応用 一般

宇宙で動くソフトウェアのつくりかた —宇宙環境での信頼性の確保—

吉田　実（三菱電機（株））

宇宙で動くソフトウェアとはどのようなものか

◆ 宇宙で動くソフトウェアを取り巻く現状

宇宙空間で動くソフトウェアとはどのようなものであろうか．2001 年宇宙の旅の HAL のような計算機は無理としても，最先端の計算機が搭載されて最先端のソフトウェアが動いている印象を持っている方が多いかもしれない．しかし，現実には宇宙環境は過酷であり，最高性能の半導体等の電子部品が使えないため，限られた CPU 性能やメモリ量で必要な機能，信頼性を満たすシステムを作りあげることになる．宇宙以外の他の分野でもコスト，消費電力，重量などの制約により性能が制限されるが，宇宙ではその制限が特に厳しい．筆者は，宇宙で動くソフトウェアで最も特徴的なものは，衛星バスと呼ばれる人工衛星全体を制御する部分で動くソフトウェアと考えている．そこでは，ある程度の自律性と高い信頼性が要求される．以下，衛星バス用のソフトウェアを中心に宇宙で動くソフトウェアを説明する．

◆ 宇宙で動くソフトウェアの特徴

筆者は多分野の組込みシステムにかかわってきたが，その視点で，宇宙で動くソフトウェアの特徴は以下の通りである．

まず，最初に挙げられるのが，物理的に孤立したシステムであることである．いったん宇宙空間に出てしまえば，地上と通信はできるが，あとはあらかじめ用意されているものを使うしかない．装置類は壊れたら修理はできないし，燃料は一度使い切れば補充されることはない．また，人間による現地調整もできなければ，ハングアップしてもリセットすることもできない．平たくいえば，だめになってしまえば，もうそれまでということである．ハッブル宇宙望遠鏡で修理が行われたことがあるが，人工衛星としては例外的である．宇宙では，電力だけが太陽電池があれば継続的に生産することができる資源といえる．

次の特徴は，信頼性を重視する比較的小規模なソフトウェアであることである．人工衛星は，その打ち上げコストも含めて高額であるが，ソフトウェアの 1 つの不具合ですべてが失われてしまう可能性がある．実際，欧州のロケットアリアン 5 はソフトウェアの不具合によって爆発したが，これは史上最も高くついたソフトウェア不具合とされる．ただし，筆者の知る限り宇宙開発の歴史で，純粋にソフトウェア製造上の問題で大きな損失が出たと断定できるのはこれだけである．むやみに，ソフトウェア製造上の信頼性を上げても恩恵は少なく，システム設計レベルで信頼性を確保することが重要である．宇宙のソフトウェア開発では，信頼性を高める技術の導入に積極的である．

第 3 の特徴は，宇宙環境の独自性である．これについては次の節で説明する．

第 4 の特徴は，自律性が必要であることである．人工衛星は多かれ少なかれ地上との通信が途絶する可能性がある．そのときに故障が起きても最低限自らを維持する必要がある．また，通常であればその人工衛星のミッションを継続することが要求される．人工衛星には静止衛星と周回衛星がある．静止衛星はいったん静止してしまえば地上から見て同じ位置にいる（衛星放送の地上受信用アンテナが固定されてことからも分かる）ため，地上とは常に通信できる状態にある．一方で，周回衛星は地表から 400 ～ 1,000km 程度とかなり低い高度で地球を周回し，ある地上局と通信でき

るのは 1 回に 5 ～ 10 分程度である．地上と通信できない期間は，衛星自らが与えられた仕事を自律的に行う必要がある．また，その間，故障が発生しても最低でも自己保持できる必要がある．なお，惑星探査などの深宇宙探査機では，往復の通信時間が長くなるため，別の意味で自律性が必要になる．

◆ 計算機・ソフトウェアから見た宇宙環境

宇宙環境は，(1) 放射線，(2) 真空，(3) 無重力，(4) 温度，(5) 打ち上げ時の強い振動の点で，地上環境と異なる．計算機のハードウェアはこれら宇宙環境に耐えるように作らねばならない．(1) の放射線は遮蔽することが困難で，CPU，メモリなどの半導体素子内部のあらゆるビットが反転するのを避けられない．宇宙環境のうち，ソフトウェア製造で直接的影響を受けるのが放射線である．放射線の影響でビット反転が起きる確率は，人工衛星の地上高度，運用時期の太陽活動などによって異なるが，耐放射線性の高い宇宙用半導体素子を使ってもなお無視できない確率で発生する．また，放射線は一時的な影響以外に徐々に半導体を劣化させる．これは 10 年間の寿命を持たせるような場合，特に問題になる．(2) の真空に関しては，冷却に空気が使えないわけで，計算機システムとしての発熱を抑制する必要がある．(4)，(5) もソフトウェアとしては直接的な影響はないが，ハードウェアとしては対策が必要である．よって，宇宙環境に耐えるためには，放射線に強いが集積度が低く，発熱の抑制のため動作周波数が低い宇宙用半導体を使い，振動に耐えるための強固なケースにしっかりと基板や電子部品を固定する必要がある．その結果，宇宙用計算機は性能が低く，容積や質量が大きくなる．ところで，(3) の無重力は一見ソフトウェアとは関係なさそうに思えるかもしれない．しかし，真空，熱，振動は地上で物理的にその環境を作って試験をすることができる．一方，地上で無重力状態を長時間つくることはできず，モデル等を使って試験をすることになる．

◆ 衛星バス用計算機・ソフトウェアの仕事

人工衛星は，いろいろな仕事（ミッション）を持っている．気象衛星であれば気象観測用のカメラとそのデータ処理装置を持ち，放送・通信衛星であれば放送・通信用の送受信機を持ち，天文観測衛星なら望遠鏡を持つ．ミッションが異なればこれらミッション用の機器に共通性はほとんどない．一方で，現代のほとんどの人工衛星は，機体を特定の方向に維持し，地上に対して衛星の状態を知らせ，衛星運用に関する基本的な指令を受ける．また，二次電池の充電状況，発電，電力消費状況をモニタ，管理し，筐体内部の温度の制御を行う．これらは衛星バス部と呼ばれる部分の仕事であり，衛星バス用計算機とソフトウェアで制御を行う（**図 -1**）．なお，衛星バスとは，構体を持ち，通信，電源，姿勢制御，推進，熱制御を行うサブシステムのことであり，USB や PCI のような計算機のバスのことではない．衛星バス用ソフトウェアは特別な信頼性が要求される．センサ故障等何らかの理由で姿勢が維持できなくなるとミッションを継続できなくなるが，もし，そのときに太陽電池を太陽の方向に向けることができず太陽電池の発電が一定期間できなくなると衛星システムが永久に復活できなくなる可能性がある．衛星バスの制御ソフトウェアはそのような場合でも，可能な限り正常なセンサやアクチュエータを使って自己保持するように設計されている．

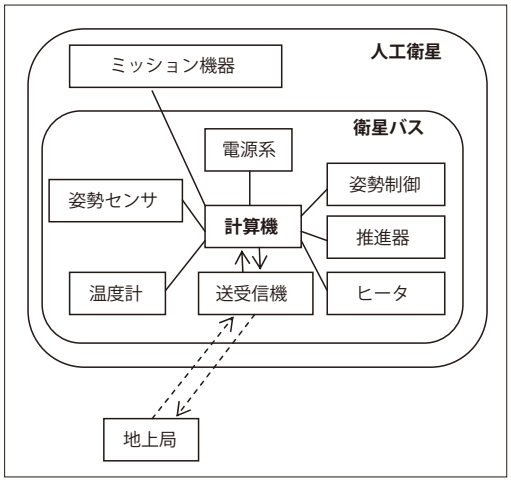

図 -1　衛星バスの構成例

人工衛星のシステム設計での信頼性に対する考え方

すでに述べたように宇宙では機器が故障しても修理できない．信頼性を確保するためには，個別の機器の

信頼性を高めるとともに，重要な機器は二重化し，故障時に切り替えられるようにする．このような冗長構成をとる場合，ある1カ所が故障すると全体の障害につながる単一故障点がないかを徹底的に検討する．たとえば，機器を二重化していても切り替え装置が機器と制御装置との通信を妨げるような故障を起こせば，その1カ所の故障が全体の障害になってしまう．このような単一故障点がないように設計する．そうすれば，たとえば，ある機器の単位時間あたりの故障する確率が 10^{-9} だとして，2個同時に故障する確率は 10^{-18} なので，大幅に信頼性を高めることができる．

ソフトウェアはどうであろうか．計算機という1つの機器と一体で考えれば，故障する確率という枠組みで扱える．すでに述べたとおり放射線の影響は，遮蔽やハードウェアで完全に取り除くことはできない．制御対象の機器がエラーになるだけであればまだよいが，計算機自体にもエラーが起きる．宇宙では，計算機にエラーが起きやすいにもかかわらず，高い信頼性が求められる．よって，計算機の中で起きるエラーを内部で処理し，なるべく故障を起こさないことが重要になる．宇宙で動くソフトウェアで特徴的な点の1つは，ハードウェアのエラーを想定し，ハードウェアとソフトウェアで協調してエラーを取り除くことである．

信頼性を向上させるために，ソフトウェア実装観点でこのようなことも考えながら作っている例として，メモリの構成方法と決定的動作について説明する．

♦ メモリの構成方法について

現代の計算機は，遅いが容量の大きい主記憶のメモリを直接読み書きするのでなく，その中間にキャッシュと呼ばれる比較的速く容量の小さいメモリを置くのが一般的である（**図-2**）．キャッシュには主記憶の一部のコピーが置かれる．CPUはキャッシュのデータをまず読み書きする．これにより，比較的速く，かつ，大容量のメモリを持っているかのような効果が得られる．キャッシュの効果は顕著で宇宙でも用いられる．ところが，メモリは放射線によりビット反転が起きやすい個所である．メモリにビット反転が起きてもエラー訂正できるメモリの構成方法について代表的な一例を用いて説明する．

まず，主記憶に1bitのエラー訂正を行えるエラー訂正符号（ECC）を使う．また，キャッシュに1bitエラーを検出できるが訂正できないパリティを使う．このような構成にするのは，ECCはパリティより速度が遅く，速度の要求されるキャッシュに使うと性能低下の原因になるからである．

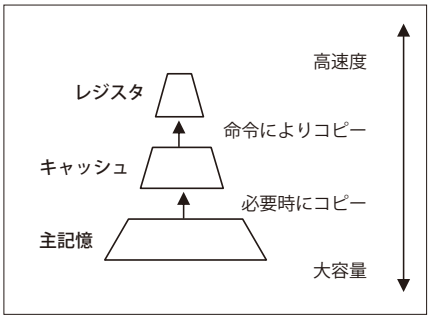

図-2　メモリ構成例

キャッシュには，書き込み時に，(1) 必要になるまで主記憶に書き込みを行わないライトバックと (2) 常に主記憶に書き込みを行うライトスルーの2つの方式があるが，ライトスルーを使う．これによって，キャッシュにあるデータは必ず主記憶にもあることになる．書き込み後，キャッシュに1bitの反転が起きた場合，パリティエラーが検出される．そのときは，単にエラーになったキャッシュのデータを破棄すればよい．すると，キャッシュの仕組みにより，主記憶からキャッシュに正しいデータがコピーされる．

このような方法をとることで1bitエラーに耐えるメモリシステムとすることができる．

なお，主記憶のうち，プログラムや定数データが置かれる個所はまったく更新されないし，通常のデータも更新頻度が低いものがある．主記憶が1bit訂正できるECCだった場合，エラーが蓄積し，2bitエラーになると訂正することができなくなる．この対策としては，定期的にデータを読んで書き戻すことを行う．これにより，1bitエラーは取り除かれ，エラーのない状態に戻る．

♦ ソフトウェアの決定的動作

宇宙開発は失敗の歴史でもある．その際重要なことは一度起こした失敗を繰り返さないことである．すでに現地で調整できないことは説明したが，さらにいえば調査もできない．できることは，宇宙から送られてくる情報から起きていることを推定することだけである．どこがおかしいかを地上で検討し，再現する必要がある．

この場合，ソフトウェアは，最初から最後まで決まったとおりに実行して終わる決定的動作が好まれる．宇宙で起こったことを地上で再現させやすいからである．逆に，動作が非決定的になる並行処理や割り込みは，主にCPUの使用効率を上げることが目的である．よって，決定的動作とCPUの使用効率のどちらをとるかというトレードオフになる．基本的には，CPUの使用効率の許す限り決定的動作をするように設計する．しかし，マイコンとして見た場合，通信に割り込みやDMA（直接データ転送）を使わないことは大きな性能劣化やCPU負荷をもたらし，システムとして成り立たないことがある．このような場合，通信は別のハードウェアで行うか，もしくは，可能な限り非決定的動作の影響が少なくなるよう考えた上で割り込みやDMAを利用する．

◆ 信頼性について他分野との比較

信頼性に関して，宇宙システムは洗練された検討がなされており，ソフトウェアもその一翼を担ってきた．一方で，近年社会の工業製品に対する安全の要求が高まっているとともに，ソフトウェアを含むことが一般的になったため，機能安全が注目を浴びている．特に，自動車産業は業界標準の安全規格ISO 26262を作っている．安全性は重要であるが，一方で，高性能，高機能，短納期，安価というトレードオフの関係にある要求もあり，どこかで折り合いをつけなければならない．ISO 26262（とそのもとになったIEC 61508）では，発生確率と起きたときの影響から必要な安全性を分類している．安全性と信頼性は異なるのであるが，類似の議論ができるものも多い．宇宙と自動車の信頼性に関して，1つのエピソードを紹介したい．それはある自動車の車種でNASAの専門家が「意図しない加速」が起きないとした報告[1]である．報告では，最初に，システム構成を説明し，意図しない加速を定義する．そして，苦情と履歴の調査，設計の理解，実装の理解，故障モードとシステムの脆弱さの識別をし，最後に確認試験を行っている．二重化されたスロットルセンサに同時にウィスカがついて同じ値を示すことはあり得ないという分析などに宇宙における信頼性確保の考え方を感じる．NASAの技術者が意図しない加速が起きないと結論付けた手法はさすがであるが，自動車メーカの安全性に対する努力もうかがい知られ，読みごたえがある．専門家でなくても読めるように分かりやすく書かれているので，興味のある方はぜひ読んでいただきたい．

なお，現実の宇宙システムは，つまらない失敗も相当数している．アリアン5の事例も単なる数値演算のオーバフローによるものである．宇宙での失敗は，まとまったものとして文献2），また，各宇宙機関からの情報がWeb上にあるのでこちらも興味のある方は読んでいただきたい．

◆ 宇宙で動くソフトウェアに求められる信頼性

宇宙空間で動くソフトウェアは，計算機内に比較的高い頻度でビットエラーが起きる環境中で，現地保守もできず，不適切な動作は人工衛星自体をだめにしてしまうという状況を考慮した上で高い信頼性を確保する必要がある．このことが宇宙で動くソフトウェアの最大の特徴である．

また，万一故障が起きたときは，それが復帰可能なものか否かにかかわらず，地上で原因究明できる必要がある．失敗に学べないといつまでも信頼性は向上できない．原因を究明する手がかりは，ほとんどの場合，衛星から電波で送られてくるテレメトリと呼ばれる比較的低速な通信データだけである．このテレメトリに想定内外のトラブルが起きたときに原因が解析できる情報を載せるとともに，トラブルが起きたときでもテレメトリが地上に届くよう可能な限り設計する．

近年の半導体の微細化や安全性要求の高まりにより，宇宙環境でしか問題にならなかったことが今後は地上でも問題になることが多くなると考えられる．そのような場合に本稿をきっかけにして宇宙での知見を参考にしていただければ幸いである．

参考文献
1） http://www.nasa.gov/topics/nasalife/features/nesc-toyota-study.html
2） Harland, D. M. and Lorenz, R. : Space Systems Failures : Disasters and Rescues of Satellites, Rocket and Space Probes, Springer (2005).

（2015年5月15日受付）

吉田　実（正会員）■ Yoshida.Minoru@ea.MitsubishiElectric.co.jp

1993年東京大学工学系研究科博士課程修了，博士（工学）．1993年三菱電機（株）中央研究所入社．以来，人工衛星，列車制御，カーナビゲーションシステム，エレベータ等のソフトウェア開発にかかわる．