの範囲を本研究の対象と定める。

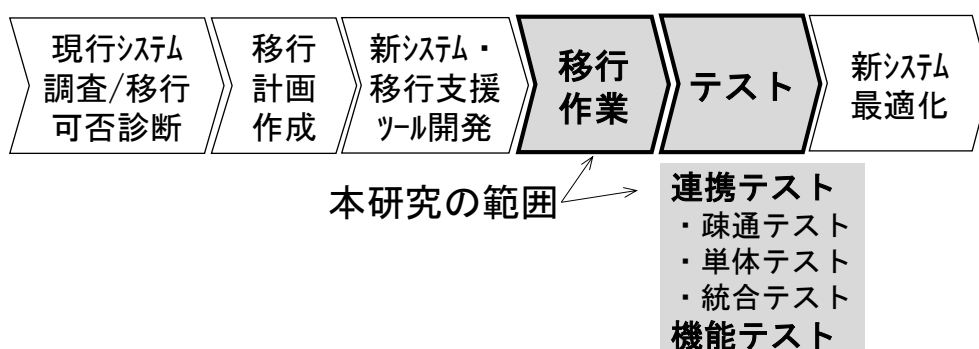


図 1.1 研究の範囲

1.2 背景

情報制御システムは、プラントや鉄道などにおける設備や機器を制御して、操業や運転を計画に沿って円滑に行うとともに、製造や運行の計画立案や実績の管理、システムの利用者に各種情報の提供などを担うためのシステムである。

1980年代に入って、デジタル計算機の高性能化、低価格化を受けて、それまで集中型が中心だったシステムの階層化や機能的な分割が行われるようになり、徐々にシステムの分散化が進展した。

2000年代に入って、クラウド上で企業の情報を集中的に管理するケースが増え、近年では人工知能技術とビッグデータ分析技術の進化に伴い、企業の経営上の意思決定や大規模な SCM (Supply Chain Management) などを行うことが一般的になってきた。また、IoT (Internet of Things) 技術の普及は、現場の機器やセンサからリアルタイムで大量のデータを取得し分析することで、現場の「見える化」を加速している。しかし、その一方で現場から得られるデータ量は急速に増大し、データ伝送速度や容量の制約から、全てをクラウドに転送することは困難となり、現場に近い場所でのデータの管理と処理を分散的に行うことが不可欠となっている。また、事業環境やシステムへの要件の変化に応じて、システムの変更に動的に行うことも求められている。現在では、集中処理と分散処理をいかに柔

軟に配置し、連携させるかが課題である。

稼働中の情報制御システムの再構築に伴う、現行システムから新システムへの移行作業には、通常はシステム設計と開発の専門家であるシステムエンジニア（SE）を主体とする長い準備期間を要する。この事前準備には、現行システムの調査と分析、新システムの構想、設計、および新規要素と移行支援ツールなどの開発、移行計画書の作成や移行作業のリハーサルなどが含まれ、これらの作業をシステムを利用するユーザのみで行うことは困難であった。また、本番の移行作業や移行後のテストもSE中心に進められ、その際には、広範囲かつ長期間のシステム停止を要することが一般的であった。

しかし、社会インフラ系の管理システムをはじめとして、情報制御システムには全停止をしないことが前提のものが多い。インフラ系以外でも、例えば工場の生産管理システムも、生産の高効率化やコスト低減のため、多くは24時間連続稼働を原則として設計、運用されている。国内においては、全停止できるのは、慣習的に盆休みと正月休みの数日間のみという工場も多いため、この期間を狙って大規模なシステム移行が計画される。チャンスが限られるだけに、移行作業を確実に成功させるため、大きなコストと長い期間をかけてSE主導での移行作業が進められる。

ところが、企業を取りまく経営環境や市場が刻々と変化する状況においては、年に数回しか規模の大きいシステム移行が出来ないのでは、事業機会を失いかねない。システムへの要件の変化に応じて、情報系から制御系までの機器やソフトウェアの更新や構成の変更などを、迅速に行うことが求められている。また、その場合にもシステム全体を止めることなく、既設部は動かし続けたいという要求は強い。さらに、小規模の移行やテストであってもSE中心でなければ行えない方式では、通常業務の合間にタイミングを見計らって作業を行うような、柔軟性や即時性は期待できない。特に製造業では、グローバルでの競争が激化し、多品種少量生産への要求が高まっている。生産の現場では、頻繁に装置替えが発生し、その都度、調整やテストを行う必要がある。多品種の製品を製造するために、セル型の分散型自律生産環境を用いて、柔軟な生産管理を行うアプローチもなされているが(杉村 1994) (Okubo 2000)、本研究で扱うようなタイヤ製造の場合は、少量といいつつも数百個を生産する必要があり、ライン型の生産環境での課題に対応する必要がある。

さらに、SEの大型のシステムの開発や刷新などへの需要や、従業員の高齢化やともあいまって、SE人材は大幅に不足している。そのため、現場では様々な作業を「レス スキルド（或いはスキルレス）」という熟練SEでなくとも実施可能な手順に移行することへの要望は高い(奥 1996)。すなわち現場の技術者でも、移行作業やテストを容易に行える仕組みが求められている。

これらのニーズに応えるため、情報制御システムのアーキテクチャ、およびシステム内の機器・装置やソフトウェアの構成と接続関係の変更などを、システムを全停止することなく、現場の技術者が柔軟かつ容易に行える方式を提案する。本研究では、システムの移行やテストなどの作業を、現場の状況に応じて任意のタイミングで行えることを

「柔軟」、システムを利用する現場のユーザのみで、SEや事前の準備に頼らず行えることを「容易」とそれぞれ定義する。

システム移行のケースを大きく分類すると、対象のシステムが情報系か制御系かという項目と、移行する新システムのアーキテクチャが既存システムと同じか異なるかという項目とがある（表 1.1）。

表 1.1 システム移行ケースの分類と既存の技術

移行先システム システム種別	同アーキテクチャ	異種 アーキテクチャ
情報システム	- エミュレータ - 仮想マシン	- 仮想マシン - クラウド
制御システム	- 互換ハードウェア - 互換ソフトウェア	本論文で提案する 移行とテストの手法

情報システムの場合には、エミュレータや仮想マシンを用いることで、既存のプログラムを変更することなく、システム移行を行う取り組みがなされてきた。異種アーキテクチャのシステムへ移行する場合も、仮想マシンやクラウドを用いることで、実装レベルでのアーキテクチャの違いを意識する必要はなくなっている（岡 2013）。

制御システムでは、リアルタイム性や信頼性への要求が高く、またベンダ独自のハードウェアやソフトウェアを用いる場合も多いため、新システムも同じアーキテクチャのシステムに移行するケースが多い。そのため、制御用の計算機やソフトウェアのベンダは継承性を意識し、既存システムのソフトウェア資産がそのまま再利用できる互換性を持つように、新製品の設計、開発を行い、また移行のためのガイドラインやツールなどを提供することが一般的である（Brisky 2007）（Rockwell Automation 2016）。

しかし、制御システムを異なるアーキテクチャのシステムに移行するケースには、先行的な研究やベンダのソリューションを見出すことができない。本研究では、既存の異種システムの連携技術やオンラインテストの方式を拡張することにより、このケースにおける移行とテストの課題をどのように解決するかを明らかにする。さらには、現場の技術者が移行やテストを容易に行える仕組みを提案する。

1. 3 目的と構成

本研究の目的は、図 1.2 に示すように、情報制御システムのアーキテクチャおよび、システム内の機器・装置やソフトウェアの構成と接続関係の変更などを、システムを全停止することなく、現場の技術者が柔軟かつ容易に行えるようにすることである。

目的を具体化した目標は、異なるアーキテクチャを持つシステムの共存と連携、システム

間連携テスト、およびシステム機能の検証を行う技術の開発である。この目標を体系的に達成するために、本研究では図 1.3 で示すシステム移行フレームワークを提案するとともに、このフレームワークに沿った技術開発を行う。

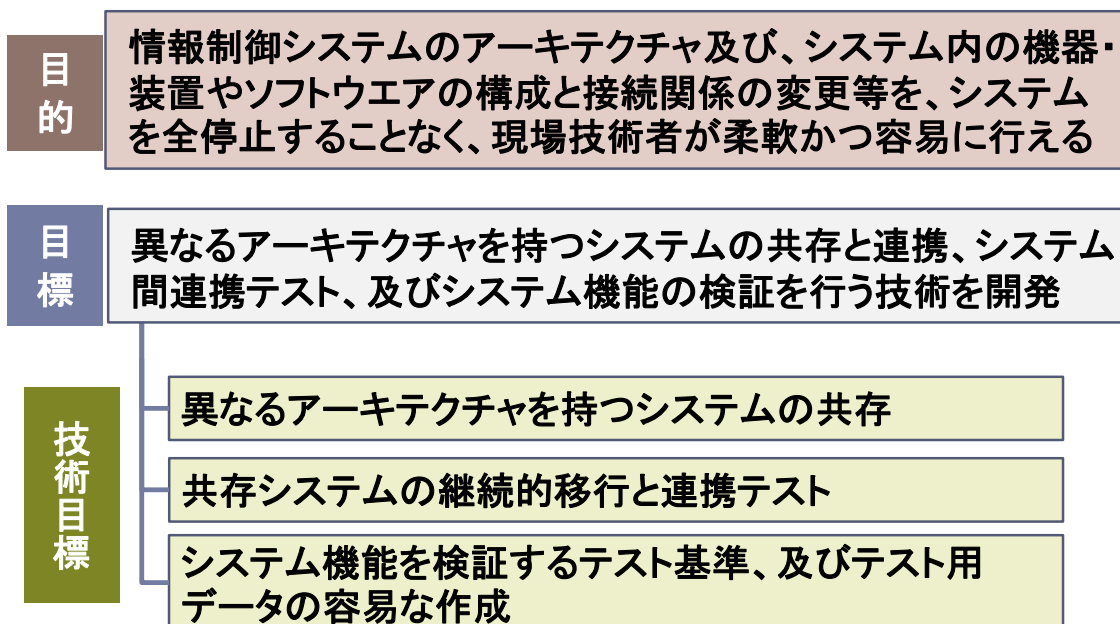


図 1.2 研究の範囲

本稿では、図 1.3 に示す通り、既存システムを集中型システム、新規システムを自律分散型システムと想定して、課題の検討や方式の提案、実システムへの適用と評価を行っている。ここで扱う、自律分散型システムとは、システム全体を統合する中枢機能を持たず、ローカルで得られる情報を用いて自律的に判断・動作する複数のシステム要素（サブシステム）の相互作用によって、全体として機能するシステムのことである。

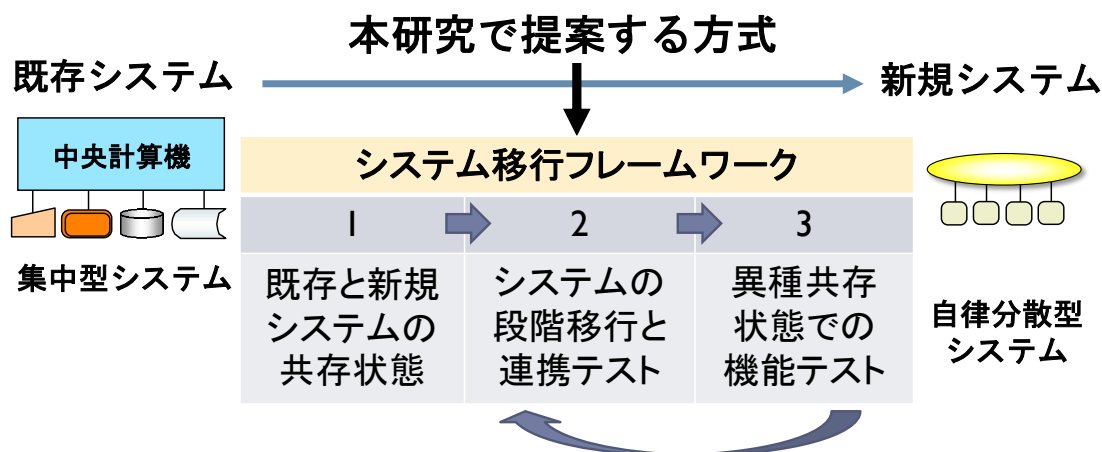


図 1.3 システム移行フレームワーク

本章以降の構成を述べる。図 1.2 で示した目標を達成するために、フレームワークに沿って3つの技術課題を設定し、以下の3つの章、第3章：異なるアーキテクチャを持つシステムの共存、第4章：共存システムの継続的移行と統合テスト、第5章：システム機能を検証するテスト基準、及びテスト用データの容易な作成、においてそれぞれ説明する。図 1.4 に本研究の流れを示すとともに、以下に提案する方式と適用結果を述べる。

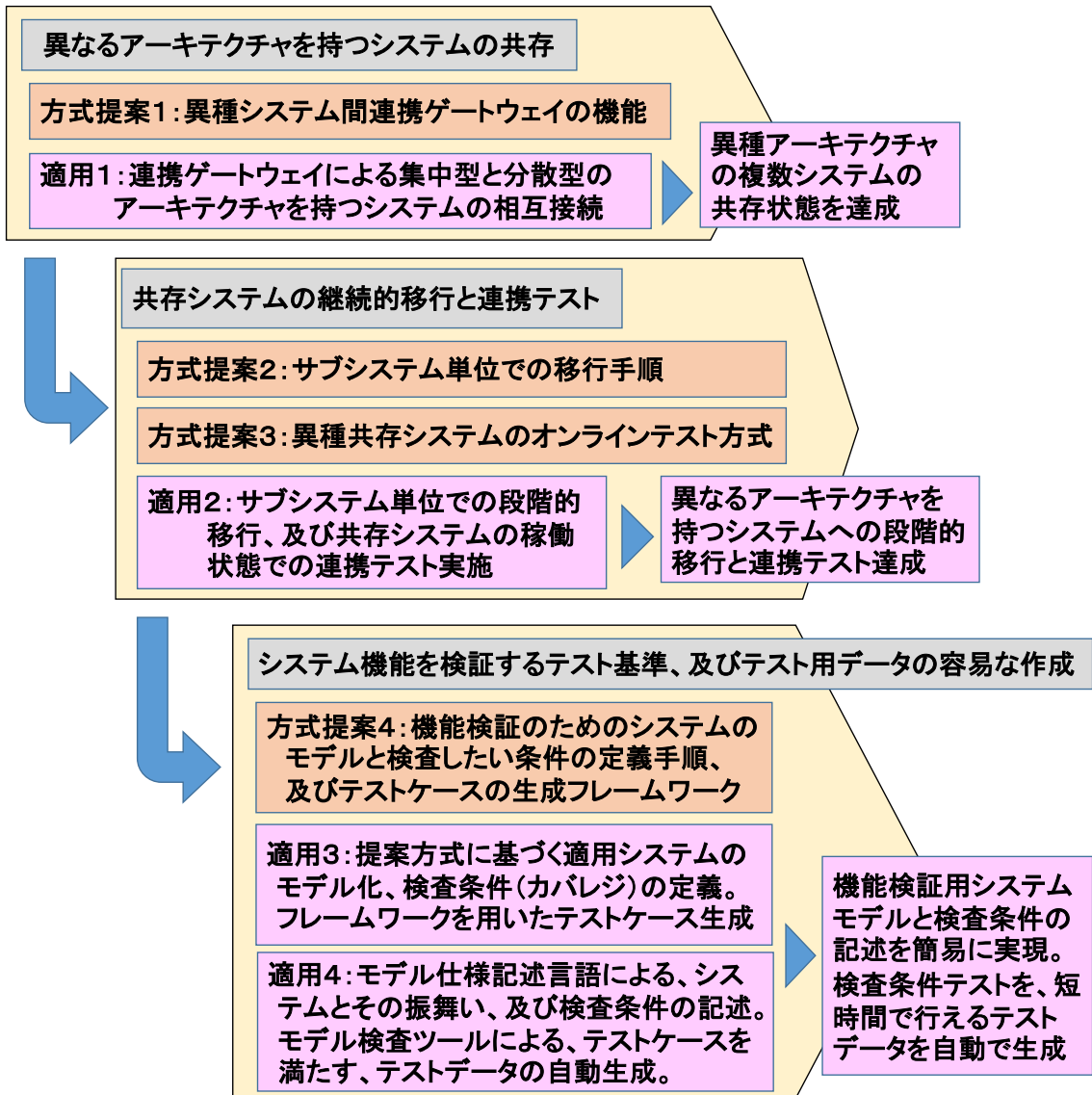


図 1.4 研究の流れ

第3章：異なるアーキテクチャを持つシステムの共存では、異種システム間連携ゲートウェイで集中型アーキテクチャのシステムと自律分散型アーキテクチャのシステムとを共存、連携させる手法を提案する。具体的には、自律分散システム技術の一環として開発さ

れた、情報システムと制御システムとを相互接続するためのゲートウェイ機能の拡張を行う。

方式提案 1：集中型と自律分散型のアーキテクチャのシステム間でメッセージを交換する場合に、相手のシステムのアーキテクチャの違いによって、それぞれのシステムが処理を変える必要がないよう、異種システム間連携ゲートウェイで、①フィルタリング機能と、②フォーマット変換機能とを提供する。集中型システムの集中管理サーバから自律分散型システムのサブシステムへ送られるメッセージには、ゲートウェイで自律分散型システムの受信処理に必要な「データの種別を表すタグ：Transaction Code (TCD)」の適切な種別を選択した上で、メッセージのヘッダに追加して、自律分散型システムのネットワークにマルチキャスト送信を行う。また、自律分散型のサブシステムからマルチキャスト送信されたメッセージのうち、ゲートウェイが集中型システムの集中管理サーバに転送すべきものを TCD によって判定し、メッセージの中から求められるデータの部分のみを抽出する。さらには、必要に応じて複数のデータの統合や形式変換などの処理を行って、集中型システムの集中管理サーバに送信する。この場合、ゲートウェイは集中型システムの下位計算機を模擬（エミュレーション）している。これによって、既存の集中型システムのプログラムなどを変更する必要はない。

適用 1：方式提案 1 の有効性を確認するために、実際に稼働しているタイヤ生産ラインのタイヤ検査工程の制御システムに適用した。集中型システムと同等の機能を持つ、自律分散型の制御システムを設計、開発するとともに、異種システム間連携ゲートウェイを実装することで、集中型と自律分散型のアーキテクチャのシステムの共存、連携の実現を達成した。

第 4 章：共存システムの継続的移行と統合テストでは、第 3 章で達成した、異種システム間連携ゲートウェイを用いた集中型と自律分散型のアーキテクチャのシステムの共存状態から、さらに集中型システムの一部を自律分散型のサブシステムへと移行し、新たな共存状態での連携テストを行う手法を提案する。具体的には、集中型システムの一つの下位計算機とその制御対象とを、自律分散型システムを構成するサブシステムの一つへと移行させるとともに、自律分散システム技術の一環として開発された、オンラインテスト技術を拡張した手法で、新たな共存状態での新しい自律分散型サブシステムの疎通テスト、単体テスト、および既存の集中型制御サーバとの連携テストを行う。

方式提案 2：集中型システムにおける、一つの下位計算機とその下位計算機が制御している機器群、装置群を段階的移行の一単位として扱う。まず、移行する下位計算機に自律分散型サブシステムのプログラムを実装し、異種システム間連携ゲートウェイのフィルタリング機能とフォーマット変換機能の設定を、自律分散型システムに新しいサブシステムが追加された状態に合わせて変更する。続いて、下位計算機に接続されている装置や機器を停止して、集中型の制御サーバから切り離す。その後、下位計算機を自律分散型システムへのネットワークへと接続し、自律分散型サブシステムのプログラムを起動する。

方式提案3：上記の提案方式2を用いて実現された、新たな共存状態での、新しい自律分散型サブシステムのオンライン連携テストを行う。オンライン連携テストは、具体的には、自律分散型システムのネットワークを流れるメッセージと、テスト対象の新サブシステムに、それぞれ「オンラインモード」と「テストモード」の2つのモードを持たせることで、モードの組合せによって、①疎通テスト、②単体テスト、および既存の集中管理サーバとの③統合テストの3段階のテストを行う。①疎通テストは、自律分散型システムの新しいサブシステムが、集中型システムの集中管理サーバとが正常に通信でき、正しく起動できることを、②単体テストは自律分散型の新しいサブシステムが所定の機能を果たすことを、③統合テストは集中型システムの集中管理サーバと自律分散型の新サブシステムとが協調して動作し、システム全体が正常に稼働することを、それぞれ確認する。

方式提案2と3の手順を繰り返すことで、異種共存状態のシステムの継続的移行が可能となり、最終的に既存の集中型システムのすべてを、新しい自律分散型システムへと移行することが出来る。

適用2：方式提案2と方式提案3の有効性を確認するために、適用1で用いたタイヤ検査工程の制御システムに適用し、サブシステム単位での段階的移行、および共存システムの稼働状態での連携テストを達成した。

第5章：システム機能を検証するテスト基準、及びテスト用データの容易な作成では、集中型と自律分散型のシステムの共存状態、及び、すべて自律分散型システムへ移行した状態で、制御対象の機器や装置が正常に動作するか、あるいは想定外の動作をしないかという、システムの機能を検証するための手順と、システム機能検証のためのテスト用データを現場の技術者でも容易に作成できる手法を提案する。具体的には、モデル検査手法における、システムのモデル化と検査条件（カバレッジ）の定義などの一連の手順を単純化、体系化して、定義設計のガイドやフレームワークとして体系化するとともに、対象とするタイヤ検査工程の生産管理システムの分析に容易に適用できるよう、手順と用語に整理し直した。そして、モデル仕様記述言語によって、対象システムとその振舞い、およびテストケースに対応する検査条件を記述し、モデル検査ツールを用いて検査条件を満たす、テストデータの自動生成を行った。

分散システムにおいては、並列動作する計算機が非同期に相互に影響しあうことによって、起こりうる状態の数が容易に組み合わせ爆発を起こす。そのため、分析的な手法を用いて抜け漏れのない状態遷移図を描くことや、検証したい状態を網羅的にリストアップして、テストしていくことは困難である。この問題を解決するための手段の一つが、形式検証技術の1つであるモデル検査による検証である。モデル検査は、システムをモデル化し、その状態をすべてシミュレーションし、設計仕様の正しさを抜け漏れなく検証する技術である。モデル検査ツールによって、検証対象のシステムの状態空間を制約条件に基づいて、網羅的に検証する。しかし、モデル記述言語によるシステムのモデル化、検証したい性質（振る舞い）

の正しい記述や、モデル検査ツールの使いこなしなどには経験を必要とし、技術者の技能への依存性が高いものだった。

本研究では、モデル検査による機能検証のためのシステムのモデル化と検査したい条件（カバレッジ）の定義の手順、及びテストケースの生成手順をフレームワークに整理するとともに、タイヤ検査システムの特徴を分析して、対象システムのモデル定義と振る舞い記述、テストケースを定義、実装した。現場の技術者は、検査工程の状況を観察、整理することで、対象システムのモデルや検証記述のパラメータを決定できる。

方式提案4：本研究では、モデル検査手法の一連の手順を以下のように体系化し、定義設計のガイドやフレームワークを提供することで、テストデータの生成までの作業の容易化を図る。

- ① 対象システムを分析し、システムを構成し、機能を実現するために必須かつ独立に動作できる要素を抽出する。
- ② 抽出された要素を「検査したい対象」と、検査したい対象の振る舞いに影響を与える「環境」の2種類に分類する。
- ③ 「検査したい対象」ごとの「視点」から検査したい条件（カバレッジ）の定義を行う。
- ④ 検査したい条件（カバレッジ）から、フレームワークを用いることで、テストケースを生成する。テストケースには優先順位をつけ、検証すべきケースを確定する。
- ⑤ モデル記述言語を用いて、検査対象システムのモデル、およびテストケースを実装し、検証記述を作成する。「環境」は検証記述のパラメータとして定義される。
- ⑥ 検証記述にモデル検査ツールを適用し検証する。
- ⑦ 検証が正しく完了したのちに、再びシミュレーションを行い、検証された状態を再現する。ランダムシミュレーションを繰り返すことで、現実のシステムの機能を検証するための、可能な限り最短のテストデータのセットを得ることができる。

適用3：方式提案4の有効性を確認するために、先ず手順①～④を適用1で用いたタイヤ検査工程の制御システムに適用し、システムの要素のうち検査したい対象を「検査装置」と「タイヤ」に、環境を「ベルトコンベア」にそれぞれ抽出、分類し、パラメータを決定した。続いて、検査装置とタイヤの視点からカバレッジを定義し、フレームワークを用いて、検証すべきテストケースのセットを生成した。

適用4：方式提案4の有効性を確認するために、適用3の結果に手順⑤～⑦を適用し、検証すべきテストケースのセットに対して、具体的なパラメータのセットを指定し、モデル記述言語 PROMELA を用いて、検証記述を作成した。この際、実際のタイヤ検査工程制御システムのサイズではなく、構成要素の数を最小にしたパラメータセットで検証記述を作成した。これは、生成されたテストデータを分析することで、方式提案4の有効性を確認するためである。

PROMELA で記述した検証記述にモデル検査ツール SPIN を適用し、自動的にモデルとテストケースの検証を行った。その後、SPIN を用いてランダムシミュレーションを行い、可能な限り最短のテストデータのセットを自動的に得た。適用システムの場合は、検査装置の動作を確認するために、ベルトコンベアに投入するタイヤの順番、すなわち、どの種類のタイヤをどの順番で何本投入すべきか、という条件である。テストデータのセットを分析することで、方式提案 4 の有効性を確認した。

第 6 章：結論と課題では、本論文で得られた結果をまとめ、本研究の貢献と今後の課題について論じる。本研究は、SE による事前の膨大な準備と立会いを必要とし、長期の工場休止時期のみに実施できる従来のシステム移行手順ではなく、現場の技術者が平常時の工場稼働状態で、任意のタイミングでシステムの段階的な移行とオンライン状態での連携テストを行い、またシステム機能の検査を行うテストデータの作成も容易に行える継続的なシステム移行のための手順と方式の体系を提案し、その有効性を確認したことに貢献がある。また、本研究で提案した方式を現実のシステムに用いることで、システム移行とテストに要する総時間の短縮、影響範囲の限定化の効果を確認した。

さらに今後の課題として、集中型から自律分散型というパターン以外のシステム移行での本提案手法の適用可能性の探索、事例として扱った製造業における生産管理システム以外のアプリケーション分野での適用可能性の探索、ランダムシミュレーションによるテストデータ生成の時間短縮、などを挙げる。

第2章 情報制御システムの移行に関する課題と研究の位置づけ

2. 1 情報制御システムの動向とシステム移行の課題

本章では、最初に本節において情報制御システムの定義とニーズを述べ、次に情報制御システムの移行作業の課題を整理する。次節以降で、既存の文献、官民の取組みなどをサーベイして、特に制御システムの移行ならびにテストに関する動向を調査する。これに基づいて、最後に本研究の流れを示し、本研究で提案する方式と実際の制御システムへの適用を設定する。

2. 1. 1 情報制御システムとは

我が国において、情報システムは、企業の業務の効率化のため 1950 年代に企業の経理や財務などの間接部門の業務処理を対象に利用が始まった。1950 年代後半にはバッチ処理に加えてオンライン処理も可能となり、サービスの適用範囲が広がった。さらに、1970 年代には製造や運行の計画立案や実績の管理、各種情報の提供などを担う情報システムや、経営意思決定支援を行う情報システムなども構築されるようになった。

一方の制御システムは、1970 年代の半ばにデジタル制御システムが登場して以降、接道や電力などの社会インフラや製造などの、高いリアルタイム性や信頼性を求められる分野に広く適用されるようになった。1980 年代までは、情報システムと制御システムとは独立して開発され、稼働していた。しかし、製造や交通などの分野において、市場の変化への迅速な対応や利用者への情報サービスへの期待などが高まるにつれ、制御システムと情報システムとを統合、連携させることで、これらのニーズに応え、競争力を高める動きが 1990 年代に現れた。これが情報制御システム、或いは情報・計測制御システムと呼ばれるものである。いまや、情報制御システムは社会・産業を支える基盤として定着し、様々なアプリケーション分野で着実に発展を続けている。

2. 1. 2 情報制御システムへのニーズ

情報制御システムは、その時代の要請や新しいシステム技術の普及などに対応し、常に発展してきた。従来の高性能、高信頼化、低価格化に加えて、システムの柔軟性、拡張性、保守性を高めつつ、特に制御システムではそれまではメーカー独自の仕様に基づくプロプライエタリなものを中心だった、計算機やネットワークのハードウェアやソフトウェアのオープン化が 1990 年代から進み、さらには情報システムと共通な製品、すなわちパソコンや Ethernet も積極的に利用されるようになってきている。

近年では、人工知能技術とビッグデータ分析技術などを活用して、クラウド上で企業の意思決定や大規模な SCM (Supply Chain Management) を集中的に行うことが一般的になってきた (岡田 2014)。また、IoT (Internet of Things) 対応機器の普及に伴い、現場のセンサや装置から得られるデータ量は急速に増大している。企業経営の指標とするため、またアセットマネジメント (設備管理) をシステム化するために、現場の状況をリアルタイムで

監視し、可視化できることへの期待は大きい。しかし、データ伝送速度や容量の制約から、すべてのデータをクラウドに集約して処理することは現実的でない。現場に近い場所でデータの管理と処理を分散的に行う構成を取る必要があるため、集中処理と分散処理を柔軟に配置し、連携させることが必須である。また、これらのデータは経営判断に直接かかわるため、システムの再構築への要件に応じて、動的にハードウェアやソフトウェアの構成の変更を行うことも求められる。経営層から現場までを垂直統合し、経済状況、市場や顧客のニーズの変化、さらには環境問題などの社会的課題にもいかに迅速に対応できるかが、競争力に大きく影響を与える（浅川 2014）。

情報制御システムは、システムへの要件の変化、ハードウェアの老朽化やソフトウェアの陳腐化、あるいは新技術の導入などに対応するため、まったく新規に構築するのではなく、稼働中のシステムを再構築していかなければならない。例えば、製造業においては、既に投資された工場の生産機能を維持しつつ設備更新を行い、併せてニーズに対応した機能改善や拡張を行っていくことが求められる。一方で、団塊世代の高齢化や少子化に伴い、熟練した技術者の数が急速に減少している。特に、システムエンジニア（SE）は、情報制御システムの高度化と複雑化の進展、次々に導入される新技術などに対応するため、需要が高まり、人材不足が深刻である（厚生労働省 2016）。

システム再構築に際して、現行システムから新システムへと移行させる作業は、一般的には広範囲かつ長期間のシステム停止を要し、システム設計と開発の専門家であるSEによる、移行計画書の作成やリハーサルなどの事前準備が必須であった（実森 2007）。ところが、情報制御システムには、止められないものの割合が高い。電力、放送、鉄道などの社会インフラ系のシステムをはじめ、生産のための制御システムも高効率生産や生産コスト低減のため、365日、24時間連続稼働を求められるものが多い。全停止ができるのは、慣習的に年に2回の盆休みと年末年始の数日間のみ、或いは特定の曜日の深夜の数時間のみという制約のあるシステムが少なくない。

ところが、企業を取りまく市場や経営環境が刻々と変化する状況においては、新たな要求に対応できる新システムへの移行を、出来るだけ速やかに行う必要がある。そのため、システムを長期間停止できるタイミングを待つのではなく、システムの稼働中に移行ができる仕組みが求められる。さらには、上述のSE不足に対処するためにも、SEによる事前の準備や立会い作業を必須とせず、現場の技術者でも実行できる移行業務とテストの手順が求められる。本研究では、システムの移行やテストなどの作業を、現場の状況に応じて任意のタイミングで行えることを「柔軟」、システムを利用する現場のユーザのみで、SEや事前の準備に頼らず行えることを「容易」とそれぞれ定義する。

以下に、情報制御システムの「柔軟」かつ「容易」な移行作業の要件を整理する。

- ① 移行作業は、現場の条件が整えば任意のタイミングで実施することができ、また、システム全体の停止を必要としない。停止する場合も、停止範囲の局所化、停止回数や停止

時間の低減を図る。

- ② システム移行作業と連携テストは、移行手順書や移行支援ツールを用いれば、現場の技術者でも可能な手順とする。
- ③ システム機能のテストも、テスト手順書やテスト支援ツールを用いれば、現場の技術者でも可能な手順とする。

2. 2 本研究で取り扱う範囲と課題

情報制御システムの移行のケースを分類すると、大きく対象のシステムが情報系か制御系かという軸と、移行する新システムのアーキテクチャが既存システムと同じか、或いは異なるか、という軸とがある（表 2.1）。

表 2.1 システム移行の分類と既存の技術

移行先システム システム種別	同アーキテクチャ	異種 アーキテクチャ
情報システム	- エミュレータ - 仮想マシン	- 仮想マシン - クラウド
制御システム	- 互換ハードウェア - 互換ソフトウェア	本論文で提案する 移行とテストの手法

情報制御システムを移行する場合は、制御やサービスを行うプログラムだけでなく、その前提となるハードウェアやソフトウェアの条件も併せて考慮する必要がある。用いられている OS やミドルウェアとそのインタフェース、通信ネットワークとプロトコル、データベースやファイルシステムの形式とそのインタフェース、制御システムの場合には共有メモリやグローバルメモリなどとのインタフェースも不可欠な確認事項である。

図 2.1 に示すシステム移行の手順の中で、このような現行システムの調査と分析が、先ず行われる。続いて、既存システムと新システムとで、これらに差異がないか、差異があった場合はどのようにそれを吸収するか、或いはプログラムのどの部分を修正するか、などを検討して、移行の可否の判断が行われる。移行が困難という判断になれば、現行システムは破棄して、まったく新しい情報制御システムを構築し、既存システムから切り替えることになる。

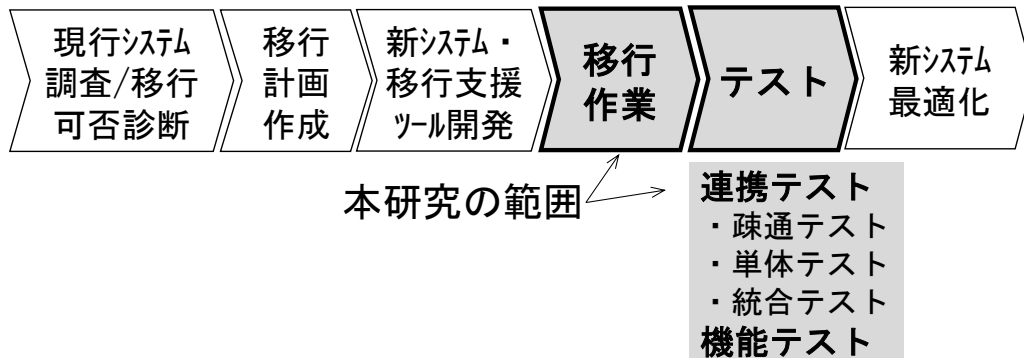


図 2.1 システム移行の手順と研究の範囲

情報システムを移行させる場合、エミュレータや仮想マシンを用いて既存システムの環境を模擬することで、既存のプログラムの変更を最小限にしてシステム移行を行う取り組みがなされてきた。2000年代に入り、クラウドコンピューティングの普及に伴って、基幹システムの移行先はクラウドが主流になりつつあり、国内外の各ベンダも様々なソリューションを提供している。欧州では、非クラウドのシステムをクラウド環境への移行をサポートするための ARTIST Project がある (Menychtas 2013-1) (Menychtas 2013-2)。欧州の IT 企業の競争力向上を目的に、非クラウドのソフトウェア資産の移行を促進する一連の方法、ツール、技術を開発し、提供している。

情報システムをクラウドへ移行する場合も、クラウドに仮想マシンを置いて、そこに既存のシステムを移行するケースと、クラウドが提供するサービスを利用し、サーバレスで既存のシステムと同等の機能を実現するケースとがある。前者は IaaS (Infrastructure as a Service)、または PaaS (Platform as a Service) と呼ばれるサービスを用いており、後者は SaaS (Software as a Service) と呼ばれるサービスを用いる。IaaS、PaaS、SaaS は、いずれもクラウドのサービスに含まれるが、近年では、SaaS とクラウドがほぼ同義で用いられることが多い。SaaS はネットワーク、ストレージ、サーバなどのシステムのインフラから、OS やミドルウェアなどの基盤となるソフトウェア、また各種のアプリケーションまで提供し、これらのハードウェアやソフトウェアの保守・運用のサービスも行っているため、SaaS を用いて新システムをクラウドへ移行する場合には、クラウド上の実装のアーキテクチャは、もはや新システムの開発者にも問題にならない。

制御システムでは、リアルタイム性や信頼性への要求が高く、またプロプライエタリなハードウェアやソフトウェアを用いている場合も多い。そのため、システム移行の際には、新システムのハードウェアやソフトウェア (OS、ミドルウェアなど) は、既存システムのプログラムがそのまま再利用できる互換性 (後方互換性) を持つものが選択される。制御用の計算機やソフトウェアのベンダは継承性を意識し、後方互換性を持つ製品を提供している (柴宮 2013)。なお、ここでいうプロプライエタリとは、ソースコード、仕様、規格、構造などがメーカ独自であり、公開されていない状態を指す。一般的には、「オープン」の反意

語として使われる。

また、制御システムでも、情報システムに近い処理は、現場の管理装置である分散サーバ（または、エッジと呼称される）に移行する選択肢もある。製造システムのサービス指向システムへの移行や、サイバー・フィジカルシステムへの移行方式の提案もなされている（Cala 2017）（Delsing 2012）。欧米では企業コンソーシアム、あるいは産官主導による、「インダストリアル・インターネット」や「インダストリー4.0」に代表されるプロジェクトで、産業システム向けのIoT基盤が提案されている。これらは、現場で稼働する様々な機器に組み込まれたセンサから生成されるデータを収集し、ビッグデータをクラウドサービスで分析し、予測を行うことで、製品の品質の向上、生産効率の向上、エネルギー消費量の削減、生産コストの削減、不良品率の低減、および生産工程管理の自動化などの達成を目的としている（OKI 2018）。この目的のため、センサなどが大量のデータを送り出すようになり、それらが通信回線の帯域を圧迫し、結果的にデータ転送の遅延を引き起こす可能性がある。この問題を回避するため、制御システムに分散サーバ（エッジ）を配置し、前処理をしてから必要なデータのみをクラウドに送信する構成を取る場合もある（岡 2013）（日本電機工業会 2017）。

このように、制御システムにおいてもシステムの構成や機能の配置を柔軟に変更することへの要求は高まっている。しかし、制御システムの移行の場合にも、現場の装置や機器を直接制御する部分を、異なるアーキテクチャのシステムに移行するケースに対しては、先行的な研究やベンダのソリューションを見出すことができない。本研究では、既存の異種システムの連携技術やオンラインテストの方式を拡張することにより、このようなケースにおける移行とテストの課題をどのように解決するかを明らかにする。さらには、SEが不在の状況でも、現場の技術者が移行作業やテストを容易に行える仕組みを提案する。

2. 3 先行研究の調査

2. 3. 1 自律分散システム

本研究では、現場の機器を制御するシステムを集中型アーキテクチャから、自律分散型アーキテクチャへ移行する手続きとテストの方式を対象としている。本項では、本研究で扱う自律分散システムとその技術について概説する。

（1）自律分散システムとは

1980年代までは、計算機リソースが高価であり、まだ通信ネットワークの速度や信頼性が高くなかったため、集中型でシステムが設計、構築される場合が多かった。ホスト機が集中的なアルゴリズムを保持し、システム全体の情報を集約して、判断を行い、システムの構成要素に指示を出す。集中型のメリットは、システム全体の状態を把握することで、解の最適化を図ることが可能という点にある。しかし、システムの大規模化に伴い、システムの一部に故障が発生する確率が上がり、またシステム全体の構造の把握やシステム全体の情報を収集することが困難になってきた。そのため、システムの無停止、段階的拡張、オンライ

ン保守などの要求に応えるには、従来の集中型システムやその延長であるツリー構造の階層型システム、或いは大規模なシステムを機能や設置場所でサブシステム（構成要素）へ分割した分散型システムでは限界が生じていた。

これらの要求に応えられるシステムとして提案されたのが、生物体をアナロジーとした、自律分散システム概念である。この概念では、システムの構築、保守、拡張は分離したプロセスではなく、同時並行的に発生するものとしてとらえ、システムの構成と制御のあり方を考えている（森 1984）（河野 1996）。自律分散概念では、システム全体を管理する集中的な管理機構を持たず、システムの構成要素である複数のサブシステムが、各自が得ることの出来る情報に基づいて自律的に判断・動作し、相互に作用しながら、全体としての機能を果たす（図 2.2）。

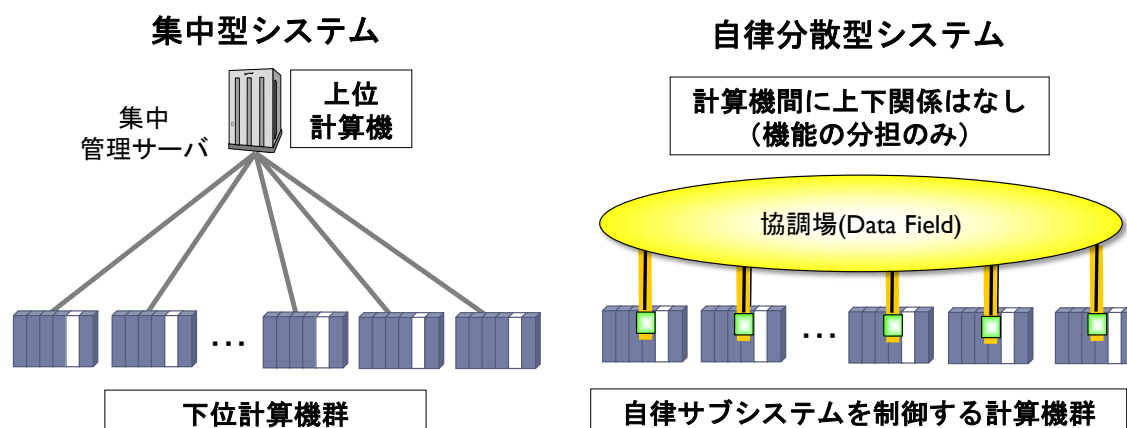


図 2.2 集中型システムと協調場を用いた自律分散型システムの模式図

この概念により、サブシステムの故障、保守、拡張などによるシステムの部分的な停止が全体の機能停止を引き起こすことはない。故障したサブシステム以外の、生存しているサブシステムが依然として機能し続けるフォールトトレランス性、稼働中のサブシステムを止めることなく拡張、保守、およびソフトウェアの移動を行うオンラインでの拡張性と保守性、およびアプリケーションプログラム（AP）の可搬性を持つことが可能となる。さらに、自律分散システム概念は、分散的で段階的な開発にも好適であることから、鉄鋼（鈴木 1990）、電力（笹嶋 1996）や鉄道（北原 1999）（OGATA 2016）（情報処理推進機構 2013）など多分野の情報制御システムで採用されている。

自律分散型システムを実現するための一つの方式として、サブシステムの自律的な判断とサブシステム間の連携とを支えるために情報共有の「場」を用いる形態がある。本研究で採用する自律分散システムの実装方式では、この「場」を「協調場」または「Data Field (DF)」と呼称する（図 2.3）。自律サブシステムは、協調場から自らの判断や動作に必要な情報を受け取り、また他のサブシステムに必要な情報を協調場へ提供する。あるサブシステムで情報が生成されると同時に協調場に提供・共有されることで、他のサブシステムは何らかのイベ

ントが発生したことを知ることが出来る。この特徴はリアルタイム性を重視する制御システムには特に重要である。

協調場をシステムに実装する方式には、幾つかの提案(石田 1993)(森 1984)(田村 1988)(林 1998)があるが、サブシステムに公平に情報が同報されるよう、放送型通信やメモリ転写などの手段を用いる場合が多い。図 2.3 に、財団法人 製造科学技術センター FA オープン推進協議会で策定、公開された「自律分散プロトコル仕様書 R3.0」(FAOP 1998)の実装手法を示す。本仕様で定義された自律分散システムは、Ethernet の同報通信プロトコルである UDP/IP を用いて協調場を実装し、サブシステムの自律的な判断のために、データの種別を示すタグである TCD (Transaction Code) をメッセージのヘッダに付与する手法を用いている。

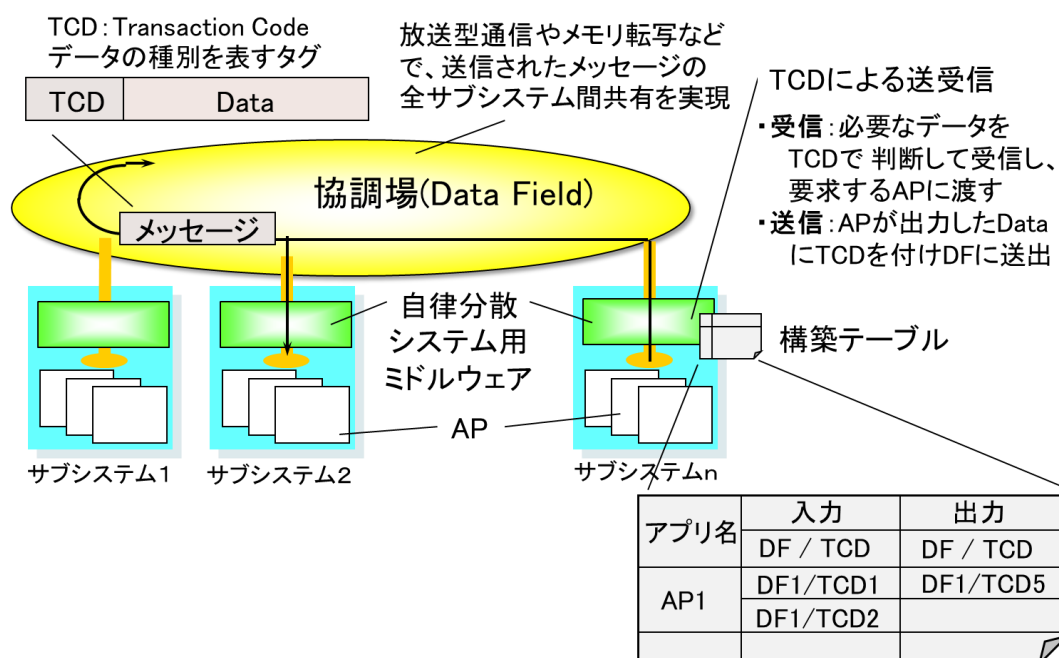


図 2.3 協調場 (Data Field) と TCD による自律分散システムの実装手法

本実装手法では、データの生成元のサブシステムが、宛先のサブシステムや AP を指定してデータを送信するのではなく、Data Field (DF) に、トランザクションコード (TCD) というデータの内容を示すコードを付与したメッセージをマルチキャストする。各サブシステムは、どの TCD のデータを受信するかを事前に構築テーブルに指定しておくことで、DF を流れるメッセージを自律的に選択受信する。これらのメッセージの送受信と、サブシステム内のアプリケーション・プログラム (AP) とのデータのやり取りは、自律分散システム用ミドルウェアが担当する。この方式では、サブシステム間は疎結合しているため、システムの構成を変更した際にも、送信側サブシステムの AP の処理内容や、送信先の設定の修正を行う必要がない。そのため、稼働中のサブシステムや AP を止めることなく、システ

ムの拡張や構成の変更が容易となる (Yoshida 2014-1) (Yoshida 2014-2) (Yasukochi 2017)。また、本方式では、同じ AP を複数の計算機の上に配置することで、多重化を容易に実現できるため、システム全体の信頼性の向上や保守などに伴う部分停止などを、柔軟に実現できる利点もある (船橋 1996)。

本研究では、この「自律分散プロトコル仕様書」に準拠した実装方法に基づいて、研究、方針提案と開発、適用を行った。以降の議論は、図 2.3 に示す方式と用語を前提とする。

2. 3. 2 本研究で対象とする分野の先行研究

本研究で対象とする、異なるアーキテクチャを持つシステムの共存と連携、システム間連携テスト、およびシステム機能の検証を行う技術の研究開発について、代表的な研究および方式を表 2.2 に示す。以下にそれぞれの研究の目的や特徴、および課題を述べる。

表 2.2 先行研究・技術の調査

	異種システムの共存	サブシステム段階的移行	システムのモデル化と検査条件の定義	テストデータ作成用テストケース生成
代表的研究や方式	協調自律分散:システム連携ゲートウェイ (林 1995)	オンラインテスト方式 (河野 2002)	検証パターンによる定義支援 (Dwyer 1999)	検査式作成ガイド (金井 2008) (早川 2012)
目的	リアルタイム性への要件やメッセージのデータ長などが異なるシステムの相互接続	新規サブシステムのオンラインデータを用いたテストと既存システム稼働への悪影響防止	検査対象システムのモデル化と検査条件の定義の容易化	検査対象システムの検査条件の定義とテストケース作成の容易化
手段	ゲートウェイにより、異種システム間のメッセージのフィルタリングと統合・分割を実施	メッセージのヘッダにテストフラグエリアを設け、オンラインとテストのモードで処理を分ける	検査式をイベント/状態の「パターン」(発生期間と順序)と「スコープ」(発生する範囲)として分類	限定した検査式のパターンをDB上に保存するとともに、定義支援ツールによる参照と再利用を容易化
課題	既存のゲートウェイ機能では、異なるアーキテクチャのシステムの共存・連携には不十分	既存方式では異なるアーキテクチャのシステムとの連携テストには不十分	パターンとスコープは汎用的すぎ、現場技術者がモデル化や検査条件定義を行うのは困難	自律分散型のシステムの定義に適したパターンの作成手順は提案されていない

(1) 異なるアーキテクチャを持つシステムの共存

図 2.4 に製造業における情報制御システムの構成例を示す。

この例では、本社の情報系のシステムから生産ラインの情報系システム、そして制御系のシステムまでが垂直統合されている。また、工場内の生産ラインの各工程の制御系システムも水平統合をされ、相互に連携している。垂直統合を実現するには、異なる階層のシステムを相互接続しているサーバ、パソコン、或いは PLC (Programmable Logic Controller) にゲートウェイ機能を持たせる (橋本 2004) (梶原 2015)、ネットワーク階層を超えてシー

ムレスに接続できる通信プロトコルを用いる（森田 2014）などの手法がある。また、制御システムの水平統合には、標準の制御系ネットワーク（フィールドネット）や Ethernetなどをベースにすることで、従来の製造工程内のプロセス制御と電気制御の統合にとどまらず、製造工程の前後の業務までを含めた統合が進められている（橋本 2004）（森田 2014）。

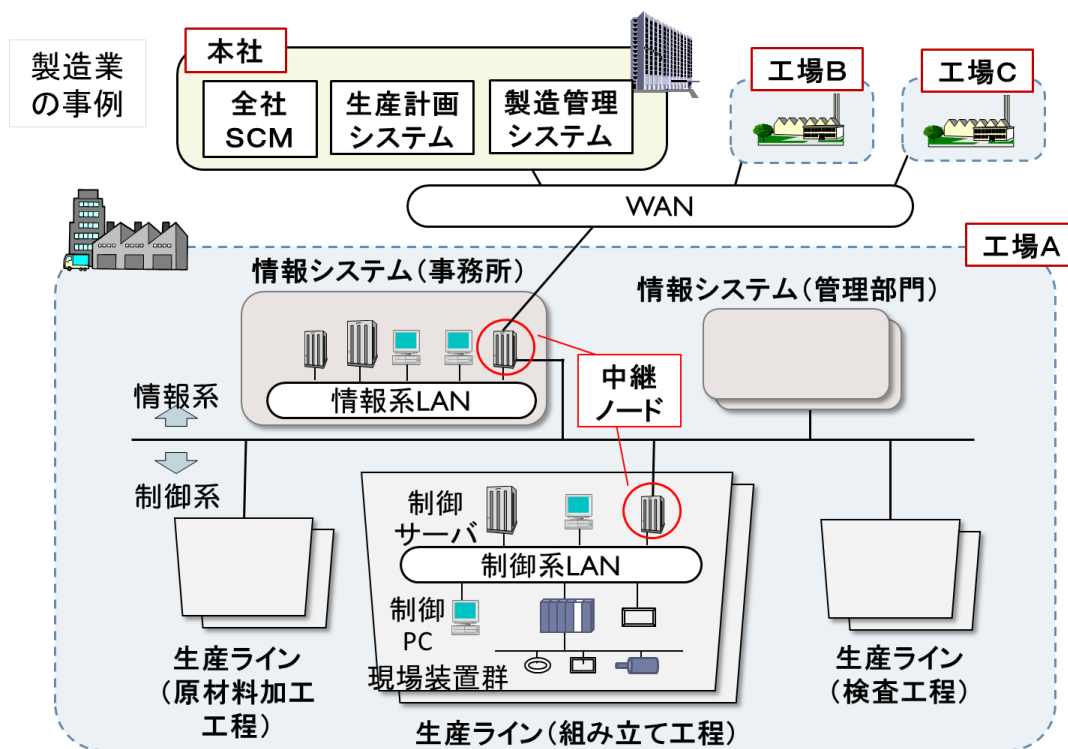


図 2.4 製造業における情報制御システムの構成例

異種の自律分散型システムを共存させる方式として、複数の DF の間を中継する計算機ノードに、TCD によるフィルタリングやメッセージの統合・分割機能をもたせることで、より柔軟な運用を可能とした協調自律分散システム概念が提案、実装されている（林 1995）（鮫嶋 1999）（山本 2013）。

図 2.3 では、単一の DF に同質のサブシステムが接続されている状況を示している。しかし、情報系システムと制御系システムとを単純に接続した自律分散型システムを一つの DF のみで運用しようとする、①情報系の不要なデータが制御系のリアルタイム処理に悪影響を与える、②情報系では利用できない制御系の膨大なデータがネットワークの負荷を増大させ、処理効率を悪化させる、という二つの問題が生じる。

そこで、協調自律分散システム概念ではシステムを構成する DF を幾つかの DF に分け、また、システムの中で複数の DF の間を結ぶサブシステムは、一方の DF から他方の DF に TCD 付きメッセージを中継するゲートウェイの役割をする。ゲートウェイは TCD 値から判別して、不要なメッセージを中継せず必要なメッセージだけを中継するフィルタリング機能を果たす。また、ゲートウェイは、異なる DF 間でのメッセージのフォーマット変換機

能も持つ。フォーマット変換機能は、具体的には制御系のメッセージのデータから必要な部分を抽出、統合することで、情報系で利用できるデータに変換して、情報系の DF に送出するメッセージの統合機能と、その逆の処理を行うメッセージの分割機能を持つ（図 2.5）。

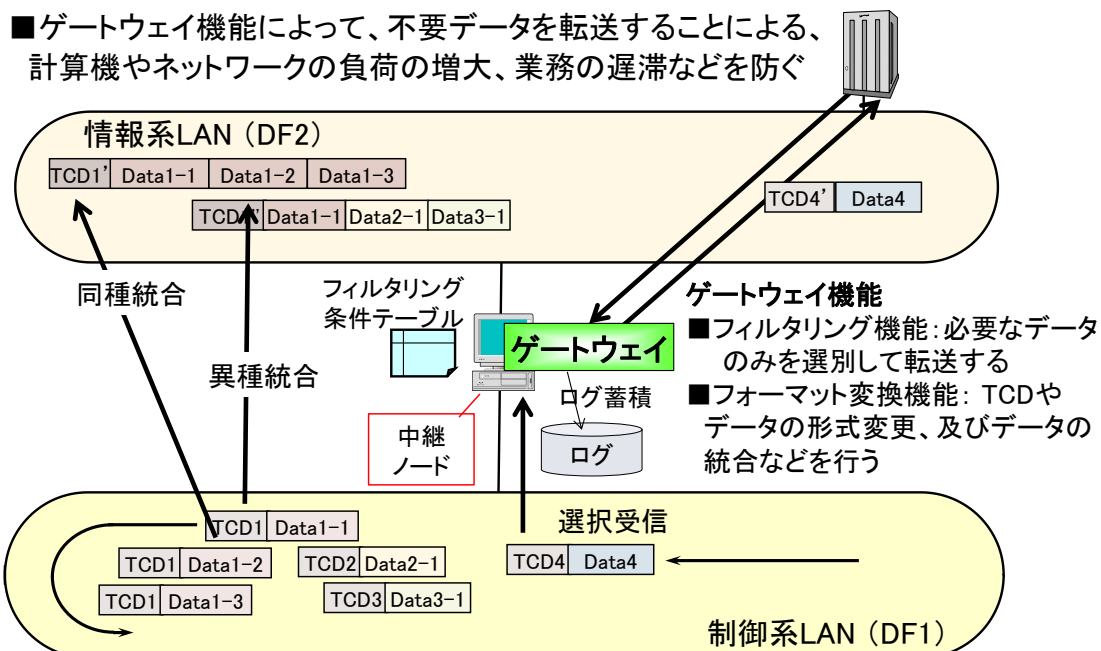


図 2.5 ゲートウェイによる情報系と制御系の自律分散型システムの統合と連携

本研究からの協調自律分散システム概念とゲートウェイ機能の課題は、これらの方式は同じ自律分散型のアーキテクチャを持つ異種のシステムを相互接続には適していても、異なるアーキテクチャのシステム、具体的には集中型のシステムとの統合や、集中管理サーバとの連携には機能が不十分な点である。

(2) 共存システムの継続的移行と連携テスト

本節で議論する連携テストを定義する。図 2.6 にドイツ政府と軍関係のプロジェクトで標準として採用されている、V モデル (V-Model 200) を示す。V モデルは IT 製品を開発する際に、設計からテストまでの一連のプロセスを定義しており、我が国のシステム設計開発やテストの現場でも一般的に用いられている。V モデルにおけるシステムテストは、システムに求められる要件を満たす動作をするか、或いは求めていない動作はしないかを確認するためのテストである。同等のテストには、ユーザテスト、運用テストなどの様々な呼称があるが、本稿ではシステム機能テストの名称に統一する。本節で議論する連携テストは、疎通テスト、単体テスト、および統合テストの 3 段階に分けられる。疎通テストは、既存の AP と新 AP とが正常に通信でき、新 AP が正しく起動できることを、単体テストは新 AP が所定の機能を果たすことを確認する。統合テストは、V モデルにおける結合テストであり、既

存 AP と新 AP とが連携して動作することを、それぞれ確認する。

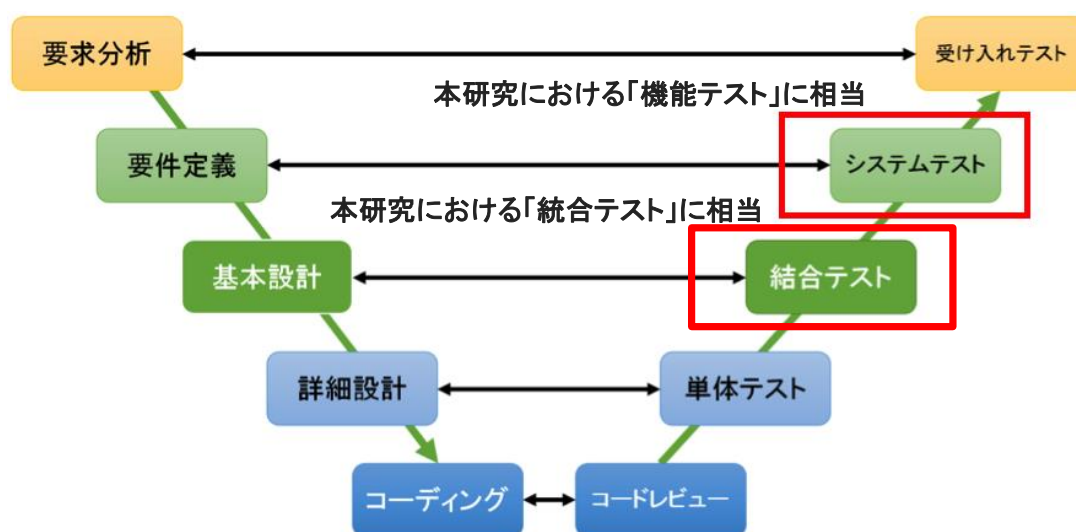


図 2.6 V 字モデルにおけるシステムテストの位置づけ

システム移行時には、既存のサブシステムと新規サブシステムとの連携テストを行う必要がある。新規サブシステムのテストが完了すれば、さらに既存のサブシステムの一部を新規サブシステムへと移行して連携テストを行うという手順を繰り返すことで、継続的なシステム移行と機能テストが可能となる。

自律分散システムでは、サブシステムの AP と、DF を流れるメッセージに、それぞれオンラインとテストの 2 つのモードを持たせることで、オンライン稼動中の連携テストを実現し、上記の検証を行っている。各サブシステムの自律分散システム用ミドルウェアが、個々の AP がオンラインモードかテストモードかを管理している。メッセージには、プロトコルヘッダの TCD エリアにさらにテストフラグエリアを付加し、このフラグが 0 か 1 かで、オンラインメッセージかテストメッセージかの識別を可能とする。

以下に、既に実用化されているテスト手法の概要を説明する (Yamamoto 1993) (Nishijima 1996) (河野 2002)。

図 2.7 の例において、テスト環境を説明する。この環境には、2 つの既存サブシステムと、1 つの新サブシステムが接続されている。既設サブシステムは、それぞれアプリケーションプログラム(AP)の A と B を持ち、新サブシステムは、AP の B をリプレースする B'を持っている。図中の右上はテスト用ツールであり、DF に流れるメッセージをオンラインメッセージかテストメッセージに関係なく受信し、テストの状態をモニタする。テストが完了し、B'の機能と他サブシステムとの連携が問題ないことが判明すれば、既設サブシステムから新サブシステムへの切り替えが行われる。

オンライン稼動中テスト方式では、テスト対象とする AP と、連携対象の AP またはテス

ト用のツール、およびデータのモードの組合せによって、テスト形態が3つのケースに分類される（表 2.3）。

表 2.3 自律分散システムのテスト形態の分類

	連携対象	新規部分または テスト用ツール	既設部分
テスト対象			
新規部分		シミュレーションテスト (ケース1)	オンライン連携テスト (ケース2)
既設部分		オンライン連携テスト (ケース3)	(通常稼働状態)

① シミュレーションテスト(ケース 1)

ケース 1 では、新サブシステムをテスト対象とする。新サブシステムは、既設稼働中のサブシステムからテスト用のメッセージは受信するが、機能的には連携せず、テスト用ツールによって動作が確認される。

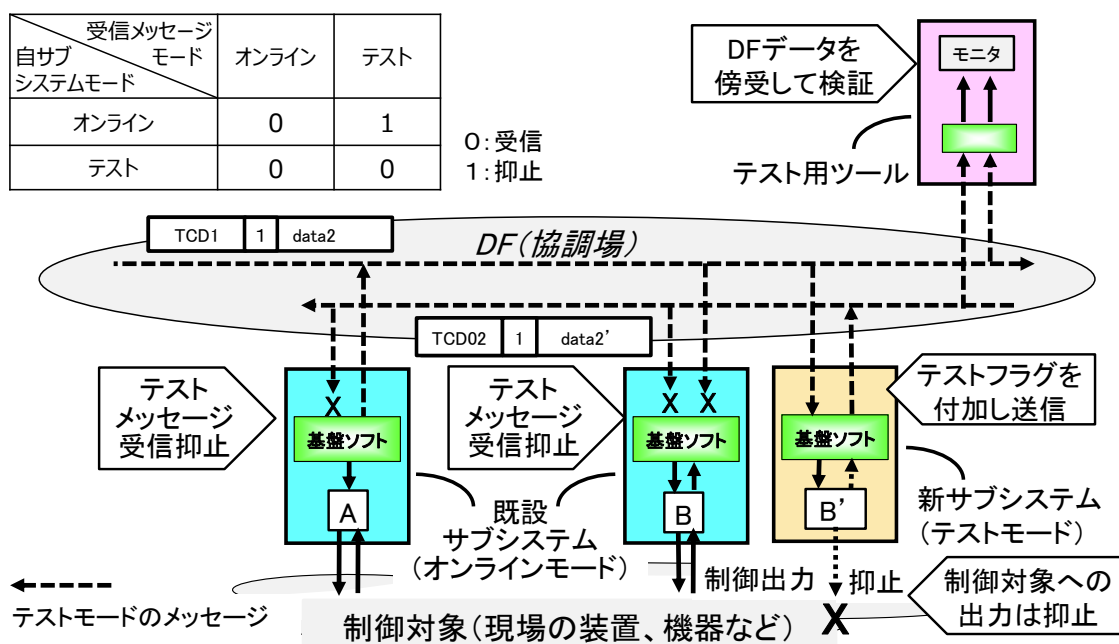


図 2.7 オンラインデータを使った自律分散システムのテスト環境例（ケース 1）

図 2.7 において、既設サブシステムはオンラインモード、新サブシステムはテストモードに設定されている。図中の左上の表にあるように、テストメッセージはテストモードのサブシステムのみが受信し、テストモードのサブシステムから送信されるメッセージはすべて

テストメッセージとなる。

新サブシステムの B' をテストするために、既設サブシステムの A からテストメッセージが送信される。このテストメッセージは、A が通常の処理を行っている場合と同じ内容を持つが、テストフラグエリアに 1 が入っている。通常の処理では、A からのメッセージを受信する既設サブシステムの B は、これがテストメッセージであるために、基盤ソフトが受信を行わず、影響を及ぼさない。

テストされる新サブシステムの B' は、このテストメッセージを受信したのちに、所定の処理を行い、その結果を DF にテストメッセージを送信する。既設サブシステムの A および B は、これがテストメッセージであるために、基盤ソフトが受信を行わず、影響を及ぼさない。テスト用ツールは、新サブシステムの B' から送信されたテストメッセージを受信し、その内容をモニタで確認することで、B' での処理が正しく行われたかを確認する。ケース 1 は、疎通テストと単体テストに相当する。

② オンライン連携テスト(ケース 2、ケース 3)

ケース 2 は、オンラインモードの新サブシステムをテスト対象とする。新サブシステムは、既設稼働中のサブシステムからリアルタイムにオンラインメッセージを受信して通常処理を行い、既設のサブシステムの出力と比較することで、動作が確認される。

図 2.8 において、すべてのサブシステムはオンラインモードに設定されている。新サブシステムの B' をテストするために、通常の処理を行っている既設サブシステムの A から、オンラインメッセージが送信される。新サブシステムの B' は、このメッセージを受信したのちに所定の処理を行い、その結果を DF にオンラインメッセージとして送信する。ただし、この時点で B' はまだテスト中であるため、制御対象の現場の装置や機器への出力は抑制される。

一方、既設サブシステムの B は、A からのオンラインメッセージを受信すれば、通常通りに処理を行い、その結果を DF にオンラインメッセージとして送信するとともに、制御対象の現場の装置や機器への出力も行う。B からの制御対象への出力を、B' からの制御対象への出力と比較することで、B' の処理が正常に行われたかどうかの判断ができる。

ケース 3 は、オンラインモードの新サブシステムと既設サブシステムとの連携のテストである。図 2.8 の例で、新サブシステム内の B' から DF に送信されたオンラインメッセージは、既設サブシステムの A が受信し、その内容に従って処理を行う。既設サブシステム B も、B' と同様の処理を行うので、と既設サブシステムの A とを連携させるテストである。このケースでは、B' から送信したオンラインメッセージを A が受信して処理を行うので、A の動作を見て、A と B' の連携が正常に行われているかを判定できる。

また、テスト用ツールは、B と B' から送信されたオンラインメッセージを受信し、その内容を比較することで、既設サブシステムと新サブシステムの機能や処理が同等であるかを確認することができる。ケース 2 とケース 3 の双方を併せて、統合テストに対応する。

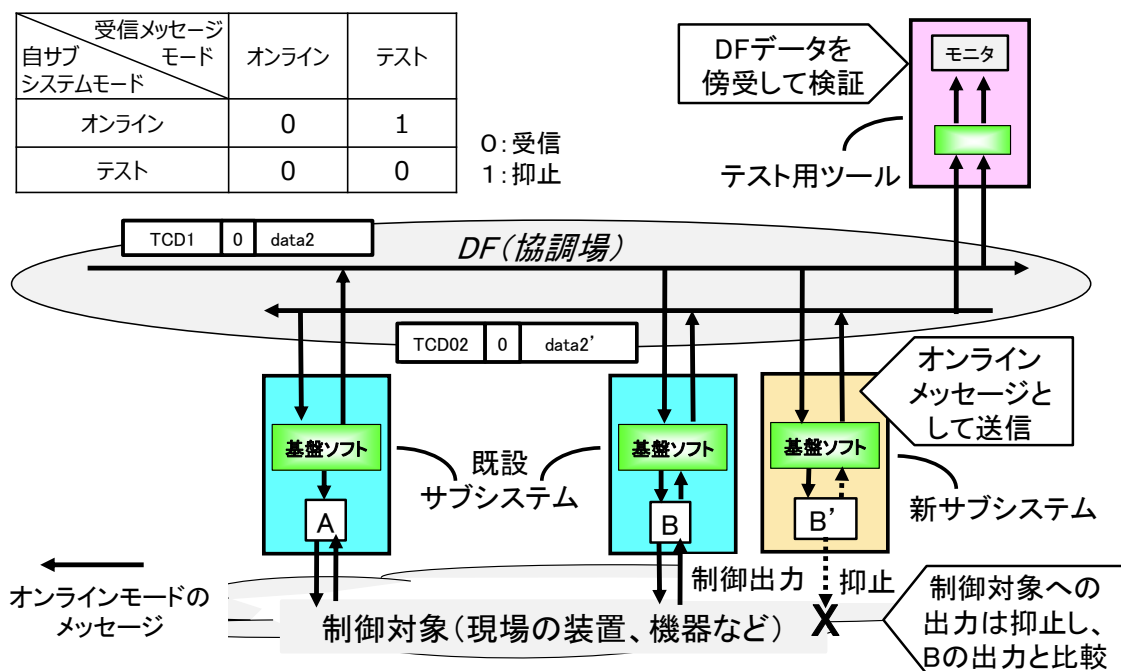


図 2.8 オンラインデータを使った自律分散システムのテスト環境例 (ケース 2、3)

このように、自律分散システム技術のオンラインテスト方式は、既存システムと新システムとの連携テストに必要な機能を備えている。しかし、本方式の本研究からの課題は、自律分散型システムの既存システムと新システムとの連携テストには適していても、異なるアーキテクチャのシステムとの連携テストには機能が不十分な点である。具体的には既存の集中型システムと自律分散型の新サブシステムとの疎通、単体、統合テストを行うための手順やケース分類、および機能の設計などがなされていない。

(3) システム機能を検証するテスト基準、及びテスト用データの容易な作成

システムの移行後に、制御対象の機器や装置が正常に動作するか、あるいは想定外の動作をしないかという、システム機能の検証の手順と、システム機能を検証のためのテスト用データの作成が必要となる。

システム機能のテスト手法は、大きくプログラム外部のテストを行うブラックボックステストとプログラム内部のテストを行うホワイトボックステストとに分類される。

ブラックボックステストは、プログラムの構造など内部を考慮せずに、システム自体の仕様を満たしているかどうかを、外部から網羅的に確認する機能のテスト手法であり (バイザー 1997)、同値分割法、境界値分析法、異常値・特異値分析法などの手法がある。一方、ホワイトボックステストは、プログラムの内部設計情報を元にしたテスト設計方法で、制御フローカバレッジ、データフローカバレッジ、モデルチェッキング (モデル検査) などの手法

がある (IPA 2010)。

これらの手法は、検査対象のシステムの機能や要件に応じて、幾つかの手法が組み合わせて利用される。システム機能の検査は、基本的には利用者からのヒアリングやシステムの仕様書をもとにテスト仕様書を作成し、そこからテストケースを選定し、テストケースを検査するためのテストデータを作成し、テストデータを用いてテストを実施する、という流れになる。ここで言うテストケースとは、検査対象システムの機能や要件に対応して、特定の入力や外部環境の変化などが発生した時に、「このように動けば正しい」、あるいは「このような状態や結果となれば間違っている」といった判断基準である (バイザー 1994) (マイヤーズ 2006) (コーブランド 2005)。

本研究では、システム機能テストの手法として、ホワイトボックステストの一種である、モデル検査手法を用いる。モデル検査は、対象システムをモデル化し、検査対象のシステムの振る舞いを仕様 (機器や装置の動作、状態遷移などの外部から観測できる挙動) を記述言語で記述し、モデル検査ツールを用いて、状態空間の網羅的な探索を行い、設計の不備を検証する技法である (Clarke 1999) (Holzman 2004) (早川 2012)。モデルの状態をすべてシミュレーションすることで、状態空間において設計意図通りの状態遷移が存在するか、あるいはデッドロックが存在するか否かなどを、一貫した手順で検証することができる (宮本 2011)。このように、モデル検査はシステム設計開発の上流工程で不具合を早期に発見するための手法として活用されているが (池田 2007) (吉岡 2014)、システム機能を検証するためのテスト用データを生成するための手段としても利用できる。

まず有限の状態遷移空間を構築し、テストすべき基準を決定した上で、検査対象のシステムの違反、すなわちデッドロックを始めとする、起きてはいけない状態を検索するために、システム検査ツールを用いて到達可能な状態空間を徹底的に探索する (Clarke 1999) (村石 2007)。違反が検出された場合は、モデル検査ツールでこの違反がどのように起こるかを示す反例を生成する。この場合、反例は、状態モデルを初期状態から違反が発生する状態にする一連の入力となる。このようなプロセスを、対象モデルの様々な遷移状態に対して繰り返すことにより、システムのテストケースが自動的に得られるとともに、シミュレーションで状態を再現することで、テストデータの自動生成を行うことができる (Heimdahl 2004)。

しかし、システム検査ツールを用いたテストケースとテストデータの生成手法の課題は、①システムモデルの作成の難しさ、②検査条件の作成の難しさにある。具体的には、モデル検証対象のシステムのモデル化と検査条件の定義、そして検査条件を満たすテストケースの生成という一連の作業の中で、整合性をもって必要十分な内容を決定し、正しく形式化して記述するには、形式手法とモデル仕様記述言語、およびモデル検査ツールの使いこなしにおいてかなりの熟練を要す。

この課題を解決するための手法として、検査式をイベント/状態の発生する「パターン」 (発生期間と順序) と「スコープ」 (発生する範囲) として分類するとともに、DB化+ツール化による再利用の容易化する汎用的手法が提案された (Dwyer 1999)。一般的な特性のパ

ターンをテンプレートとして提供する方式(Smith 2002)、さらには、対象をソフトウェア設計に限定した検査式のパターンをデータベース上に保存し、モデル検査の実施者が参照できるようにする手法(金井 2008)などが提案されている。

しかし、既存の研究のパターンやスコープは抽象的かつ汎用的すぎて、SEや現場の技術者では、テストケースとテストデータの生成を行うことは困難であること。また、先行研究で参照できる事例もソフトウェア設計であり、異なるサブシステムの連携で状態空間を遷移するモデルを扱っていることから、本研究が対象とする製造ラインのような、類似の分散型サブシステム(装置や機器)が大量、並列、非同期に動作するような状況に向いていない。そのため、自律分散型の管理システムを持つ、製造ラインの設備や機器の動作テスト、およびシステムの機能テストには、適したシステムモデルを新たに定義する必要がある。

第3章 異なるアーキテクチャを持つシステムの共存

3.1 はじめに

本章では、第2章で整理した先行研究の課題のうち、異なるアーキテクチャを持つシステムの共存を実現する手法について提案するとともに、その有効性を確認するために、実際のシステムに適用し、その効果を評価する。

まず、以下に本研究の提案方式の適用対象となる、タイヤ生産ラインのタイヤ検査工程の制御システムについて概説する。

まずタイヤの生産工程を説明する。前工程において、タイヤの素材となるゴム、ベルト、ワイヤー類を製造する。生産ラインでは、ゴムの原料を混ぜ合わせる「混練」、素材を中間部材として生成する「部材」、部材を組み合わせて生タイヤにする「成型」、金型に入れて加熱・加圧し、化学反応させることで完成品のタイヤの形状に仕上げる「加硫」、形状やバランスなどをチェックする「検査」の大きく5つの工程を経て、出荷に至る。生産ラインの各工程を管理するシステムは、工場全体の情報制御システムの構成要素であり、その主体は各工程の生産計画に基づいて工程内の機器の稼働スケジュールを作成し、機器にタスクの割り当て（ディスパッチング）を行う情報システム、ラインの情報を把握するためのモニタリングシステム、およびラインの機器を操作するための制御システムから成る。

図3.1に検査工程の概観を示す。上流工程から送られてきた完成品のタイヤは、多品種混合でベルトコンベア上を流れている。コンベアから適切な自動検査装置に送り込み、サイズやバランスが適正かを評価する。検査に合格したタイヤは倉庫に運ばれて、出荷を待つ。

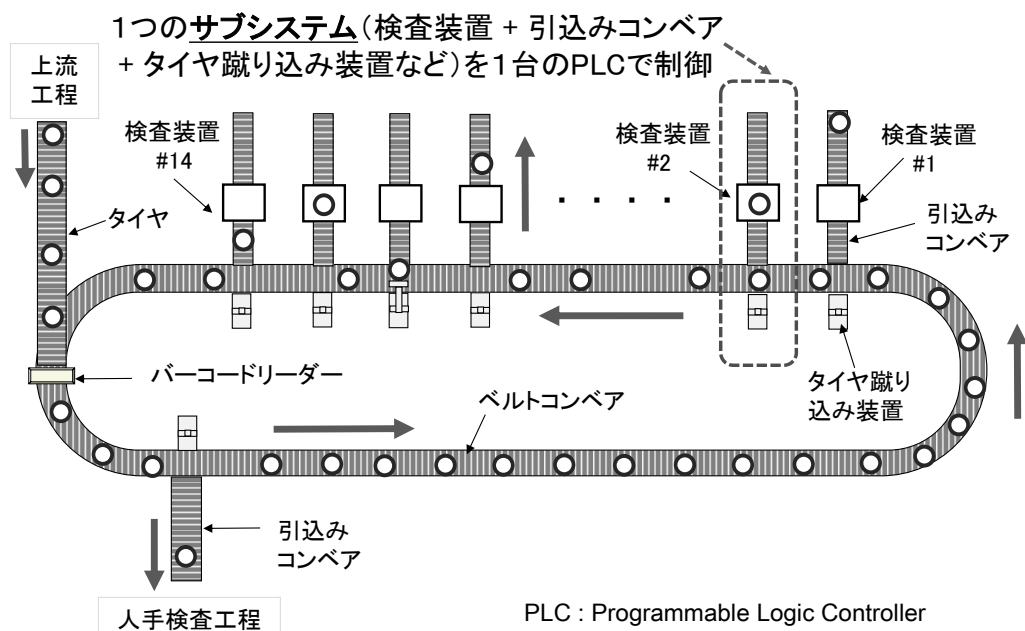


図 3.1 タイヤ生産ラインの検査工程の概観

この工場で製造可能なタイヤは約 2000 種類あるが、1 日の間にラインで作るのは、高々 10～20 種類である。タイヤの種類は貼り付けられたバーコードにより、区別できる。検査装置は、セッティングの調整を行うことで複数の種類のタイヤを自動検査することが出来るが、これはタイヤの構造、内径、外径、幅、などによって制約される。通常、セッティングした、1 台の検査装置で同時に検査可能なタイヤは 2～3 種類である。

各工程を管理する既設のシステムは、当該の工程を統括する集中管理サーバ（上位計算機）と、現場の機器を制御するコントローラや現場の監視・操作などを行う制御 PC や PLC などの下位計算機からなる階層構造を持っている。集中管理サーバ（上位計算機）は、先述の機器の稼働スケジュール作成し、機器へのタスクの割り当て（ディスパッチング）、およびラインのモニタリングを担当する。

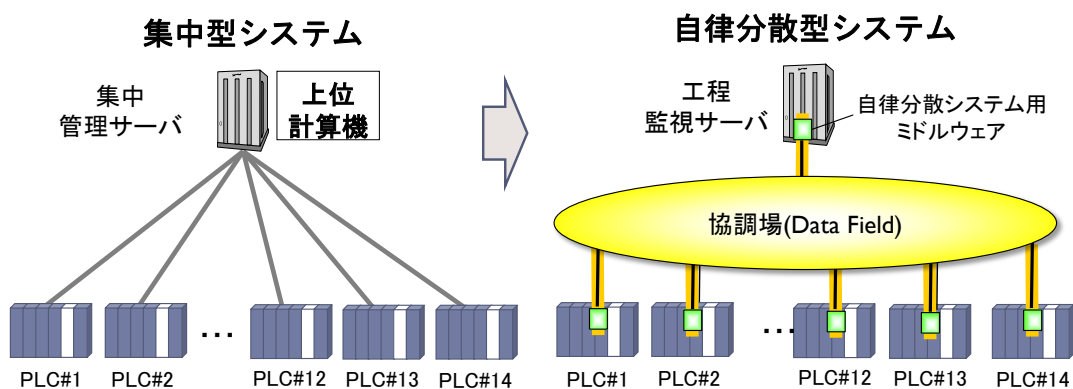
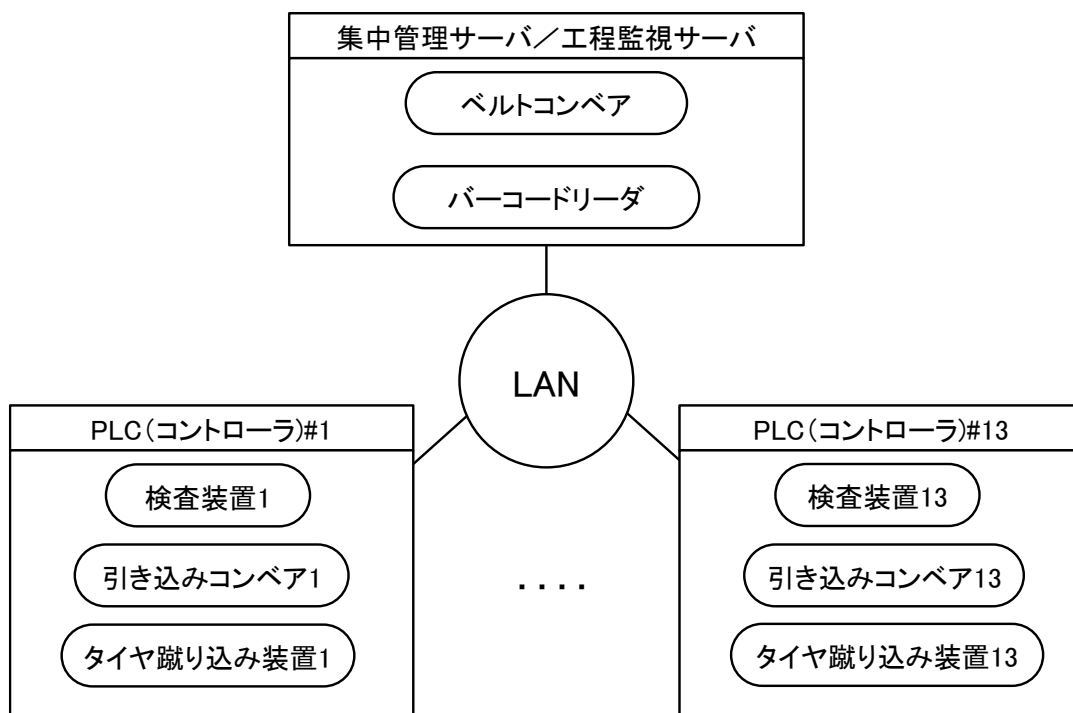


図 3.2 タイヤ検査工程を管理する計算機システムの模式図

図 3.1 のタイヤ検査工程を管理する制御システムの模式図を図 3.2 に示す。14 機の検査装置と、引き込み用のベルトコンベア、タイヤの蹴り込み装置などは、それぞれ 1 台の PLC（下位計算機）が制御している。

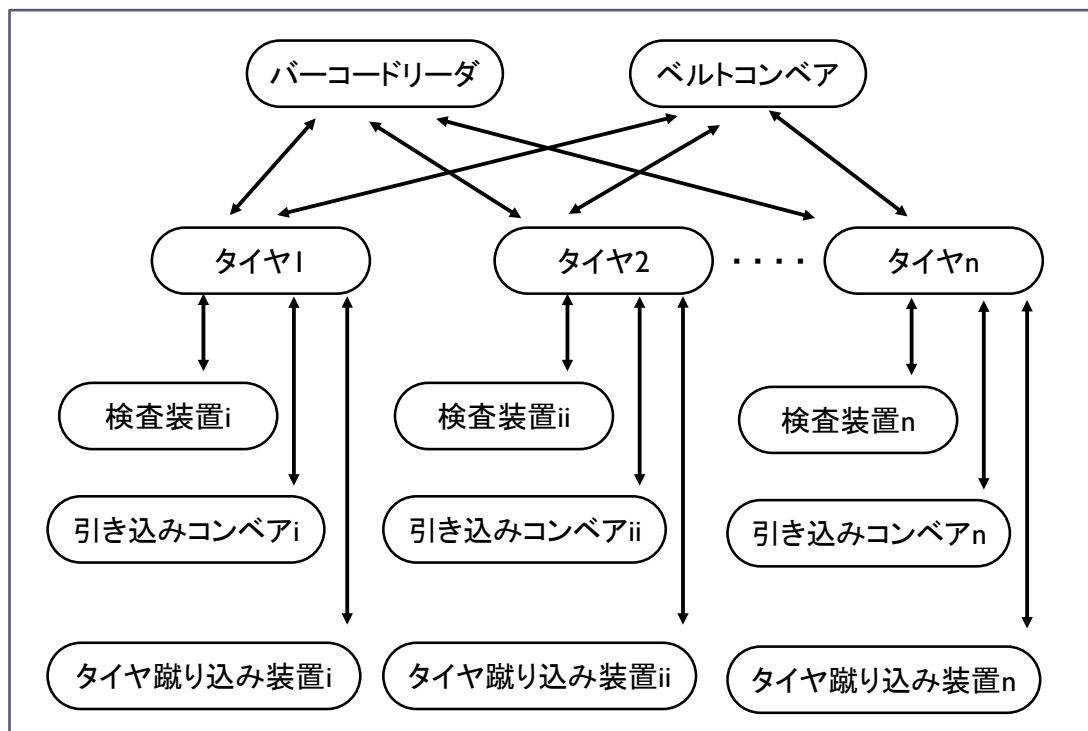
既設の集中型の制御システムの場合、集中管理サーバは、ベルトコンベアに流れてくるタイヤのバーコードを読むことで、検査工程に入ったタイヤの種類と順番を把握する。14 台ある検査装置の稼働状況、ベルトコンベアの速度などは、常に集中管理サーバがモニタしているため、集中的なアルゴリズムで、どのタイヤを、どの検査装置に送り込めば、もっとも効率的に検査を行うことができるかの、最適スケジュールを作成することができる。

図 3.3 に、図 3.1 のタイヤ検査工程の物理的な構成と図 3.2 の工程管理システムとを論理的な構成要素にモデル化し、マッピングした、仮想的な検査工程モデルと、工程管理システムのモデルとを示す。



工程管理システムモデル

↑ システムへのマッピングと実装方式



仮想的な検査工程モデル

図 3.3 タイヤ検査工程と工程管理システムの論理的な構成

本研究では、既存の集中型システムを自律分散型システムに継続的に移行し、テストすることを事例として取り扱う。

生産管理システムのタスク割り振り（ディスパッチング）は、製造システムの分散化が進み始めた 1990 年代から、大きな課題である（鳩野 1995）（藤井 1995）（藤井 1996）。先行研究では、柔軟な生産環境を実現するため、タスクの待ち状況と分散型の生産サブシステムの稼働状況などを把握することで、タスク割り当ての優先度を動的に更新する、分散されたスケジューリング方法が提案されている（翁 2003）（大久保 2005）（Fujii 2012）（Morinaga 2014）（Morinaga 2015）。しかし、本研究で適用例として扱う検査工程のような、タイヤが数秒毎にコンベアに到着し、また多品種混合のため事前にどの種類のタイヤが到着するかを予測できず、一つのタスクも数秒～数十秒という短い時間で完了するタイプの工程の再スケジュールを行うには、先行事例で提案されている方式は適していない。以下に、タスクの割り当て（ディスパッチング）も含めて、検査工程における集中型システムと同等の機能を、自律分散型システムで実現する方式を述べる。

自律分散型システムでは、1 機の検査装置と、引き込み用のベルトコンベア、タイヤの蹴り込み装置などを制御する PLC とで 1 つのサブシステムを構成する（図 3.3）。各 PLC は、自サブシステムの検査装置が検査できるタイヤの種別や稼働状況などサブシステムの状態をリアルタイムに把握している。工程監視サーバは、検査工程全体の集中的な管理や調整を行うのではなく、検査工程の動的な状態を監視し、共有することが業務である。具体的には、ベルトコンベアの速度やコンベア上を流れるタイヤの種別などの情報を協調場を用いて全てのサブシステムに共有すること、検査工程で検査を完了したタイヤの種類と数や検査装置の稼働時間などを集約し、実績を管理して工場の情報システムに報告すること、などが工程監視サーバの通常の役割である。ただし、工程の設備に異常が発生した際には、工程監視サーバがアラームを発信し、異常が重大な場合にはラインの緊急停止などを行う。

工程監視サーバは、上流工程からベルトコンベアに流れてくるタイヤ、および検査されずにベルトコンベア上を回っているタイヤのバーコードを読み、そのタイヤの種別を DF に送出する。各サブシステムを管理する PLC は、ベルトコンベアのスピードを把握しているので、自サブシステムの検査装置が検査できるタイヤが、いつ蹴り込み装置の前に来るかを予測できる。検査装置が検査のために稼働中かどうか、稼働中ならば、あとどれくらいの時間で検査が完了する予定かという予測と突き合わせることで、各サブシステムは、次にどのタイヤをいつ取り込むかという計画を立てることが出来る。また、ベルトコンベアから、タイヤを引き込んだ際には、PLC はそのイベントをメッセージとして DF に同報する。この情報を得て、各 PLC はベルトコンベア上のタイヤの種別と位置とを把握できる。そして、自サブシステムが取り込む予定だったタイヤを別のサブシステムが先に取り込んだ場合にも、自己のタイヤ取り込み計画を自律的に修正できる。

このような仕組みを各サブシステムに実装することで、タイヤ検査タスクの自律的な割り当て（ディスパッチング）を分散システム上で実現でき、また、他のサブシステムの停止

や故障、セッティングの変更などに関係なく、自サブシステムの業務を継続できる。

3. 2 提案方式

方式提案 1：異種システム間連携ゲートウェイで集中型アーキテクチャと自律分散型アーキテクチャのシステムを共存、連携させる手法を提案する。

集中型と自律分散型のアーキテクチャのシステム間でメッセージを交換する場合は、相手のシステムのアーキテクチャの違いによって、それぞれのシステムが処理を変える必要がないよう、異種システム間連携ゲートウェイで、①フィルタリング機能と、②フォーマット変換機能とを提供する。さらには、必要に応じて複数のデータの統合や形式変換などの処理を行って、集中型システムの集中管理サーバに送信する。この場合、ゲートウェイは集中型システムの下位計算機を模擬（エミュレーション）している。これによって、既存の集中型システムのプログラムなどを変更する必要はない。

以下に、方針提案 1 の詳細を述べる。サブシステム単位でアーキテクチャの異なる新サブシステムにリプレースを行うには、まず、既存システム用のネットワークと、新規システム用のネットワークとをゲートウェイ（中継ノード）を介して相互接続することで、異種システムの共存状態を実現する。この場合、必要ないデータを互いのネットワークに転送することによる、計算機やネットワークの負荷の増大、それに伴う本来の業務の遅滞などを防ぐため、中継ノードの計算機には先行研究の協調自律分散システムのゲートウェイ機能を拡張した、①フィルタリング機能と②フォーマット変換機能を提供する。集中型のサーバから自律分散型システムのサブシステムへ送られるメッセージには、ゲートウェイで自律分散型システムの受信処理に必要な「データの種別を表すタグ：Transaction Code (TCD)」の適切な種別を選択の上、メッセージのヘッダに追加して、自律分散型システムのネットワークにマルチキャスト送信を行う。また、自律分散型のサブシステムからマルチキャスト送信されたメッセージのうち、ゲートウェイが集中型システムのサーバに転送すべきものを TCD によって判定し、必要なデータのみを抽出、必要に応じて複数のデータの統合や形式変換などの処理を行って集中型システムのサーバに送信する（表 3.1）。

表 3.1 集中型と自律分散型のシステムを接続するゲートウェイ（中継ノード）の機能

送信元 \ 受信先	集中型	自律分散型
集中型	(既設システム内 通常処理)	・T C D付与 ・D Fへ送付
自律分散型	集中サーバに送るべき データを抽出し転送	(新規システム内 通常処理)

図 3.4 において、ゲートウェイは自律分散システムのサブシステムの構成要素の一つであり、協調場に流れるデータを受信できる。一方、集中型システムの上位計算機の制御サーバからは、ゲートウェイは下位の計算機、サブシステムの一つを制御する PLC としてエミュレートされて見えている。

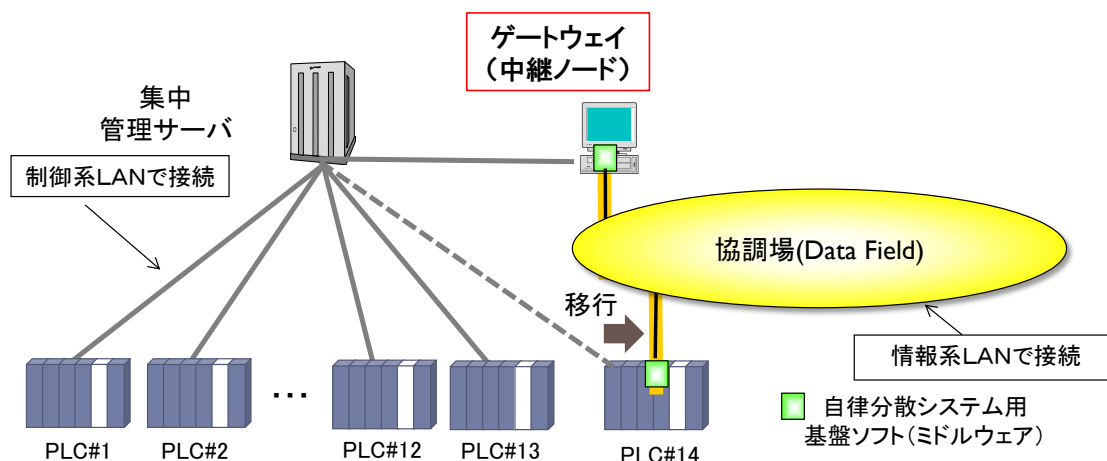


図 3.4 集中型と自律分散型のシステムの共存

集中型システムでは、集中管理サーバは刻々と変化するシステム全体の状況と、各サブシステムの状態を、常に把握している。そして、稼働目的や計画に沿って、システムが最適に動作をするように、個々のサブシステムが行うべき動作を決定し、逐次、下位の計算機に指示を出す。制御システムの場合は、下位の計算機は、その指示に従ってプロセスを制御し、その結果は集中管理サーバに伝達される。

ゲートウェイを集中型システムに接続する際には、集中管理サーバからは、下位計算機と同様に見えるように、入出力を定義する。集中管理サーバからゲートウェイへは、動作の指示を出し、また、ゲートウェイからは、サブシステムの状況や動作の結果、例えば生産管理システムの場合には、制御対象の機器の状態、生産ラインで処理をした製品の数や動作の結果などを集中管理サーバに伝達する。これらの情報は、自律分散システムに接続されている、

新サブシステムを制御する計算機から同報されたメッセージからデータを収集して、作成する。

これらの仕組みによって、自律分散型システムと共存しても、集中管理サーバのプログラムには変更を加える必要がない。また、既存のサブシステムを制御する下位計算機も、自律動作を行うためのプログラムを追加するだけで、既存の制御プログラムの変更は必要なく、受信処理負荷が増えることもない。

図 3.4 では、集中管理サーバに接続されていた PLC#14 を協調場に接続し、自律分散型システムの構成要素に移行した状況を示している。PLC#14 が既存システムの下位計算機であった時と同様の機能を発揮し、また他システムにも悪影響を及ぼさないことを確認して、PLC#14 は移行完了とする。続いて、PLC#13 を協調場に接続し、集中管理サーバからはゲートウェイが PLC#14 に加えて PLC#13 もエミュレートする、という手順でサブシステムを段階的に集中型から自律分散型へ移行する。

3. 3 適用結果

方式提案 1 の有効性を確認するために、実際のタイヤ検査工程の制御システムに適用した。分散型システムでは、集中管理サーバが持っていた、機器の稼働スケジュール作成機能を廃し、各サブシステムを制御する PLC が機器へのタスクの割り当て(ディスパッチング)を自律的に行うよう、機能の分担を変更した。

集中型システムで、例えば PLC#14 が制御しているサブシステム#14 は、分散システムでも引き続き PLC#14 が制御を行うこととした。サブシステム#14 を制御する新しい PLC を準備することも立案されたが、システムアーキテクチャを変更するという大きな試みの際に、可能な限り他の変更部分を減らしたかったことと、新 PLC の調達コスト削減などの理由から、既存の PLC#14 を再利用し、制御対象の検査装置やタイヤ蹴り込み装置などの制御プログラムもそのまま用いる選択を行った。

自律分散システムの協調場は、Ethernet の UDP/IP プロトコルによる同報通信(マルチキャスト通信)により実現した。PLC#14 に Ethernet に接続できるボードを追加し、ここに自律分散システム用基盤ソフトウェアをインストールし、3. 1 節で述べた、自律的なタスクの割り当て機能を実装した。さらに、Windows PC に異種システム間連携ゲートウェイ機能と、集中管理サーバが持っていたラインのモニタリング機能を実装した。3. 2 節の手順で集中型と自律分散型のアーキテクチャのシステムの共存、連携状態での稼働を、特段の問題を起こすことなく達成した。

3. 4 考察

本適用例では、検査工程システムを構成する 14 個のサブシステム(検査装置、引き込み用ベルトコンベア、タイヤの蹴り込み装置などで構成)は、集中型システムの時点でも、検査処理自体は、他のサブシステムからほぼ独立して動作していた。しかし、集中管理サーバ

で、機器の稼働スケジュール作成やタスクの割り当て（ディスパッチング）を集中的に行っていたために、1つのサブシステムの停止や故障などが発生した際に、工程全体の計画やタスク割り当ての再計算が発生し、他のサブシステムへの影響は避けられなかった。

タイヤ製品は、グローバルでの競争が激化し、製品の種類が年々増加し、多品種少量生産への対応が必須となっている。このため生産ラインの装置替えの頻度が上がり、従来の集中型システムでは、その都度、稼働スケジュール作成とタスクの割り当てをやり直す必要があり、工程全体のシステムの停止が避けられなかった。また、装置替えの度に、調整やテストを行う必要があるため、これらの作業負荷の低減も求められていた。

そのため、工程管理システムの自律分散型システムへの移行、およびシステム移行フレームワークの適用は大きな効果を挙げた。具体的な効果は、以下の通りである（奥 1996）。

- ・従来技術では、サブシステムの移行には少なくとも3回（全体システム変更、サブシステム停止、再試験）のシステム停止が必要だった。また、移行準備と移行作業にのべ10か月を要していた。
- ・システム移行フレームワークの適用により、移行準備と移行作業の全体期間が約半分で完了した。全体停止は一度だけ。他は部分停止のみで達成できた。
- ・従来、サブシステムの改造中に全システムの停止が必要であり、1つのサブシステムの移行作業には数日から数週間の時間と多くの作業員が必要としたが、本方式の採用で、新人中心のシステム構築に成功。ソフトウェア設計から、試運転に工数低減。ビルディングブロック化、オンラインテストにより、部分停止を行いながら、約3分の1の期間（実績のべ4日間）で達成。

3. 5 おわりに

本適用例では、検査システムを構成するサブシステムのタイヤ検査処理は、他のサブシステムの業務からほぼ独立して動作しており、集中型から自律分散型の移行の際の課題の中心は、タスク割り当ての問題であったこと。また集中型のシステム構造が単純なツリー型（1個の下位計算機は1個の上位計算機のみと接続されている）であった（図 3.3）。そのため、システム移行フレームワークを適用した集中型から分散型のシステムへの移行に適していた。単純なツリー型ならば、階層構造を持っていても、同様に適用可能と予測される。

一方で、システム移行フレームワークの適用には制約があり、例えば以下のような特徴を持つシステムでは、実施することは困難と予想される。

- ・ツリー型構造でも、下位の計算機が、複数の上位計算機と接続されている場合
- ・下位計算機間で直接通信を行う場合

本章で適用を行ったタイヤ検査工程の従来の集中管理システムでも、下位計算機間での直接通信は存在したが、協調場へのメッセージ送信に切り替えることで、分散型システムへの移行を実現できた。しかし、リアルタイム性への要求が厳しいシステムにおいては、隣接

した PLC 間で通信を行って、動作のタイミングを調整するようなケースは少なくないため、通信性能の要件達成が課題になりうる。

今後、システム移行フレームワークの適用限界、および適用への制約の明確化、また適用範囲を広げるための方式の検討などが求められる。

第4章 共存システムの継続的移行と統合テスト

4.1 はじめに

本章では、第3章で達成した、異種システム間連携ゲートウェイを用いた集中型と自律分散型のシステムの共存状態から、集中型システムの一部を段階的に移行し、新たな共存状態での連携テストを行う手法を提案する。具体的には、集中型システムの一つの下位計算機を、自律分散型システムのサブシステムの一つへと移行させるとともに、自律分散システム技術の一環として開発された、オンラインテスト技術を拡張した手法で、新たな共存状態での新しい自律分散型サブシステムの疎通テスト、単体テスト、および既存の集中型制御サーバとの連携テストを行う。

4.2 提案方式

方式提案2:集中型システムにおける、一つの下位計算機とその下位計算機が制御している機器群、装置群を段階的移行の一単位として扱う。まず、移行する下位計算機に自律分散型サブシステムのプログラムを実装し、異種システム間連携ゲートウェイのフィルタリング機能とフォーマット変換機能の設定を新しいサブシステムが追加された状態に合わせて変更する。続いて、下位計算機と関係する装置や機器を停止して、集中型の制御サーバから切り離して、自律分散型システムへのネットワークへと接続し、自律分散型サブシステムのプログラムを起動する。

方式提案3:上記の方式提案2を用いて実現された、異種システム共存の新たな共存状態での、新しい自律分散型サブシステムのオンライン連携テストを行う。オンライン連携テストは、具体的には、自律分散型システムのネットワークを流れるメッセージと、テスト対象の新サブシステムに、それぞれ「オンラインモード」と「テストモード」の2つのモードを持たせることで、モードの組合せによって、①疎通テスト、②単体テスト、および既存の集中型制御サーバとの③統合テストの3段階のテストを行う。①疎通テストは、集中型の制御サーバと自律分散型システムの新しいサブシステムとが正常に通信でき、正しく起動できることを、②単体テストは自律分散型の新しいサブシステムが所定の機能を果たすことを、③統合テストは集中型システムの制御サーバと自律分散型の新サブシステムとが連携して動作し、システム全体が正常に稼働することを、それぞれ確認する。方式提案2と3の手順を繰り返すことで、共存システムの継続的移行が可能となり、最終的に既存の集中型システムの全体を、自律分散型システムへと移行することが出来る。

3章で、アーキテクチャの異なるサブシステムを共存する方式と適用を述べた。このような、異種アーキテクチャが共存するシステムで、1つのサブシステムを移行する毎に、移行されたサブシステムが正常に動作し、既存のシステムに悪影響を及ぼさないかをテストするため、2.2節のオンライン稼働中テスト方式を、次のように拡張し、対応することを提

案する。

まず、既存システム用のネットワークと、新規システム用のネットワークとを中継ノードを介して相互接続することで、異種システムの共存状態を実現した上で、共存状態における自律分散型の新規サブシステムと、既存の集中型システムの連携テストを、表 4.2 のように 4 つのケースに分類し、順次テストを行う。

以下では、図 3.4 の構成を例に、個々のケースの内容と、テストの手順を述べる。

表 4.2 集中型と自律分散型の共存システムにおけるテスト形態の分類

連携対象 テスト対象	集中型	自律分散型
集中型	既設システム内通常テスト	(ケース3) シミュレーションテスト (ケース4) オンライン連携テスト
自律分散型	(ケース1) 抑制状態テスト (ケース2) オンライン連携テスト	前出の自律分散 システムのテスト形態

(1) 自律分散型サブシステムの抑制状態テスト (ケース 1)

ケース 1 は、自律分散型の新サブシステムをテスト対象とする。新サブシステムをテストモードでテストすることにより、既存の集中型システムには影響を与えない。

本テストでは、ゲートウェイと移行した新規のサブシステム (図 3.4 の場合は、PLC#14) をテストモードにして実行する。PLC#14 はゲートウェイを介して、集中型システムの集中管理サーバからの指示をメッセージとして受信し、プログラムをテストモードで実行する。制御対象への出力は抑止されるが、実行の結果を、テストモードのメッセージとして DF に送信する。ゲートウェイは PLC#14 からのメッセージを受信し、その内容は集中管理サーバには送信せず、検証のためログファイルに保存する。保存されたログファイルの内容を確認することで、PLC#14 で適切な処理が実行されたかを、判定する。この一連の手順で、既存のシステムや制御対象に悪影響を及ぼすことなく、自律分散型の新サブシステムをテストできる。

(2) 自律分散型サブシステムのオンライン連携テスト (ケース 2)

ケース 2 は、自律分散型の新サブシステムをテスト対象とし、既設サブシステムと連携させたテストを行う。本テストでは、ゲートウェイと新規のサブシステムをオンラインモードにして実行する。

図 3.4 において、ケース 1 と同様に、PLC#14 はゲートウェイを介して、集中型システムの集中管理サーバからの指示をメッセージとして受信し、実行の結果をオンラインモードのメッセージとして DF に送信する。ゲートウェイは PLC#14 からのメッセージを受信す

るが、その内容は制御サーバには送信せず、検証のためログファイルに保存する。

本テストで、ケース 1 と異なるのは、既に移行を完了した自律分散型サブシステムとの連携が正常に行われるかどうかを確認される点にある。自律分散型サブシステムの PLC からオンラインモードで送出されたメッセージは、既に移行を完了した他の自律分散型サブシステムがあった場合、そのサブシステムに受信されて、処理が実行される。メッセージを受信した自律分散型サブシステムの内部状態をモニタすることで、連携が正常に行われたかどうかを判定できる。

(3) 集中型システムのシミュレーションテスト (ケース 3)

ケース 3 は、既設システムをテスト対象とし、新サブシステムとの連携をシミュレーションで行うテストである。

本テストでは、ゲートウェイをテストモードで実行する。テスト用ツールを用いて、PLC#14 の状態や動作結果を模擬 (エミュレート) したデータを作成し、集中型システムの制御サーバへと伝達する。制御サーバが、この模擬されたデータを見て、正しく新サブシステムの状態を把握し、次の指示を作成するかどうかを確認する。ここで用いる模擬データは、ケース 1 やケース 2 でゲートウェイが保存したログファイルなどを用いて作成する。

(4) 集中型システムのオンライン連携テスト (ケース 4)

ケース 4 は、実稼働状態での新サブシステムと既設サブシステムとの連携のテストである。新サブシステムをオンラインモードで稼働させるとともに、ゲートウェイも集中型と自律分散型間のメッセージの変換や転送を逐次行う。通常の稼働時と同様の環境で、ソフトウェアの動作や出力データをモニタすることで、以下の問題が発生しないことを確認する。

(i) 制御対象に対する誤制御指示

(ii) 共存する集中型システムと自律分散型システムのサブシステムの処理への悪影響。

タイヤ検査工程の場合は、タイヤの検査タスクの各サブシステムへの割り当てが正常に行われるかどうかを確認する。

方式提案 2 に従って、サブシステム単位の移行作業を行う度に、上記の方式提案 3 のケース 1 から 4 までのテストを実施していく。最終的に全ての PLC を自律分散型システムの構成要素へと移行し、全てのサブシステムの機能を検証したところで、集中管理サーバの集中的アルゴリズムを停止し、自律分散型システムへの移行は完了となる。

4. 3 適用結果

適用 2 : 4. 2 節の方式提案 2 と 3 の有効性を確認するために、タイヤ検査工程の制御システムに適用し、サブシステム単位での段階的移行、及び共存システムの稼働状態での連携テストを達成した。

本方式の適用することで、現場の技術者により、サブシステム単位の移行作業と連携テストが実施できるようになった。現場の技術者は、最初のサブシステムの移行の際には、安全

を期してSEの立会いと助言のもと、約一日をかけて、移行作業と連携テストを行った。しかし、その後、同様の作業を繰り返すことで、SEの立会いは不要となり、現場の技術者のみで、1つのサブシステムの移行とテストと2～3時間の作業で実施できるようになった。

その後、現場の状況を見て、任意のタイミングで、順次PLCを1台ずつゲートウェイにつなぎ変え、自律分散型システムへの段階的移行を達成した。最終的に全てのPLCを分散型システムに移行し、集中管理サーバとの接続を切断することで、工程監視サーバは異種システム間のゲートウェイとしての役割を終え、移行作業は完了した。ここで、集中管理サーバが持っていたシステムの集中管理機能やタスクの割り当て機能は、すべて工程監視サーバと各サブシステムを管理するPLCとに分散配置された。

4. 4 考察

従来、システム移行は、工程全体、全てのサブシステムに及ぶ膨大な作業と、全体停止が必要で必須であった。また熟練のSEが長い準備期間をかけて、移行手順とテスト仕様書を作成していた。

本方式の適用によって、現場技術者向けの移行手順書と連携テスト手順書がいったん整えば、現場の技術者のみでサブシステム単位の移行作業と、連携テストを実施できるようになった。各サブシステムを停止してもいいタイミングや、現場の技術者が手すきになる時間を事前に予想するのは困難であり、現場の判断によって任意のタイミングでサブシステムの移行とテストを実施できることは、移行作業の柔軟度と効率を大幅に向上させた。

特に、年に2回の工場停止期間（年末年始とお盆休み）にしか出来なかった規模のシステムの移行が、移行手順とテスト手順、およびツールなどが整えば、即時に着手できること、また、現場の状況が許せば段階的な移行作業が可能になったことは、市場の変化スピードが速い製造業にとって、大きな利点であり、単に移行作業をSEレスで実施できる以上の効果があった。

本方式を適用したタイヤ製造メーカーは、国内に約10箇所、海外に約40箇所の製造拠点を持っている。今回適用した分散型の工程管理システムと、システム移行フレームワークは、同じ製造拠点の他工程のシステムのみならず、他の製造拠点の類似のアーキテクチャを持つ既存システムにも順次適用されている。

4. 5 おわりに

本事例では、単純なツリー構造を持つ集中型システムから、協調場を用いた分散型システムへゲートウェイを用いて移行する手順と、異なるアーキテクチャを持つシステムの共存状態でのシステム連携テストの方式をタイヤ検査工程のシステムに適用し、その有効性を確認した。

製造業の工程管理システムを集中型システムから分散型システムへと移行させることは、大きなトレンドであるが、同じ企業の同じ製造拠点の他工程の管理システムや他の製造拠

点の工程管理システムでは、既存のシステムが単純なツリー構造ではないアーキテクチャを持つケース、移行したい分散システムが自律分散型ではない、あるいは自律分散型であっても協調場を用いる実装ではないケースなどは少なくない。

今後、システム移行フレームワークの適用限界、および適用への制約の明確化、また適用範囲を広げるための方式の検討などが求められる。

第5章 システム機能を検証するテスト基準、及びテスト用データの容易な作成

5.1 はじめに

システム機能を検証するテスト基準、及びテスト用データの容易な作成では、集中型と自律分散型のシステムの共存状態、及び、すべて自律分散型システムへ移行した状態で、制御対象の機器や装置が正常に動作するか、あるいは想定外の動作をしないかという、システムの機能を検証するための手順と、システム機能検証のためのテスト用データを現場の技術者でも容易に作成できる手法を提案する。

5.2 提案方式

モデル検証対象のシステムのモデル化と検査条件（カバレッジ）の定義を行い、フレームワークを用いて満たすべきテストケースを生成する。そして、モデル仕様記述言語によって、対象システムとその振舞い、およびテストケースに対応する検査条件を記述し、モデル検査ツールを用いて検査条件を満たす、テストデータの自動生成を行う。

分散システムにおいては、並列動作する計算機が相互に影響しあうことによって、起こりうる状態の数が容易に組み合わせ爆発を起こす。そのため、分析的な手法を用いて検証したい状態を網羅的にリストアップし、テストしていくことは困難である。この問題を解決するための手段の一つが、形式検証技術の1つであるモデル検査による検証である。

しかし、システム検査ツールによるテストケースとテストデータの生成手法の課題は、①システムモデルの作成の難しさ、②検査条件の作成の難しさにある。具体的には、モデル検証対象のシステムのモデル化と検査条件の定義、そして検査条件を満たすテストケースの生成という一連の作業の中で、整合性をもって必要十分な内容を決定し、正しく形式化して記述するには、形式手法とモデル仕様記述言語、およびモデル検査ツールの使いこなしにおいてかなりの熟練を要す。

本研究では、現場の技術者が現場の状況、例えば、タイヤ検査工程の場合には、どの検査装置が稼働可能か、個々の検査装置がテスト可能なタイヤの種別、ベルトコンベア上に存在する、或いはテストに用いることの出来るタイヤの種別と数などを観察し、整理することが出来れば、それを用いてテストの手順とテストデータが生成できることを目標とした。そのために、事前に対象のシステムを分析し、システムのモデル化や検査条件の定義、テストケースの生成などを、あとはパラメータさえ入力すれば、自動的に計算できるツールの開発を行った。

しかし、先行研究で参照できる対象システムの事例は、ほとんどがソフトウェア設計であり、異なる機能を持つコンポーネント間の連携で状態空間を遷移するモデルを扱っていることから、本研究が対象とするタイヤ検査工程のような、類似のサブシステム（装置や機器）が大量、並列、非同期に動作するような状況に向いていなかった。本研究では、対象とするタイヤ検査工程の生産管理システムの分析やモデル化に適するよう、モデル検査手法の手

順と用語を整理し直した。

方式提案4：本研究では、モデル検査の一連の手順を、定義設計のガイドやフレームワークとして以下のように体系化し、類似、或いは同等の機能を持つサブシステムが大量、並列、非同期に動作するようなシステムに適した手順と用語に整理し直した。

- ① 対象システムを分析し、システムを構成し、機能を実現するために必須かつ独立に動作できる要素を抽出する。
- ② 抽出された要素を「検査したい対象」と、検査したい対象の振る舞いに影響を与える「環境」の2種類に分類する。
- ③ 「検査したい対象」ごとの「視点」から検査したい条件（カバレッジ）の定義を行う。
- ④ 検査したい条件から、フレームワークを用いることで、テストケースを生成する。テストケースには優先順位をつけ、検証すべきケースを確定する。
- ⑤ モデル記述言語を用いて、検査対象システムのモデル、およびテストケースを実装し、検証記述を作成する。「環境」は検証記述のパラメータとして定義される。
- ⑥ 検証記述にモデル検査ツールを適用し検証する。
- ⑦ 検証が正しく完了したのちに、再びシミュレーションを行い、検証された状態を再現する。ランダムシミュレーションを繰り返すことで、現実のシステムの機能を検証するための、可能な限り最短のテストデータのセットを得ることができる。

5. 3 適用結果

適用3：方式提案4の有効性を確認するために、先ず手順①～④を適用1で用いたタイヤ検査工程の制御システムに適用した。システムの要素のうち検査したい対象を「検査装置」と「タイヤ」に、環境を「ベルトコンベア」にそれぞれ抽出、分類し、パラメータを決定した。続いて、検査装置とタイヤの視点からカバレッジを定義し、フレームワークを用いて、検証すべきテストケースのセットを生成した。以下に詳細を述べる。

- (1) システムを構成する必須かつ独立に動作できる要素を抽出する
対象システムの場合：タイヤ、検査装置、ベルトコンベアの3要素のみとする。
タイヤ蹴り込み装置や引き込みコンベアは、タイヤと直接の関与を持つ要素であるが(図3.3)、タイヤの種別に関係なく動作する装置であり、テストケースの定義にも関係しない要素であるため、モデル化の要素からは除外した。
- (2) 構成要素をシステムの①制御対象となるものと②環境を成すものとに分類する
対象システムの場合：①タイヤ、検査装置 ②ベルトコンベア
検査工程の主要業務は、タイヤを適切に検査することであるため、ベルトコンベアは、多品種混合のタイヤを検査装置まで運んでくる「環境」と分類した。
- (3) 制御対象となる要素のそれぞれの視点で、満たすべき条件（カバレッジ）を優先順位の高い順に列挙する

例（一部。全体は後述）：

■検査装置視点：全ての検査装置が（どの種類のタイヤでもよいので）少なくとも一度は検査する

■タイヤ視点：全てのタイヤ種類で、少なくとも一度は検査される

(4) シミュレーションを行うための、構成要素のパラメータを決定する

・カバレッジの具体化

(3) のプロセスを以下のように、具体的な対象システムを前提に詳細化した。これらの条件を満たすことで、全ての検査装置が適切に検査を行い、また全ての種別のタイヤが適切に検査されるとともに、不適切な検査は行われぬ。

■検査装置視点：【前提】各検査装置で検査可能なタイヤの種類は定義済み

1. 全ての検査装置が（どの種類でもよいので）少なくとも一度は検査する
2. 全ての検査装置が、その装置で検査可能な種類のタイヤを各種類について少なくとも一度は検査する
3. 2に加え、全ての検査装置が、その装置で検査不可能な種類のタイヤをどの種類でもよいので、少なくとも一度は検査しないで見送る
4. 2に加え、全ての検査装置が、その装置で検査可能な種類のタイヤをどの種類でもよいので、少なくとも一度は検査中の状態で見送る
5. 全ての検査装置が、その装置で検査可能な種類のタイヤの任意の順序について、少なくとも一度は検査する
6. 4に加え、全ての検査装置において、一度見送られたタイヤがベルトコンベアを回って次回以降に検査される

■タイヤ視点

1. 全てのタイヤ種類で、少なくとも一度は検査される
2. 全てのタイヤ種類で、少なくとも一度は検査不可能な検査装置に見送られる
3. 全てのタイヤ種類で、少なくとも一度は検査可能な検査装置が検査中のため見送られる

適用 4：方式提案 4 の有効性を確認するために、適用 3 の結果に手順⑤～⑦を適用し、検証すべきテストケースのセットに対して、具体的なパラメータのセットを指定し、モデル記述言語 PROMELA を用いて、検証記述を作成した。この際、実際のタイヤ検査工程制御システムのサイズではなく、構成要素の数を最小にしたパラメータセットで検証記述を作成した。これは、生成されたテストデータを分析することで、方式提案 4 の有効性を確認するためである。

PROMELA で記述した検証記述にモデル検査ツール SPIN を適用し、自動的にモデルとテストケースの検証を行った。その後、SPIN を用いてランダムシミュレーションを行い、

可能な限り最短のテストデータのセットを自動的に得た。適用システムの場合は、検査装置の動作を確認するために、ベルトコンベアに投入するタイヤの順番、すなわち、どの種類のタイヤをどの順番で何本投入すべきか、という条件である。テストデータのセットを分析することで、方式提案4の有効性を確認した。以下に詳細を示す。

まず、(1) 応用システムの構成要素を最小にしたパラメータセットでモデル化とテスト用データの生成を行い、提案手法の有効性を確認する。その後、(2) 応用システムの実際のスケールに合ったパラメータに変更し、実適用可能なテストデータを得る、という2段階を踏む。

(1) 構成要素を最小にしたパラメータセットでモデル化とテスト用データの生成シミュレーションを行うための、最小構成要素を以下の通り決定する (図 5.1)。

- ・タイヤ種類は3種：A、B、C
- ・検査装置は3台：検査装置#1はAとBを、検査装置#2はBとCを、検査装置#3はAとCをそれぞれ検査可能
- ・タイヤ検査にかかる時間：タイヤがコンベアを2つ分動く間（平均）
- ・検査装置の間隔：タイヤ5つ分

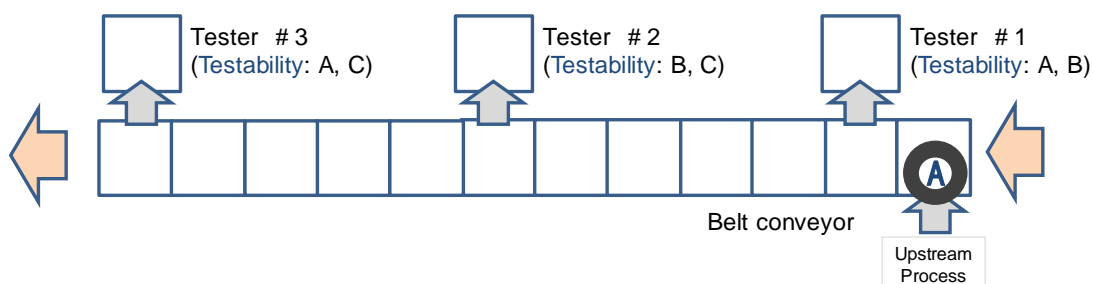


図 5.1 タイヤ検査工程の最小セットの構成要素

SPIN でランダムシミュレーションを行うことで、カバレッジを満たすタイヤの投入順序を求めると、一例として、AACBCB が得られた。

(2) 応用システムのパラメータに変更し、実適用可能なテストデータを得る

(1) のパラメータを以下のような実システムで最も複雑な構成や条件に拡張し、提案の手法でテストデータを生成した：

- ・検査装置 14 台（各装置で検査可能なタイヤは 2 種類、または 3 種類）
 - ・タイヤの種類：10 種類
- シミュレーション結果：テスト用タイヤ投入順序：50～70本でカバレッジを充足

5. 4 考察

本方式の適用によって、システム機能テスト手順書が整えば、現場の技術者がサブシステムの移行完了後に、システム機能テストを実施できるようになった。

従来は、工程を全停止した状態で、大規模なシステム移行を行っていたので、SE作成したテスト仕様書に従ってテストを実施していた。しかし、現場の技術者がサブシステムの移行作業を行い、同時にテストも行う状況となると、その場でテストデータを作成する必要が生じる。それは、ラインがどのような状態であり、どのタイヤがベルトコンベア上やストックに存在するか（テストに使えるか）などは、事前に予測することは不可能だからである。

そのため、現場の技術者が見て、確認できる内容、すなわち動作可能な検査装置の台数と、それぞれの装置が検査できるタイヤの種類を確認すれば、テストデータの作成できること。そして、現場の技術者が実際に操作できる内容、すなわちベルトコンベアの速度を調整し、指示された種類と順番と間隔でタイヤをコンベアに投入しさえすれば、テストを実施でき、システム機能を確認できることは非常に重要であり、本方式の適用の意義が大きい。

なお、本方式では、タイヤ検査装置が3台、タイヤ種別が3種類の最小構成のケースで、シミュレーション深さ1万の場合は、Windows ノート PC で608秒で計算が完了した。

しかし、検査装置14台、タイヤの種別が10種類という、実際の工程で起こりうる最も複雑な条件の場合は、Windows Server で平均して約10時間半の計算時間を要し、現実的な時間内に解を得ることは出来なかった。現実の検査工程では、このような複雑な条件が発生する場合はまれだが、今後、テスト条件を間引く（検査装置やタイヤの数を一部のみでテストする）などの運用を検討し、その場合でもシステム機能の検証として問題ないかの確認を行っていききたい。また、シミュレーションの際に、重複する状況を排除するなどの改善を行い、計算時間短縮方式も併せて検討していききたい。

5. 5 おわりに

本方式は、先ずタイヤ検査工程に適用されたが、他の工程にも援用が容易と考える。今後、他工程のシステム機能テストの要件を確認し、本方式の適用と拡張を検討していく。

また、本方式は、類似の機能を持つサブシステムが大量、並列、非同期に動作する分散型システムに適した、モデル検査手法の手順を整理したものである。このような特徴を持つ分散システムならば、生産管理システムに限らず、適用可能と考える。

今後、本手法の適用限界、および適用への制約の明確化、また適用範囲を広げるための方式などを検討していききたい。

第6章 結論と課題

6.1 結論

1980年代までに構築された、情報制御システムは集中型のアーキテクチャを持つシステムが多かった。集中型のメリットは、上位の集中型管理計算機がシステム全体の状態を把握することで、業務のスケジューリングやリソースの割振りの最適化を図ることが可能な点にある。しかし、システムの大規模化に伴い、システムの一部に故障する確率が上がり、またシステム全体の情報を収集することが困難になってきた。このような状況では、システムを分散化し、下位の装置でローカルで情報を収集し、タスク割振りを自律的に行う方式が有効となってきた。集中型から分散型へのシステム移行は大きなトレンドである。

システムの主な移行方式は3種類あり、大きく(1)一括移行方式、(2)段階移行方式、(3)並行運用方式に分類される。一括移行方式は、ある時点で現行システムを休止して切り離し、全機能を新システムへと一気に切り替える方法である。段階移行方式は、業務や機能といった、ひとまとまりの単位ごとに現行システムを休止し、順次、新システムに切り替えていく方法である。並行運用方式は、現行システムと新システムを同時並行で稼働させ、結果を比較検証しながら移行する方式である。

タイヤ生産システムの移行の場合、停止期間を長く取れないことと現場のスペースなどの理由から一括移行方式は、またコストの問題から並行運用方式は、それぞれ実現が困難である。そのため、段階移行方式、具体的には工程の各ラインを管理するサブシステムを一つずつ移行していく方式を採用している。しかし、従来の移行とテストの方式では、少なくとも、新サブシステムの接続、新サブシステムのテスト、新サブシステムへの切り替えの3回のタイミングで、現行システムも含めた工程全体のシステムを停止する必要があった。そのため、これまでは工場が長期に休業する年に2回の機会(夏休みと年末年始)にしか、工程単位の移行作業は実施できなかった。

しかし、本論文で提案する方式を採用することによって、既存サブシステムから新サブシステムに切り替える時のみ1回の停止で可能になり、またその際にも全体システムを停止する必要はなく、移行するラインのサブシステムのみ範囲を狭めることができた。適用事例の場合には、14ラインが稼働しているため、影響範囲は14分の1である。

また、新サブシステムのテストも、工程の他のラインは通常の稼働をさせつつ、行うことができるため、通常操業時にも工程単位の移行を徐々に実施できることが出来るようになった。これらの効果により、移行とテストにかかるのべ時間を約3分の1に圧縮することが出来た。

本研究の貢献は、集中型システムの機能を自律分散型システムで実装し、タスク割り当て処理の分散化を実現すると同時に、障害や変更へのロバストネスを高めたこと。また、SEによる事前の膨大な準備と立会いを必要とし、長期の工場休止時期のみに実施できる従来のシステム移行方式ではなく、現場の技術者が平常時の工場稼働状態で、システムの段階的な移行とオンライン状態での連携テストを行える方式を実現したこと。さらに、現場の技術

者によって、システム機能の検査を行うテストデータの作成も容易に行える、継続的なシステム移行のための手順と方式の体系を提案し、その有効性を確認したことにある。

また、システム移行とテストに要する総時間の短縮、影響範囲の限定化を確認した。

6. 2 今後の課題

今後の課題を、以下の項目に整理する。

(1) システム移行フレームワークの適用可能性の探索

システムアーキテクチャレベルで、どのようなシステムからどのようなシステムへ移行するか、適用可能性の探索が期待される。

本稿では、単純なツリー構造を持つ集中型から協調場を用いた自律分散型への移行という事例を扱ったが、このケースにおいても、既存システムと新規システムでの機能分担のパターンは複数考えられる。また、移行途中のシステム機能や性能の縮退を許すか、既存と同等のものを維持する必要があるかによっても、移行手順の実装設計に制約が生じるであろう。

(2) 適用事例以外の実システムの適用可能性の探索

タイヤ検査ラインの管理システム以外の、生産管理システムをはじめ、他分野のアプリケーションの実システムへの適用可能性の探索が期待される。

また、実システムにフレームワークを適用する際の制約と回避手法にも様々な課題がある。適用事例であるタイヤ検査工程の既存の集中管理システムでは、下位計算機間での直接通信が存在したが、協調場へのメッセージ送信に切り替えることで制約を回避することが出来た。リアルタイム性や処理性能への要件が厳しい場合に、制約の回避を行う手法の検討が求められる。

さらに、モデル検査手法の適用限界、および適用への制約の明確化、また適用範囲を広げるための方式の検討も求められる。

(3) ランダムシミュレーションによるテストデータ生成の時間短縮方式

システムの大規模化や複雑化に伴い、ランダムシミュレーションによるテストデータ生成のための計算時間は階乗的に増大する。

シミュレーションの組合せの枝刈りや、テスト条件の間引きなどによって、計算量を圧縮し、現実的な時間範囲内で計算が完了する方式が期待される。同時に、枝刈りや間引きなどを行った場合も、システム機能の検証として問題ないかの検証が必須である。

謝辞

本論文の研究指導をいただいた寺野隆雄先生には、長期の在学中にわたり御指導と御鞭撻をいただき、深く御礼申し上げます。寺野研究室の吉川厚先生、ならびに最後の1年間にご指導をくださった出口弘先生には、異なる視点からのご助言をいただき、検討を深めることができました。

2015年に東京工業大学に入学して以降、寺野研究室の社会人ゼミにて、先輩の國上真章さんを筆頭に、ゼミのメンバの皆様より、多様な経験と知見に基づいた議論と、多面的かつ忌憚ないご意見をいただきました。ここでの議論とご意見がなければ、本研究を論文として纏めることは困難でした。また、企業での研究開発活動では得られない経験となりました。

株式会社日立製作所で、本論文に関わりの深い研究開発をともに行った、鮫嶋茂捻さん、矢野浩仁さん、小川秀人さんには、本学での論文執筆にご理解、ご協力をいただきました。特に、モデル検査手法に関する検討と提案に関しては、小川さんのご助言とご支援なしには決して形にならず、寺野研のOBでもある矢野さんの継続的な叱咤激励がなければ、間違いなく途中で挫折していたと思います。重ねて御礼申し上げます。

また、本論文の審査員である、山村雅幸先生、三宅美博先生、石井秀明先生、小野功先生には、論文の構成が十分に定まらぬ早期の段階から、長期間にわたってご議論、ご意見をいただき、感謝の念に堪えません。また、各種の論文を投稿させていただいた、論文誌、各学会で査読を担当して下さった先生方からは、大変丁寧かつ的確な意見をいただき、本研究の不十分な部分を補足することが出来ました。

ここに、本研究を支えてくださった全ての方々に、改めて御礼申し上げます。

最後に、家庭にあって、私を終始支えてくれた妻 有多子と、執筆の応援をしてくれた子どもらに感謝します。

文献

(浅川 2014) 浅川 直輝、北川 賢一：「2015 年問題」が IT 業界に迫る覚悟、日経コンピュータ、2014 年 2 月 6 日号、pp.26-28

(石田 1993) 石田 好輝：自律分散システムの情報処理の特徴、J.SICE、Vol.32、No.10 (1993 年)、pp.830-836

(稲荷 2015) 稲荷 将、高柳 洋一、東 隆男：IoT のハブを担う高信頼かつ高付加価値の産業用コンピュータ、東芝レビュー、Vol.70、No.10 (2015 年)、pp.18-22

(翁 2003) 翁 嘉華、大久保 寛基、大成 尚：工程割り付けと作業順序付けを同時に考慮する組み立てショップスケジューリングの提案、日本経営工学論文誌、Vol.54、No.4 (2003 年)、pp.245-252

(大久保 2005) 大久保 寛基、翁 嘉華、中村 敬一、大成 尚：オンライン生産制御におけるジョブの工程ルート設定のための設備選択ディスパッチングルールの研究、日本機械学会論文集 C 編、Vol.71、No.706 (2005 年)、pp.2028-2034

(岡田 2014) 岡田 叔之：社会情報システムの変遷と今後の展望、三菱電機技報、Vol.88、No.9 (2014 年)、pp.36-39

(岡 2013) 岡 大勝、矢口 竜太郎：動き出した 基幹系×クラウド [3] クラウドへの移行手順、日経 SYSTEMS、2013 年 7 月号、pp.44-51

(奥 1996) 奥 雅春、伊藤 俊彦、溝上 宏、大村 賢、吉沢 隆司、阿部 雅樹：市場変化への即応をこたえるオープン自律分散情報制御システム -株式会社ブリヂストンにおけるタイヤ生産システムへの適応事例-、日立論評、Vol.78、No.4 (1996 年)、pp.37-42

(鬼澤 2016) 鬼澤 亮、高村 稔子、田中 将貴、本橋 修一：グローバル時代の多品種少量製品における IoT を利用した次世代生産システム -日立製作所大みか事業所での取り組み-、日立評論、Vol.98、No.03 (2016 年)、pp.45-48

(小野 1998) 小野 力、清宮 英雄、海東 健一、菊地 邦行、角本 喜紀：広域鉄道システムにおけるソフトウェア保守のための分散管理方式、電気学会論文誌 D (産業応用部門誌)、118 巻 5 号 (1998 年)、pp.637-644

- (梶原 2015) 梶原 繁、梅田 裕二、三村 昭弘：計測・制御システムの動向と展開、東芝レビュー、Vol.70、No.10 (2015 年)、pp.2-5
- (金井 2008) 金井 勇人、岸 知二：ソフトウェア設計に対するモデル検査のための検証パターン、情報処理学会論文誌、Vol.49、No.10 (2008 年)、pp.3493-3507
- (河野 1988) 河野 克己、森 欣司、織茂昌之：高信頼化のための自律適用制御技法の一考察、計測自動制御学会 第 6 回自律分散シンポジウム(1988 年)、pp.97-100
- (河野 2002-1) 河野 克己、鮫嶋 茂稔、足達 芳昭、薦田 憲久：情報制御システムにおける制御系保護を狙いにした異種データのマルチモード・フィルタリング方式、電気学会 D 分冊論文集、Vol.122、No.1 (2002 年)
- (河野 2002-2) 河野 克己、藤原 和紀、西島 英児、野里 雅也、薦田 憲久：自律分散オンラインテストにおける制御プログラム検証方式、電気学会論文誌D (産業応用部門誌)、122 巻 6 号 (2002 年)、pp.575-581
- (北原 1999) 北原 文夫、自律分散型列車運行管理システム、電気学会誌、119 巻 2 号 (1999 年)、pp.80-82
- (倉田 2013) 倉田 祐介、久一 大介：大規模ミッションクリティカルシステムにおける移行計画、UNISYS TECHNOLOGY REVIEW、第 118 号 (2013 年)、pp.129-140
- (栗林 2016) 栗林 健、粕谷 祥一、竹内 拓也、秋田 佳稔、高橋 創：IoT 時代における鉄鋼制御システムの進展、日立評論、Vol.98、No.03 (2016 年)、pp.33-36
- (厚生労働省 2016) 厚生労働省：求人倍率の高い職業の動向、労働市場分析レポート 第 61 号 (平成 28 年 1 月 29 日)
- (小林 1993) 小林 重信：自律分散システムからの新たなる飛翔、計測と制御、Vol.32、No. 10 (1993 年)、pp.858-861
- (コーブランド 2005) リー・コーブランド=著/宗雅彦=訳：『はじめて学ぶソフトウェアのテスト技法』、日経 BP 社/ (『A Practitioner's Guide to Software Test Design』の邦訳)、(2005 年)

(笹嶋 1996) 笹嶋 広和、大森 勝美、西川 淳彦、谷 正之：分散型電力情報制御システムを支えるハードウェア・基本ソフトウェア、日立論評、Vol.78、No.2 (1996 年)、pp.9-14

(柴宮 2013) 柴宮 理、弘田 達夫、柴田 浩司：高度化するユーザーニーズに応える産業用コントローラの新技术と展開、東芝レビュー、Vol.68、No.10 (2013 年)、pp.14-17

(情報処理推進機構 2013) 独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター：アシュアランス技術を用いた鉄道信号の革新、先進的な設計・検証技術の適用事例報告書、2013 年度版 PART II 設計事例、II-7-II-15

(杉村 1994) 杉村 延広：自律分散型スケジューリング、計測と制御、第 33 巻 第 7 号 (1994 年)、pp.585-589

(鈴木 1990) 鈴木 靖雄、堀 真司：鉄鋼プラント用コンピュータシステム、電子情報通信学会誌、Vol.73、No. 11(1990 年)、pp.1190-1196

(田中 1997) 田中 玲子、新 誠一：階層型分散システムにおける自律可制御性、電気学会 C 分冊論文誌、Vol.117、No.7 (1997 年)、pp.850-855.

(中村 2016) 中村 和海、森永 英二、若松 栄史、新井 栄司：高度分散フレキシブルジョブショップスケジューリングの一手法、日本機械学会 2016 年次大会

(日本電機工業会 2017)：飛躍するプログラマブルコントローラ、2017~2018 年版パンフレット・日本電機工業会

[https://www.jema-net.or.jp/jema/data/S5210\(20171122\).pdf](https://www.jema-net.or.jp/jema/data/S5210(20171122).pdf)

(バイザー 1994) ボーリス・バイザー＝著／小野間彰、山浦恒央＝訳：『ソフトウェアテスト技法——自動化、品質保証、そしてバグの未然防止のために』、日経 BP 出版センター／(『Software Testing Techniques, 2nd Edition』の邦訳)、(1994 年)

(バイザー 1997) ボーリス・バイザー＝著／小野間彰、山浦恒央、石原成夫＝訳：『実践的プログラムテスト入門——ソフトウェアのブラックボックステスト』、日経 BP 社／(『Black-Box Testing: Techniques for Functional Testing of Software and Systems』の邦訳)、(1997 年)

(萩谷 2008) 萩谷 昌巳、吉岡 信和、青木 利晃、田原 康之：SPIN による設計モデル検証

ーモデル検査の実践ソフトウェア検証、近代科学社 (2008 年)

(橋本 2004) 橋本 親、中島 千尋、黒江 潤一：情報制御システムの現状と展望、富士時報、Vol.77、No.6 (2004 年)、pp.2-7

(鳩野 1995)鳩野 逸生、橋 一成、馬野 元秀、田村 坦之：フレキシブル生産を対象とした分散型リアルタイムスケジューリング、計測自動制御学会論文集、Vol.31、No.1 (1995 年)、pp.108-115

(林 1995) 林 慶治郎、河野 克己、綿谷 洋、佐々布 昭義：産業情報制御システムの成長を支える協調自律分散システム、日立論評、Vol.77、No7 (1995 年)、pp.9-14

(早川 2012) 早川 浩司：モデル検査の検査式作成手法、JAXA/IPA 10th Workshop on Critical Software System (WOCS)、(2012 年)

(藤井 1995) 藤井 進、森田 浩、松本 博忠、高田 芳治、龍田 康登：FMS におけるジョブ及び工具配分問題に対するオークション方式の適用、第 5 回インテリジェント FA シンポジウム講演論文集 (1995 年)、pp.25-28

(藤井 1996) 藤井 進、森田 浩、松本 博忠：工具配分制約型 FMS におけるスケジューリング法、1996 年度日本オペレーションズ・リサーチ学会 周期研究発表会

(船橋 1996) 船橋 誠壽、林 利弘、佐々布 昭義、諸岡 泰男：産業社会のグローバル化と変革にこたえる新情報制御システムの技術動向、日立評論、Vol.78、No.10 (1996 年)、pp.4-8

(マイヤーズ 2006) グレンフォード・J・マイヤーズ、トム・バジェット、テッド・M・トーマス、ユーリー・サンドラー＝著／長尾真＝監訳／松尾正信＝訳：『ソフトウェア・テストの技法〈第 2 版〉』、近代科学社 (『The Art of Software Testing: 2nd ed』の邦訳)、(2006 年)

(三菱電機技報 2018) 三菱電機技報：特集：最新の FA 技術・システム、2018 年 04 月号
<http://www.mitsubishielectric.co.jp/corporate/giho/1804/index.html>

(実森 2007) 実森 仁志：最後の難関システム移行、日経 SYSTEMS、2007 年 5 月号、pp.15-35

(宮本 2011) 宮本 直樹、和崎 克己：UML シーケンス図の構造記述から線形時相論式への自動変換技術、第 10 回情報科学技術フォーラム(FIT2011、B-028 (2011 年)、pp.311-314

(村石 2007) 村石 理恵、服部 彰宏、野村 秀樹、山本 訓稔：Model Checking を適用した実践的非同期制御検証、ソフトウェアテストシンポジウム(JaSST)、(2007 年)

(百武 2015) 百武 博幸、GUAN Zhenzhen、浜口 博一：MES ソリューションとの連携でプラントの高度情報化を実現する産業コントローラ、東芝レビュー、Vol.70、No.10 (2015 年)、pp.14-17

(森田 2014) 森田 英昭(2014)：FA 用コントローラの技術革新と適用分野拡大、三菱電機技報、Vol.88、No.9 (2014 年)、pp.22-27

(森 1984) 森 欣司、宮本 捷二、井原 廣一：自律分散概念の提案,電気学会 C 部門論文誌、Vol.104、No.12 (1984 年)、pp.303-310.

(森脇 1992) 森脇 俊道、杉村 延広、Yatna. Y. MARTAWIRYA、蜷川 達也：自律分散型生産システム設計と運用に関する研究、日本機械学会論文集 (C 編)、58 巻 549 号 (1992 年)、pp.340-345

(森脇 1993)森脇 俊道、上田 完次、杉村 延広、Yatna Y. MARTAWIRYA、蜷川 達也：自律分散型生産システムの設計と運用に関する研究 (第 2 報)、日本機械学会論文集 (C 編)、59 巻 568 号 (1993 年)、pp.372-377

(山本 2013) 山本 秀典、加藤 博光、鮫嶋 茂稔：制御システムにおける異種システム間相互接続のためのミドルウェア連携、計測自動制御学会論文集、Vol49、No8 (2013 年)、pp.780-786,

(吉岡 2014) 吉岡 信和、田辺 良則、田原 康之、長谷川 哲夫、磯部 祥尚：モデル検査による設計検証、コンピュータソフトウェア、Vol.31、No.4 (2014 年)、pp.44-65

(FAOP 1999) 財団法人 製造科学技術センター FA オープン推進協議会 (FAOP: FA Open Systems Promotion Forum) 分散型製造システム専門委員会：自律分散プロトコル仕様書 R3.0 (1999 年)

(ITZBund 2005) ITZBund V-Model XT (2005 年), <http://www.v-modell-xt.de>

(IPA 2010) 独立行政法人 情報処理推進機構 (IPA) ソフトウェア・エンジニアリング・センター: 高信頼化ソフトウェアのための開発手法ガイドブック -予防と検証の事例を中心に - (Ver.1.0)、(2010 年)

(OKI 2018) OKI テクニカルレビュー : IoT 時代のモノづくりイノベーション特集、第 231 号 Vol.85、No.1 (2018 年)

(Brisky 2007) Russ Brisky, Siemens Energy & Automation, Inc.: simatic s5 to s7 migration whitepaper (March 2006, Updated June 2007) Public deliverable.
<https://www.siemens.com/content/dam/webassetpool/mam/tag-siemens-com/smdb/regions/japan/di/customer-service/s5/s5-simatic-s5-s7-renewal-manual-en.pdf>

(Cala 2017) A. Cala, A. Luder, A. Cachada, F. Pires, J. Barbosa, P. Leitao, and M. Gepp: Migration from Traditional towards Cyber-Physical Production Systems, 15th International Conference on IEEE Industrial Informatics (INDIN),(July 2017), pp. 1147-1152

(Clarke 1999) E.M. Clarke, O. Grumberg, and D.A. Peled: Model Checking. MIT Press, (1999)

(Delsing 2012) J. Delsing, F. Rosenqvist, O. Carlsson, A. W. Colombo, and T. Bangemann: Migration of Industrial Process Control Systems into Service Oriented Architecture, 38th Annual Conference on IEEE Industrial Electronics Society (IECON), (Oct. 2012), pp.5786- 5792

(Dwyer 1999) Matthew B. Dwyer, George S. Avrunin, and James C. Corbett: Patterns in property specifications for finite-state verification, ICSE '99 Proceedings of the 21st international conference on Software engineering (1999), pp.411-420

(Fujii 2012) Fujii, S., Fujii, N., Iwamura, K., Morinaga, E., Tsumaya, A., Inoue, T., and Mariyama, T: A Basic Study on A Highly Distributed Simulation of Manufacturing Systems under Ubiquitous Environment, Proceeding od ASME/ISCIE 2012 International Symposium on Flexible Automation (2012)

(Heimdahl 2004) Mats P E Heimdahl, Devaraj George, Robert Weber: Specification Test

Coverage Adequacy Criteria = Specification Test Generation Inadequacy Criteria?, Eighth IEEE International Symposium on High Assurance Systems Engineering (2004), pp.178-186

(Holzman 2004) Holzman: The SPIN Model Checker, Addison Wesley, (2004)

(Kawano 1987) K. Kawano, K. Mori, and M. Koizumi: A Fault Tolerant Decentralized Control System ensured by Autonomous Controllability, in Proc. of the 10th World Congress on Automatic Control (IFAC 87), Vol.VII (1987), pp.137-142

(Kawano 1989) K. Kawano, M. Orimo, and K. Mori: Autonomous Decentralized System Test Technique, in Proc. of the 13th Annual International Computer Software & Application Conference (COMPSAC 89), (1989), pp.52-57

(Menychtas 2013-1) Andreas Menychtas, Christina Santzaridou, George Kousiouris: ARTIST Methodology and Framework: A Novel Approach for the Migration of Legacy Software on the Cloud, 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE (2013), pp.424-431

(Menychtas 2013-2) Menychtas, A. (Ed): Deliverable D6.1. Analysis of current migration approaches, Public deliverable, ARTIST Project (2013),
<http://www.artist-project.eu/>

(Morinaga 2014) Eiji Morinaga, Akira Takagi, Yuki Sakaguchi, Hidefumi Wakamatsu and Eiji Arai: Basic study on production scheduling method for highly-distributed manufacturing systems, Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol.8, No.5 (Oct 2014), pp.JAMDSM 0072,

(Morinaga 2015) Eiji Morinaga, Tomomi Nakamura, Hidefumi Wakamatsu and Eiji Arai (2015): A Method of Distributed Job-Shop Scheduling for Highly-Distributed Manufacturing Systems, Proceedings of the 23rd International Conference on Production Research, Paper No.1100 (Aug 2015)

(Nishijima 1996) E. Nishijima and K. Kawano: On-line Testing for Application Software of Widely Distributed System, Proc. of 15th Symposium on Reliable Distributed Systems (1996), pp.54-63,

(OGATA 2016) Masaki OGATA: Management and Technology Innovation in Rail Industry as Social Infrastructure for Improved Quality of Life, IEICE TRANS. COMMUN., VOL.E99–B, NO.4 (2016)

(Okubo 2000) H. Okubo, H. Onari: Characteristics of distributed autonomous production control, International Journal of Production Research, Vo.38, No.17 (2000), pp.4205-4215

(Rockwell Automation 2016) Rockwell Automation, Inc. - Allen Bradley, Migration Profile: PLC-5 Controllers and 1771 I/O to ControlLogix, Public deliverable (2016), http://literature.rockwellautomation.com/idc/groups/literature/documents/pp/migrat-pp003_-en-e.pdf

(Smith 2002) Rachel L. Smith, George S. Avrunin, Lori A. Clarke, and Leon J. Osterweil: Propel: an approach supporting property elucidation (Jan 2002)

(Yamamoto 1993) H. Yamamoto, A. Yoshizawa, K. Kawano, K. Mori, K. Oshima, and T. Kawamura: On-line Software Test Technique based on Autonomous Decentralized System Concept, in Proc. of the 2nd Workshop on Future Trend of Distributed Computer Systems (FTDCS 93), (1993), pp.291-296

(Yoshida 2014-1) Yasuhiro Yoshida, Kouhei Akutsu, Yasuhiro Yumita, Ryuji Yamagiwa, Hideki Osumi, Mitsunori Okada, Kiyosumi Fukui: Full Upgrade of ATOS to Improve Safety and Reliability of Railway Services, Hitachi Review, Vol.63, No. 8 (2014)

(Yoshida 2014-2) Y. Yoshida, K. Akutsu, Y. Yumita, R. Yamagiwa, H. Osumi, M. Okada, K. Fukui: Full Upgrade of ATOS to Improve Safety and Reliability of Railway Services, Hitachi Review, Vol.63, No.8 (2014), pp.45-51

(Yasukochi 2017) T. Yasukochi, F. Sato, T. Furuta: Autonomous Control Technique for Gradual Systems Expansion, the IEEE 13th International Symposium on Autonomous Decentralized System (ISADS), (March 2017), pp.169-174

研究業績一覧

■査読付き学術誌

○継続的移行とオンラインテスト手法

- 1) Junichi Toyouchi, Kojin Yano, and Takao Terano: Proposing an Online Testing Method for Gradual Migration of Information and Control System, Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII), Vol.23 No.1, pp.52-59, January 20, 2019

■査読付き国際会議

○異種システム共存とテスト手法

- 2) Junichi Toyouchi, Kojin Yano, Takao Terano: Proposing an Online Testing Method for Continues Replacement of Information and Control System, 2017 International Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII), Nov.4 2017, MS2-2-2

○テストデータ作成手法

- 3) Junichi Toyouchi, Hideto Ogawa, Kojin Yano, Takao Terano: Test Data Creation by Simulation for Replaced Information and Control System, The Sixth Asian Conference on Information Systems (ACIS 2017), Dec.13 2017, WE1-4-2

■その他国内会議

○背景とニーズの整理

- 4) 豊内順一、鮫嶋茂稔、寺野隆雄：協調自律分散システムの製造システムへの適用 -モノづくり現場の見える化によるサプライチェーンの改善-, 第 30 回 人工知能学会全国大会、2016 年 6 月
- 5) 豊内順一、伊藤昭博、望月智之、森正勝、川上賢一郎、後藤正宏：ユビキタス機器と情報システムとの連携ミドルウェアの研究、平成 18 年電気学会 電子・情報・システム部門大会講演論文集, 2006 年 9 月

■その他国際会議

- 6) Junichi Toyouchi, Motohisa Funabashi: Service Integration Platform based on TINA 3-tier Model and Interfaces, TINA 2000 Conference, Sep. 2000
- 7) Junichi Toyouchi and Motohisa Funabashi: Development of Service Integration Platform for One-stop Service Applications” International Workshop on Advanced issues of. E-Commerce and Web-Based Information Systems IEEE conference (WECWIS 2001), Jun. 2001.

8) Junichi Toyouchi, Motohisa Funabashi, Shinsuke Honjo, Norihisa Komoda: Web Service Integration Platform with Privacy Preferences for One-stop Application, International Conference on Web Services (ICWS'03), Jun. 2003

■その他解説

- ・「行政&情報システム」2007年12月号（行政情報システム研究所）
「セキュア・ジャパン2007の解説」
- ・第2回セキュアVMシンポジウム「仮想化とセキュリティ」（2008/03/19）
「我が国のセキュリティ政策の現状と方向性」
- ・第3回セキュアVMシンポジウム「仮想化とセキュリティ」（2009/03/24）
「我が国の新しい情報セキュリティ政策について」
- ・セマンティック Web コンファレンス（2001/10/29）
「メタデータ応用システム」

【書籍】

- ・「ネットベース・アプリケーション」（2002/08）裳華房 NetCom ライブラリ
共著（編著：船橋誠壽）
- ・「世界を動かす技術思考 要素からシステムへ」（2017/05/21）講談社ブルーバックス
共著（編著：木村英紀）

【報告書】

- 国立研究開発法人科学技術振興機構 研究開発戦略センター（JST CRDS）発行
- ・戦略プログラム「統合サービスシステムとしての都市インフラ構築のための基盤研究」、
Feb. 2012
- ・戦略プログラム「エネルギー政策のための科学：技術・経済モデルの研究開発」、Mar. 2012
- ・平成 24 年度報告書「社会的期待と研究開発領域の邂逅に基づく「課題達成型」研究開発
戦略の立案」Mar. 2012
- ・平成 23 年度 報告書「社会的期待に関する検討 -CRDS 研究開発戦略立案プロセスに
おける利用を中心として-」、Mar. 2012
- ・海外動向報告書「中国の科学技術力について世界トップレベル研究開発施設」Jun. 2012
- ・戦略プロポーザル「将来水問題の解決に向けた統合モデリングシステムの研究 ～システ
ム構築戦略研究の具体化～」、Mar. 2013
- ・研究開発の俯瞰報告書「システム科学技術分野（2013年）」、Mar. 2013
- ・研究開発の俯瞰報告書 システム科学技術分野（2013年）第2版、Mar. 2014
- ・戦略プロポーザル「課題解決型研究開発の提言(2) 強靱で持続可能な社会の実現に向け
た 社会インフラ統合管理システムの研究」、Jun. 2014
- ・次世代ものづくり ～基盤技術とプラットフォームの統合化戦略～<中間とりまとめ>、
Dec. 2014