第三分科会 機能安全グループ

安全関連システムの分割要件の考察

Consideration of requirement of decomposition for a safety related system

主査：飯泉 紀子（(株）日立ハイテクノロジーズ）
副主査：田所 孝文（(株）山武)、吉澤 智美（NEC エレクトロニクス（株））
研究員：大野康昭

機能安全規格 IEC61508 の個別適用規格 ISO 26262 における、システム分割の概念に着目して、ソフトウェアを内包するシステムの分割概念を考察した。
安全性関連システムを分割して、安全性要件を各サブシステム／コンポーネントに分割するには、これらの独立性を要件としているが、現状では相互監視の観点で不十分である。本研究では分割の条件を考察し、その条件を検討している。その結果、システム分割の為には、分割されたサブシステム／コンポーネントの独立性及び故障検出率が重要であるとの結論に達した。そして独立性及び故障検出率をマトリックスで表現して、システム分割の条件とする事を提案する。

We considered the concept of system decomposition paying attention to the decomposition concept of the system which includes software in individual application standard ISO 26262 of functional safety standard IEC61508
In order to divide a safety related system and to deliver the requirements for safety into each subsystems/components, such independency is made into requirements, but under the present circumstances, it is insufficient in view of bilaterally supervising.
This research considers the attribute of decomposition and is examining the attribute.
As a result, the conclusion that the independency and the diagnostic coverage of the divided subsystem/component were important for system decomposition was reached.
And it proposes making independency and the diagnostic coverage into a matrix, and considering it as the attribute of system decomposition.

1.背景

近年電気･電子･プログラマブル電子を応用した、組み込み機器に関する安全性への関心が高まっている。その結果安全性に関する B 規格である、IEC61508 及び、その個別適用である C 規格及び指針が次々と制定されている(付表 B.1)。
自動車産業においても、ISO SC3/TC22/WG16 において、2005 年より、2007 年のコミッティードラフト制定に向けて、ワーキングドラフト(ISO WD26262)の作成作業が開始されており、日本からは(社)自動車技術会が窓口になり、規格制定作業を行っている。
自動車のみならず組み込み用電気／電子／プログラマブル電子機器において、ソフトウェアの果たす役割は、年々増加しており、ソフトウェアの安全性確立が急務の課題である。

2.研究の目的

機能安全性規格 IEC61508 においても ISO WD26262 においても、リスクは重大性とその起き得る頻度の組み合わせと定義される。すべてのシステムはそのリスクに応じて、4 段階の安全尺度にクラス分けする。この尺度を IEC6150 では SIL(Safety Integrity Level)、ISO WD2626 では ASIL(Automotive Safety Integrity Level)と呼ぶ。安全性に関連したシステムは SIL/ASIL に応じて、リスクの発現する可能性を低減する為の様々な手法の導入を要求される。危険事象の起き得る頻度を許容可能な頻度(tolerable risk)まで低減する事により、対象システムの安全性を確保しようとする思想である。リスク低減後に残った残余のリスクは残存リスク(Residual Risk)と呼ばれ、このリスクは許容可能なリスク以下で無ければならない。しかしソフトウェアの性質を考えると、ソフトウェア工学における、プロセスや各種手法を導入して、ソフトウェアの信頼性を向上させたとしても、その結果が期待する残存リスク以下である事を証明する事は極めて難しい。

一方 ISO WD26262 には Decomposition と言う概念がある。これは一つのシステムを機能分割して相互に監視し合う事により、対象システムの安全性要求である ASIL を下げる（D→C+A、C→B+A 等）事を可能としている。この概念を図示すると、図１の様になる。しかし現状の ISO26262 では、独立性の記述は有るものの曖昧であり、相互監視の概念も不明瞭である。これでは安全性関連システムが野放図に分割される事となり、全体の安全性を損ねる懸念がある。そこで本研究では、現状の ISO WD26262 をベースとして、ソフトウェアを含んだシステムにおける Decomposition の要件を考察する事とした。

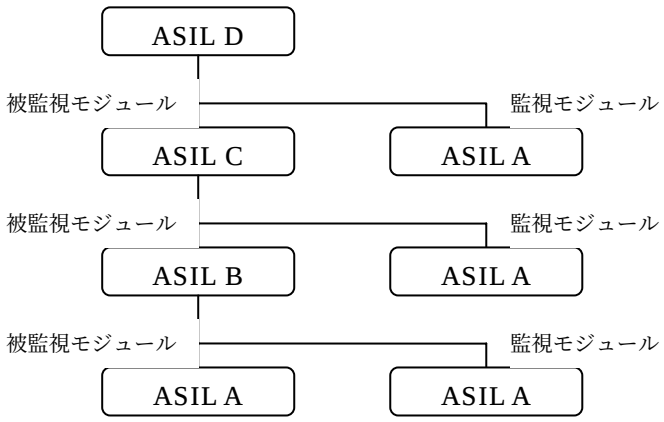


図 1 Decomposition の概念

論文の構成

本研究は IEC61508、ISO26262 を勘案した上で、以下の構成となっている。

1. 監視システムの考察
2. ソフトウェアによる監視機構の考察
3. システム分割の要件の提案
4. 結論及び考察
5. 終わりに
6. 付録 A　IEC61508/ISO26262 の概要
7. 付録 B　付図･付表

1. 監視システムの考察

Decomposition の前提となる、現状の ISO WD26262 ではこれらの手法はハーウェアのアーキテクチャ設計要件として記述されているが、その実現には、ソフトウェアの介在が必要不可欠である。システムの監視アーキテクチャは以下が考えられる。各々固有の検出率(Diagnostic coverage)と、独立性をそのアトリビュートとして持っている。

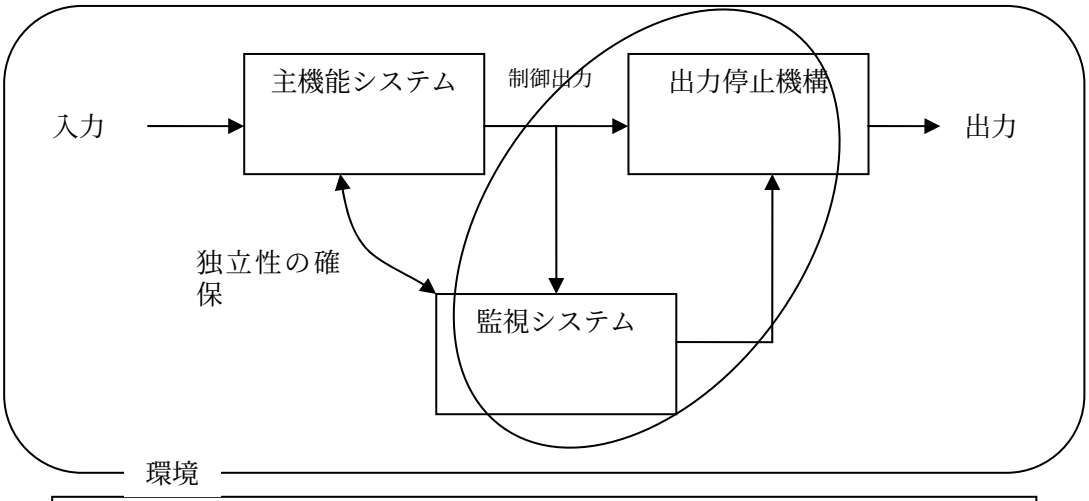
(ア) 他のシステムによる監視システム

(イ) 同一のシステム内の他のハードウェアに実装された監視システム。(監視 IC 等)

(ウ) 同一のハードウェアに実装された監視システム。(内蔵ウォッチドッグ等)

(エ) ソフトウェアのみによる監視システム。

監視システムを考える時、被監視システムである、主機能システムの障害により、監視システム自体が機能を停止する、或いはその逆で監視システムの障害が原因となって主機能システムが阻害されるモードを CCF(共通因子故障：Common cause failure)と呼ぶ。CCF の排除の為には、監視システムと主機能システムの独立性が重要になる。システムの独立性は一般に（ア）が最も高く、(エ)が最も低い。次に一般的な監視機構をもったシステムを考える。



同一因子故障：被監視システムと監視システムが同一の原因で、機能障害に陥る事。例えば、監視システムが監視の為に、被監視システムと同一の入力を使っていて、入力が原因でシステム障害に陥る場合。

図 2 監視機構を持つシステム

図 2 は一般的な監視機構を表しているが、このシステムでは主機能システムの異常を監視システムが検出し、あらかじめ設定された安全状態(Safety State)にシステムを遷移させ、その状態を維持する。同様に監視システムの異常を主機能システムが検出し、Safety State に遷移・維持する。これは監視システムの喪失が単一故障で即危険状態に至る潜在故障(Latent failure)を避けるためである。例えば ABS/VSA 等の電子制御のブレーキシステムにおいて故障を検出した時に、警告灯の表示で運転者に異常を知らせると共にシステムを機械式ブレーキシステム相当に戻した時の状態が、安全状態である。

監視機構のアーキテクチャは IEC61508 のパート 6 で種々提案されているが、ここでは省略する。この様な監視機構を持つシステムが危険事象に至る確率は、FT 図で図 3 の様に表現できる。すな

わち、単一故障が原因の危険事象の発生は、主機能部と監視部が同時に故障した時か、あるいは CCF 発生のときである。図 3 においてシステム全ての危険事象の発生率を λ_S、主機能部の故障率を λ_f、監視部の故障率を λ_d、CCF の発生率を λ_{CCF} とした時、λ_f、λ_d、λ_{CCF}、は十分小さいので、λ_S は次式で近似される。

$$\lambda_S = \lambda_f * \lambda_d + \lambda_{CCF}$$

ここで、CCF の発生率は、独立性(ｉ)に依存するので、$\lambda_{CCF} = f(i)$ となり、このシステム全体の危険事象発生確率は、独立性(ｉ)が関連する事が判る。

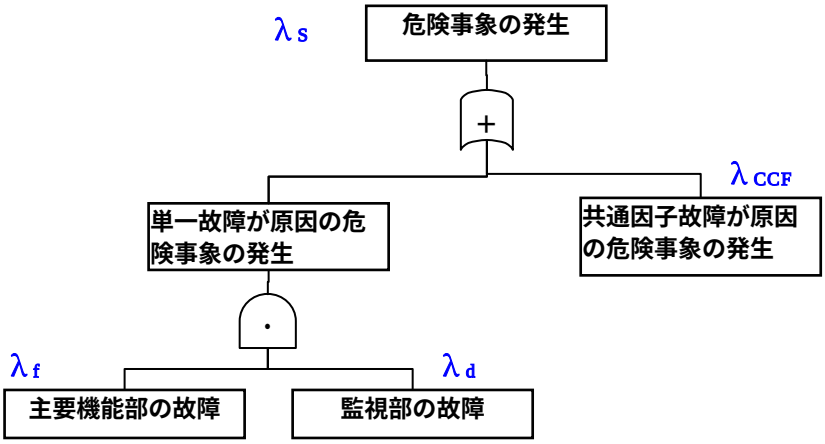


図 3　監視機構を持つシステムが危険事象に至る時の FT 図

次に各々のシステムの故障検出率を考慮して、単一故障が原因の危険事象の発生確率を FT 図で表すと図 4 の様になる。故障を検出できればシステムは安全状態に移行できるので、危険事象が起きるのは、監視部が故障しているのに主機能部が検出できず、その後主機能部が故障した場合か、検出できない主機能部の故障が発生した時である。

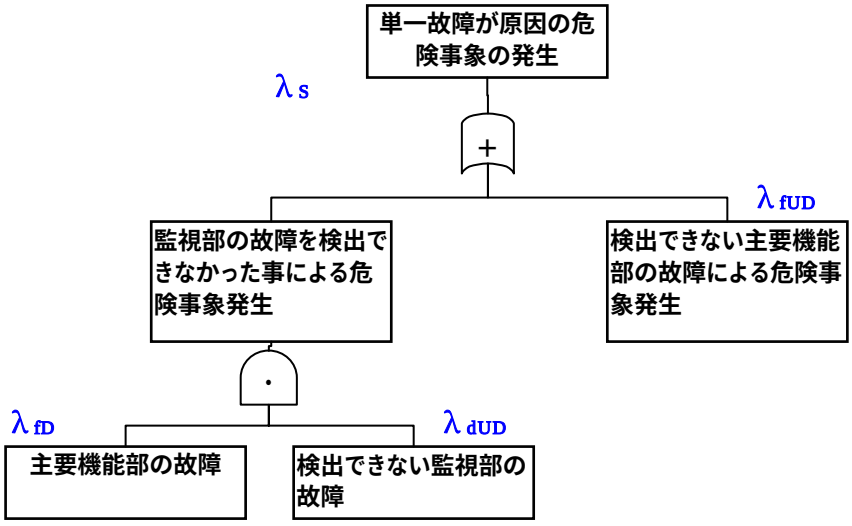


図 4　監視機構を考慮した田につ故障による危険事象発生の FT 図

ここで検出可能な主機能部の故障率を λ_{fD}、検出不可能な監視部の故障率を λ_{dUD}、検出不可能な主機能部の故障を λ_{fUD} とした時、λ_S は同様に次式で近似される。

$$\lambda_S = \lambda_{fD} * \lambda_{dUD} + \lambda_{fUD}$$

これにより、危険事象の発生には、故障検出率が関与している事が判る。
以上の考察より、相互監視機構を持つシステム全体の危険事象発生率は、主機能システムと、監視システムの独立性と、監視機構の検出率に強く関連する事が判る。

2. ソフトウェアによる監視機構の考察

次に同一ハードウェアにおける相互監視を実現するソフトウェアに関して考察する。当然ながら監視システムと被監視システムは同一環境の中で動作しており、これらは環境から自由になる事は不可能である。従ってここで言う独立性はあくまでも環境を除外した、被監視システムと監視システムの独立性を指す。環境として勘案されるべき項目の中で以下の項目は、対象システムとは独立して、検証可能である。

① CPU、電源等のハードウェア
② OS／API 等の基本的なソフトウェア
③ コンパイラ等のツール群

ソフトウェアの独立性を考える時、以下の項目はソフトウェア相互作用に依存する。従って本稿では主に以下の項目についてソフトウェアの独立性を考察する。

④ 使用するメモリ／IO 等
⑤ 仕様書
⑥ プログラムモジュール

以上の概念を図式化すると、図 5 の様になる。

図 5 ソフトウェア監視構成の概念図

上記項目④～⑥の独立性を立証する為には、少なくとも以下の事項が立証される必要がある。

① メモリ・IO の競合が起こった場合、安全性構想（Safety Concept）に基づいてシステムを安全状態(Safety state)に移行し、この状態を維持する為の手段が完備している事。
② モジュールが暴走・無限ループに陥った場合の、監視・回復機構が完備している事。
③ 同一のサブモジュールを使用していない事。
④ 同一のグローバル変数あるいは IO を使用している場合、相互作用の影響によって各モジュールの一方或いは両方が、機能上の影響を受けない事。

上記ソフトウェアの領域と同時に、以下の事項は共通因子故障の排除の為に独立性の定義に含める必要がある可能性が有る。

① 開発人員・組織
② 仕様書の管理
③ 開発プロセス

④ 検証手段

ソフトウェア上の独立性が確立した後に、ソフトウェアでも前述の監視モジュール、被監視モジュールの故障検出率を証明しなければならない。

対象となるソフトウェアは特定ハードウェア及び OS（API 等も含む）上に構成された相互監視機能を持った、独立したモジュールである。当然ながら各モジュールはハードウェアからは切り離せないので、ハードウェア故障は独立した事象である。同様に OS の故障（残存バグへの遭遇確率）も独立した事象として扱う。これを FT 図で表現すると図 6 のように表す事が出来る。

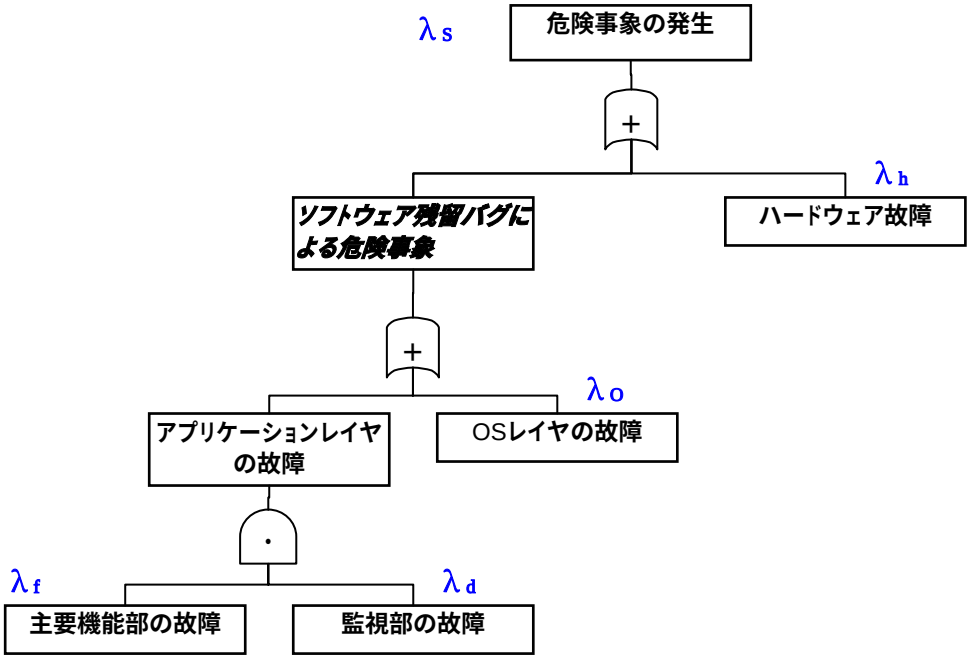


図 6 ソフトウェア残留バグを考慮した危険事象発生の FT 図

ハードウェア故障率を λ_h、OS 残存バグ確率を λ_o、主機能残存バグ確立を λ_f、監視ソフトウェア残存バグ確率を λ_d とした時、この系の故障確率 λ_s は、以下の様に表す事が出来る。

$$\lambda_s = \lambda_f + \lambda_o + \lambda_f * \lambda_d$$

ここで各付録 A に示した各 ASIL の目標故障率を当てはめて見る。すなわちハードウェア及び OS が ASIL-D で作成されていたと仮定すれば各々の故障率は 10^{-8} 以下が実現されている事から、

$$\lambda_h + \lambda_o < 10^{-7} \text{ 且、 } \lambda_h * \lambda_d < 10^{-7}$$

となり、系全体の故障確率は ASIL 要件 C 以上を満足できる事になる。以上の考察より、ソフトウェアのみによる故障検出においても、独立性と検出率を満足すれば、Decomposition による分割が可能である事が判る。またこの場合の独立性及び、検出率の関連性は、前述のシステム分割の時と同様である事は自明である。

3. システム分割の要件の提案

以上の考察により安全性に関連するシステムは、適切な独立性と検出率を実現する事により、Decomposition による分割が可能である事が判った。

WD26262-5 においては、分割されたサブシステム／コンポーネント内部のハードウェア検出率が

規定されている事から、Decomposition の場合においてもこの規定を考慮すると、ASIL の高位のシステムは下位のシステムよりも、検出率においてより厳しい規定が必要である。同様に CCF の残存率に対しても、ASIL 高位のシステムは下位のシステムよりもより厳しい規定があるため、ASIL 高位のシステムの独立性は、下位のシステムの独立性よりも厳しい規定が必要である。これを表 1 にまとめる。

表 1 検出率と独立性のマトリックス

独立性 / 検出率	None 同一ハード	Low 同一ハード(独立手法の確保)	Medium 同一システム･別ハード	High 別システム
None (< 60%)	規定せず	規定せず	ASIL A	ASIL B
Low (>=60%)	規定せず	ASIL A	ASIL B	ASIL C
Medium (>=90%)	ASIL A	ASIL B	ASIL C	ASIL D
High (>=99%)	ASIL B	ASIL C	ASIL D	ASIL D

この表は、例えば ASIL D を分割するためには、分割後のサブシステム／コンポーネントの独立性が少なくとも Medium 以上で、且システム間の故障検出率が High あるいは、独立性が High であれば、故障検出率が少なくとも Medium 以上でなければならない事を表している。

4. 結論及び考察

以上の考察により、ASIL Decomposition による分割は、ソフトウェアハードウェアに関わらず実行可能であり、分割の条件として、表 1 の様な、独立性と検出率の相関関係により、分割のガイドラインとする事を提案する。

Decomposition の条件に独立性と検出率の相関関係を示す事により、安全関連システムを無秩序に分割する事による、安全性の低下を防ぐ事ができる。

今後の課題として、システムレベルの独立性のランク付け、検出率のランク付けを検討して行きたい。

5. 終わりに

ISO WD26262 は本稿執筆時点で、'07 年末の CD(Committee Draft)の部分的リリースに向けて作業中であり、ソフトウェアの信頼性、監視機構による ASIL 分割は各国から、反論や訂正提案が出ている状況である。Decomposition の確率的な考え方に関しては、2006 年 10 月に本稿主旨とほぼ同様の提案がなされて、この主張の正当性を裏付けている。ソフトウェアに関しては英国 MISRA が中心となり、日本からは自技会のソフトウェア小委員会が主体となって、現在ドラフト作成中である。その様な状況で本テーマを選択した事は、議論を深める上でも大変有意義であった。Decomposition の条件に独立性と検出率をマトリックスとして加味する考え方は、第 6 回 ISO TC22/SC3/WG16 国際会議に日本案として提出し、ハードウェアアーキテクチャの初期設計段階に加える事を検討中である。

最後に本テーマ研究に助力いただいた、主査の飯泉様、副主査の吉澤様、田所様に謝意を表して、結びとしたい。

参考文献：　IEC61508 (JIS C 0508)

ISO WD26262-3　Concept (Working draft)（注 1）

ISO WD26262-4　System (Working draft)（注 1）

ISO WD26262-5　Hardware (Working draft)（注 1）

ISO WD26262-6　Software (Working draft)（注 1）

注 1 ISO WD26262 は 2006 年 12 月現在の版を用いており、現在規格作成中である。

MISRA Safety Analysis Guidelines (MISRA SA)Draft Ver.13J（注 2）

注 2 MISRA SA Ver.13J は 2006 年 6 月の版を用いている。

付録 A　機能安全性企画の安全性の考え方

安全性規格 IEC61508 の中では、機器の機能失陥により起き得る予知可能な危険事象の重大性と頻度の組み合わせをリスクと呼んでいる。システムの内包するリスクを準定量的に定義し、そのリスクの度合いを SIL（安全尺度： Safety Integrity Level)と呼ぶ。対象システムは SIL に対応する、リスク低減手法を定めている。

SIL は 4 段階に規定されており、SIL の算出手法は IEC61508 のパート 5 に詳しい。自動車の場合は IEC61508 で規定された SIL に、自動車特有の条件を追加しており、安全尺度には ASIL (Automotive Safety Integrity Level)を用いる。ASIL は SIL と同様４段階(A~D)に規定されているが、SIL４の相当する ASIL は無く、SIL1~3 に表 A.1 の様に対応している。ASIL の算出手法は現在議論中であり、2007 年中に確定する予定である。IEC61508 においても WD26262 においても、リスクの重大性に対して起き得る可能性を様々な手法で低減する事により、許容可能なリスク(tolerable risk)まで低減する事により、対象システムの安全性を確保しようとする思想である。この概念を図示すると図 A.1 の様になる。リスク低減後に残った残余のリスクは残留リスク(Residual Risk)と呼ばれ、このリスクは許容可能なリスク以下で無ければならない。蛇足ながら安全性の向上は必ずしも信頼性の向上には直結しない。例えば「危険が予知される時に頻繁に停止する制御装置」は安全性上、問題ないが、信頼性上は好ましくない事を考えれば明らかである。

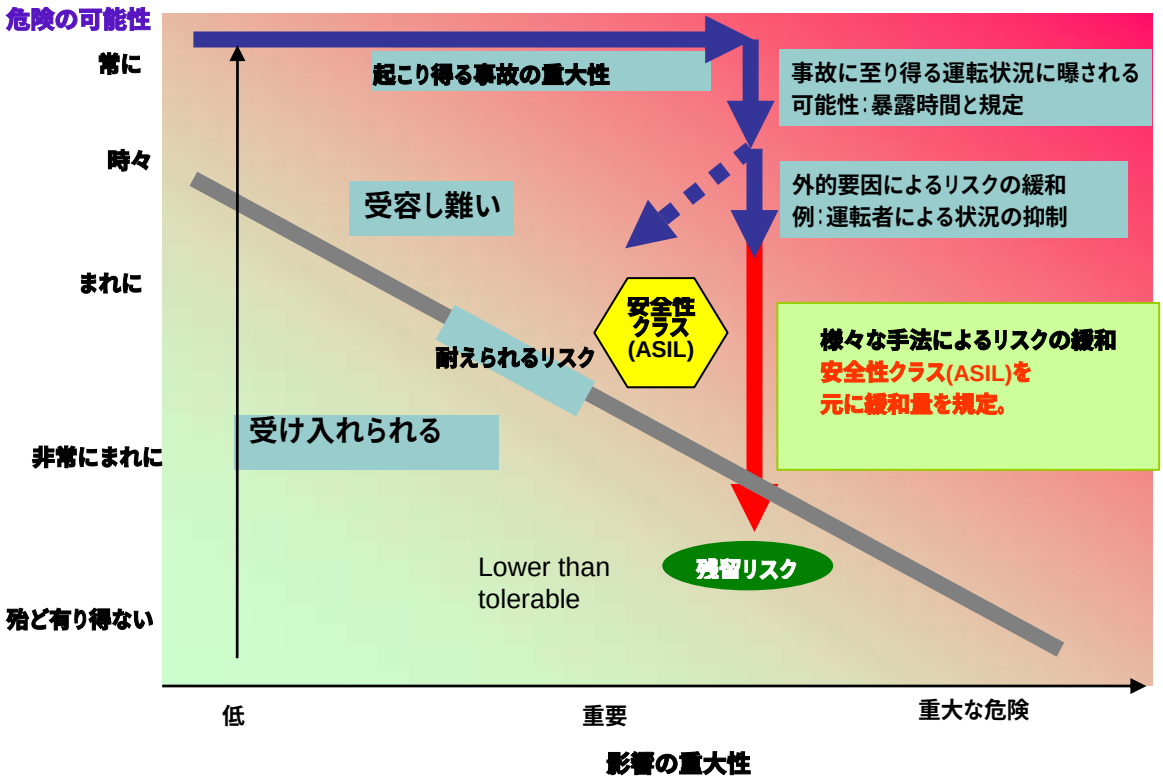


図 A.1. リスク低減の考え方

ASIL	対応する SIL	概要(詳細は各々規定された評価手法に従う)
A	1	安全性に影響するが、結果が極めて軽い。或いは発生する頻度が低い。
B	2 or 3	安全性に関連するが、結果が中程度。或いは発生する頻度が低い。
C	3 or 2	安全性に関連するが、結果が中程度。或いは頻度が低い。
D	3	安全性に関連し、重大な影響がある。或いは頻度はきわめて低いとは言えない。
該当なし	４	壊滅的な打撃。（大規模プラントの災害など）

表 A.1 ASIL の概要

システムを構築する上でリスクとして考え得るものは以下に分される

1. システムの内包するリスク
   (ア) 偶発故障(Random mode failure)
   (イ) 欠陥(Systematic failure)
2. 外部要因に起因する障害（Hazard）
   (ア) ユーザーの誤使用によるリスク(Usability)
   (イ) 外来ノイズ等の障害(Hazard)

ソフトウェアにおいては、演算機、メモリー、IO などの故障に起因する障害をハードウェアの偶発故障に分類するならば、全ての障害は欠陥(Systematic failure)に起因する。これをソフトウェア開発プロセスに照らして以下の様に分類できる。

1. 仕様の不備による障害
   (ア) 対象物の理解不足による障害
   (イ) 対象物の物理的特性が未知の場合のリスク
   (ウ) Hazard の分析不足による障害
   (エ) 要件定義の不備による障害
2. 実装の不備による障害
   (ア) 要件定義の不備による障害
   (イ) アーキテクチャの設計上の障害
   (ウ) コンパイラ等の開発環境による障害
   (エ) OS 等の実装環境による障害
   (オ) ハードウェアの理解不足
   (カ) プログラミング技法における障害
3. 管理の不備による障害
   (ア) 日程管理
   (イ) 変更管理

機能安全規格ではこれらのソフトウェアを含んだシステムの安全性を確保する為に、SIL/ASIL 毎に、システムのライフサイクル全般に渡って、種々の規定がある。

IEC61508 におけるライフサイクルの概念を図 A.2 に示す。

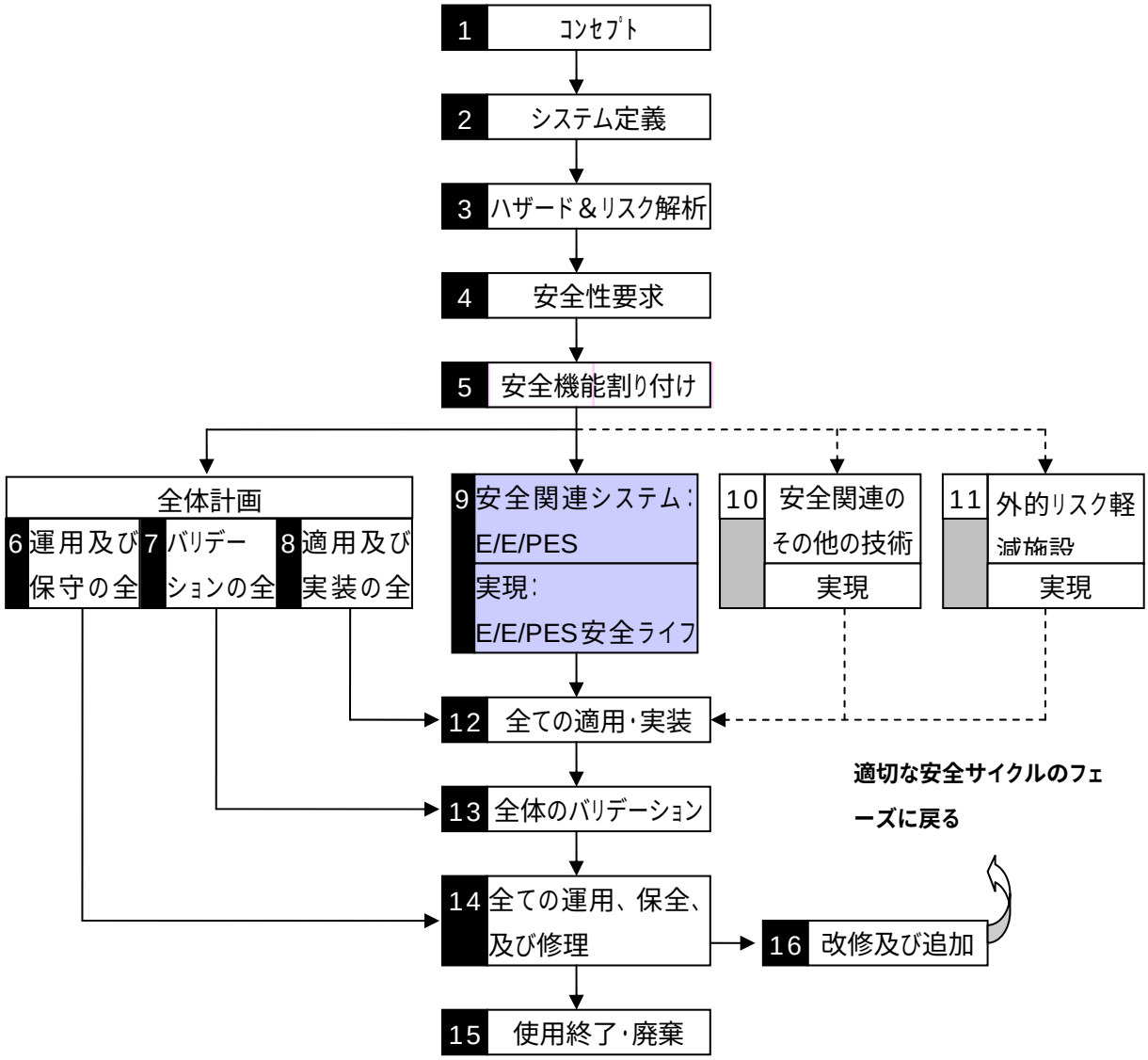


図 A.2 IEC61508 における安全性ライフサイクル

IEC61508 においても、ISO WD26262 においてもシステムの故障率は対応する SIL/ASIL 毎に決まっている。その例を表 A.3,A.4 に示す。

表A.2 IEC61508における高頻度作動システムの偶発故障発生率要件（１時間当たり）

Safety integrity level	**High demand or continuous mode of operation** (Probability of a dangerous failure per hour)
4	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$
NOTE See notes 3 to 9 below for details on interpreting this table.	

表 A.4 ISO WD26262 における偶発故障の発生率要件

ASIL Level	Random HW failure target values
D	$< 10^{-8}/h$
C	$< 10^{-7}/h$
B	No requirement($<10^{-6}$)
A	No requirement（$<10^{-5}$）

付録 B 付図・付表

付表 1 これまでに制定された **IEC61508** 関係の個別適用分野規格

EN 規格	**EN61508**	一般（計装）
	EM50126	鉄道
IEC 規格	**IEC6151**	プロセス産業
	IEC61513	原子力
	IEC62061	産業機械
	IEC62304	医療
	IEC61800	モータ
ISO 規格	**ISO/WD26262**	自動車
指針類	**EEMUA**	アラーム指針
	DFT STAN	防衛（英国）
	海軍規格(英国)	
	MISRA-SA	自動車（英国）
	計量ソフトウェア指針	モータ
	EMC ガイドライン	**IEE**（英国）

付表 2 IEC61508 における形式手法の推奨例

（IEC61508 Part1 Annex A Table A.1 — Software safety requirements specification)

Table A.1 — Software safety requirements specification (see 7.2)

Technique/Measure	**Ref**	**SIL1**	**SIL2**	**SIL3**	**SIL4**
1 Computer-aided specification tools	B.2.4	R	R	HR	HR
2a Semi-formal methods	Table B.7	R	R	HR	HR
2b Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	---	R	R	HR

a) The software safety requirements specification will always require a description of the problem in natural language and any necessary mathematical notation that reflects the application.

b) The table reflects additional requirements for specifying the software safety requirements clearly and precisely.

c) Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.

Table A.4 — Software design and development:

detailed design (see 7.4.5 and 7.4.6)

(This includes software system design, software module design and coding)

Technique/Measure	Ref	SIL1	SIL2	SIL3	SIL4
1a Structured methods including for example, JSD, MASCOT, SADT and Yourdon	C.2.1	HR	HR	HR	HR
1b Semi-formal methods	Table B.7	R	HR	HR	HR
1c Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	---	R	R	HR
2 Computer-aided design tools	B.3.5	R	R	HR	HR
3 Defensive programming	C.2.5	---	R	HR	HR
4 Modular approach	Table B.9	HR	HR	HR	HR
5 Design and coding standards	Table B.1	R	HR	HR	HR
6 Structured programming	C.2.7	HR	HR	HR	HR
7 Use of trusted/verified software modules and components (if available)	C.2.10 C.4.5	R	HR	HR	HR

Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.