

アリアン5の爆発事故と ソフトウェア安全性に関する国際規格

横浜国大 清水久二

1. 始めに

ESA (欧州宇宙機構) が 1996 年に打ち上げたロケット・アリアン5が、発射後まもなく空中で爆発、落下した事故は当時テレビで多くの人が目撃した。しかしその原因が飛行制御用のコンピュータ・ソフトウェアの欠陥であったことを関係者以外で知る人は少ない。それ以来、欧州ではESA加盟国を中心にソフトウェアに対する安全要求の国際規格作りが精力的に続けられている。特に英国はHSEや民間航空局を中心に非常に完成度の高い規格を既に作成し、一般に推奨している。筆者はISO/IEC TC65/SC65/WG9,10の国内対策委員会の主査としてこの問題に関わっており、特にIEC61508 制定後の規格改訂作業委員会(MT-12)に2度ほど出席しているので、現在の規格周辺の動向を紹介しておきたい。現在の改訂作業委員会の委員達の問題意識は、

- ・ 産業界で日々開発されるソフトウェアの量は指数関数的に増大している。
- ・ それらのソフトウェアは既存のソフト部品の組み合わせで構築されることが多く、過去の不確定な要素が蓄積される傾向にある。
- ・ これらのソフト部品は一般市場に広く出回る傾向にある。

ことであり、これが「潜在リスクの主要原因」になっていると指摘する。これらに対処すべく一部の第三者検査機関では安全機能用ソフトの認定業務を検討しており、未来技術におけるコンピュータが果たす役割の重要性に鑑み、このような動向が国際的な産業活動の中枢に影響を与える可能性もある。

2. アリアン5の爆発事故⁽¹⁾

2.1 事故の概略

1996年6月4日の朝、アフリカのESA宇宙センターの空は快晴であり、打ち上げに天候上の障害は無かった。所定の時刻に打ち上げられたアリアン5は、点火シーケンス後40秒にして高度3700mに達したが、その後突然軌道を外れ分解、爆発した。事故後直ちに加盟各国の専門家を網羅した原因調査委員会が組織され、事故原因の究明に当たった。事故の究明は明らかに判る以下の事実、即ち

- ・ H0(点火時)+36秒に至るまでの打ち上げ挙動は正常であった。
- ・ 正規の慣性標準装置(Inertial Reference System)が故障し、少し遅れてバックアップ用が故障した。
- ・ 二つの固体ブースタの振れ角(ジンバル角)と液体水素エンジンの振れは最大の位置にあった。

等の事実を出発点にして、事故の連鎖を詳細に追跡した分析の結果、慣性標準装置に指令を与えるコンピュータソフトに明らかなミスが認められた。

2.2 装置の概要と事故原因

アリアン 5 の飛行制御システムは標準設計のものであり、打ち上げ機(Launcher)の宇宙空間での姿勢とその運動は慣性標準装置(Inertial Reference System : IRS)によって計測されていた。IRS は内部にコンピュータを持ち、角度と速度とは"Strap-down"の慣性プラットフォーム上のレーザージャイロと加速時計からの情報を基に計算されていた。IRS からのデータはデータベースを経由して搭載コンピュータ(OBC)へ伝送され、飛行プログラムと固体ブースタと液体水素エンジンのノズルを制御していた。信頼性の向上のため装置レベルで適切な冗長性が考慮されていた。即ちハードとソフトにおいて同一の2つの IRS が並列に動作していた。一方は能動状態にあり、また他は常時稼働の待機状態にあって、故障時には OBC がその切り替えの任務に当たっていた。また OBC も2式あり、飛行制御システムの他のユニットも同様に多重化されていた。アリアン 5 の IRS 周辺に関する設計はアリアン 4 の設計を踏襲しており、とくにソフトウェアについては再利用の比率は大きかった。事故原因の精査の結果、以下の事故連鎖が明らかにされた。

- ・打ち上げ機は H0+ 39 秒において切り離しを開始したが、それは迎え角(angle of attack) が 20 度以上にも達して空力的負荷が増大し、ブースタの主ステージからの分離が始まっていたからである。
 - ・この迎え角は固体ブースタと主エンジン・ノズルの傾きが最大になったためによる。
 - ・またこの傾きは常時稼働の IRS-2 によって伝送されるデータを基にした OBC ソフトウェアの指令下にあったが、そのデータには適切な飛行データが含まれていなかった。
 - ・その理由はユニットがソフトウェアミスによる故障宣言状態にいたことに因る。
 - ・OBC が待機 IRS-1 への切り替えに失敗した理由は、72 ミリ秒の予備サイクルにおいてユニットは既に同じ理由で動作を休止していたことに因る。
 - ・内部 IRS ソフトウェアの異常は 64 ビットの浮動小数点を 16 ビットの符号付整数へ変換するルーチンで起きた。浮動小数の大きさは 16 ビットの符号付整数で表し得る数よりも大きく、これがオペランドエラーを起こした。その上 ADA 言語で書かれたデータ変換命令は適切な保護がなされていなかった。
 - ・他に保護の下にない変数が 3 つ見つかった。
- 結論として、これはソフトウェアにおける典型的なシステム・エラーであることが判った。その他の事故状況について過度に技術的なため、本稿では割愛する。

2.3 試験と認証手順、及び教訓

アリアン 5 の飛行制御システムに対する認定(qualification)手順は以下の標準の方式によっていた。即ち

- ・装置認定(Equipment qualification)

- ・ソフトウェア認定（搭載コンピュータのソフトウェア）
- ・ステージの統合
- ・システム確認試験（validation test）

適用された論理は、それ以前に達成されなかった問題を各段階毎にテストすることであり、この様にしてそれぞれのサブシステムと統合されたシステムの全てのテストを網羅していた。しかし打ち上げ直後の過渡期の IRS の挙動を確認する試験は省略された。その理由は IRS を飛行環境においてブラックボックスの状態で試験することが物理的には不可能であった。つまり加速度形の出力信号を取り入れた再現試験は地上では原理的に出来ないという点にある。

さらに IRS のシステム仕様には運転上の制約が記載されていなかった。

事故調査委員会が得たその他の膨大な知見は省略するが、直接的な原因はアリアン 4 で使用されたソフトウェアの有るプログラム要素には implicit な定義のものが含まれており、それがアリアン 5 では見逃され誤動作した、ということである。

3. IEC 61508 のソフトウェア安全要求⁽²⁾

本規格は全 7 部より成るが、全体として E / E / P E S（電気・電子・プログラム電子系）を安全機能実現に使用した時の安全要求を定めている。ここで PES とはプログラム可能なシステム（programmable electronic system）の略称であり、具体的には PLC のような電子システムを指す。この内第 3 部（Pt.3）がソフトウェア安全要求に関する章であり、また付録としての第 7 部にはソフトウェア検査に関わる各種の方式、概念、用語の解説がある。これに対応する翻訳 J I S が J I S 0 5 0 8 であるが、第 7 部はまだ翻訳されていない。

第 3 部のソフトウェア安全要求の枠組みは第 2 部のハードウェアに対する枠組みに準拠しており、その骨子は

- ・設計から使用停止に至るまでの安全ライフサイクルの全期間での管理。
- ・S I L（安全統合度）に対応したソフトウェア検査方式を推奨または義務化。
- ・文書化の義務付け。
- ・第 3 者による査察

等である。

ところでソフトの欠陥（エラー）はハードウェアの欠陥の様な時間的に偶発する性質を有さず、むしろ製作時に混入するシステムティック・エラーであることが多い。従って一度受け入れ検査に合格すれば安全性は保証されたと考える向きが多い。つまり使用実績によって安全要求が満足された（proven-in-use）との判断である。しかしアリアン 5 の様な事故では使用条件が微妙に変化し、ソフト・モジュールの implicit な意味の相違が顕在化して事故となる場合もあるので、絶対安全の保証として proven-in-use を認めるがどうかは重大な問題となる。

また近年 Visual Basic や SCADA のようなプログラムの部品化が急ピッチで進行しており，製作時の身元が判らないソフトが大量に使用される傾向にある．このような SOUP (Software of uncertain pedigree) の安全認証も重要な問題になりつつある．これについては後で述べる．

4．英国民間航空局のソフトウェア安全保証⁽³⁾ SW01

現在 IEC61508-3 は MT-12 委員会で改訂作業中であるが，作業部会は SW01 の考え方をかなり参考にしている．本規制は航空機に搭載されるコンピュータ，或いは航空管制センターで使用するコンピュータのソフトウェアの安全保証義務を規定している．その枠組みの中核は SIL 概念に良く似た AELs (Assurance Evidence Levels) が 3 段階に定められていることである．その危険性に対応した各段階において要求される責務は討論 (Argument) と立証 (Evidence) の 2 つである．

と言うのは，ソフトの安全性は，ハードの安全性が材質や構造等に依存するのと異なり，専らプログラムのアーキテクチャやコンフィグレーションに依存することに因る．従ってそれらの依存関係を口頭，或いは文書で合理的に説明できるものでなければならない．またそれらを何らかの方法で立証することが要求される．

1．要約

本研究プロジェクトの目標は「血統の定かでないソフトウェア (software of uncertain pedigree: SOUP)」を安全関連系へ応用する場合に派生する諸問題を検討することにある．ここで本報告のアプローチの基礎は SOUP の安全保証 (safety assurance) であり，これは IEC61508 の文脈，即ち 5 つの段階～予備，アーキテクチャ化，具体化，設置，及び運転のライフサイクル段階に於いて安全の正当化 (justification) を求めることにある．

これらの中で，安全の正当化が SOUP の安全要件として最も重要であり，これに関連する活動は IEC61508, pt.1 の 7.6 が規定する項目に対応する．この段階での設計上の選択は SOUP を含むシステムの安全保証に重大な影響を与える．その選択を行うに当たり安全立証 (safety evidence) を得るコストや，入手可能な安全立証や論証 (argument) の適切性，この立証をシステムの生涯時間にわたり有効とするコストをも勘案しなければならない．

SOUP に対する安全立証はブラックボックス法や，ホワイトボックス立証の方法もあるが，多くの事例では，SOUP 要素をブラックボックスとして見れば適切な立証を得ることが出来，本報告はその必要にして十分な要件を示している．

2．背景

2.1 SOUP の特性

- ・既に存在しているもの．
- ・ユーザ側において re-engineer 出来ないもの．

- ・ 普遍性があり，システムへの応用において不必要な関数を含むことが多いもの．
- ・ 継続的な修正を受けることが多いもの．これは量販志向の SOUP 市場においては消費者の需要を満たし，競争に生き残るためである．

一方，SOUP は欠陥 (faults) が有るにも係わらず，その市場性によりそこで実証されたと受け取られる事が多い．SOUP が用いられてトラブルが生じた典型例を以下に示す．

- ・ マイクロソフト社 Window 95 & 98 の時間アルゴリズムの中にバグがあり，50 日経過後にコンピュータを停止させたことがあった．恐らく純粋なブラックボックス手法，或いは統計学的テストではこの問題点は発見されなかった．また発現頻度の高いバグの陰に隠れてしまうことを有り得た．

- ・ SOUP の難しい問題の一つは，特殊な用途では使用されない追補的な機能の影響を事前に予測出来ないことである．通常では上記の機能は記録文書から同定できる性質のものであるが，完全には文書化出来ない様なソフトウェア製品もある．これらの諸事実は，SOUP の扱いをいわゆる「特注で製作されたソフトウェア (bespoke software)」よりも難しくする理由であり，さらに

- ・ 開発プロセスを記述した文書
- ・ 設計書
- ・ ソースコード
- ・ 不具合 (欠陥) の時間履歴

等の情報へのアクセスが難しい事にもよる．製造業者は品質管理規格 ISO 9001 に準拠した開発プロセスの記録文書は公開するかもしれないが，これでは不十分である．

4 . 安全正当化のアプローチ

本節では以下の5つのアプローチについて具体的に記述する．その内容は割愛する．

- ・ 予備的段階での安全正当化
- ・ アーキテクチャ上の安全正当化
- ・ 具体化 (implementation) における安全正当化
- ・ 設置時の安全正当化
- ・ 運転時の安全正当化

であり，それぞれの安全ライフサイクルの中での位置付けを図 10.1 に示す．

5 . 安全正当化の構造

5.1 保証のための設計

5.1.1 PES アーキテクチャ

5.1.2 ソフトウェア故障解析

5.1.3 要素故障への対策

ほかである．

5.2 安全正当化の立証

ここでは SOUP に適用され得る立証（証拠）の一般的な形式について考察する．
アーキテクチャ上の安全正当化には，機能的及び非機能的要求への割り付けが指定された安全性能に適合しているという正当化（justification）プロセスが含まれる．
第 2 図に示される様に安全正当化は以下の項目で構成される．

- ・ PES の安全要求に合致しているという主張
- ・ この主張を支持する論旨（argument）
- ・ 論旨で使用される立証（証拠）

であり，この立証の形式としては

- ・ 試験により立証
- ・ 解析的立証（製品及びインプレメンテーション過程について）
- ・ フィールド経験（製品が既往の使用実績を有する場合）

この立証はシステム・インプレメンテーションにより生成されることもあり，また SOUP の供給業者や有資格の第 3 者により提供されることもある．表 3 はそれぞれのカテゴリーにおける安全属性に対して得られた立証形式を要約したものである．同時にここではブラックボックスかホワイトボックス精査かが識別されている．

参考文献

1. <http://www.esrin.esa.it/htdocs/tidc/Press/Press96/ariane5rep.html>
2. 福田・清水「機械安全工学」、養賢堂、2000-2
- 3 Regulatory Objective for Software Safety Assurance in Air Traffic Service Equipment SW01, Excerpt from CAP 670-Civil Aviation Authority

雑誌「安全工学」2002年1月号掲載

図表は本誌を参照の事