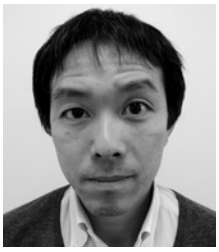
安全性評価に基づき演算量を削減するFail-operational E/Eアーキテクチャ評価手法


大塚 敏史※1


中西 健二※2


櫻井 康平※2

社会インフラにおける制御システムは，制御の高度化を目的とし大規模化を続けている．大規模システムのアーキテクチャを効率良く設計するためには，システムエンジニアリングプロセスに従い，要求を実現する論理アーキテクチャを構築後，複数の物理アーキテクチャ案のトレードオフを考慮して選定し，論理アーキテクチャを物理アーキテクチャに統合する．しかしアーキテクチャの統合には無数の代替案があるため，評価の演算量削減が必要である．本報告では統合時のアーキテクチャ評価効率化を目的とし，Fail-operationalの実現に必要な安全要件の充足判定のモデル化を行い，統合時に自動的に安全性の評価を行うアーキテクチャ評価手法を提案する．提案手法について自動運転システムを例題に評価を行い，アーキテクチャパターンを7,776通りから600通りへと92%を削減可能なことを確認した．

Safety-based E/E Architecture Evaluation Method for Fail-operational Systems to Reduce Calculation Complexity

Satoshi Otsuka※1, Kenji Nakanishi※2, Kohei Sakurai※2

Control systems for infrastructure continue to grow the scale to realize intelligent control. To design system architecture for large-scale control system efficiently, a system architect applies system engineering process to construct logical architecture which satisfies requirements of the system, to select physical architecture with considering tradeoffs, and to synthesize the logical architecture to physical architecture. However, synthesis of logical and physical architecture have numerous patterns. Therefore, the calculation complexity for evaluations of architecture needs to be reduced. In this paper, we propose architecture evaluation method which can evaluate safety automatically by modeling formula of safety requirements for fail-operational systems. Furthermore, the proposed method had been evaluated with an automated driving system and confirmed that the architecture pattern can be reduced by 92% from 7,776 to 600 patterns.

※1 株式会社日立製作所 研究開発グループ　※2 日立オートモティブシステムズ株式会社

1　はじめに

近年，社会インフラにおける制御システムは，制御の高度化や自動化を目的とし，CPS（Cyber Physical Systems）やIoT（Internet of Things）などのサイバー空間との連携や機器間制御による知能化が加速している．社会インフラで用いられる制御システムは，故障による不安全事象への影響が大きく，高度化や自動化が進むほど，故障時も制御を安全に継続する（Fail-operational）ことが重要となる．

制御システムの安全性については電気・電子・プログラマブル電子に関する機能安全規格IEC61508［IEC2010］及び自動車分野における機能安全規格ISO26262［ISO2011］がそれぞれ定められており，安全目標を達成するためのシステム，ハードウェア，ソフトウェアの設計プロセスの定義や，Fail-operationalを実現するハードウェア冗長系のパターンが示されている．

一方で，制御範囲が拡大し大規模化する制御システムの設計においては，システム全体を効率良く設計可能とするシステムエンジニアリングプロセス［INCOSE2015］の適用が必要となる．例えばIEEE1220［IEEE2005］で定義されているシステムエンジニアリングプロセスに従い，要件分析（Requirement Analysis），論理アーキテクチャ分析（Functional Analysis），統合（Synthesis）と，各プロセスでの比較検証を行い，システムの要件を満たす論理及び物理構成を決定する．本プロセスにおいて，前述するシステムの安全性についても評価及び検証が必要となる．

そこで本論文では，システムエンジニアリングプロセスにおけるアーキテクチャ評価時間を削減し，かつ自動運転システムなどFail-operationalが要求される高信頼システムでの安全要件を満たす安全性評価に基づくアーキテクチャ評価手法（Safety-Based Architecture Evaluation Method：SBAE）［Otsuka2016］の検討結果を報告する．本手法では，安全要件の充足判定のモデル化により安全要件の充足を自動的に判定しアーキテクチャを選定する．本手法について自動運転システムを例題としてアーキテクチャ評価の実証を行った．結果として，組み合わせ件数を7,776通りから600通りへと92％削減可能なことを確認した．

2節で本研究の背景，3節では関連研究，4節では提案手法の全体像とFail-operationalの要件を充足するための統合時の安全要件の充足判定方法，5節では本提案手法による自動運転システムの評価結果を述べ，6節にて考察を行い，7節でまとめを述べる．

2　背景

2.1　自動車E/Eシステム動向

自動車のE/E（Electrical and Electronic）システムは，安全性及び快適性の向上や環境負荷低減を目的とした高度な制御を実現するために，その制御範囲を拡大している．とくに，自動運転システムの実現に向け，E/Eシステムの複数の構成要素（センサ，アクチュエータ，コントロールを行うECU（Electronic Control Unit））の連携により車両全体の統合制御を行う必要がある．

自動運転レベル［SAE2016］が向上するにつれ，ドライバが行う運転操作がシステムにより代替され，より利便性が高まる一方で，運転に関する責任がドライバから自動運転システムへと委譲される．とくに自動運転中にドライバによる監視責任がない自動運転レベル3以上では，ハードウェアやソフトウェアに障害が発生した場合でも，不安全な状態を引き起こさないため，Dynamic driving task（動的運転タスク）について，Fallbackによる動作継続（Fail-operational）が要求されている［SAE2016］．

2.2　自動運転E/Eアーキテクチャ設計の課題

IEEE1220で定義されているシステムエンジニアリングプロセスの概要を図1に示す．ここでSynthesisプロセスにおいて，論理（Functional）アーキテクチャが物理（Physical）アーキテクチャに統合される．この際に代案との比較評価を行い，アーキテクチャを決定する．

機能連携が複雑化する自動車システムでは，システムを構成する物理エレメント（演算装置，ネットワークなど）及び論理エ

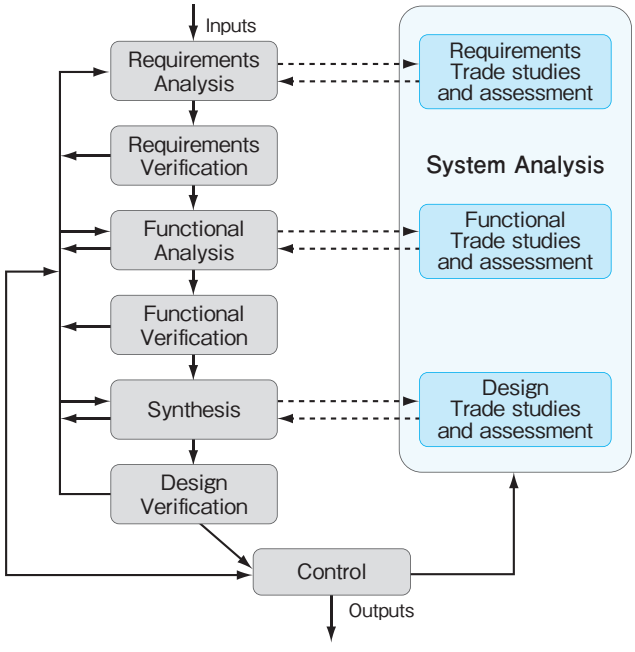


図1　IEEE1220システムエンジニアリングプロセス概要

レメント（論理機能，データリンクなど）の数，また論理エレメントを物理エレメントに対し配置するパターンの組み合わせが増加し，アーキテクチャ決定に必要な評価の組み合わせが増大する．

例えば論理エレメント数がm，物理エレメント数がnの場合，配置の組み合わせはnのm乗となり，それぞれ論理及び物理のエレメント数が10で，評価時間1秒の場合，すべてのアーキテクチャパターンを評価する時間は10の10乗秒となる．

システム設計及び評価は，システム要件や設計の変更など，システムの構成要素が変化するごとに繰り返し行う必要があるため，上記時間の削減を目的とした評価手法が重要となる．

3　アーキテクチャ安全判定の関連研究

アーキテクチャを評価する手法としてATAM（Architecture Tradeoff Analysis Method）[Kazman2000]があり，トレードオフポイントを定めステークホルダと議論を行い，安全性を含めてアーキテクチャを評価する手法を提案している．

自動車E/Eシステムの安全設計・評価手法は，欧州プロジェクトのSAFE（Safe Automotive soFtware architEcture）[Voget2012]があり，アーキテクチャ記述言語のEAST-ADL [Blom2013] を用いた，モデルベース開発によるシステム全体の安全設計・検証方式を提案している．また航空・自動車分野で用いられるアーキテクチャ記述言語のAADL（Architecture Analysis and Design Language）[Feiler2006] を用い，故障の伝播モデルを構築して安全分析を行う手法が提案されている [Delange 2014]．

アーキテクチャの自動最適化を実現するための手法としてHiP-HOPS [Adachi2011] があり，アーキテクチャ及び故障伝播のモデル化を行い，安全要件を検証し，要求コストと信頼度を満たすパレート最適解の導出を行う．

アーキテクチャの最適解を設計空間から探索するため，システムモデルをCSP（Constraint Satisfaction Problems）として定式化し，最適解を導出する研究がある [Li2014]，この中では各種要件（コスト，スケジューラビリティ，リソース）を制約条件として定式化しソルバにより最適解を導出している．

これら研究においては，システムの安全性評価及び検証を行う手法を構築しているが，本研究の課題としているアーキテクチャ統合時の組み合わせ量の削減手法，及び安全要件の充足判定の定式化については述べられていない．そこで本研究ではアーキテクチャ統合において，安全要件の充足判定の定式化により演算量削減を行うことを目的とする．

4　提案手法

4.1　提案手法の全体像

Synthesisプロセスで実施する，安全性評価に基づくアーキテクチャ評価手法の全体構成を図2に示す．提案手法であるSBAEでは高信頼システムで必須となる安全要件の充足を最初に評価して要件を充足しないアーキテクチャを排除し，アーキテクチャ評価全体での演算量の削減を行う．自動運転のように高信頼が要求されるシステムでは，安全要件を満たすことが必須である．そこで，安全要件を満たさないアーキテクチャをトレードオフの評価から排除し，演算量を削減する．

本プロセスでは，最初に物理アーキテクチャのエレメント（演算装置）に対して論理アーキテクチャのエレメント（論理機能）を割り当てる統合を行い，統合を行ったアーキテクチャに対して，安全性評価として後述する安全要件の充足判定を行う．

安全性評価では，機能安全規格ISO26262に倣い安全要件の充足判定を実施する．機能安全規格では，安全機能を実行する物理エレメントに対して信頼性（安全度水準：Safety Integrity Level）が要求される．これはハザード分析（Hazard Analysis）及びリスクアセスメントにより導出される．導出された安全度水準に対し，各エレメントが対応しているか，要求から実装までのトレーサビリティが要求されている．

ISO26262ではASILのレベルごとに異常検出率の要求値が定義されている．論理機能で要求される信頼度を，配置先の物理エレメントが実現可能か否かを判定することにより，前記論理機能を実現する物理エレメントで故障が発生した場合に，十分な信頼度で故障を検出することが可能か否かを評価することが可能となる．

また一方で，単一故障発生時にFail-operationalを実現するためには，機能安全規格に記載のある，共通原因故障（Common Cause Failure）やカスケード故障（Cascading Failure）などの従属故障（Dependent Failure）について，機能及びその情報伝達経路の独立性を確保することが必要となる．

これらより，下記がFail-operationalを実現するアーキテクチャの安全要件の充足判定における必要条件となる．

1. 配置時信頼性
2. 機能独立性
3. 経路独立性

図3を例にアーキテクチャ配置（統合）の例を示す．L_i及びDL_mは論理エレメントであり，それぞれ論理機能と論理機能間をつなぐデータリンクを示しており，P_j及びNW_nは物理エレメ

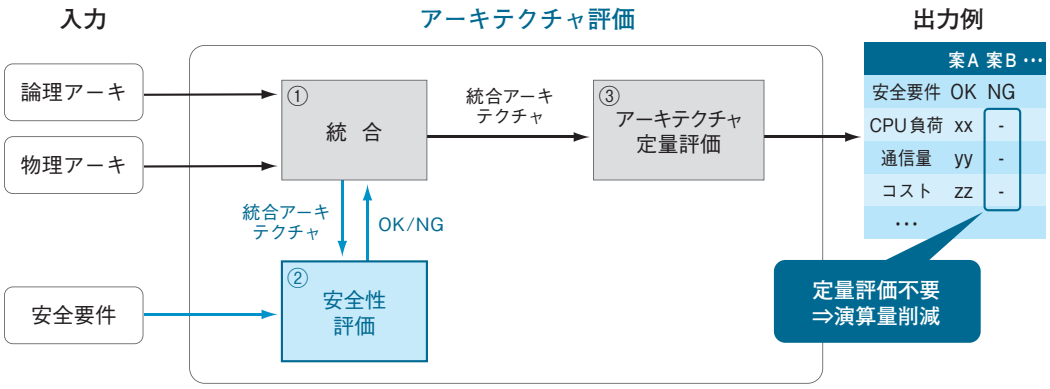


図2　安全性評価に基づくアーキテクチャ評価手法

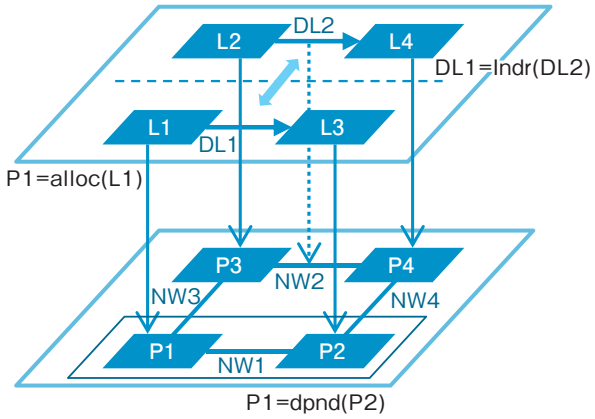


図3　論理アーキテクチャの物理アーキテクチャへの配置（統合）

ントの演算装置とネットワークを示している．

ここではL_1とL_3の連携により実現する機能と，L_2とL_4の連携により実現する機能が冗長系となっており，いずれか一方の機能連携の実行が保証されることにより，Fail-operationalが実現可能となる例を示している．

ここでアーキテクチャ統合における論理アーキテクチャの物理アーキテクチャへの配置について以下のように定義する．ここでは，論理エレメントL_iが物理エレメントP_jに配置されることを下記のように定義する．

$P_j = \mathrm{alloc}(L_i)$・・・(1)

また，論理エレメント（論理機能及びデータリンク）の独立性要求について下記の通り定義する．ここでは，論理エレメントL_iとL_jについて独立性が要求されている場合である．

$L_j = \mathrm{Indr}(L_i)$・・・(2)

また同様に，物理エレメントの依存関係について下記の通り定義する．ここでは物理エレメントP_iとP_jについて依存関係がある場合である．

$P_j = \mathrm{dpnd}(P_i)$・・・(3)

4.2　安全要件の充足判定方法

以下それぞれの安全要件の充足判定方法について説明する．

配置時信頼性

論理機能に割り当てられる要求信頼度に対し，統合後のアーキテクチャが要件を満たしているかを判定する．例えば物理エレメントで実現可能な信頼度に対し，配置された論理エレメントの要求信頼度がより高い場合にはNGとして判定する（図4）．

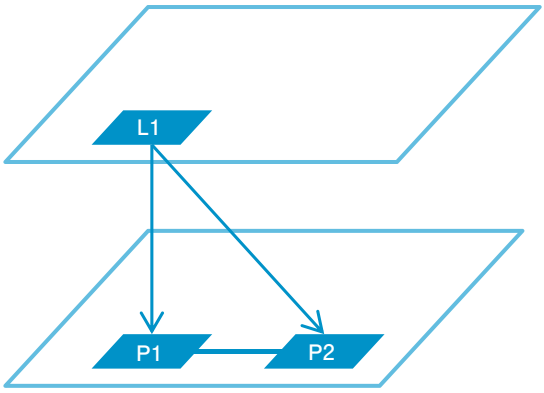


図4　配置時信頼性の判定

判定式はそれぞれ$L_i.\mathrm{asil}$をL_iに要求される信頼度，$P_j.\mathrm{asil}$をP_jが実現可能な信頼度とした場合に，

$(P_j = \mathrm{alloc}(L_i)) \wedge (P_j.\mathrm{asil} \geqq L_i.\mathrm{asil})$・・・(4)

となる．

機能独立性

冗長化を行っている機能が同時に故障することを防ぐため，論理機能間の独立性要求の充足可否を統合アーキテクチャ上で評価する．

図5の例では，論理機能L_1とL_2が独立に動作継続を要求される例を示している．ここで物理エレメントP_1とP_2が例えば同じリソース（電源など）を共有している場合，L_1とL_2を同じリソースを共有している物理エレメントに配置していないことを判定する．

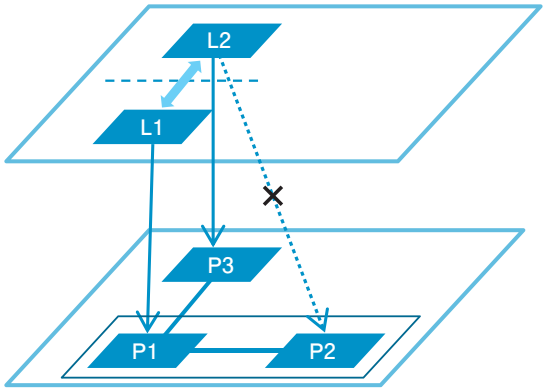


図5　機能独立性の判定

判定式は以下の通りとなる．

$(L_j = indr(L_i)) \wedge (alloc(L_j) \neq dpnd(alloc(L_i)))$・・・(5)

経路独立性

故障発生時にも連携して動作すべき論理機能について，単一故障の発生時でもそれぞれが連携され，動作継続可能か否かを経路の独立性の観点で判定する．

ここではL_1とL_3の連携で動作する機能と，L_2とL_4の連携で動作する機能がそれぞれ独立して動作することを要求される機能である．例えばL_1とL_3の連携で動作する機能が主機能，L_2とL_4の連携で動作する機能が縮退機能である．

この場合に，L_1とL_3の機能連携で使用している物理エレメント（演算装置またはネットワーク）と同じ物理エレメントをL_2とL_4の機能連携で使用した場合，共通部分が単一故障点となる．そのためそれぞれの機能連携で使用する物理エレメントで共通点がないことを判定する（図6）．

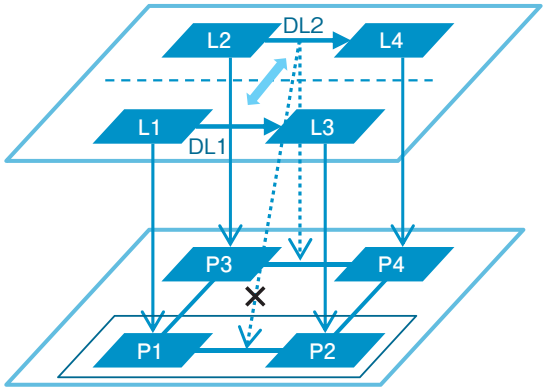


図6　経路独立性の判定

判定式は以下の通りとなる．

$(DL_j = indr(DL_i)) \wedge (alloc(DL_j) \neq dpnd(alloc(DL_i)))$・・・(6)

5　評価

5.1　実験

提案手法について，自動運転システムのアーキテクチャを例題に評価を行った．

評価に用いた論理アーキテクチャについて図7に示す．ここでは主機能を自動運転システムにおける認知・判断・操作とし，安全機能は主機能の診断及び縮退制御の実行，及び通常時制御から縮退制御への切替えを行うとした．

本実験ではFail-operationalを実現する安全コンセプトとして，IEC61508に記載のある1oo2D（1 out of 2 with Diagnostics）を用い，主機能と安全機能の系を独立で有し，それぞれの動作を診断して出力を切替える構成としている．

論理アーキテクチャの各エレメントには，演算量，ROM/RAM使用量など定量評価に必要なパラメータ，機能の要求ASIL，独立性要求の対象（機能独立性が必要な論理エレメントの関係）を付与した．

また評価に用いた物理アーキテクチャについて図8に示す．ここでは各四角は物理エレメントであるコントローラ，コントローラ内部の角丸四角は演算装置を示しており，内部に複数の演算装置があるコントローラも存在する．また青色で示すエレメントは，入力となるセンサまたは出力となるアクチュエータを示しており，それらに論理エレメントは配置せず，本評価の対象となる演算装置は含まないこととした．

各物理エレメントには，パラメータとして，実行可能な演算量や，有しているROM/RAM量など定量評価に必要なパラメータ，対応可能なASIL，物理依存対象（従属故障を引き起こす物理エレメントの関係）を付与した．またネットワークについてはここではピアツーピア型の構成とした．

これら評価モデルの実験諸元について表1に示す．論理アーキテクチャにおける論理エレメントである論理機能数は18，データリンク数は23であり．また物理エレメントである演算装置数は12，ネットワークの数は12である．ここで示す論理機能の数は，図7において，切替えを除く検出から制御までのブロックである．

本評価に適用したアーキテクチャでは，すべての統合パターンを網羅的に作成した場合，演算装置と論理機能のすべての組み合わせは，12の18乗通りである．

しかし評価においては，例えば特定の論理機能を特定の演算

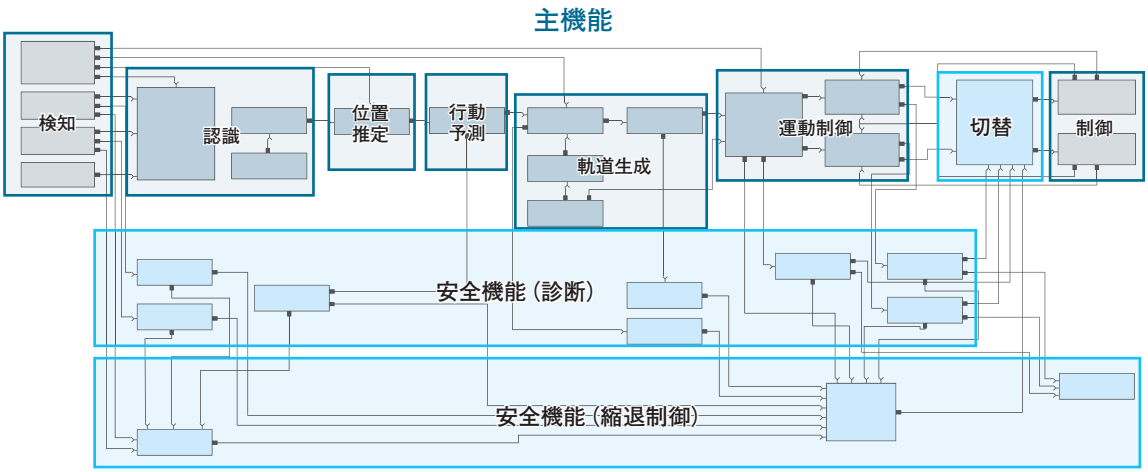


図７　論理アーキテクチャ

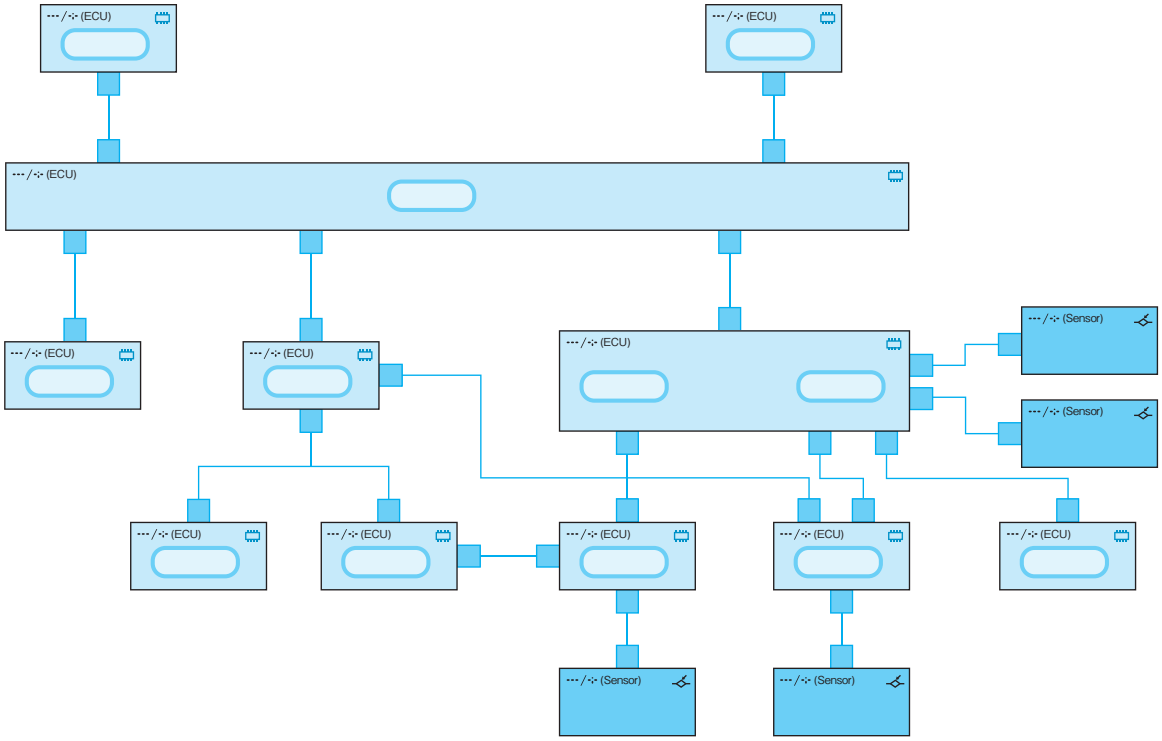


図８　物理アーキテクチャ

装置に固定し，また特定の論理機能を同じ演算装置に配置するようにグループ化を行った．

これは例えばアクチュエータを制御する論理機能について，アクチュエータと演算装置間でネットワークを経由すると要求レイテンシを満たさない場合や，大容量のデータを送受信するため，同じくネットワークを経由するとスループットの要求を満たさない複数の論理機能など，レイテンシやスループットなどあらかじめ与えられている制約について反映して配置の固定化を行い，組み合わせをあらかじめ7,776通りまで削減した．

安全要件については，ASILを付与された主機能の論理エレメント（例えばASILが付与される論理機能），安全機能の論理エレメント（図７の水色枠で示す安全機能），機能連携の論理エレメント数（例えば縮退に用いられる安全機能の論理エレメント）が安全要件の充足判定に使用されるため，その合計数を記載した．

表1　実験諸元

論理エレメント数	論理機能	18
	データリンク	23
物理エレメント数	演算装置	12
	ネットワーク	12
安全要件数		47

5.2　結果

以上のアーキテクチャについて，ツールを用いて，提案手法によりアーキテクチャの評価を行った結果について表2に示す．

ここでは前記記載したグループ化及び配置の固定化により削減した組み合わせの総数が7,776通りに対し，提案手法による安全性評価で排除した組み合わせ数がそれぞれ表2に記載の排除数となり，安全性評価の結果，組み合わせを600通りに削減した．

表2　評価結果

総組み合わせ数	7,776
配置時信頼性判定による排除数	5,184
機能独立性判定による排除数	1,692
経路独立性判定による排除数	300
評価組み合わせ数	600

6　考察

6.1　実験結果に対する分析と考察

提案手法により，アーキテクチャとして評価すべき組み合わせ数について，ツールの自動実行により7,776通りから600通りに削減することが可能になった．

この安全性評価でOKと判定されたアーキテクチャのパターンについては，高度な自動運転システムなど，安全面でのシステム責任が重要となり，かつ故障時も安全性確保のために動作を続ける必要があるシステムについて，機能安全の観点からFail-operationalの必要要件（配置時信頼性，機能独立性，経路独立性）を満たす組み合わせである．これら必須の要件を満たすアーキテクチャの中から，更に定量的なトレードオフ評価を行い，最適なアーキテクチャを選定する．

とくに制御システムでは，アーキテクチャの定量評価においては，システム全体でのレイテンシの要件充足性や，各演算装置及びネットワークにおけるスケジューラビリティなど，一つ一つのアーキテクチャで多くの評価項目が必要となる．

本手法により，定量的評価を実施するアーキテクチャのパターンを削減することは，定量評価での演算量が大きくなるほど，アーキテクチャ評価時間削減の効果は大きくなると考える．

6.2　限界と妥当性への脅威

本提案手法については，独立性の判定について多重故障は想定せず，独立性の判定のみを実施している．これはIEC61508などの機能安全規格でも，N重の同時故障は（独立性を有していれば）同時に発生することは極めて確率が低いと考え，同時故障は考慮外としている．ただし，潜在故障（Latent fault）は検出する必要があるため，潜在故障の検出手段は必要となる．本手法でも潜在故障の検出手段について安全要件の充足判定を実施することにより．安全性確保が可能になる．

また故障に起因しない不具合については本手法の対象外である．ただし大規模システムの場合にはコンポーネントの相互作用による不具合も発生し得ることから，システム理論に基づく事故モデル及び安全分析手法（STAMP/STPA［Leveson2004］）を適用し，その分析結果に基づく安全機能について，本手法を用いて判定することが重要である．

6.3　内的妥当性への脅威

今回の実験では，物理アーキテクチャは安全設計を実施し，冗長系や安全機構を設計したため，安全要件の充足判定によりパターンの排除を行えた可能性がある．本実験は，モデルの作成は自動車システムの知見を有した開発者のレビューを通じて構築したものであり，実開発に合わせた適用という点で信頼性があると考える．

一方で，手法としてはすべてのパターンが安全要件の充足判定によりOKと判定され，排除ができない可能性もある．しかし，パターンが排除できないということは，すべての論理機能の配置が許容されることから，物理設計の信頼度が過剰であることや，冗長性が高いなど，過度に安全な設計がなされていることが想定される．冗長系は信頼性が高まる一方でコストが高くなるため［Armoush2010］，適切な冗長系と信頼度の設計を行い，本手法により可能な配置を探索することが必要であると考える．

6.4　外的妥当性への脅威

提案手法は自動運転システムとして自動車システムへの適用を評価したが，機能安全規格であるIEC61508の観点は各製品分野で同様であるため，他分野への適用も同様に実施可能であると考える．

しかし，System of Systemsなどの事前に全体システムの構成

が決定しないシステムでは，事前に安全要件のすべてが決定されない可能性がある．そのような場合にはConditional Safety Certification［Schneider2013］のように，それぞれのシステムが事前に契約に基づく設計により安全制約を導出し，安全制約を安全要件として判定することにより適用が可能であると考える．

7　おわりに

7.1　まとめ

本報告では安全性評価に基づくアーキテクチャ評価手法を提案し，とくにFail-operationalを実現するための安全要件について，ツールでの自動実行が可能となるように安全要件の充足判定を定式化し，アーキテクチャを評価するための安全性評価手法を構築した．

本提案手法について自動運転システムの例題を用いて評価を行い，アーキテクチャの評価パターン数を，安全性を満たす必要な組み合わせのみを残すことをツールで自動実行可能にすることにより，アーキテクチャ評価の演算量を削減可能な見通しを得た．

7.2　今後の課題

安全性の評価については，機能安全規格にて定められているFTTI（Fault Tolerant Time Interval）など時間制約も制御システムの評価においては重要である．物理アーキテクチャにおける時間制約の判定も含めて，アーキテクチャの安全性を評価可能とすることが今後の課題である．

また安全性評価により600件に削減した組み合わせについて，定量的な評価を用いて更に組み合わせを削減することが必要となる．そのためには，例えば物理アーキテクチャのコストの比較評価や，システムとしてのEnd to Endでの要求レイテンシと実行時間による判定，製品展開を考慮した場合の拡張性の評価により，更にアーキテクチャ候補を絞り込むことについても今後の課題である．

【参考文献】

［Adachi2011］Adachi, Masakazu, et al. “An approach to optimization of fault tolerant architectures using HiP-HOPS.” Software :Practice and Experience 41.11 (2011):1303-1327.

［Armoush2010］Armoush, Ashraf. Design patterns for safety-critical embedded systems. Diss. RWTH Aachen University, 2010.

［Delange 2014］Delange, Julien, et al. AADL fault modeling and analysis within an ARP4761 safety assessment. No. CMU/SEI-2014-TR-020. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2014.

［Feiler2006］Feiler, Peter H., David P. Gluch, and John J. Hudak. The architecture analysis & design language (AADL):An introduction. No. CMU/SEI-2006-TN-011. Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, 2006.

［Blom2013］Blom, Hans, Henrik Lönn, Frank Hagl, et al. :EAST-ADL :An Architecture Description Language for Automotive Software-Intensive Systems. EAST-ADL WhitePaper, Volume 1, 2013.

［IEC2010］IEC 61508 ed2.0, Functional safety of electrical/electronic/programmable electronic safety-related systems, 2010.

［IEEE2005］IEEE Std. 1220-2005 Systems engineering — Application and management of the systems engineering process, 2005.

［INCOSE2015］INCOSE Systems Engineering Handbook :A Guide for System Life Cycle Processes and Activities, fourth edition, 2015.

［ISO2011］ISO International Standard, Road vehicles – Functional safety, ISO Standard 26262, Rev. Nov. 2011.

［Kazman2000］Kazman, Rick. Mark Klein, and Paul Clements. ATAM :Method for architecture evaluation. No. CMU/SEI-2000-TR-004. Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, 2000.

［Leveson2004］Leveson, Nancy. “A new accident model for engineering safer systems.” Safety science 42.4 (2004):237-270.

［Li2014］Li, Shuo. Ahmed Hemani. “Three-dimensional design space exploration for system level synthesis,” in Digital System Design(DSD), 2014 17th Euromicro Conference on, Aug 2014, pp. 419–426.

［Otsuka2016］大塚 敏史, 中西健二, 櫻井康平, 安全性評価に基づくE/Eアーキテクチャ評価手法, 第14回クリティカルソフトウェアワークショップ, 2016.

［SAE2016］SAE International J3016A, Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2016.

［Schneider2013］Schneider, Daniel, and Mario Trapp. “Conditional safety certification of open adaptive systems.” ACM Transactions on Autonomous and Adaptive Systems (TAAS) 8.2 (2013):8.

［Voget2012］Voget, Stefan ;SAFE project presentation ;ARTEMIS Technology Conference on interoperability ;Nuremberg/Germany ;March, 2012.