機能安全対応に向けた AUTOSAR OS の安全分析手法

毛利 守男[†1,†2]　佐藤 秀昭[†1,†3]　平林 寛崇[†1]　山下 映[†1,†4]　松原 豊[†1]　高田 広章[†1]

概要：AUTOSAR を用いた車載制御システムを機能安全対応させる場合，OS 及び OS 上で動作するアプリケーションの間で，安全要求の違反に繋がる伝播故障が存在しない事を示す必要がある．その手段として，AUTOSAR OS はメモリ保護機能を含む複数の保護機構を提供している．

我々は，機能安全対応の AUTOSAR OS（TOPPERS/ATK2）を開発する際の安全分析において，OS の保護機構に対する逸脱に関する分析を行った．その際，OS で対応すべき要求とアプリケーション開発者が対処すべき要求に分類した．前者の逸脱の原因について，仕様から設計内容まで掘り下げの分析が必要と判断・実施し，追加の安全機構の必要性を検証し，安全方策を導出した．

キーワード：AUTOSAR，RTOS，ISO 26262，機能安全，安全分析，HAZOP，SHARD

Safety analysis method for functional safety support of AUTOSAR OS

Morio MORI[†1,†2]　Hideaki SATO[†1,†3]　Hirotaka HIRABAYASHI[†1]
Akira YAMASHITA[†1,†4]　Yutaka MATSUBARA[†1]　Hiroaki TAKADA[†1]

Abstract: To conform the automotive application using AUTOSAR Platform to the functional safety standards, it is necessary to realize the FFI between applications running on the AUTOSAR OS. In order to realize FFI, AUTOSAR OS provides multiple protection facilities including memory protection scheme. The authors analyze about risk with violating the safety requirements of the protection facilities of AUTOSAR OS. And the safety requirements are divided between those responsible for OS developer and for application developer. The authors decided more detail analysis in architecture design level must be done, and put into practice. Finally the authors verified whether it needs additional safety mechanisms and defined the safety measures.

Keywords: AUTOSAR, RTOS, ISO 26262, Functional safety, Safety analysis, HAZOP, SHARD

1. はじめに

近年，車載制御システム開発において AUTOSAR[1]への注目が高まっている．ソフトウェアプラットフォームとして AUTOSAR を用いる事により，アプリケーションの複雑性の軽減と再利用性の向上が期待できる事がその理由である．名古屋大学大学院情報科学研究科附属組込みシステム研究センターでは AUTOSAR OS[2]準拠のリアルタイム OS として，TOPPERS/ATK2[3]（以下，ATK2 と表記）SC1～SC4 を開発し，リリースしている．SC は Scalability Class の略で，AUTOSAR OS 機能群の実装状況を SC1～SC4 の 4 段階で表している．SC1 は機能群の最小セット，SC2 は SC1 の機能及びタイミング保護機能を中心とする機能群，SC3 は SC1 の機能及びメモリ保護機能を中心とする機能群に対応する．SC4 は SC2，SC3 の全機能に対応する事を意味する．

†1 名古屋大学
Nagoya University
†2 SCSK 株式会社
SCSK Corporation
†3 株式会社ジェイテクト
JTEKT CORPORATION
†4 日本電気通信システム株式会社
NEC Communication Systems, Ltd.

AUTOSAR を用いるシステムは，車載制御システム用途で利用される事から，車載電気・電子システム向け機能安全国際規格 ISO 26262[4][5]対応を求められる機会が増えている．そこで我々は，ATK2 を機能安全規格に対応させる活動を進めている．

AUTOSAR では ISO 26262 への対応状況について，文献[6]において情報を提供しているが，この 3.4 節において，AUTOSAR 自身が対応しない項目を表明している．これらについては，AUTOSAR 利用者自身が対応する必要がある．本論文では，非対応事項の中で，安全分析に関連する事項について述べる．

安全関連システムを開発する場合，実装者は安全要求（システムの安全に関する要求事項）を満たすように設計，実装し，その上で実際に満たしている事を検証する必要がある．安全要求を仕様化し，実装するために必要となる取り組みが安全分析である．安全分析では，フォールト（故障を引き起こす可能性のある異常な状態）と故障の因果関係を分析する．安全分析を行う事により，安全要求の侵害を引き起こす原因の特定や，その対策に当たる安全方策の決定が可能となる．安全方策は，安全要求の侵害を防止するための活動（安全分析，レビューやテスト，利用者に対するマニュアル作成等）と安全機構（ウォッチドッグタイマやメモリ保護ユニット等の技術対策）で構成される．

AUTOSAR OS に対して安全分析を行う上での課題について二点述べる．一点目は公知の情報の入手が困難であるという点である．AUTOSAR OS には ISO 26262 対応を行った事を表明している製品が存在するが，対策にあたっての詳細な情報は開示されていないのが現状である．二点目は，リアルタイム OS（AUTOSAR OS を含む）や通信ミドルウェア等の共通ソフトウェアに対する安全分析の手順，方針，対策の検討方法が確立されていないという点である．前述の文献[6]や文献[4] Part 9 においても，複数の分析手法が提示されているだけで，方法論までは提示されていない．さらに，本論文で対象とするメモリ保護機能を有するリアルタイム OS のように，より高機能化した OS を対象とする安全分析手法や事例は，過去に報告されていない．

本研究では，AUTOSAR OS の安全要求の違反に繋がるアプリケーションの動作の原因と影響を，安全分析によって明らかにする．また，安全分析によって導出される安全方策によって，AUTOSAR OS の安全要求違反を検出または防止できる事を確認する．なお，AUTOSAR OS 自身が有する機能だけでは対策できない場合には，開発段階のレビューや，アプリケーションへの安全方策の追加により対処する．

本論文の構成は，以下の通りである．第 2 章では，本安全分析における方針の整理，目的の定義，前提の整理，分析対象の整理，用いる手法の選定を行う．第 3 章では，ATK2-SC1 に対する分析を行う．第 4 章では，保護機構を実装している ATK2-SC3 について，ATK2-SC1 との仕様の違いを示し，安全分析の方針を再整理した上で，メモリ保護の仕様を例に，実際に行った安全分析を示す．第 5 章では，まとめとして全体を総括する．

2. 安全分析の前提と方針

2.1. 安全分析の方針

ATK2 はリアルタイム OS である事から，様々なアプリケーションを動作させる事が可能である．そのため，幅広い製品からの利用が想定される．そこで我々は，ATK2 を文献[5] 第 9 節に規定されている SEooC（Secure Element out of Context）として扱う事とした．

SEooC は，具体的な製品を前提とせず汎用部品として開発する事を指す．そのため，安全分析にあたっては，安全要求を仮定する必要がある．

ATK2 は様々な制御システムで用いられる事が想定されるので，対象システムが決まらない段階で，OS の提供する機能を安全要求と非安全要求（安全に関係しない要求）に分類する事は難しい．そこで，全ての正常系機能（例えば，タスク管理機能や排他制御等）について，その機能が正しく動作しない場合に，製品の安全目標の侵害を引き起こす可能性があると考える．すなわち，ATK2 に対する安全要求を「全ての正常系機能が仕様通りに動作する事」と仮定する．

2.2. 安全分析の前提

ATK2 を SEooC とみなして行う安全分析の流れを図 1 に示す．はじめに ATK2 外部仕様書[7]に含まれる各仕様を分類し，安全要求仕様整理表と呼ぶ表に記述する．その中で安全分析対象に該当する仕様について，想定した通りに OS が動作しない場合（逸脱）による影響を分析する．影響が安全要求を侵害する場合，逸脱の原因を分析し，その対策として安全方策を導出する．各仕様から安全方策を導出するまでの過程を安全分析シートにまとめる．安全方策は，アプリケーション開発側で実施するものと，OS 開発側で実施するものに分ける．前者は，アプリケーションの開発工程への入力，後者は OS の設計工程への入力となる．

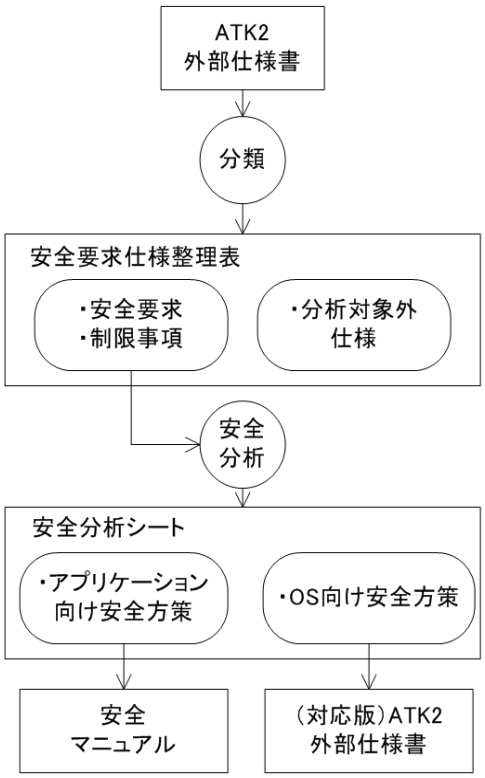

図 1 安全分析の流れ

文献[4] Part 6 第 7 節において，安全要求に割当てられるレベルを ASIL（Automotive Safety Integrity Level）指標を用いて A～D の四段階で定義する方法が示されている（未対応を示す QM を含めると五段階となる．ASIL D が最も厳しい）．ATK2 については，あらゆる ASIL のアプリケーションがその上で動作する事を想定し，ASIL D と仮定した．

2.3. 安全分析の対象

正常系機能を抽出する対象は，AUTOSAR OS 仕様をベースとする ATK2 外部仕様書とした．ATK2 には正常系機能以外の仕様も含まれる為，分析対象となる仕様を分類の上，安全分析を行った．

AUTOSAR OS 仕様をベースとして作成した ATK2 の外部仕様書には，正常系機能以外に，異常系機能や補足的な説明等の記述も含まれる．そこで，我々は ATK2 外部仕様書を分析するにあたり，各外部仕様を表 1 のように分類し，安全分析対象を特定した．なお，今回の分析対象は SC1 および SC 3 とする．

表 1 ATK2 外部仕様書の機能分類

分類	説明
正常系機能	・ATK2 の安全要求 ・安全分析対象 ・ATK2 に実装される
異常系機能	・制限事項違反の対策 ・安全機構とみなす（SC1） ・安全分析対象外（SC1）（誤動作対策であり，二重故障を検討しないため） ・保護機構に位置づけられる場合安全要求とみなす（SC3） ・安全分析対象（SC3） ・ATK2 に実装される
制限事項	・ATK2 の使用に当たってアプリケーションが守るべき事項 ・安全分析対象 ・アプリケーションに実装される
非安全機能	・デバッグ用途の機能等，安全関連システムでは使用されないと想定している機能 ・安全分析対象外
ジェネレータ関連機能	・ジェネレータに関する仕様 ・ジェネレータへの不正入力の対策 ・ジェネレータの出力結果が入力に対して正しい事をユーザが検証する前提で分析対象外とする
例外処理要求	・システムの誤動作の対策として，OS に要求されると一般的に想定される要件（スタックモニタリング，CPU 例外等が該当） ・安全分析対象外（既に導出済の誤動作対策であり，二重故障を検討しないため）
補足説明	・安全要求，及び安全機構の補足 ・安全分析対象外

この分類に基づき，安全分析の対象を正常系機能と制限事項の二種類とする．SC3 については，保護機構に該当する異常系機能も安全要求とみなし，分析対象とする．その理由は，4.2 節で詳細に述べる．そして以降は，ここで分析対象とした仕様についてのみ安全分析作業を行う事とした．

2.4. 安全分析の手法

本項では，安全分析の手法について述べる．ATK2 の安全要求は「全ての正常系機能が仕様通りに動作する事」であるが，正常系機能の性質や，仕様からの逸脱状況によっては，必ずしも製品の安全目標を侵害しない場合も考えられる．このため，どの機能をどのように逸脱した場合に安全目標が侵害されるか，を精査することが必要となる．

また，正常系機能からの逸脱を防ぐためには，安全分析の過程で，その逸脱がどのような原因で発生するかを特定し，各原因について対策を立てる事が必要となる．

これらの要求を満たす分析手法として，我々は HAZOP ベースの手法を選択する事とした．HAZOP は，正常な状態に対して，その状態の逸脱を示すガイドワード(「～ない」や「～の値が小さい」)を当てはめる事で，実際に起こりうる逸脱を導出し，その逸脱に伴う影響と，原因並びに対策を分析する手法である．類似の分析手法に FTA や FMEA があるが，FTA は故障を引き起こす原因の導出，FMEA は各障害原因によって発生する影響の導出が目的であり，両方の目的を実現する事を考えた場合，HAZOP がより適切であると判断した．

ただし，HAZOP 自身は元々化学プラントにおける薬品の流れを対象としているため，ガイドワードがリアルタイム OS の仕様に着目した分析に向かないという欠点がある．そこで，我々は HAZOP をベースにし，ガイドワードをソフトウェア向けに整理した SHARD（Software Hazard Analysis and Resolution in Design）[8]を用いて分析を行った．SHARD のガイドワードは，真偽値や処理のタイミングを重視する傾向がある事から，論理的なコンポーネントの組合せで構成される OS に適していると判断した．

使用したガイドワードを以下に示す．なお，本分析では，SHARD のガイドワードに加えて，リアルタイム OS の要求に対する逸脱を分析するためにガイドワードを新たに付け加えている．AUTOSAR OS は，車載システムの特に制御用途で用いられる事が前提であり，処理によっては数ミリ秒の制約違反が事故に繋がってしまう可能性がある．そこで，我々は処理時間が長すぎたり，短すぎたりするリアルタイム制約違反に対応するため，ガイドワードとして Longer, Shorter を追加した．

- Omission： 機能が提供されない
- Commission： 要求されていない時に機能が提供される
- Early： 期待されるタイミングより早く機能が提供される
- Late： 期待されるタイミングより遅く機能が提供される
- Value： 機能の出力値が間違っている
- Longer： 機能の提供までにかかる時間が長すぎる
- Shorter： 機能の提供までにかかる時間が短すぎる

3. AUTOSAR OS SC1 に対する分析

3.1. AUTOSAR OS SC1 に対する分析方針

我々は，まず ATK2-SC1 の分析を行った．SC は，AUTOSAR OS における保護機構への対応状況を示す．AUTOSAR OS SC1 は，OS の基本機能のみ実装し，保護機構は実装していない．ISO 26262 の要件として，文献[4] Part 9 第 6.4.5 節に(OS を含む)各アプリケーション間の FFI（Freedom From Interference の略．2 つ以上のエレメント間において安全要求の侵害に繋がるカスケード故障が存在しない状態を指す）を確保する仕組みが存在しない場合に，各アプリケーションは共存しているアプリケーションの中で最も高い ASIL で開発される必要がある旨の記載がある．そのため，SC1 は事実上単一 ASIL のアプリケーションのみを扱う．

図 2 に，ATK2-SC1 上でアプリケーションを動作させる場合の想定構成を示す．

ATK2 上で ASIL D のアプリケーションを動作させる場合は，ATK2 とアプリケーションの全体で，ASIL D を満たすように開発する必要がある．文献[4] Part 6 第 7.4.14 節において，ソフトウェアが ASIL D を満たすためには，ソフトウェアアーキテクチャレベルで以下のエラー検出要求を満たす必要がある旨の記述がある．

- 制御フローモニタリング
- 入出力データの範囲チェック
- 外部の動作監視機構
- プラウシビリティチェック
- ダイバース設計

このうち，制御フローモニタリングについては，AUTOSAR の基盤ソフトウェアに当たる AUTOSAR Basic Software(以下，BSW と表記)モジュールの一つ，WdgM（Watchdog Manager）[9]の利用により対応する事が可能である．外部の動作監視機構については，システムの対処が必要であるため，対象から除外する．

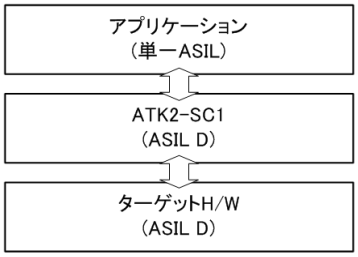

図 2　ATK2-SC1 が動作する想定システムの構成

入出力データの範囲チェック，プラウシビリティチェック，及びダイバース設計については，アプリケーション側でも対応を実施するため，OS でも対応を行うと冗長となりパフォーマンスが低下する．よって，アプリケーションにて対応できないエラーについてのみ OS 側に安全機構として追加する．

3.2. 基本機能の外部仕様レベル分析

ATK2-SC1 に対する SHARD を用いた実際の分析について，基本機能の 1 つであるリソース機能の仕様を例に説明する．リソース機能は，AUTOSAR OS における処理単位(タスク等)間での排他制御を行う．本節では，ATK2 外部仕様書のリソースの章から，以下の仕様を選び分析を行った．

- タスク，ISR2 は同一のリソースをネストして獲得することはできない

本仕様は，OS が提供する排他処理用関数 GetResource に関するものである．表 1 の分類では ATK2 の使用に当たってアプリケーションが守るべき事項，すなわち制限事項に該当する．分析結果を表 2 に示す．表の各列の読み方は以下の通り．

- 仕様 ： 分析対象となる仕様情報を ATK2 外部仕様書から抜粋して記述する
- ガイドワード ： ガイドワードを記述する．今回の例ではスペースの都合で Omission 以外のガイドワード記述を省略している
- 逸脱 ： 仕様にガイドワードを当てはめた結果を記述する．ガイドワードを当てはめた結果，無意味な文章が生成された場合は逸脱に追加しない(例 ： 本仕様に Longer を当てはめた場合，「タスク，ISR2 は同一のリソースをネストして獲得することはできない時間が長すぎる」となる)
- 影響 ： 逸脱に伴う影響を導出する．もし，そこで導出された影響が，安全目標を侵害しうる内容ではない場合は，そこで当該逸脱に対する分析を打ち切る
- 原因 ： 逸脱の原因を導出する．今回の例では，表 2 に抜粋した「OS 外の不具合」の他に「H/W の不具合」も原因として想定する．もし，原因を掘り下げる必要がある場合は，その理由を何段階にもわたって掘り下げる事がある
- 対策 ： 導出された個々の原因に対して，対策を立案し，表に記述する．今回の例では，まず OS が既に実装している安全方策として，ATK2 外部仕様書に記載されている異常系機能から，今回の仕様の安全機構に該当する仕様を抽出して記述した．次に，アプリケーション向けの安全方策としてエラーコードが返された後の処理に関する記述を作成した．なお，後者については，アプリケーション開発者が利用できるように開発者向け安全マニュアルに追記している

上記の方針に従い，SHARD 表に記載した各仕様について，左側から右側の列に順に記述を行っていく事により，逸脱の影響の精査から，原因の分析，対策の導出を行った．

3.3. AUTOSAR OS SC1 に対する分析結果

ATK2-SC1 の安全分析を行った結果を以下に記す．ATK2 外部仕様書に記載されている ATK2-SC1 に関連する 935 件の仕様に関する記述について，分類を行った．その結果をもとに，ATK2-SC1 に対して安全分析を行った結果を以下に示す．

- 各異常系機能が，安全要求（正常系機能，制限事項）の逸脱に対応する安全機構として機能する事を確認した
- 各制限事項をアプリケーション側で守るべきルールとしてまとめ，安全マニュアルに記述した
- 逸脱原因のうち，OS 自身の不具合や，ハードウェア不具合によるものについては，アプリケーション側で実装すべき安全機構の指針としてまとめ，安全マニュアルに記述した

以上の対策を実施する事により，ATK2-SC1 の安全要求の検証結果を後工程に反映できる事を確認した．

4. AUTOSAR OS SC3 に対する分析

4.1. AUTOSAR OS SC1 との仕様の違い

次に ATK2-SC3 の分析を行った．AUTOSAR OS SC3 は，同一のアプリケーションを構成する OS オブジェクトをグループ化するための仕組みである OS アプリケーション（以下，OSAP と表記）を導入している．

表 2 リソース機能の外部仕様分析例(抜粋)

仕様	ガイドワード	逸脱	影響	原因	対策
タスク，ISR2 は同一のリソースをネストして獲得することはできない	Omission	タスク，ISR2 が同一のリソースをネストして獲得する	OS が意図しない動作となる (安全要求を侵害する可能性がある)	OS 外の不具合	GetResource 呼出し時に指定されたリソースが，既に占有されている場合，E_OS_ACCESS を返す (COS3837)
					E_OS_ACCESS が返された場合の適切な対応をユーザ側で検討する(マニュアル ID0010)
					・・・
				H/W の不具合	・・・

OSAP は権限レベルに応じて信頼 OSAP と非信頼 OSAP の二種類に分類される. 信頼 OSAP は原則 OS 上の全ての OS オブジェクトにアクセス可能だが，非信頼 OSAP は原則自 OSAP に属する OS オブジェクトへのアクセスしか認められず，システム全体に影響を及ぼす操作(全割込み禁止，OS シャットダウン等)を行う事も認められない. この権限レベルを活用する事により，信頼度が異なる複数のアプリケーションを管理する事が可能になる.

SC3 では OSAP の他に，保護機構を導入している. AUTOSAR は，複数の開発元による OSAP が同一のプロセッサ上で共存する事をプラットフォームの要求に含めている. 保護機構は，ある OSAP のフォールトが無関係な OSAP に伝播する事を防ぐ事により，OSAP の共存を可能にするための仕組みである. SC3 で追加される保護機構は，主にサービス保護とメモリ保護の二つに大別される.

(1) サービス保護

サービス保護は，API レベルでのアクセス保護を行う. サービス保護は大きく三種類に大別される. 一つ目は，非信頼 OSAP 等，OSAP の処理単位(タスクまたは ISR2(Interrupt Service Routine Category 2))から，信頼 OSAP を含む他 OSAP に対し，アクセス権を付与されていない状態でシステムサービスを実行した場合に，その処理要求を拒否する仕組みである. 二つ目は，OS が提供するシステムサービスを不適切な形(割込み禁止状態での呼び出し，不適切な引数の使用等)で呼び出した場合，その処理要求を拒否する仕組みである. 三つ目は. 非信頼 OSAP から他の全 OSAP に影響を与えるシステムサービスの実行(OS シャットダウン要求，全割り込み禁止処理等)を拒否する仕組みである.

(2) メモリ保護

メモリ保護は，メモリレベルでのアクセス保護を行う. 各非信頼 OSAP の単位で，アクセス(読み，書き，実行)可能なメモリ領域を予め定義しておき，非信頼 OSAP がアクセスを許可されていない領域に対して，アクセスを行った場合に，MPU(Memory Protection Unit)によりこれを検知し，アクセスを遮断する仕組みである.

4.2. AUTOSAR OS SC3 向け安全分析の方針

複数の OSAP が ATK2-SC3 上で共存しつつ動作する構成について分析を行う. ここでは，2 種類の ASIL アプリケーション，具体的には低 ASIL(ASIL A 等)と高 ASIL(ASIL D 等)のアプリケーションが共存するケースを想定する. 想定構成を図 3 に示す.

OSAP 共存にあたっては，高い ASIL を割り付けられた OSAP に対し，より低い ASIL を割当てられている OSAP が影響を及ぼさない事を担保する必要がある. 図 3 において，低 ASIL のソフトウェアが，高 ASIL のソフトウェアに対してアクセスするパターンは二種類存在する.

① 低 ASIL 側の OSAP⇒高 ASIL 側の OSAP
② 低 ASIL 側の OSAP⇒OS(ASIL D)

それぞれのパターンについて，AUTOSAR が提供する保護機構を整理し，FFI を満たすための条件を確認する. なお，高 ASIL 側の OSAP が OS より ASIL が低い場合，高 ASIL 側の OSAP(ASIL C)⇒OS(ASIL D)

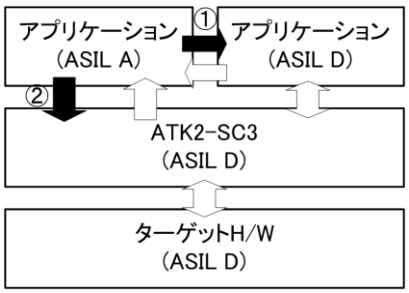

図 3　ATK2-SC3 が動作する想定システムの構成

のようなパターンも生じうるが，今回は②に包含する形で分析を行う．図 3 における①と②に対してそれぞれサービス保護，メモリ保護を実現する事により，権限を持たない非信頼 OSAP に対して，API レベルでのアクセスを拒否した上で，API に依らない手段で直接メモリにアクセスしてくる事を排除する事が可能となる．

今回の分析は，FFI の実現にあたり，保護機構に含まれる安全要求を分析し，安全方策を導出する事を目的とする．保護機構の仕様は異常系機能に分類されるが，仕様から一度逸脱するだけで即座に安全要求に違反する事から，安全分析の対象として逸脱時の影響を分析した．分析の結果導出した安全方策に対応する事により，ATK2 上で複数 ASIL のアプリケーションを安全に共存させる事が担保可能になる．

4.3. メモリ保護の外部仕様レベル分析

ATK2-SC3 に対する SHARD を用いた実際の分析について，メモリ保護の仕様を例に説明する．本論文では，ATK2 外部仕様書のメモリ保護の章から以下の仕様を選び分析を行う．

- 非信頼 OSAP の専有コード領域に対して，他の非信頼 OSAP が読出し，書込み，実行アクセスする事を禁止する

外部仕様レベルの分析結果を表 3 に記載する．分析手順を以下に示す．仕様に対し，ガイドワードを用いて逸脱に伴う影響及び逸脱の原因を導出する．例では，書込み処理について，omission を当てはめた場合の影響を導出している．影響から，アクセス先の OSAP に干渉し，安全要求に反する事が確認できるため，原因の導出を行った．表 3 では，原因 1～原因 3 まで 3 段階分掘り下げて分析を行っている．ここで導出した原因 2 について，原因 3 において OS に起因する不具合と H/W に起因する不具合（H/W の中で MPU については独立で扱う）に分ける．H/W に関する対策はユーザに対応を委ね，OS に関する対策は OS 開発者側に対応を委ねる方針とする．

分析対象の安全要求の粒度によっては，原因を分析した結果，有効な対策が導出できない場合がある．この場合，更に詳細な設計情報を参照して，原因の分析，対策の導出を行う．その時点で具体性を伴った対策を導出できた場合は，当該安全要求に対する分析を終了する．導出できなかった場合は，対策が導出できるまで，設計の粒度を細かくして分析を繰り返す．

4.4. メモリ保護のアーキテクチャ設計レベル分析

外部仕様に対する分析の結果，OS の不具合に起因する逸脱の可能性が明らかになった．OS の不具合の内容をより明確にし，具体的な対策を検討するためには，保護機構の設計に対する分析が必要である．本節では外部仕様の分析で導出された原因をより詳細に分析するため，アーキテクチャ設計を分析する事とした．アーキテクチャ設計レベルのメモリ保護の概要を図 4 に示す．

表 3 メモリ保護の外部仕様分析例（抜粋）

仕様	ガイドワード	逸脱	影響	原因 1	原因 2	原因 3
非信頼 OSAP の専有コード領域に対して，他の非信頼 OSAP が読出し，書込み，実行アクセスする事を禁止する	Omission	書込みアクセスを禁止しない	タスク，ISR2 が自身の所属する非信頼 OSAP とは異なる非信頼 OSAP の専有コード領域に対して書込みを行い，アクセス先が意図しない動作となる	当該アクセスを許可すべきと誤認した	アクセス元の OSAP が信頼 OSAP であると誤認した	OS の不具合
						H/W 故障
					別の信頼 OSAP からのアクセスであると誤認した	OS の不具合
						H/W 故障
				書込みアクセスを禁止しなかった		OS の不具合
						MPU 故障
						H/W 故障

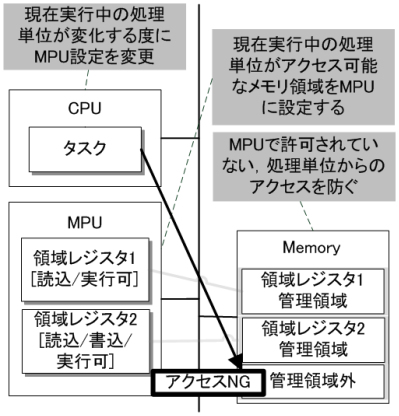

図 4　メモリ保護のアーキテクチャ設計情報

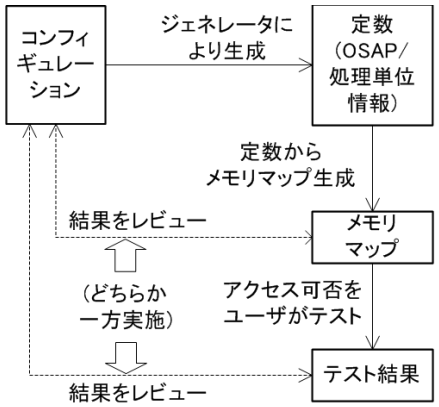

図 5　定数，ジェネレータの安全方策

メモリ保護に関する設計情報の中で，本外部仕様に関連する内容をまとめた内容を以下に示す．

- ATK2-SC3 では，タスクをはじめとする OS の処理単位は MPU を介してメモリ領域にアクセスする．MPU で許可されていない領域にアクセスした場合，アクセスが遮断され CPU 例外が発生する．
- 非信頼 OSAP の処理単位を実行中は，非特権モードでプロセッサを動作させ，その処理単位からアクセス可能なメモリ領域を MPU に設定する事で，メモリアクセスを制限する．

これらは，OS 機能の一つであるディスパッチャが処理単位を切り替える際に以下の一連の処理を順に実行する事により実現する．

1) MPU にアクセス可能なメモリ領域をセット
2) 動作モードを CPU のステータスレジスタにセット
3) return 命令を用いて動作モードを切り替える

切替え後の処理単位からアクセス可能なメモリ領域は，以下の二種類である．

- その処理単位の持つユーザスタック
- その処理単位が属する OSAP からアクセス可能なメモリ領域

以上の設計を分析した結果，メモリ保護仕様の逸脱の原因となり得る不具合を（ア）～（ウ）に分類した．

（ア）定数設計の誤り
（イ）ジェネレータ設計の誤り
（ウ）処理設計の誤り

（ア）と（イ）は，システムコンフィギュレーションにおける不具合である．（ウ）は，メモリ保護機構の設計の不具合である．後者については，具体的な対策を検討するために，次節で，さらに詳細な設計に対して分析する．

システムコンフィギュレーションの不具合とは，具体的には，処理単位と OSAP の管理データが意図と異なるという不具合である．例えば，OSAP に所属するタスクを設定するパラメータ「OsAppTaskRef」にあるタスクを設定し，コンフィギュレーションを実施した結果，OSAP の管理データにもタスクの管理データにも当該関連を示す情報が保持されず，OSAP に所属するタスクの関連を確認できなくなるような不具合が該当する．

図 5 に示すように，AUTOSAR OS では，処理単位と OSAP の管理データは，システムの設計段階で，コンフィギュレーション情報として記述され，ジェネレータによって C 言語ソースファイルに変換される．OS は，その C ソースファイルを参照して動作する．アプリケーション開発者の意図通りの管理データが生成されている事を，以下のいずれかの方法で確認する事を提案する．これらの方法は，図 5 に示されるように，レビューの対象とタイミングが異なるだけであり，本質的には同じである

- 生成されたメモリマップ情報（アクセス可能なメモリ領域情報）を取得しレビュー
- 生成されたメモリマップ情報に基づきアクセス可否をテスト（領域境界に対してアクセスし，メモリ保護例外が発生するかどうかをテストする）

本対策の実施により，コンフィギュレーションの意図に基づくメモリ保護を行う事が可能となる．概要を図 5 に示す．

（ウ）の対策について検討したが，対策にあたり抑えるべき内容が「切替え先に応じて正しく動作モード，MPU 設定を行う事」であるため，まだ設計情報として粒度が粗く，対応する安全対策を導出する事は困難と判断した．このため，ユニット設計レベルについて，さらに分析を行う事とした．

表 4 メモリ保護のユニット設計レベルの分析例(抜粋)

逸脱	原因 2	原因 3	原因 4	原因 5	対策
書込みアクセスを禁止しない	アクセス元の OSAP が信頼 OSAP であると誤認した	OS の不具合	ジェネレータ設計誤り		メモリマップレビュー又はテストをユーザにて実施し，意図通りである事を確認する
			定数設計誤り		
			処理設計誤り	所属 OSAP の属性情報取得間違い	左記観点のレビュー・テストを実施する
				・・・	
	別の信頼 OSAP からのアクセスであると誤認した	OS の不具合	ジェネレータ設計誤り		メモリマップレビュー又はテストをユーザにて実施し，意図通りである事を確認する
			定数設計誤り		
			処理設計誤り	所属 OSAP の情報取得間違い	左記観点のレビュー・テストを実施する
				・・・	
	書込みアクセスを禁止しなかった	OS の不具合	ジェネレータ設計誤り		メモリマップレビュー又はテストをユーザにて実施し，意図通りである事を確認する
			定数設計誤り		
			処理設計誤り	MPU レジスタへの設定間違い	左記観点のレビュー・テストを実施
				・・・	

4.5. メモリ保護のユニット設計レベル分析

メモリ保護機構の処理設計の誤りの原因について，ユニット設計レベルでの分析を行った．ここでは，アーキテクチャ設計における「1) MPU にアクセス可能なメモリ領域をセットする」について，処理の流れを取り上げる．

ユニット設計レベルの分析結果を表 4 に記載する．原因 4 にはアーキテクチャ設計レベルで導出した原因，原因 5 にはユニット設計レベルで導出した原因，対策には導出された安全方策を記載している．以下，原因 4 から原因 5 を導出するまでの流れを説明する．

ユニット設計によると，MPU へのメモリ領域セット処理は以下の流れで行われる．

1) 実行中の処理単位が属する OSAP が切り替え先の OSAP と等しいかどうかをチェックする．等しい場合は，切り替え処理を行わずに終了する．
2) MPU にメモリ保護対象として設定した処理単位と現在動作している処理単位が等しいかどうかをチェックする．
3) 2) の結果，等しくない場合は，処理単位レベルの MPU 設定レジスタに切替え先処理単位の MPU 設定情報をセットする．
4) MPU にメモリ保護対象として設定した OSAP 情報と現在動作している OSAP 情報が等しいかどうかをチェックする．
5) 4) の結果，等しくない場合は，OSAP MPU 設定レジスタに，切替え先 OSAP の MPU 設定情報をセットする．

上記過程において発生し得る不具合は，以下の二点に収束する．なお，動作モード設定処理で発生し得る不具合も，分析の結果以下の二点に収束している．

- 変数判定(所属情報の確認処理)の間違い
- レジスタの設定間違い

この二点の根本原因に対する対策として，以下の二種類が考えられる．

(ア) 安全機構の追加
(イ) 設計とソースコードに対する検証(レビュー及びテスト)で確認

今回我々は，この対策として(イ)を選択した．これは，以下の理由による．

- 十分レビューで確認できる内容，量であると判断した
- 実装の複雑化を避ける必要があると判断した

そこで，これらのレビューの観点をまとめたレビューチェック表を作成し，OS の設計工程のインプットとした．

4.6. AUTOSAR OS SC3 の安全分析結果

ATK2 外部仕様書に記載されている ATK2-SC3 に関連する 514 件の仕様について，分析を行った．

ATK2-SC3 は，FFI を実現するための保護機構を持つ．SC3 における保護機能はメモリ保護とサービス保護という 2 つの側面から構成される．本章では，主にメモリ保護について取り上げた．メモリ保護はハードウェア(MPU)の機能と連動して動作するため，外部仕様レベルだけでなく，ユニット設計レベルまで掘り下げて分析を行った．その結果，さらなる技術的な安全機構を追加するのではなく，メモリ保護機構の逸脱を防ぐために必要となるレビュー項目を洗い出し，チェックリスト化する事により，対応する事とした．

サービス保護についても，メモリ保護と同様の方針で設計まで掘り下げる形で分析を行い，仕様の逸脱を防ぐためのレビュー項目を抽出し，チェックリスト化を行った．また，直接保護機構に属さない SC3 の機能(OSAP 等)についても，同様の方針で分析を行った．

これらの作業により，複数 ASIL のアプリケーション向けに SC3 で拡張された安全分析の対象についても SC1 と同様に安全分析を行い，安全要求の検証を行える事を確認出来た．

5. おわりに

我々は，ATK2 に対して，今回提案する安全分析方法を適用する事により，安全方策を導出した．

ATK2-SC3 の保護機構に対して分析を行い，安全分析によって得られるレビュー・テスト表を OS 開発工程へのインプットとする事により，保護機構の安全要求からの逸脱を防止できる事を確認した．以上の対策により，SC3 の保護機構について，新規の安全機構を追加する事なく FFI を実現し，OS 上で複数のアプリケーションを共存できる事を確認した．

本研究の課題について示す．今回分析した内容は AUTOSAR OS に限定されるため，Platform 全体として機能安全対応を行うには，AUTOSAR RTE[10]と連携して SW-C(AUTOSAR Platform におけるアプリケーションの基本単位)を動作させるケースの分析や，通信ミドルウェア等のリアルタイム OS 以外の共通ソフトウェアの分析が必要になる．

参考文献

[1] AUTOSAR, http://www.autosar.org/（参照 2016-09-07）
[2] AUTOSAR, Specification of Operating System AUTOSAR Release 4.2.2 (2015).
[3] TOPPERS プロジェクト, TOPPERS/ATK2，入手先（https://www.toppers.jp/atk2-download.html）（参照 2016-06-21）
[4] ISO, ISO 26262:Road vehicles - Functional safety - , Part 1～9 (2011).
[5] ISO, ISO 26262:Road vehicles - Functional safety - , Part 10 (2012).
[6] AUTOSAR, Overview of Functional Safety Measures in AUTOSAR (2014).
[7] AP コンソーシアム, ATK2 外部仕様書，入手先（https://www.toppers.jp/docs/tech/ATK2-0010_ATK2_spec_120.pdf）（参照 2016-06-21）
[8] David J.P., The Principled Design of Computer System Safety Analyses, PhD Thesis, The University of York (1999).
[9] AUTOSAR, Specification of Watchdog Manager AUTOSAR Release 4.2.2 (2015).
[10] AUTOSAR, Specification of RTE AUTOSAR Release 4.2.2 (2015).