Safety/Assurance Case ガイド（Ver. 1.0）

2012/03/16

本資料は JST CREST DEOS Project での石川裕研究チームの成果物の一つです。

本文書は、Safety/Assurance Case に関する資料を抜粋し、初学者でも分かる単独の読物として作成されたものである。

目次

１．　本資料について

本資料は、最終報告書から Safety/Assurance Case に関する、初学者でも理解できる資料として作成されたものである。

２．　Safety Case

Safety Case は、 Assurance Case の一つの例であり、それ以外にも Maintainability Case や Dependability Case など、様々なシステム特性に基づく Case が作成されている。基本的には、システム特性 X に対して X-Case の作成が、そのシステム特性を保証されるために作成することが可能である。

安全性の保証のための枠組みである Safety Case を主な分析対象にする主な理由としては、Safety Case が様々な規格、ガイドラインにおいて最も利用されているので、D-case の利用について一番参考になると考えられるからである。

Safety Case の作成、提出を義務付けている規格はいくつかあり、本資料においては、以下のものを主に参考にした。

・Yellow Book [1]
・EUROCONTROL [2]
・ISO 26262 [3]
・Def-Stan 00-56 [4]

ISO 26262 以外は略称であり、正式名称は参考文献を参照されたい。

Yellow Book は、鉄道システムに対して改変が行われる場合の安全管理に関するガイドラインである。

EUROCONTROL はヨーロッパの 29 カ国が加盟している、航空管制の安全に関与する組織であり、安全で機能的な航空管制管理（Air Traffic Management） を提供するものである。そこでの安全管理のための Safety Case 作成のガイドラインが文献 [2] である。

ISO 26262 は自動車の電気、電子システムに関する機能安全規格である。機能安全の完全性の保証のために Safety Case の作成が義務付けられている。

Def-Stan 00-56 は英国防衛省が策定した、防衛システムの安全管理システムに関する規格である。

これらのガイドラインとは別に、 Safety Case の設計に関する資料としては、以下を参考とした。

・Modular Software Safety Case Process [5]

上記の資料は 英国における二つの主要な航空機システムインテグレーターである BAE Systems と General Dynamics UK で構成される、Industrial Avionics Working Group が、 Safety Case 構築のより効率的な方法論の確立のために行われた研究の成果資料である。

本資料においては、Safety Case と、Safety Case Report と呼ばれる成果物を明確に区別する。Safety Case は対象システムの安全性を保証するための議論の構造を意味するのに対して、Safety Case Report の定義としては以下のものを採用する。

> *A **Safety Case Report** is a key deliverable that summarises the Safety Case at a particular instant in time. It provides **assurance** to the Duty Holder that safety is being managed effectively, highlights areas of safety-related project risk requiring management attention and gives stakeholders visibility of the status of the Safety Case.*
>
> ([4], page 9)

Safety Case と Safety Case Report を混同して利用している例は多い(例えば、[2])。本資料においては、 Safety Case は議論の構造であり、 Safety Case Report はそれを元に作成された、安全性の保証のための審査に利用される文書であるとする。Safety Case Report については第３章で述べられる。

次章において、Safety Case がどのような状況で、どのような目的に対して、誰が利用するかについて述べる。

２．１．　Safety Case の利用方法

Safety Case の利用方法については、いくつかの分類方法が考えられる。本資料においては、規格やガイドラインで作成が義務付けられているかどうかを最初の大きな分類として扱うことにする。

【作成される前提条件】

- 規格、ガイドライン、法規により作成、提出が義務付けられている。
- 義務付けられていないが作成を行う。

義務付けられていない Safety Case に関しては、記述方法などは自由であり、またどのようなドキュメントをそこから作成するかは任意であるので、本報告書においては触れない。このような利用方法の目的としては、従来の安全規格などにおける prescriptive （処方箋的）なアプローチが不十分であることから、目的ベースで安全に対する主張を行うた

めに利用するものである。これは、1988 年 7 月における、北海油田における Piper Alpha 事故（167 名死亡（229 名中）、270 億円の被害）に対する、Cullen 卿による事故調査レポートにおける以下の言葉にもうかがえる。

Compliance with detailed prescriptive regulations was not sufficient to ensure safety

しかし､現在においては規格に従い､与えられたパターン(テンプレート)を用いて Safety Case を記述すれば安全が保証される、といった傾向があり、それはある意味 prescriptive なアプローチであり、反省すべき点であると言える。

規格、ガイドライン、法規において作成、提出が義務付けられている例としては、列車システムに関する Railway Yellow book [1] 、航空管制システムについての EUROCONTROL [2]、車載組み込みシステムの機能安全規格である ISO 26262 [3] 、防衛関係 Defence Standard 00-56 [4] などがある。また、これらのガイドライン、規格においては、詳細度の程度の差があるが、どのような Safety Case を作成する必要があるかが規定されている。

Safety Case にも様々な種類があり、規格、ガイドラインにより異なる規定が行われている。さらに、単一の Safety Case だけを作成すれば良いのではなく、複数の性質の異なる Safety Case の作成が義務付けられている場合もある。

例えば、 Railway Yellow Book においては、Engineering Safety Case （変更に関する安全性）と Railway Safety Case（組織の安全管理に関する Safety Case）の両者の作成が義務付けられている。

また、注意しなければならないのは、これらのガイドライン（例：[1]、[2]）は、 Safety Case 作成の規定ではなく、より包括的なシステムの安全性の保証、管理に関するガイドライン、規格であり、Safety Case はその一部であることである。Yellow Book の目的については、以下のように記述されている。

> *The main purpose of the Yellow Book is to help you set up a process that protects you and others from mistakes and gives documented evidence (the engineering safety case) that risk is at an acceptable level.*
>
> ([1] , Vol. 1, Page4)

さらに、Yellow Book では、どのような内容について記述するかを明確に規定している。

> *Among other things, the railway safety case must describe:*

・*the operator's safety policy and arrangements for safety management;*
・*the operator's assessment of the risk;*
・*how it will monitor safety;*
・*how it organises itself to carry out its safety policy; and*
・*how it makes sure that its staff are competent to do safety-related work*

([1] , Vol. 1, Page5)

上記の記述により、 Yellow Book における Safety Case の書き方、議論の構造を規定している。

ISO 26262 [3] では、 Safety Case の記述は、 Part 1、Part 2、そして Part 10 において述べられている。Part 1 の Safety Case の定義では、以下のように記述されている。

1.106
safety case
argument that the safety requirements for an item (1.69) are complete and satisfied by evidence compiled from work products of the safety activities during development

NOTE Safety case can be extended to cover safety (1.103) issues beyond the scope of ISO 26262.

([3] Part 1, page 14)

Safety Case の定義として標準的であると言えるが、 item というシステム（もしくはサブシステム）を表す ISO 26262 独自の用語については注意が必要である。特に注意として記されているのが、ISO 26262 のスコープを超えて利用することも可能である点である。その場合には、規格に規定された内容とは異なる議論を独自に構築する必要がある。

ISO 26262 における Safety Case の記述に関しては、以下のように規定されている。

6.4.6 Safety case

6.4.6.1 *This requirement shall be complied with for items that have at least one safety goal with an ASIL (A), B, C or D: a safety case shall be developed in accordance with the safety plan.*

6.4.6.2 *The safety case should progressively compile the work products that are generated during the safety lifecycle.*

([3], Part 3, page 13)

すなわち、本要件（Safety case）は、ASIL (Automotive Safety Integrity Level) A、B、C、D が割り振られた少なくとも一つの安全ゴールを持つ item のために作成され、Safety Case は 規格で規定された safety plan にしたがって開発されなければならない、という点と、安全ライフサイクルの間に生成された work product （規格で規定された提出すべき成果物）を、漸次的に集めなければならない、ということが示されている。

ここで、分かるのは、Part 3（Concept Phase）（Part 3 では安全分析、評価、安全ゴール、安全機能要件の作成が行われる） で作成される、ASIL が割り当てられた安全ゴールに関する議論に対して利用され、根拠として利用されるのは work product である、ということである。

次に考慮すべき点は、Safety Case の作成から破棄されるまでのライフサイクルである。Safety Case は作成された後、レビューを受け、その完全性についての検証が行われる（[3] Part 2, Table 1, page 15)。その後、受理され、改定が必要になれば保守が行われ、最後に必要ななくなると放棄される。ここから分かるのは、Safety Case は通常のシステムライフサイクルと同等のものを持っている、ということである。

１） 作成
２） レビュー
３） 受理
４） 保守
５） 放棄

保守に関しては、変更部分に関するインパクト分析を行い、その結果に従って既に作成された Safety Case を改訂し、それに合わせて、 Safety Case から作成されたドキュメントに変更を行う必要がある。

Safety Case のレビューには様々な条件があるが、ここでは ISO 26262 で規定されたレビューの方法を示す。

C.2 Review of the completeness of the safety case (see 6.5.3)

C.2.1 *Confirmation that the work products referenced in the safety case are available and sufficiently complete, so that the item's achievement of functional safety can be adequately*

evaluated.

NOTE The referenced work products can be the work products that are identified as relevant to support the safety case.

***C.2.2** Confirmation that the work products referenced in the safety case:*

- *are traceable from one to another,*
- *have no contradictions within or between work products, and*
- *either have no open issues that can lead to the violation of a safety goal, or have only open issues that are controlled and have a plan for closure.*

([3], Part 2, page 21)

ISO 26262 においては、Safety Case の中で work product （提出すべき成果物）が参照されており、レビューにおいては、それらが利用可能であり十分に完全であるか確認することで、item の機能安全の達成が、適切に評価されること、さらに、 Safety Case において参照される work product については、互いにトレーサブルであり、相互に矛盾が無く、安全ゴールを侵害するようなオープンイシューが無いことを確認する必要があること、と記されている。機能安全規格においては、対象システムの開発プロセスにおける成果物間の前方、後方トレーサビリティを必要条件としている。しかし、ここで興味深いのは、認証に関わる成果物間のトレーサビリティを述べている点であり、そのために Safety Case が利用されている点である。

次に Safety case の作成、利用に関して、どのような人間（ロール）が関与するかを分析する。

【Case に関わりがあるロール】

一般的には以下を考慮する必要がある。

１） 開発者
２） プロジェクト・マネジャー
３） 安全マネジャー
４） 安全アセサ、オーディター

ここでの開発者はシステムの開発者という意味である。プロジェクト・マネジャーは、開発チームのコンタクト先であり、規格で要求される要件が満たされていることについての責任を持つ。安全マネジャーは、安全性についての責任者である。そして、最後にアセサ（オーディター）は、安全性が効率的に管理されており、安全管理が法規に沿って施行されていることを審査する者である。

様々な規格においては、これらのロールに対して厳密に規定しているものもあるし、非常に一般的に規定している場合もある。また、ロールの独立性について記述している場合もある。例えば、 ISO 26262 では、ASIL のレベルにより独立性（I0 から I3）を規定している（[3] Part 2, Table 1, page 15)。Def-Stan 00-56 [4] においては、内部的に審査が行われる場合には、審査対象となる部門とは別のチームにより審査は実行されるべきであると述べている。

ここに上げた全てのロールは Safety Case に関与している。本案件の Safety Case から文書を作成する拡張機能に関わりがあるのは、開発者、プロジェクト・マネジャー、安全マネジャーである。どのようなフォーマットの書類（例えば、 Safety Case Report）が必要になるかの決定権を持っているのは、安全マネジャーであり、実際に書類を書くのは、開発者と安全マネジャーであると考えられる。以下のものは Yellow Book における Project Safety Manager の定義である。

> *Person responsible for safety on a project and for producing all safety-related documentation*
>
> （[1], Page A-4）

最後にシステムの開発ライフサイクル（企画、要求分析、設計、実装、検証、運用）に沿って、 各プロセスに関する Assurance Case を作成する、という考え方がある（すなわち、企画に関する Assurance Case、要求分析に関する Assurance Case、など)。しかし、このような考え方はまだ、確立されておらず、本案件におけるユースケース分析の参考にするのには、十分な資料が無いので、本資料においては取り扱わないこととする。

【Case 作成のプロセス】

Safety Case の作成プロセスを規定している規格も存在する。例えば、EUROCONTROL [2] においては、二つの異なる Safety Case （Unit Safety Case と Project Safety Case ）の作成と、開発プロセスが示されている。二種類の Safety Case の定義は以下のものである。

・Unit Safety Case

現在、進行中で定常的な運用が安全、かつこれからも安全を保持できること示す Safety Case

・Project Safety Case

現在ある安全に関連するサービスやシステム（新しいサービス、システムの導入を含む）

が実施されるときに作成される。

図１のプロセスで示されているのは、安全管理システム（安全計画、安全分析、安全アセスメントと二種類の Safety Case との関連である。

本プロセスでは、安全への考察をしてから、初期的な安全議論を構築し、それから安全計画を立案している。初期的な安全議論を構築した後は、それを利用して Project Safety Case の構築を行う。安全計画に従って、安全のライフサイクルが回り（中央の黄色い部分）、その結果は Project Safety Case の根拠資料になる。

安全アセスメントに関しては、FHA（Functional Hazard Assessment）、PSSA(Preliminary System Safety Assessment)と、SSA (System Safety Assessment)の三種類のアセスメントを安全計画、操作状況を元に行う。SSA においては、実装、システム結合に関するアセスメントの結果は Project Safety Case において利用され、その結果により許可が降りると、運用への移管、運用・保守に関するアセスメントが行われ、それが Unit Safety Case に用いられる、というプロセスになっている。

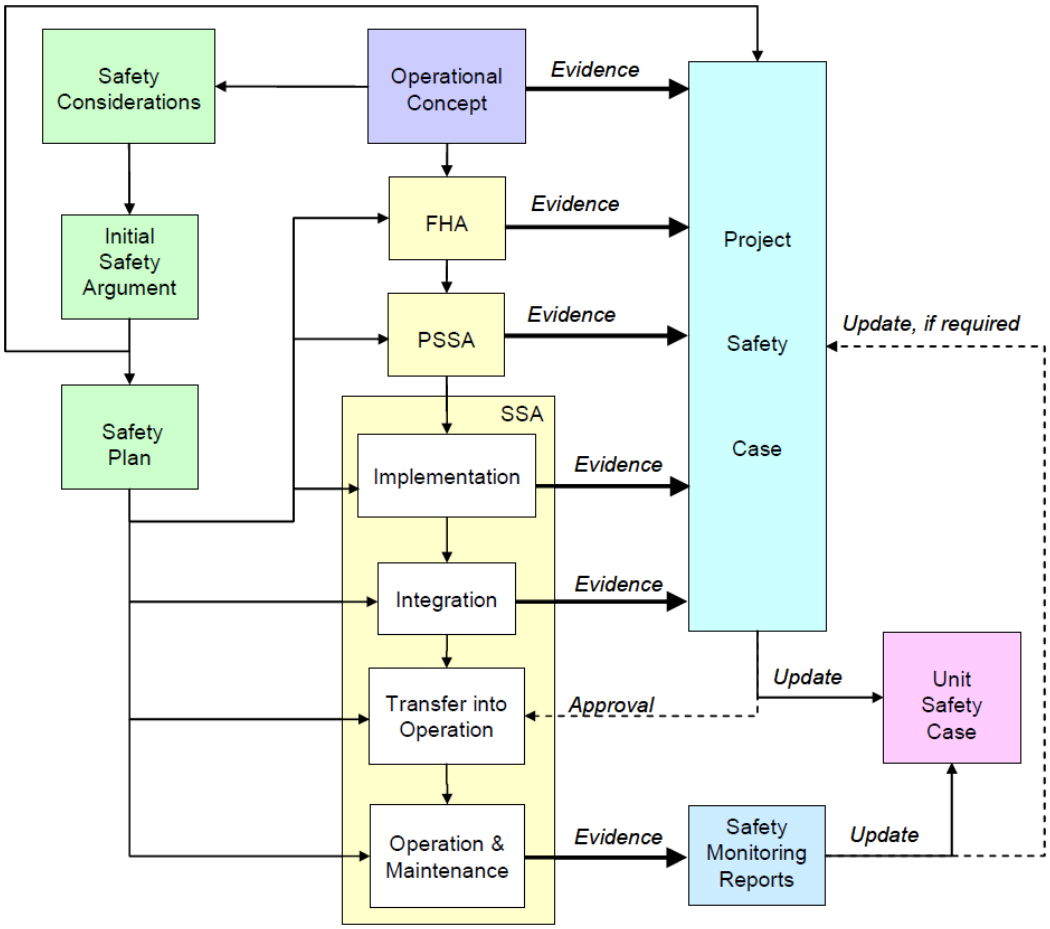

図 1．EUROCONTORL における Safety Case 開発プロセス（[2], page 8）

これに対して、 Safety Case 自身の設計についてより詳細に述べているのが Modular Software Safety Case Process [5] である。

[5] においては、プロセスは、「製品のライフサイクルの分析」から始まり、「ソフトウェア設計と Safety Case アーキテクチャの最適化」に続き、「モジュラー Safety Case の作成」へと移るプロセスを提案している。最適化は、「安全に関する議論の設計」と「ソフトウェア Safety Case アーキテクチャの定義」という二つの部分プロセスに分解される。このような設計プロセスが可能なのは、まず Safety Case の構造がモジュール化されているので、議論の構造の分割統治が容易であることである。

このような考え方は、当然 D-case 図の作成においても利用することが可能である。例えば、 ISO 26262 の Safety Case の考え方からすると、work product をどのように取り扱うかが非常に重要であり、そのためには、初期的な議論の構造は以下の図に示されるようになるはずである。

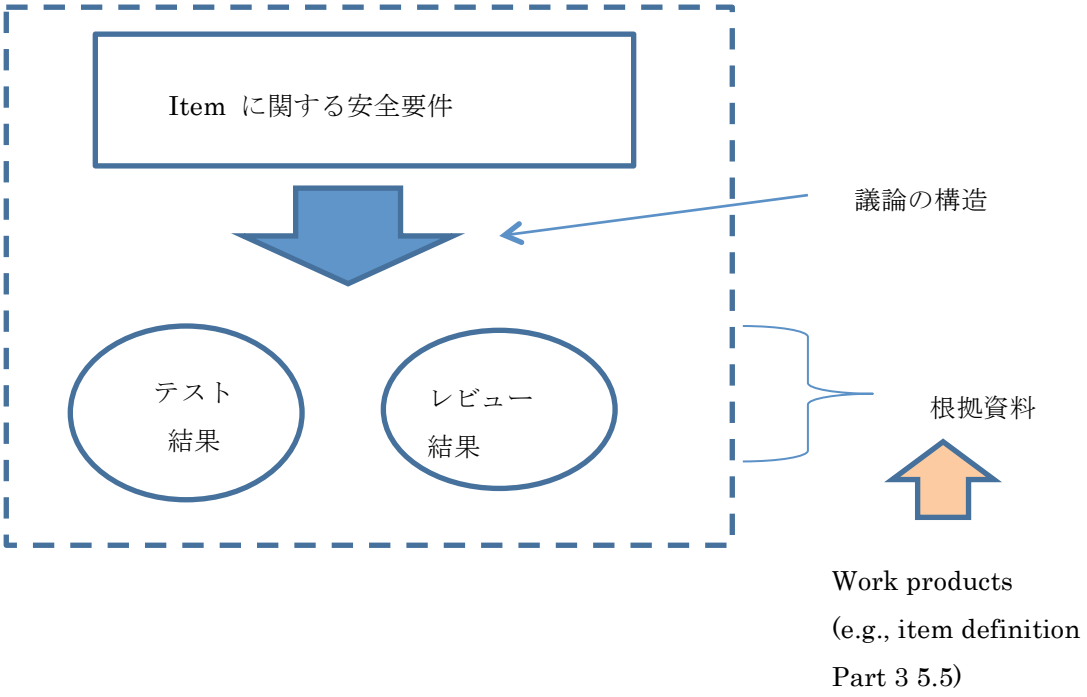

図 2. ISO 26262 の Safety Case の初期的な議論の構造

このような初期の議論の構造を決め、アーキテクチャを決めることで、根拠が正しいテンプレートの開発が可能となる。

２．２．　Assurance Case の国際標準化

ここでは、簡単に現時点における Assurance Case に関する国際標準化の動向について述べる。OMG (Object Management Group) の System Assurance Platform Task Force (SysA PTF) においては、ARM (Argumentation Metamodel) FTF beta [9] と SAEM (Software Assurance Evidence Metamodel) FTF beta [10] が策定され、現在は両者を統合するための SACM (Structured Assurance Case Metamodel) [10] の策定が進行中である。OMG は 1989 年に設立されたソフトウェアシステムに関連する規格の標準化団体であり、これまでも UML (Unified Modeling Language) や CORBA (Common Object Request Broker) などの標準化で知られている。 SysA PTF は、システム保証に関する標準化を議論する OMG の一部会である。SysA PTF の座長は、B. Calloni (Rockheed Martin), D. Campara (KDM Analytics) と田口研治（産業技術総合研究所／シーエーブイ・テクノロジーズ）である。

ここで FTF (Final Task Force) とは OMG の用語で、規格の最終稿を意味する。

ARM は議論の構造、 SAEM は根拠資料についての規格という分け方から、その統合へと規格の策定は進んでいる。これらの規格は、利用方法などについての規定をしている訳ではなく、Assurance Case の概念的構造を規定しており、本案件とは直接的には関連したものではない。しかし、どのような規格であるかを、背景知識として知ることは必要なので、簡単に SAEM と ARM について説明する。

OMG における規格は UML をモデル記述言語としてメタモデルを定義することで行われる。SAEM は 20 のクラス図から構成されており、論理構造としては、 Exhibits、Fact Model、 Properties、 Evidence Evaluation、Administration から構成されている。

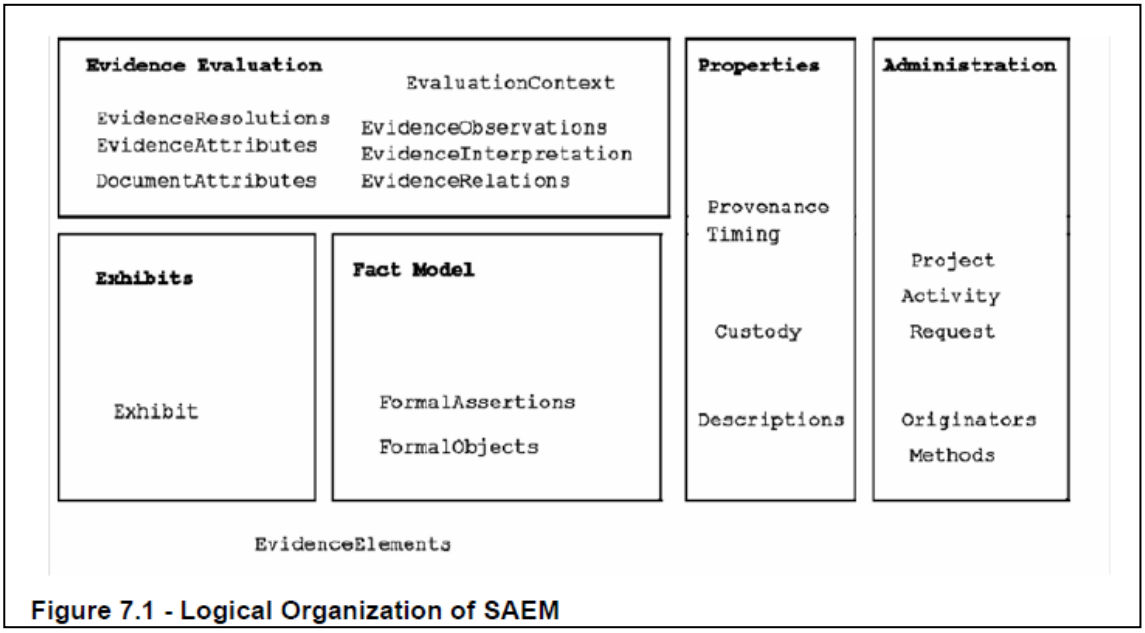

Figure 7.1 - Logical Organization of SAEM

図 3. SAEM [9], p15, Figure 7.1

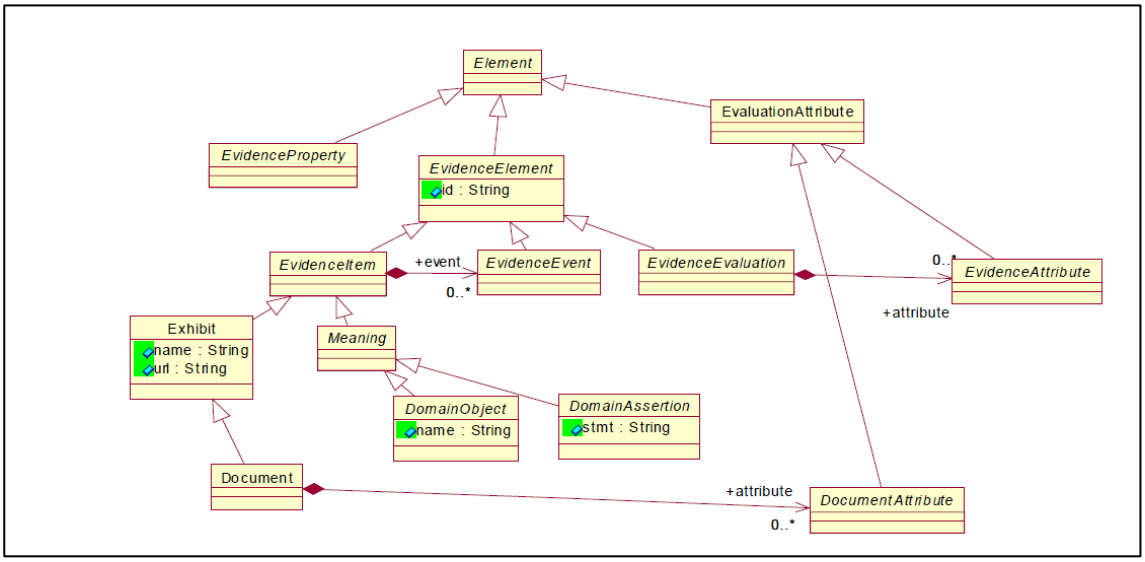

図 4. SAEM [9], p16, Figure 7.2

図 4 は SAEM における主なクラスを示している。EvidenceElement は根拠（証拠）資料として提出された物理オブジェクトを示すもので、主要構成要素になっている。EvidencProperty は根拠（証拠）の主要構成要素の属性を示している。EvidenceItem は根拠（証拠）として収集された対象を表す。Exibit は根拠（証拠）である物理オブジェクトを示す。DomainAssertion は根拠（証拠）に関する命題である。EvidenceAttribute は評価において主張された根拠（証拠）資料の特性を示す。

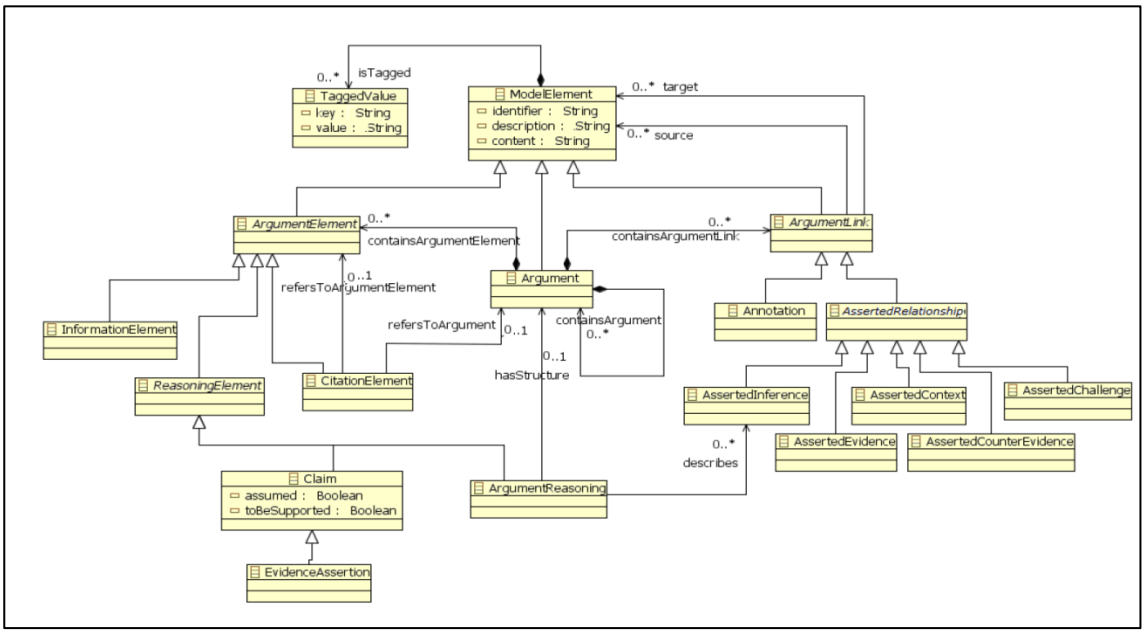

図 5. ARM [8]、p21、Figure 8.1

図 5 で示されたものが、 ARM のメタモデルであり、 SAEM とは異なり、簡潔に一つのクラス図で表されている。ここでは、Argument クラスが、構造化議論全体を統括している。ArgumentElement において、Argument 全体を構成するクラスを表し、ArgumentElement を ArgumentLink がつないで、構造化議論になるという解釈をしている。ReasoningElement クラスは、推論の基礎になるクラスであり、その部分クラスとして Claim クラス（CAE [7] でいう Claim、GSN [6] でいう Goal）と、 Claim 間の推論関係を規定する ArgumentReasoning クラスにより構成されている。根拠資料のクラスとしては、 EvidenceAssertion クラスがあり、本クラスは Claim のサブクラスとして規定されている。

ARM の特徴としては、その簡潔さから拡張が容易になっている点が上げられるが、GSN と CAE のコア部分だけを規定しており、例えば GSN におけるモジュールや、パターン言語への拡張は全く入っていない。今後、Assurance Case を支援するツールにおいては、相互互換性を考えた場合、 ARM への準拠が必要になると思われる。

２．３．　ゴール構造の作成方法

本章では、D-case 図の基礎である GSN 記法による議論構造の基本的な作成方法について述べる。ここで示されている作成方法は D-case 図作成にも用いることが出来る。

高度な Case の作成方法としては、アーキテクチャの決定方法やパターンの利用などがあるが、ここではふれない。ここで述べられる方法は、 GSN Community Standard [6] に記述されているものである（元々のソースは、 T. Kelly の博士論文 [12] に記述されたも

のである)。

D-case 図を作成する作成者として最初に注意すべき点は、以下の点である（[6])。

・文書化された議論の明確さ
・文書化された議論の理解しやすさ
・文書化された議論の真実性（veracity）

議論の明確さとは、各々の主張と参考資料は容易に理解可能であり、議論の論理的流れが明確であることである。議論が理解しやすいとは、作成者と読者の両者が議論における主張に対して、共通理解を持っていることである。議論の真実性とは、議論は根拠資料と推論の真実の状態を正確に反映していることである。

議論構造を作成する場合には、構造に対してトップダウンのやり方とボトムアップのやり方がある。ここでは、トップダウンによる作成方法について述べる。

【トップダウンアプローチ】

ステップ１） 支持すべき Goal （ゴール）を同定する
ステップ２） ゴールが述べられている Context （文脈）について定義をする
ステップ３） ゴールを支持するために利用される Strategy （戦略）を同定する
ステップ４） Strategy が述べられている基礎について定義する
ステップ５） Strategy を洗練化する（そして、ステップ１に戻り新しいゴールを同定する）かステップ６に移る
ステップ６） Evidence（根拠資料）を同定する。

議論の構造を作成する際に最初に行うのは、トップゴールを同定することである。トップゴールの記述は、適度なレベルの詳細度を持つ必要がある（ステップ１)。主張が真であるためには、それを支持するゴール構造を作成する必要がある。ゴールを支持する大切な情報が、どのような文脈において、その主張がされたかを示す Context ノードである。Context ノードには、以下の情報を示す必要がある（ステップ２)。

・議論の対象となるシステムに関する情報
・システムの環境に関する情報
・議論に関する情報（用語、規格、等）

次には主張をどのように根拠付けるための戦略を決める必要がある（ステップ３)。戦略

には、様々な種類がある。例えば、次章において述べられる最初のテンプレートにおいては、リスク指向の戦略が用いられている。これは、ハザード分析をした結果を用いて、各ハザード毎にその安全性を保証する議論を構築する戦略である。他にも分割統治（divide and conquer） では、大きなゴールと小さなゴールに分解することで行われる。

ゴールと同様に Strategy がどのような文脈において用いられるかを示す必要がある（ステップ４）。もし、Strategy によるゴールの分解の粒度が荒すぎる場合には、Strategy を複数作成し、さらなるゴールの分解を行う必要がある（ステップ５）。最後に、それ以上、議論構造にゴールを付加することが出来ない段階になったら、 Evidence （根拠資料）を付け加えることで、ゴール構造の作成は終了する（ステップ６）。

参考文献

[1]Railtrack, Engineering Safety Management Issue 3, Yellow Book 3, Volume 1 and 3, Fundamentals and Guidance, 2000

[2] European Air Traffic Management, Safety Case Development Manual, European Organisation For the Safety of Air Navigation, Ed. 2.2, Nov 2006.

[3] ISO 26262, Road Vehicles – Functional Safety, 2011

[4] Ministry of Defence, Defence Standard 00-56, Issue 4 Publication Date 01, June 2007

[5] Industrial Avionics Working Group, Modular Software Safety Case Process, Part A: Process Definition, Oct 2007

[6] GSN Community Standard, 2011

[7] ASCAD, Adelard Safety Case Development Manual, 1998, Adelard

[8] OMG SysA PTF, Argumentation Metamodel (ARM), FTF beta, 2010, Aug

[9] OMG SysA PTF, Software Assurance Evidence Metamodel) (SAEM), FTF beta, 2010, Oct

[10] OMG SysA PTF, Structured Assurance Case Metamodel (SACM), 策定中

[11] D-Case エディタ機能仕様書, Ver. 0.7, 2011, 6 月

[12] T. Kelly, Arguing Safety: A Systematic Approach to Managing Safety Cases, D. Phil Thesis, University of York (1998), http://www-users.cs.york.ac.uk/~tpk

[13] I. Habli, et. al., “Model-Based Assurance for Justifying Automotive Functional Safety”, in the Proceedings of the 2010 SAE World Congress, Detroit, Michigan, USA, April 2010