

超小型人工衛星におけるシステム検証の可視化

Visualization of System Verification and Validation in Nano/Micro Satellite

宮崎・山崎研究室

Miyazaki-Yamazaki Laboratory

福島のだみ

Nozomi Fukushima

Recently, there is a growing interest in Nano/Micro Satellite system over the past decade because of their short development time and reduced cost. However, Nano/Micro satellites of over 60% have not been able to achieve the all mission due to failure on orbit. And it is reported that about half of the cause of failure was able to take measures on the ground test. That is, it is thought that a failure was overlooked in the ground test. In this paper, in order to solve the problems and perform efficient verification, we visualize the verification by expressing the system using the matrix. By doing this, we clarify the verification item, the timing and the procedure of the verification, and support the verification without omission. And applying the process to Prelude and discussed. the results show that it is an effective tool for visualizing the verification.

1. 序論

1.1. 超小型人工衛星の現状

超小型人工衛星は民生品の利用により開発期間やコストを抑えることが可能なため、リモートセンシングや災害監視、GPS、通信など様々な分野での利用拡大が進んでいる。SpaceWorks 社の試算では、複数の衛星を連携し運用するコンステレーションの拡大などにより、2019～2024 年の 5 年間で最大 2800 機の打上げが予想されている^[1]。しかしその一方で、約 60%以上の超小型人工衛星は不具合の発生により、全てのミッションを達成することが出来ていない (Fig. 1)。

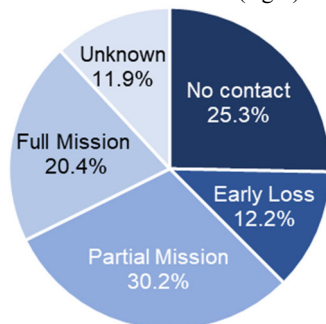


Fig. 1 CubeSat Mission Status (2000-2015) ^[2]

超小型人工衛星の軌道上における不具合についての調査結果によると、不具合原因の約半数を占める過充電やソフトウェアバグ、誤作動などは、地上で対策を行うことが出来たとされている^[3]。つまり、不具合の見逃しによる故障が、超小型人工衛星のミッション達成を阻害する大きな要因になっていると考えられ、地上試験で漏れなく確実に検証を行うことがより一層重要である。

地上試験で不具合を発見できなかった要因としては、ミッションの高度化によるシステムの複雑化が挙げられる。これに伴い、検証項目は多数かつ、サブシステム間を跨ぐような項目も存在してくるため、項目の見落としや、優先度を認識することが出来ず、重要な検証に十分な時間をかけることができなかったことなどが考えられる。また、開発段階における検証において、いつ・どのような順序で検証を行うかといった、検証手順を明示化するプロセスが存在しないことも不具合が見逃された一因として挙げられる。超小型人工衛星の検証は、衛星によ

りミッション内容やシステム構成が異なるため、標準的な枠組みが存在しない。そのため、設計者の頭の中で暗黙的に検証プロセスが構築されており、どのタイミングで、どのように検証を行うべきかが不明確なまま開発が進み、不具合が作りこまれてしまうといったことも考えられる。

そこで、「ほどよし信頼性工学」^[4]においては、暗黙的に行われていた判断や検証内容を可視化する試みとして、アシュアランスケースの考え方に基づいて開発された D-case が提案されている。この手法は、システムが正しく動作することを、ノードと呼ばれるブロックを用いて構造化及び可視化することにより体系立てて保証する手法である。そのため、システムが正常に動作するために必要な検証項目を抽出し可視化することは可能であるが、ツリー状の図によって可視化を行うため、システム全体に対して適用をおこなうと、コンポーネントなどの詳細な検証項目に分解するまでに膨大なノード数になり、分かりづらくなってしまふ。また、この手法はシステム開発者と利用者間での妥当性検証や、システムの品質保証を行う点に主眼が置かれているため、各検証を行う順序や、どの開発段階のタイミングで検証を実施するかといったことを表現することはできない。

そのため、通常は暗黙的に行われている開発段階の検証について、検証項目を洗い出し、タイミングや順序、また、検証項目同士の相互関係を可視化することと、洗い出しから可視化までの一連のプロセスを明示化することが求められる。また、システム検証の項目や箇所、手順を明示化することが出来れば、検証時に不具合が発生した場合においても、手順を遡っていくことにより原因の追及や、不具合によって影響を受ける機能を明らかにすることができるため、開発の効率化に役立てることができると思う。

1.2. 研究目的

前節を踏まえ本研究では、超小型人工衛星の検証を抜け漏れなく、確実に実行するため、機能とシステムをそれぞれ階層化し、分解することに加え、マトリクスを用いて関係を可視化することによって、検証項目や箇所、検証を行うタイミング・順序、影響範囲を明らかにする方法を提案する。また、現在開発中の地震先行現象検知衛星「Prelude」に提案する手法を適用し、議論することにより、有効性を示す。

2. システム検証の可視化

2.1. システム検証

超小型人工衛星の検証では、システムを統合していく過程でコンポーネントの動作確認やサブシステムやシステム全体の機能確認を行ったのち、振動や真空などの運用環境を模擬した状態での機能確認を行う。本研究では、システムを統合していく過程における検証漏れの防止を図るため、システム統合時のコンポーネントの動作確認から衛星システム全体の機能確認を対象とし、「システム検証」と呼ぶこととする。INCOSEの定義によると、システム検証とは「対象とするシステムの全ての要素が意図された機能をはたし、割り当てられた性能要求を果たしていることを確認する」とされている^[9]。そこで、本研究はこのINCOSEの考え方にに基づき、衛星システムと機能を段階的に詳細化していくことによって検証項目及び箇所を明らかにしていく。

2.2. 可視化プロセス

提案するプロセスは、以下の4つの手順で構成される(Fig. 2)。本節では概略手順および目的を示し、次節以降に詳細を示す。

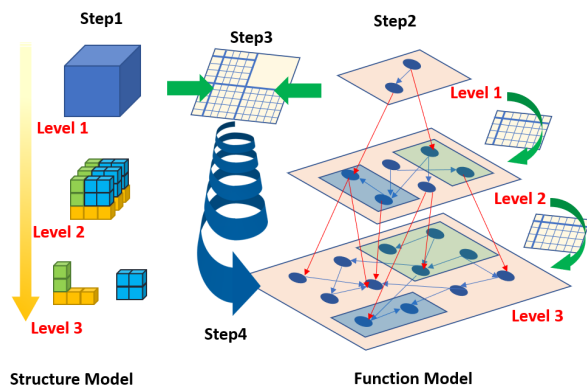


Fig. 2 Model of visualization process

【Step1:モデルの階層化】衛星システムを構成する要素の階層を定義し、システムを分解することにより、検証を行うタイミングを決定する。

【Step2:運用モード・機能の分解】衛星システムに要求されている運用モードや機能を Step 1 の階層をもとに分解することにより、各開発段階で検証を行う機能を洗い出す。

【Step3:マトリクスによる関係の可視化】Step 1 及び Step 2 で洗い出したシステムと機能の関係をマトリクスにより可視化することによって、検証を行う箇所及び手順、また、検証項目同士の依存関係を明らかにする。

【Step4:各階層でのマトリクスの作成】Step 3 のマトリクスを各階層で作成する。

2.3. Step1:モデルの階層化

ここではシステムを段階的に検証していくために、システムの階層を定義し、分解していくことにより、検証を行うタイミングを決定する。単純な製品であれば、その製品に対する機能も少ないため比較的簡単に機能に対応するシステムやコンポーネントを把握することができる。しかし、超小型人工衛星は複数のコンポーネント等の要素が相互に関わることで、衛星として求められる機能を満たしている。そのため、Table 1 に示すように対象とするシステムの階層(以下、システムレベル)

を定義することにより、各開発段階でやるべき検証を明確にし、検証の抜け・漏れの防止を図る。

Table 1 Structure of Satellite Model

System Level	Definition	Example of satellite
Level 1	対象システム	Satellite
Level 2	主要構成要素	Subsystem
Level 3	コンポーネント	MPU, Sensor

2.4. Step2:運用モード・機能の分解

ここでは、システムに要求される機能の洗い出しを Step 1 の階層をもとに行う。また、機能の表現については「写真を撮影して、SD に保存する機能」などと表現するのではなく、「写真を撮影する機能」や「SD へ保存する機能」などシステムやコンポーネントの単位の機能として定義することにより、どのシステムがその機能を担っているかを明確にする。

Level 1 では、Step 1 で検証の対象を衛星システムと定義したことから、開発後期で行う End-to-end 試験^[6]を想定し、まず運用モードの洗い出しを行う。また、加えて衛星を運用するにあたって最低限必要となる機能をシステム要求やロケット側からの要求より洗い出しを行う。Level 2 においては、level 1 で定義した運用モード中のシーケンスを考え、サブシステムレベルでの機能を洗い出す。さらに、level 3 においては、さらに詳細なコンポーネントレベルでの機能の洗い出しを行う。また、Fig. 3(a)の Function 2 のように機能は様々なモードで繰り返し使われることが想定されるため、Fig. 3(b)に示すようなマトリクスを用いて整理をしながら、機能展開を行う。これにより、検証のタイミングが異なる機能同士の関係性を明らかにする。

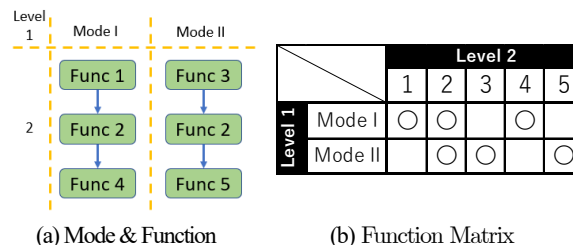


Fig. 3 Function Expansion Matrix

2.5. Step3:マトリクスによる関係の可視化

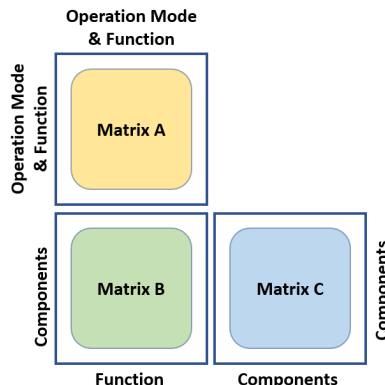
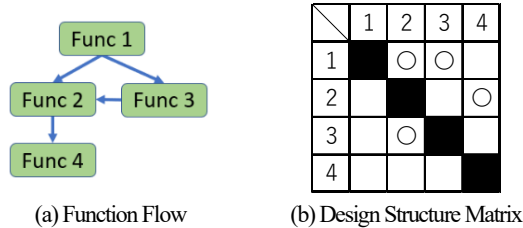


Fig. 4 Multidomain Matrix Model

ここでは、モデルと機能の関係を Fig. 4 に示すようなマルチドメインマトリクス^[7](Multidomain Matrix: 以下 MDM)により

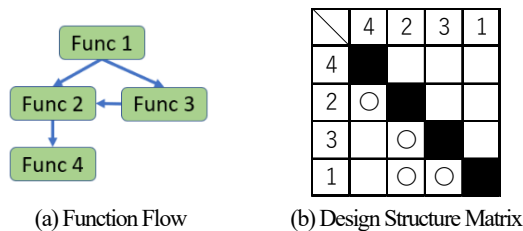
可視化する。MDM とは 2 つ以上の要素からなるマトリクスのことであり、今回は機能同士、機能とコンポーネント、コンポーネント同士の関係を表す 3 つのマトリクスを用いて表現する。これにより、検証時に必要となる検証箇所と検証項目を一つのマトリクスで表現することが可能となる。また、マトリクス上でシステムを構成する要素とそれらの相互作用を表現することができることから、機能やコンポーネントなどの要素を並び替えることによって検証の順序を明らかにすることや、他の機能への影響も見て取ることが可能となる。各マトリクスの詳細な構成及び見方は以下で示す。

■ Matrix A/ 機能 DSM (Design Structure Matrix)



(a) Function Flow (b) Design Structure Matrix
Fig. 5 Function DSM (Before)

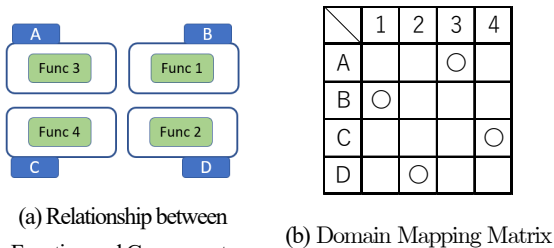
Matrix A は機能間の入出力関係を示す。Fig. 5(a)は機能の流れを表し、DSM で表現すると Fig. 5(b)のようになる。要素の行に注目すると出力先、列に注目すると入力元を知ることができる。そのため、Fig. 5 において Function 1 が失われたとき、行に注目すると出力先である Function 2, 3 へ影響が及ぶことが分かる。また、同様にして Function 2・3 の出力先をたどることによって故障の影響を受ける機能を明らかにすることができる。



(a) Function Flow (b) Design Structure Matrix
Fig. 6 Function DSM (After)

また、Fig. 6(b)に示すように機能の並び替えを行うことによって、機能の流れを示すことができる。そのため、列に注目すると Function 1 のように入力元が存在しない要素、すなわち機能を構成する大元となる要素を特定することが可能となる。これによって、根本となる機能から順番に検証を行うことができるため、検証手順の最適化を図ることができる。また、不具合発生時には機能を遡って検証していくことで、原因を特定する指標とすることができる。

■ Matrix B/ 機能・コンポーネント DMM

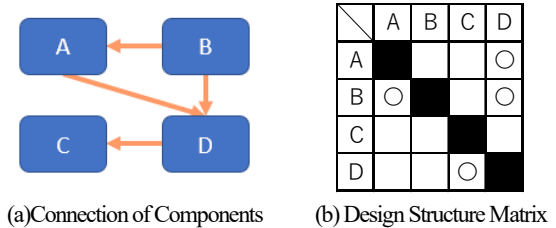


(a) Relationship between Function and Components (b) Domain Mapping Matrix

Fig. 7 Function / Component DMM

Matrix B は行にコンポーネント、列に機能が割り当てられており、コンポーネントが担っている機能を示す。Fig. 7(a)は青ブロックがコンポーネント、緑ブロックが各コンポーネントが担う機能を表し、この関係を DMM で表現すると Fig. 7(b)のようになる。これにより、各コンポーネントの満たすべき機能が明らかになるため、検証すべき項目を洗い出すことができる。また、列に注目すると、機能を構成するコンポーネントを特定することが可能となる。そのため、不具合が発生した際には機能を構成するコンポーネントを検証していくことにより故障元となる箇所を特定する指標とすることができる。

■ Matrix C/ コンポーネント DSM



(a) Connection of Components (b) Design Structure Matrix
Fig. 8 Components DSM

Matrix C はコンポーネント間の入出力関係をしめす。Fig. 8(a)はコンポーネント間の入出力関係を表し、DSM で表現すると Fig. 8(b)のようになる。Matrix A と同様に要素の行に注目すると出力先、列に注目すると入力元を知ることができる。例えば、Fig. 8 において Component D の検証を行いたい場合には、Component A 及び B からの入力が必要であることがわかる。すなわち、開発初期段階などにおいて Component A 及び B が無い状態においても、入力の値や負荷を予測・再現することが出来れば、Component D の検証を行うことができることを示す。これにより開発後期の手戻りを減らすことが期待できる。

2. 6. Step4 : 各階層でのマトリクスの作成

Step 1 及び Step 2 で洗い出した各階層のシステムと機能をもとに、Step 3 のマトリクスを各階層で作成する。

3. 検証プロセスの適用例

本章では、地震先行現象検知衛星「Prelude」に対して、本研究で提案するプロセスを適用し、有効性の評価を行う。Prelude は VLF 帯の電波強度の観測を行い、地震先行現象とされている電波の減少を検知し、地震と電波強度との因果関係を明らかにすることをミッションとした 6U CubeSat である。

3. 1. Step1 : モデルの階層化

システムの階層は衛星システム全体、サブシステム、コンポーネントの各開発段階で検証を実施することができるよう、Table 2 に示す 3 階層とした。

Table 2 Structure of Prelude Model

System Level	Definition
Level 1	Prelude (衛星)
Level 2	EPS, C&DH, COM, ADC, VLF
Level 3	OBC, Sensor, Memory, Battery, Solar Cell, etc.

3.2. Step2：運用モード・機能の分解

今回はミッションの一つである VLF 帯の電波強度の測定を行うシーケンスを想定し、機能展開の Level 1 は「VLF センシングモード」とした。また、Level 2 で洗い出しを行った機能を Table 3 に示す。また、Level 2 に示した M1～4 はミッションを行う上で必要な機能(ミッション機能)、R1～2 は衛星として必要な機能(バス機能)を示す。

Table 3 Function of Prelude			
Level1	機能	No.	VLF センシングモード
Level2	ミ ッション 機 能	M1	地上局からのアップリンクを処理する
		M2	姿勢制御を行う
		M3	ミッションデータを取得する
		M4	地上局にデータを送信する
Level2	バ ス 機 能	R1	電力を供給する
		R2	HK データを取得する

3.3. Step3・4：マトリクスによる関係の可視化

Level 3 における MDM を Fig. 9 に示す。Level 3 においては、level 2 より 23 個のコンポーネント、47 の機能に分解をした。

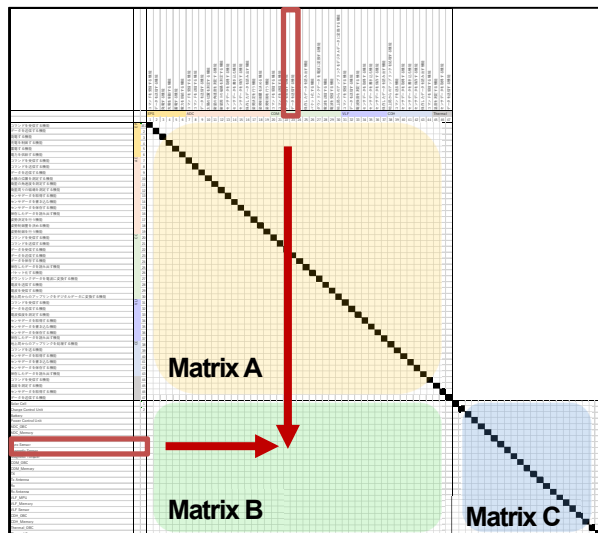


Fig. 9 Prelude MDM

3.4. 考察

■ 検証項目の可視化

Matrix B に示される機能とコンポーネントの関係より検証を行う箇所と項目を知ることが可能である。検証の際は、Fig. 9 中に示す赤い矢印のように各コンポーネントに割り当てられている機能をひとつひとつ確認することによって、検証の漏れを防止する。

■ 検証の手順

また、検証を行う順番については Matrix A に示される機能 DSM を並び替え、機能の流れを整理することにより手順の最適化を行う。今回は並び替えをおこなった結果、「電力を供給する」機能が最もシステムの構成の根幹にあることから、電気系のシステムから検証を行うことが最適であることが分かった。加えて、機能の流れが明らかになったことにより、不具合が発生した場合においては、手順を遡っていくことによって原因を追究することが可能となる。

■ 検証を行うタイミング

どの開発段階でどのような検証を行うかについては、機能の展開を行う際に、定義したシステムレベルに合わせてレベルダウンを行うことによって、検証を行うタイミングを明らかにする。たとえば、level 2 で定義した「M3 / ミッションデータを取得する」機能は level 3 において 7 つの機能に分解を行っている。そのため、まずは level 3 においてコンポーネントレベルで 7 つの検証を行ったのち、level 2 で VLF サブシステムとして、「センシングを行い、データを取得できているか」の検証を行う。これによって、各開発段階での検証項目を明確にし、不具合が作りこまれることを防止する。

■ 検証に必要な要素

また、マトリクスから検証時に必要な要素を得ることができ。例えば、Level 2 において定義した姿勢制御系の「姿勢制御を行う」機能について検証を行うのであれば、Level 3 で分解をした 13 の機能と 6 つのコンポーネントが必要となることが分かる。これによって、開発の初期段階においてもシステム統合後の負荷などを予測した上でコンポーネントの検証を行うことが可能となるため、開発の手戻りを防ぐことが期待できる。

4. 結論

本研究では、超小型人工衛星の検証における抜け漏れを防ぐため、機能とシステムで構成した MDM モデルによるシステム検証の可視化法を提案した。また、地震先行現象検知衛星「Prelude」に対して提案するプロセスを適用することによって、検証項目、検証箇所、検証を行う順序、タイミングを明示化できることを示した。これによって、検証の抜け・漏れの防止や不具合発生時の原因追究、システム検証の効率化を図ることが可能である。今後はこの手法を実際開発に適用し、コストや検証工数の観点から定量的に有効性の評価を行うことが必要である。

参考文献

- [1] Space Works : 2019 Nano/Microsatellite Market Forecast, 9th Edition, At <https://www.spaceworks.aero/nano-microsatellite-forecast-9th-edition-2019/>, 2019
- [2] Michael Swartwout, : CubeSats and Mission Success 2016 Update, 2016 Electronics Technology Workshop, At <https://nepp.nasa.gov/workshops/etw2016/talks/14TUE/20160614-0900-Swartwout-etw2016.pdf>, 2016, p.14.
- [3] Mike Tolmasoff, et.al. : Improving Mission Success of CubeSats, U.S. Space Program Mission Assurance Improvement Workshop, 2017, p.31.
- [4] 白坂成功, 中須賀真一 : ほどよし信頼性工学とその適用, 第 57 回宇宙科学技術連合講演会講演集, 3001, 2013
- [5] INCOSE : Systems Engineering Handbook , INCOSE-TP-2003-002-03.1 , August 2007
- [6] 宇宙航空研究開発機構チーフエンジニアオフィス : システムズエンジニアリングの基本的な考え方 初版, At <https://ssl.tksc.jaxa.jp/isasse01/kanren/BDB/BDB06007BSEkih on.pdf>, 2007, P.25.
- [7] S.D.E.a.T.R.Browning : Design Structure Matrix Methods and Applications, Massachusetts Institute of Technology, 2012, P234.