ゴール木とパターンを用いたISO26262における安全要求のモデル化

青木 利晃[1]　トライチャイヤポーン クリアンクライ[1]　千葉 勇輝[1]　松原 正裕[2]　西 昌能[2]
成沢 文雄[2]

概要：車載システム向け機能安全標準 ISO26262 では，安全要求は，最上位の Safety Goal から始まり，段階的にハードウェアやソフトウェアへの要求へと詳細化を行う．このような段階的な詳細化は，当初の安全目標から追跡可能になるように要求を細かくし，最終的に実装レベルの概念に引き継がれていることを確認するためのものである．一方で，文書が曖昧であったり，矛盾していると，安全要求の追跡が困難となり，結果として漏れや抜けが発生し，安全性の達成が不確実となってしまう．そこで，我々は，要求分析手法 KAOS におけるゴール木の考え方に基づいて，開発中である電子制御ステアリングシステムの安全要求の分析を行った．その結果，多くの暗黙の前提，および，曖昧な記述を指摘することができた．そして，これらの問題点を解決するために，ゴール木とパターンに基づいて，安全要求を記述する手法を提案した．この手法を電子ステアリングシステムの安全要求の一部に適用することにより，高い有効性を確認することができた．本論文では，以上の事例と提案手法について紹介する．

キーワード：機能安全，安全要求，ゴール指向，車載システム

Modeling Safety Requirements of ISO26262 using Goal Trees and Patterns

Toshiaki Aoki[1]　Kriangkrai Traichaiyaporn[1]　Yuki Chiba[1]　Masahiro Matsubara[2]
Masataka Nishi[2]　Fumio Narisawa[2]

Abstract: In ISO 26262, safety requirements are constructed from general goals to be achieved into system, hardware and software requirements step by step. Such stepwise construction of the saftey requirements allow us to confirm that the system realizes the goals by making them traceable. The traceability also helps us to exhaustively extract requirements which are necessary to achieve safety. On the other hand, it depends on the quality of the documents. If the documents contain ambiguities, contradictions and many of requirements are missing, those lead to the unsafety of the system. We conducted a case study to make safety requirements in which traceability of the requirements is realized using the goal tree of KAOS. Although the document is still under construction when we analyzed it, we found many of implicit assuptions missing and ambigous requirements. To solve this problem, we proposed a method to describe the safety requirements based on the goal tree and its patterns. Then, we confirmed the effectivenes of the method by applying it to an electronic power steering system. In this paper, we show the proposed mentod and the case study of the electronic power steering system.

Keywords: Functional safety, safety requirements, goal-oriented analysis, automotive systems

1. はじめに

車載ソフトウェアの安全性に関する問題は，社会において非常に大きな関心となりつつある．自動車は，従来は機械的に制御されてきたが，近年，コンピュータ制御技術の発展と利便性や性能の追求により，多くの部品の電子化が進んできている．これにより，車載ソフトウェアの規模の急速な増大と複雑化がもたらされ，主に，電子制御部分の安全性に関する問題が取り上げられつつある．世界標準においては，一般の電子システム向けの機能安全標準 IEC61508

[1] 北陸先端科学技術大学院大学
JAIST
[2] (株) 日立製作所 研究開発グループ 制御イノベーションセンタ
Hitachi, Ltd., Research & Development Group, Center for Technology Innovation - Controls

だけでなく [2]，車載システムに特化された ISO26262 が策定されている [1].

ISO26262 は，IEC61508 を基本として，車載システム向けに特化したものである．ISO26262 では，車載システム開発サイクル全体にわたり機能安全の考え方や基準について定義している．その基軸となっている活動に，開発対象システムが安全である根拠を，システムの安全性を実現する要求 (安全要求) として，文書化することがある．この安全要求は，最上位の SG(Safety Goal) から始まり，段階的にハードウェアやソフトウェアへの要求へと詳細化を行う．SG とはハザード分析やリスクアセスメントに基づいて設定される安全性を達成するための目標である．このような段階的な詳細化は，当初の安全目標から追跡可能になるよう要求を具体化し，最終的に実装レベルの概念に引き継がれていることを確認するためのものである．安全要求を追跡可能にすることにより，安全であるという根拠を示すだけでなく，安全要求の漏れを回避することも可能になる．一方で，安全要求が追跡可能であることは，文書の品質に大きく依存する．もし，文書が曖昧であったり，矛盾していると，安全要求の追跡が困難となり，結果として漏れや抜けが発生し，安全性の達成が不確実となってしまう．よって，安全要求を厳密に記述することが重要である．

そこで，我々は，ISO26262 における安全要求の形式化について共同研究を行っている．この共同研究では，電子制御ステアリングシステム (EPS, Electronic Power Steering system) の安全要求を対象に，問題点の分析および解決策の提案を行った．当初，EPS の安全要求は，スプレッドシートを用いて，自然言語 (英語) で記述されていた．スプレッドシートの表により，安全要求の間を対応づけていたのである．この安全要求の文書を分析するために，規範となる文書の考え方を導入した．下位の安全要求が上位の安全要求を満たすという関係に基づいて記載されていたので，要求分析手法 KAOS[3] におけるゴール木とゴール木における完全性の考え方に基づいて文書の分析を行った．文書はドラフト段階で作成途中のものであったが，これにより，多くの暗黙の前提，および，曖昧な記述を指摘することができた．そして，これらの問題点を解決するために，ゴール木とパターンに基づいて，安全要求を記述する手法を提案した．また，この手法を EPS の安全要求の一部に適用することにより，高い有効性を確認することができた．本論文では，以上の提案手法とケーススタディについて紹介する．

2. 関連研究

我々は，安全要求を記述するために，KAOS のゴール木を採用した．他にも，Tim Kelly らが中心となって提案している GSN(Goal Structuring Notation) [4] がある．GSN では，木構造に基づいてセーフティケースを記述する．セーフティケースとは，安全性を示すための文書であり，安全要求や安全を裏付けるエビデンスなどが記述される．また，Tim Kelly らは，セーフティーケースを再利用するため，パターン [7] も提案している．

我々が，KAOS のゴール木を採用した理由は，その単純さと明確な意味論が存在するためである．我々は，安全要求を形式的に検証することも目的にしており，明確な意味論は不可欠である．GSN のパターンやその形式化の研究も存在する．E.Denny らは，GSN，および，そのパターンの形式化行っている [5], [6]．主に GSN の構造的な取扱いに関する形式化であり，個々の項目の内容を取り扱ってはいない.

また，KAOS にも要求のパターンを取り扱う手法が提案されている [9]. このパターンは，ゴール木に汎用なものである．一方，我々のパターンは，安全要求，さらには，安全の仕組みに特化したものである．本論文では，典型的な安全の仕組みが存在し，その安全要求は，ゴール木上でパターン化可能であることを示した.

3. 安全要求とゴール木

3.1 安全要求

ISO26262 は車載システムに特化された機能安全に関する国際標準であり，車載システム開発サイクル全体にわたり，安全性に関する概念，基準や指針，について記述されている．ISO26262 では，安全要求は，ハザード分析やリスクアセスメントにおいて導出される概念的なものから，ハードウェアやソフトウェアへの要求へと段階的に詳細化される．最上位の安全要求は SG(Safety Goal) と呼ばれる．SG は FSR(Functional Safety) に詳細化される．FSR は機能的な安全要求であり，SG を実現するよう，機能の観点から詳細化し獲得される．次に，FSR は，技術的な観点から，TSR (Technical Safety Requirement) に詳細化される．そして，最終的には，TSR は HSR(Hardware Safety Requirement) と SSR(Software Safety Requirement) に詳細化される．車載システムは，ハードウェアとソフトウェアにより実現されるが，HSR と SSR は，それぞれに対する安全要求である．このように，ISO26262 では，概念的な安全要求は段階的に実装可能な安全要求へと詳細化される．また，段階的な詳細化は，PAA(Preliminary Architecture Asssumption) と呼ばれる，対象車載システムの構造に関する情報を参照しながら行う．PAA は，開発の初期段階で作成される対象システムの抽象的な構造である．安全要求は，PAA に出現する特定のコンポーネントやモジュールを参照したり，その構造を前提として記述される.

3.2 ゴール木

ゴール木はゴール指向要求分析手法の 1 つである KAOS などで採用されている，木構造に基づいて要求を構造化す

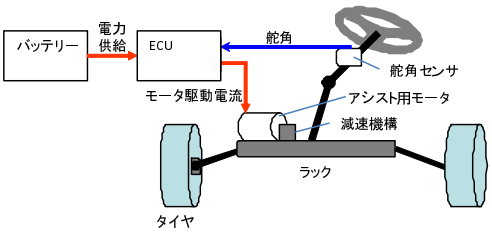

図 1 電子制御ステアリングシステム

るモデルである．ゴール木では，木構造において，子に相当するゴール (子ゴール，または，サブゴールと呼ぶ) が親に相当するゴール (親ゴールと呼ぶ) を満たすことを表現している．すなわち，親ゴールを G，サブゴールを $G_1, \cdots, G_n$ とすると，G_1 から G_n に記述されている要求から，G の要求を導けるように，親ゴールから子ゴールに分解するのである．ゴール木では，完全性と呼ばれる性質が定義されている．KAOS では，それぞれのゴールは，命題に基づいた線形時相論理により記述される．そして，子ゴールが親ゴールを満たすことを完全性と呼ばれる性質として定義している．形式的には，ゴール木における，それぞれの親子関係において，親ゴールを G，サブゴールを $G_1, \cdots, G_n$ とすると，$\{G_1 \cdots, G_n\} \models G$ が成立することが，完全性である．安全要求を分析するためには，その文書に関する，なにかしらの規範が必要である．そこで，我々は，ゴール木を，ISO26262 における安全要求の記述に関するモデルとして採用した．ISO26262 における安全要求の段階的な詳細化の考え方が，ゴール木の考え方と適合しているからである．また，安全要求では，要求の漏れや誤りの検出がとても重要であり，完全性がその検出に適していると考えた．

4. 電子制御ステアリングシステムの安全要求

4.1 電子制御ステアリングシステムの PAA

電子制御ステアリングシステムはモータでステアリングのアシストを行うシステムである．従来は油圧によるアシストが行われてきたが，その代わりにモータを用いているのである．図 1 にその概要を示す．ステアリングのアシストが必要な際，バッテリからモータに電力を供給し，ステアリングの操舵を支援する．電力供給は，ECU で制御されており，今回対象とするのは，ECU における電力喪失の際のフェールセーフ機能に関する安全要求である．

我々が対象とした電子制御ステアリングシステムの PAA を図 2 に示す．ここで，図の右端の 1,2,3 への線は，それぞれ，図の上部の 1,2,3 へ接続されており，電力を遮断するリレーを表現している．EPS では，Power Supply Unit が Mortor に電力を供給している．Pre-Driver と Inverter は電力における電圧と波形を変更している．このシステムでは，電力の供給が失われると安全に動作を停止するフェールセーフ機能を実現している．電力は，Pre-Driver Voltage Monitor と Inverter Voltage Monitor により，その電圧が監視されている．Diagnostic Function モジュールは，Pre-Driver Voltage Monitor と Inverter Voltage Monitor が獲得した電圧に基づいて，電力が失われているかどうか判断する．もし，電力が失われたならば，Fail-safe Action Function モジュールにメッセージ'Manual Steering' を送信し，Fail-safe Action Function モジュールは，それを受けて，Mortor への電力供給を停止する．ここで，確実に電力供給を停止するために，Pre-Driver, Inverter Relay, Motor Relay のすべてにおいて，電力をカットする (図 1 の 1,2,3)．以降，Current Control Unit, Pre-Driver, Inverter, Diagnostic Funtion, Fail-safe Action Function を，それぞれ，CCU, PD, Inv, DF, FSF と略記する．

4.2 安全要求の分析

電子ステアリングシステムの安全要求は，表形式 (スプレッドシート) を用いて自然言語 (英語) で記述されていた．表では，それぞれのセルに安全要求が記述されており，SG，FSR, TSR, HSR/SSR の関係は，セル同士の横の関係として表現されていた．そこで，まず，この表形式の文書に基づいて，ゴール木を作成した．ここでは，オリジナルの文書の問題を分析するために，追加や修正を行わず，単に，表形式のものを，木構造で表現しなおしたのみである．その一部を図 3 に示す．この図では，G1 が FSR で，G2 から G5 までが TSR である．G1 から G5 が表現する安全要求は以下のとおりである．

G1 System shall make transition to 'Manual Steering' if failure of voltage supplied to Current Control Unit has been detected.

G2 Demand for transition to 'Manual Steering' shall be sent to ECU Processing Unit if failure of voltage supplied to inverter has been detected.

G3 Demand for transition to 'Manual Steering' shall be sent to ECU Processing Unit if failure of voltage supplied to Pre-Driver has been detected.

G4 ECU Processing Unit shall send 'Stop Demand' to Pre-Driver if ECU Processing Unit has received demand for transition to Manual Steering.

G5 Pre-Driver shall stop according to 'Stop Demand'.

ゴール木の完全性では，G2 から G5 が成立するならば，G1 が成立しなければならないことを定めている．この完全性に基づいて，安全要求の分析を行った．その結果，以下の問題点が存在することが明らかになった．

- 暗黙の前提の存在.
  安全性に関する事実を記述するためには，一般に，多くの前提が必要となる．さらに，基本的な前提は，多くの安全要求の記述と関連しており，追跡可能とするためには，何度も文書中に出現させる必要がある．そ

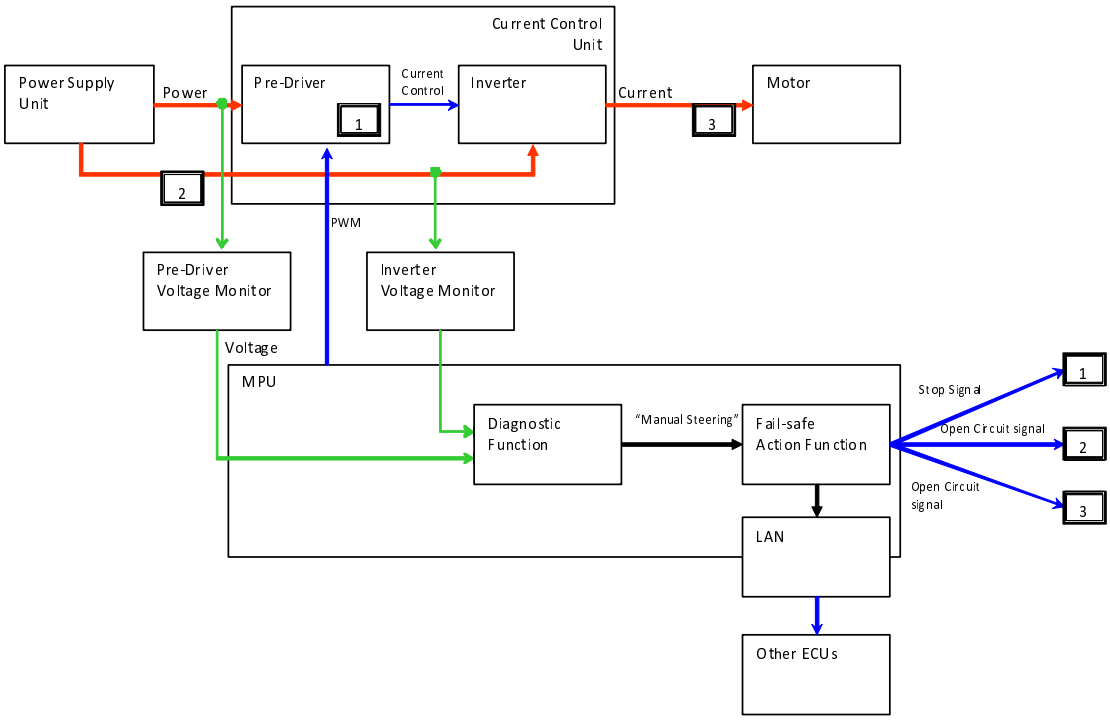

図 2　電子制御ステアリングシステムの PAA

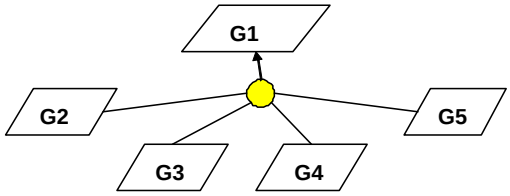

図 3 安全要求のゴール木による表現

のため，冗長な作業と感じられ，暗黙の前提となってしまっているのである．一方で，暗黙の前提は，必ずしもエンジニア間で共有されているとは限らないため，安全性の実現への脅威となりえる．

- 記述の不統一.
  FSR, TSR, HSR で抽象度が階層的になっているが，それぞれにおいても，ばらばらの抽象度で記述されている．さらに，それらの間の抽象度に大きな乖離があり，追跡が困難である．
- 安全要求の不足.
  暗黙の前提と類似した問題であるが，基本的な安全要求が省略されていたり，そもそも，安全要求が不足している部分が存在する．

図 3 の例で，それぞれの問題について説明する．想定されている挙動は，Inv における電力供給の失敗と PD における電力供給の失敗により，CCU への電力供給が失われ，それにより，メッセージ'Stop Demand' が PD に送られる．そして，PD が停止することにより，システムが'Manual Steering' モードとなり，モータによるアシストが停止する，ということである．しかしながら，Inv の PD の電力喪失と，CCU の電力喪失の関係が暗黙の前提となっており，親ゴールと子ゴールの関係がわからない．また，親ゴールでは CCU に関する安全要求であるのに対して，子ゴールでは Inv と PD に対するものである．これらは，大きく抽象度が異なり，親ゴールと子ゴールの間を追跡するのは困難である．さらに，CCU, Inv, PD の間の通信について何も言及が無い．図 3 で示した以外の安全要求で，通信が存在する場合は，それについて明確に言及してあり，通信の失敗無しでメッセージが送信されることについて規定されている．この通信に関する安全要求が不足しているのである．

使われている語彙にも統一性が無い．ECU Processing Unit は，PAA には出現していないが，メッセージを送受信する際の DF と FSF の総称として使われている．また，別の安全要求の記述では，それらの総称は MPU となっている場合もあった．

5. ゴール木とパターンに基づいた安全要求開発

5.1 ゴール木による安全要求の記述

ISO26262 における段階的な安全要求の詳細化は，ゴール木の考え方と適合する．例えば，TSR は FSR を満たすように，HSR/SSR は TSR を満たすように詳細化されなければならない．また，このような詳細化の正しさは，完全性という性質で形式的に定義されている．よって，ゴール木のノードを論理式で記述できれば，形式的に詳細化の正しさを保証することができる．そこで，提案手法では，それぞれのノードを命題論理式で記述することにした．対象とした EPS では，安全要求は含意 (ならば) の形のものが多く，複雑な計算を伴うものはなかった．このような安全要求で懸念されることは，親ゴールが Modus Ponens(三段論法など) により論証されるかどうかである．ここで，Modus Ponens とは，A と $A \Rightarrow B$ から，B を導く推論のことである．安全要求は大量にあるので，暗黙の前提などにより省略が重なってしまうと，安全要求のつじつまが合わなくなってしまう危険性があるのである．この論証を行うためには，命題論理で十分であり，安全要求の品質を向上させるために有効であると考えた．また，KAOS におけるゴール木では，時相論理式を用いてゴールを記述するが，対象とする安全要求では，時間的なタイミングに基づいた推論は必要ではなかった．つまり，時間的なタイミングにより導出できたり，できなかったりする安全要求は，対象とした範囲では無かったということである．例えば，図 3 の G1 と G2 は，以下のように記述することができる．

- G1:(CCU.VoltFailureDetected ⇒ S.State=Manual Steering)
- G2:(Inv.VoltFailureDetected ⇒ DF.Send(Manual Steering, DF, MPU)

以上の記述では，命題変数は CCU.VoltFailureDetected, S.State=Manual Steering, Inv.VoltFailureDetected, DF.Send(Manual Steering, DF, MPU) である． =の演算子や DF.Send のような関数が含まれているように見えるが，それらは，命題であり，演算や計算の評価は行わず，真偽値を持つ事実を表現しているのみである．CCU.VoltFailureDetected と VoltFailureDetected は，それぞれ，CCU および Inv への電力供給の失敗を表現している．S.State=Manual Steering は，システム状態がマニュアルステアリングに移行すること，DF.Send(Manual Steering, DF, MPU) は，メッセージ Manual Steering が DF から MPU に送信される事実を表現している．

自然言語による記述では，メッセージの送信元もしくは送信先として，ECU Processing Unit と記述されているが，それが指すものは，DF もしくは FSF である．また，抽象的な記述では，それらを含む全体のモジュールを表

現したい場合も存在した．そこで，具体的なモジュールを指定する時は DF もしくは FSF，全体のモジュール場合は，PAA に出現している MPU を用いることにした．このように，記号を用いることで，安全要求を統一的にかつ厳密に記述することができる．また，モジュール間の関係も明確に記述することができる．例えば，Inv の PD の電力供給の失敗と，CCU の電力供給の失敗の関係は，CCU.VoltFailureDetected == (Inv.VoltFailureDetected || PD.VoltFailureDetected) として記述できる．

5.2 安全要求パターン

以上のようなゴール木により完全性を保証するためには，親ゴールが導けるくらい十分な事実を子ゴールに記述しなければならない．これにより，暗黙の前提や安全要求の欠如を発見することができる．また，命題論理で推論を行うためには，命題変数を統一的に，かつ，適切な抽象度で導入しなければならないため，安全要求の記述を統一させることができる．4.2 節で指摘した問題点を解決することが期待できるのである．

一方で，多くの事実を記述しなければならないことが容易に想像できる．暗黙の前提が存在した理由は，基本的な前提が何度も出現することによる煩わしさに起因しているものと考えられる．命題論理とゴール木の完全性により，より，多くの事実を記述しなければならず，煩わしさが増大する恐れがある．そこで，安全要求パターンを提案した．安全要求パターンは，論理式の一部を置き換え可能なゴール木の部分木である．図 4 に例を示す．このパターンは，メッセージ M を S から D に安全に送信するための要求を表現している．パターンに出現する論理式，もしくは，命題変数の一部はパラメータ化されている．図 4 では，M, S, D は命題変数をパラメータ化したものであり，それぞれ，メッセージを表現する文字列，送信元を表現する文字列，送信先を表現する文字列に変換される．C, TC は論理式をパラメータ化したものであり，それぞれ，メッセージを送信する条件，および，メッセージ送信後の条件に置き換えられる．$C \Rightarrow D.Send(M,S,D)$ は，C が成立する時にメッセージ M を S から D に送信すること，$D.SendWithoutFailure(M,S,D)$ は失敗なしに送信することを意味している．$D.Send(M,S,D) \wedge D.SendWithoutFailure(M,S,D) \Rightarrow D.Received(M,D)$ は，メッセージ M を S から D に送信して，かつ，通信が失敗しないなら，メッセージ M は D に受信されること，また，$D.Received(M,D) \Rightarrow TC$ は，受信したら TC が成立することを意味している．

安全要求パターンでは，任意の置き換えに対して，子ゴールが親ゴールを満たすことも証明されている．図 4 では，任意の M, S, D, C, TC に対して，$C \Rightarrow D.Send(M,S,D)$, $D.SendWithoutFailure(M,S,D)$ と $D.Send(M,S,D) \wedge D.SendWithoutFailure(M,S,D) \Rightarrow D.Received(M,D)$ から，$C \Rightarrow D.Received(M,D)$ が導ける．そして，$C \Rightarrow D.Received(M,D)$ と $D.Received(M,D) \Rightarrow TC$ から，$C \Rightarrow TC$ が導け，子ゴールから親ゴールが導けることを示すことができる．安全要求パターンとして子ゴールが親ゴールを満たすことが示されているので，パラメータを置き換えてパターンを実体化し，安全要求に組み込んでも，その部分の完全性は保証されているのである．

また，安全要求パターンは，デザインパターン [10] などの記述にならって文書化している．安全要求パターンの文書では，図 4 の木構造に加えて，適用可能性，説明，適用例，さらには，以上の完全性の証明が記述されている．

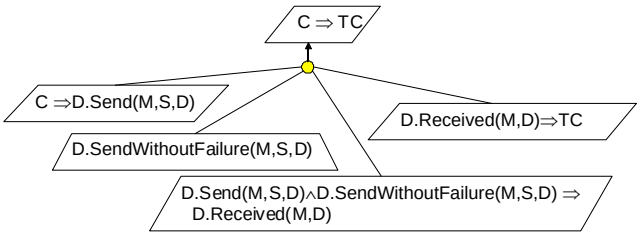

図 4 安全要求パターン

5.3 安全要求パターンの適用

安全性を保証する仕組みのことを安全機構と呼ぶ．安全機構を実現する手段は，それほど，バリエーションがあるものではない．よって，典型的な安全機構に関する安全要求群を，安全要求パターンとして事前に準備することができる．我々が対象とした電子制御ステアリングシステムの安全要求は，電力供給失敗の際のモジュール間の通信に関する部分であった．この部分では，通信を同一チップ内の通信 (In-chip)，チップ間の通信 (Inter-chip)，コントローラ間の通信 (Inter-controller) の 3 種類が存在する．通信の信頼性は，In-chip がもっとも高く，Inter-chip が中程度，Inter-controller がもっとも低い．また，流れるデータは単一のデジタルデータ (Digital)，アナログデータ (Analog)，一連のデジタルデータ (DigitalCom) の 3 種類が存在する．通信の信頼性は，Digital がもっとも高く，Analog が中程度，DigitalCom がもっとも低い．必要な安全機構は，これらの組により決められていることがわかった．表 1 に，それらの組と，信頼性の対応関係を表で示す．例えば，In-chip で Digital は信頼性は高く，Inter-chip で DigitalCom は低い．通信に関しては， 通信の信頼性の程度 (高，中，低) の応じて，3 種類の典型的な安全機構 (Hight, Mid, Low) が存在した．そこで，それぞれの安全機構に対して，安全要求パターンを準備した．図 4 は，高い通信の信頼性に対する安全機構を表現するものである (High)．実際の信頼性目標は数値 (例えば，x 掛ける 10 のマイナス n 乗という形式) で定義されてる．一方で，対象とするシステムや領域

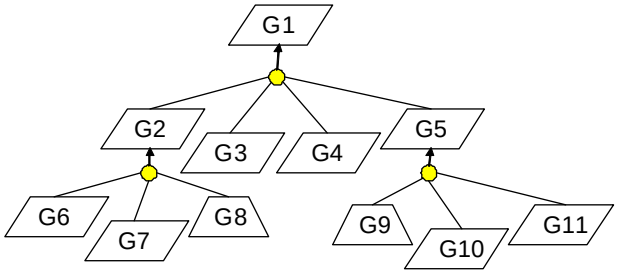

図 5 安全要求の記述

を特定すると，使用するハードウェアやコンポーネントのバリエーションは限定的であり，信頼性目標を達成する組み合わせも限定的になる．このような組み合わせを導出する際，本ケーススタディで対象とした EPS では，三段階の信頼性を識別することで十分であり，これにより系統的な安全要求パターンの適用が可能になっているのである．

安全要求を作成する際には， PAA を元に通信する場所と流れるデータを特定して，表 1 に基づいて，適用する安全要求パターンを求めることができる．表 2 は，適用する安全要求パターンを求め方を示している．例えば，この表の 3 行名 PD から MPU への通信で，流れるデータが電圧 (Voltage) であれば，安全性が中程度のための安全要求パターンを適用することが示されている．このように，安全要求の作成の際，系統的に適用する安全要求パターンを選択でき，効率的に安全要求を作成することができる．

場所/データ	Digital	Analog	DigitalCom
In-chip	高	高	-
Inter-chip	高	中	低
Inter-controller	中	-	低

表 1 通信の信頼性

6. ケーススタディ

6.1 安全要求の獲得

図 3 に示した当初の安全要求に，5 節で提案した手法を適用した結果の一部を図 5 に示す．この安全要求は，図 4 に示されている安全要求パターンにおけるパラメータ S, D, M, C, TC を，それぞれ，DF, MPU, 'Manual Steering', CCU.VoltFailureDetected, S.State = 'Manual Steering' に実体化している．そして，G2 と G5 を，それぞれ，オリジナルの安全要求に基づいて，手作業で形式化行った．それぞれのゴールの詳細は，付録に示した．ゴールが表現する自然言語による記述と，命題論理による記述が書かれている．G2 から G5 の子ゴールから，Modus Ponens を用いて，G1 が導かれるのがわかる．G6 から G8 の子ゴールから，G2 についても，同様に導くことができる．

この安全要求では，G9 から G11 は，MPU から PD に停止要求を表現するメッセージ'Stop Demand' を送信することを記述している．表 1 と表 2 から，この場合も，図 4 が適用できることがわかる．しかしながら，図 5 には，メッセージを失敗無しで送信するという安全要求は記述されておらず，安全要求が欠如していることがわかる．

図 4 の安全要求パターンを適用するためには，パラメータ S, D, M, C, TC をそれぞれ，MPU, PD, 'Stop Demand', DF.Received('Manual Steering', MPU), S.State = 'Manual Steering' で置き換える．安全要求パターンを実体化すると，親ゴールは DF.Received('Manual Steering', MPU) ⇒ S.State = 'Manual Steering'，子ゴールは以下になる．

G10 DF.Received('Manual Steering', MPU) ⇒ PD.Send('Stop Demand',MPU,PD)

G12 PD.SendWithoutFailure('Stop Demand', MPU, PD)

G13 PD.Send('Stop Demand', MPU, PD) ∧ PD.SendWithoutFailure('Stop Demand', MPU, PD) ⇒ PD.Received('Stop Demand', PD)

G14 PD.Received('Stop Demand',PD) ⇒ S.State = 'Manual Steering'

そして，図 6 のように，G9 と G11 を G14 の子ノードとして追加すれば，追跡可能な安全要求を獲得できる．以上により，図 3 に記述されている部分の安全要求を，形式的，かつ，追跡可能に記述することができる．

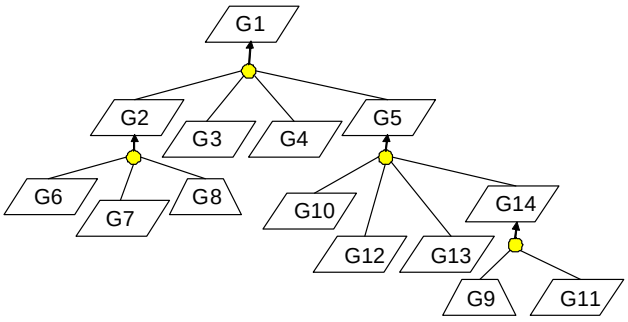

図 6 完全性が成立する安全要求

6.2 評価

提案手法を用いて，EPS における電力供給失敗に対処する安全機構の通信部分の安全要求を作成した．安全要求は開発途中のものであったが，対象部分は一通り記述されていた．対象とした部分は，オリジナルの安全要求記述では 3 つの FSR により記述されている．この安全要求に基づいて，安全要求パターンの適用と手作業による記述を繰り返しながらゴール木を作成した．オリジナルの安全要求に記述されている安全要求だけでは追跡可能にならない部分が多くあったので，適宜，追加，および，修正しながら，形式化を行った．結果を以下に示す．

オリジナルの安全要求には 24 個の項目（ゴール木におけ

送信元	送信先	場所	データ種類	データ	パターン
DF	FSF	in-chip	Digital	Signal	High
PD	MPU	Inter-chip	Analog	Voltage	Mid
MPU	PD	Inter-chip	Digital	Signal	High
MPU	PD	Inter-chip	Digital	PWM	High
⋮	⋮	⋮	⋮	⋮	⋮

表 2 適用パターン

るノード）が存在した．一方，形式化を行い追跡可能，すなわち，ゴール木の完全性が成立する安全要求では，53 個のノードが存在した．ここで，形式化した安全要求において，オリジナルの安全仕様の項目から修正が必要なかったものは 17 個，削除したものは 3 個，修正したものは 4 個，追加したものは 32 個であった．すなわち，形式化の際，67%の安全要求の項目を追加，もしくは，修正を行ったことになる．次に安全要求パターンに関係している部分についても分析を行った．形式化を行った安全要求においては 29 個のノードが安全要求パターンから生成されたものであった．安全要求全体の 55%をカバーしているのである．それらのうち，オリジナルの安全要求に出現しているノードは 12 個，修正が必要であったものは 4 個，安全要求パターンにより新たに追加されたものは 13 個であった．すなわち，安全要求パターンにより，それに関係する部分では，59%の安全要求の項目を追加，もしくは，修正を行ったことになる．また，安全要求全体に関しては，32%となる．

6.3 考察

提案手法では，多くの安全要求の項目を追加することができた．オリジナルの安全要求の記述では，追跡可能であるためには，多くの項目が不足していることがわかる．追加した項目のほとんどは，暗黙の前提となっていたものである．しかしながら，このような暗黙の前提が多く存在すると，本質的に必要な安全要求の項目を見逃してしまいがちである．例えば，6.1 節で指摘した，「メッセージを失敗無しで送信する」という安全要求は，他の通信に関する部分では記述されていたことから，暗黙の前提ではなく，見逃していたものであると考えられる．提案手法では，追跡可能とするために，暗黙の前提を明確にして多くの安全要求を記述しなければならないが，それにより，重要な安全要求の見逃しを回避することができたと言える．

ケーススタディで準備した安全要求パターンは 3 つであり，それらにより，安全要求の半分以上をカバーすることができている．本来，パターンは典型的な構造であり，すべてをカバーすることを目的とはしていないが，記述の効率化に関してパターンの効果は大きいと言える．暗黙の前提を置いてしまう原因は，繰り返し出現する安全要求を毎回記述するという冗長な作業に起因していると考えられる．のような安全要求は，安全要求パターンから実体化することにより効率的に記述することができ，暗黙の前提を回避することに繋がったと考えている．一方，安全要求パターンにより追加された項目は全体に対して 32%であり，全体で追加した割合 67%と比較すると，それほど多くは無い．このことから，パターンに該当する項目が，オリジナルの安全要求において出現していた部分が比較的多いことがわかる．つまり，安全要求パターンは，安全要求の本質的な項目を捉えることができていると考えられる．本質的な項目が骨格となり，次に，その関連項目が明確となり，全体として，多くの項目が明確になったと言える．

7. 議論

我々は，安全要求における追跡可能性を保証するために，ゴール木における完全性の考え方を導入した．これにより，安全要求記述のための明確な規範が導入され，完全性が成立するように注意深く項目を記述することができた．そして，多くの暗黙の前提を明確にでき，また，重要な安全要求の欠如を回避できることがわかった．ISO26262 に記載されている追跡可能性という考え方は，非常に抽象度が高く，曖昧なものである．本提案で導入した完全性のように，明確な規範を導入することにより，安全要求の分析で注意しなければならないことが明確になり，安全要求の品質を向上させることができるのである．

提案手法では，ゴール木と命題論理に基づいた浅い形式性を導入した．実際，KAOS のゴール木では時相論理を用いて要求を記述するが，安全要求の追跡可能性に関しては，本質的に，時間の前後関係に基づいた推論は必要無いと考えている．また，モデル検査では，時相論理を用いて性質を記述するが，意図に合った時相論理を記述するのは難しいことが知られている [8]．そこで，浅い形式化により実践的であり，かつ，十分な効果を期待できる手法にすることが本研究の狙いである．安全要求は，最低限，Modus Ponens に基づいて論理的に一貫しているべきであり，今回のケーススタディでは命題論理で十分であった．命題論理は単純な論理である．それにより追跡可能性を保証することは，理解されやすく，安全要求の品質を説明する際にも有効であると考えている．複雑な論理は，理解するのが難しく，確信も獲にくい傾向にある．一方で，単純な Modus Ponens は，多くの人に受け入れられやすい．それにより，追跡可能性を形式的に証明できているので，追跡可能であ

るという確信は獲やすいであろう．

一方，手法の一般性を考えると，数量を伴う安全要求も存在するであろうが，追跡可能性の保証の際，複雑な算術計算などに基づいた推論が本質的に必要になるとは考えにくい．そのような算術計算の詳細に立ち入らず，単純なModus Ponens で説明できる要求まで洗練するのが安全要求の策定では重要であろう．もちろん，システムの挙動を記述するためには，より高度な論理が必要であるが，それらは安全要求を実現する際など，別の工程で，考慮されるものであると考えている．

提案手法では，追跡可能性を実現するために，多くの項目を記述しなければならない．暗黙の前提を明確にできる一方で，当たり前と思える事実も記述しなければならない．安全要求パターンにより，繰り返し出現する重要な項目を取り扱うことは可能になった．一方で，安全要求パターンのカバー率は 55%であることから，残りの部分に関しては手作業で記述しなければならない．パターンを増やすことも考えられるが，それは，パターンの意義を低下させることになると考えている．つまり，パターンは典型的な構造や重要な構造を記述するためのものであり，無闇にその数を増やすことは，重要ではない構造も含まれてしまい，その観点からの意義を低下させることになるのである．よって，実際に開発現場で使っていくためには，パターンによる支援の他に，安全要求の補完支援なども必要であろう．

8. まとめ

本論文では，ゴール木と命題論理に基づいた，安全仕様記述手法を提案した．そして，電子制御ステアリングシステムの安全要求に適用し，様々な効果があることを確認できた．安全要求に関して，効果が上がって，かつ，実践しやすい形式仕様記述法を発見したと言える．現在，提案手法に基づいて，要件の入力から検証までを支援する統合ツールを開発中である．今後は，この統合ツールを電子制御ステアリングシステムの安全要求全体に適用する予定である．また，コストをかけて品質を担保した形式仕様は，システム開発において，積極的に参照すべきであり，安全要求の他工程での利用法などについても検討を行う予定である．

参考文献

[1] ISO 26262 Road vehicles - functional safety, 2011.
[2] IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems, 1998.
[3] Axel van Lamsweerde:Requirements Engineering: From System Goals to UML Models to Software Specifications, Wiley, 2011.
[4] R.A. Weaver and T.P. Kelly: The goal structuring notation-a safety argument notation, Workshop on Assurance Cases, Dependable Systems and Networks, 2004.
[5] Ewen Denney, Ganesh Pai, and Iain Whiteside: Formal Foundations for Hierarchical Safety Cases, International Symposium on High Assurance Systems Engineering, pp.52–59, 2015.
[6] Ewen Denney and Ganesh Pai: A Formal Basis for Safety Case Patterns, SAFECOMP 2013, LNCS 8153, pp.21–32, 2013.
[7] Tim P. Kelly and John A. McDermid: Safety case construction and reuse using patterns, Safe Comp 97, pp.55–69. Springer, 1997.
[8] Matthew B. Dwyer, George S. Avrunin and James C. Corbett: Patterns in property specifications for finite-state verification, International Conference on Software Engineering, pp.411–420, 1999.
[9] Robert Darimont and Axel Van Lamsweerde: Formal refinement patterns for goal-driven requirements elaboration, ACM SIGSOFT Software Engineering Notes, 21(6), pp.179–190, 1996.
[10] Erich Gamma, et.al: Elements of reusable object orientated software, Addison-Wesley Professional, 1995.

付　　録

A.1 獲得した安全要求の詳細

G1 System shall make transition to 'Manual Steering' If failure of voltage supplied to Current Control Unit has been detected.
CCU.VoltFailureDetected ⇒ S.State = 'Manual Steering'

G2 Demand for transition to 'Manual Steering' shall be sent to ECU Processing Unit if failure of voltage supplied to Current Control Unit has been detected.
CCU.VoltFailureDetected ⇒ DF.Send('Manual Steering', DF, MPU)

G3 Demand for transition to 'Manual Steering' shall be sent without failure.
DF.SendWithoutFaulure('Manual Steering', DF, MPU)

G4 Demand for transition to 'Manual Steering' shall be received if it is sent without failure.
(DF.Send('Manual Steering', DF, MPU) ∧ DF.SendWithoutFaulure ('Manual Steering', DF, MPU)) ⇒ DF.Received('Manual Steering', MPU)

G5 System shall make transition to 'Manual Steering' if Demand for transition to 'Manual Steering' shall be received.
DF.Received('Manual Steering', MPU) ⇒ S.State = 'Manual Steering'

G6 Demand for transition to 'Manual Steering' shall be sent to ECU Processing Unit if failure of voltage supplied to inverter has been detected.
Inv.VoltFailureDetected ⇒ DF.Send('Manual Steering', DF, MPU)

G7 Demand for transition to 'Manual Steering' shall be sent to ECU Processing Unit if failure of voltage supplied to Pre-driver has been detected.
PD.VoltFailureDetected ⇒ DF.Send('Manual Steering', DF, MPU)

G8 failure of voltage supplied to Current Control Unit has been detected if failure of voltage supplied to inverter or Pre-Driver has been detected.
CCU.VoltFailureDetected ⇔ (Inv.VoltFailureDetected || PD.VoltFailureDetected)

G9 System shall make transition to 'Manual Steering' if Pre-Driver stops.
PD.Status = 'Stop' ⇒ S.State = 'Manual Steering'

G10 ECU Processing Unit shall send 'Stop Demand' to Pre-Driver if ECU Processing Unit has received demand for transition to Manual Steering.
DF.Received('Manual Steering', MPU) ⇒ PD.Send('Stop Demand', MPU, PD)

G11 Pre-Driver shall stop according to 'Stop Demand'.
PD.Received('Stop Demand', PD) ⇒ PD.Status = 'Stop'