

1 技術継承問題に取り組むに至った背景

超小型衛星開発は今後継続的に発展していくことが予測されている。

超小型衛星開発のように複雑で統合度が高いシステムを一括で開発を行っている多くの大学研究室，教育・研究機関が抱える問題点として，人の入れ替わりが激しいために生じる，” 暗黙知の損失” という技術継承に関する問題がある。日本大学においても人の入れ替わりによる暗黙知の損失が，卒業した当時の開発メンバーにシステム仕様を聞かざるを得ないケースの発生や再開発を要求されるなどプロジェクトのスケジュールを圧迫している。他大学ではシステムの失敗の原因が技術継承に問題があったとされる報告も出てきている。

このように超小型人工衛星開発における技術継承問題は，プロジェクトのミッション成否にかかわる重要な問題であり，組織内において技術継承が適切に行わないと，組織の技術力が低下し，継続的に発展できないことが予測される。本研究では，技術継承問題を定義し，どのようにこの問題に対処すべきかを提案する。

2 超小型人工衛星開発における技術継承問題の問題提起

2.1 語句の定義

- システム (境界) : 定義された目的を成し遂げるための，相互に作用する要素を組み合わせたもの。ハードウェア，ソフトウェア，ファームウェア，人，情報，技術，設備，および他の支援要素を含む
- 粒度 : 階層に渡るシステム(境界)の粗さ。
- 抽象度 : システム(境界)の見方に関する切り分け。システムに課せられた要求，システムの機能的振る舞い，システムの構成要素に関するアーキテクチャなどの順に具体的であるとする
- 暗黙知 : プロジェクトや組織において個人あるいは当事者間で暗黙の了解とされている，システム(境界)のアーキテクチャ
- 形式知 : プロジェクトや組織において明示的に示されていて個人が参照可能な，システム (境界) のアーキテクチャ
- 工数 : 時間×人
- 相互関係 : システム内での要素の関係性。システムの

性質に依存して様々な種類がある。

- 開発当事者 : システムの開発に携わった者で，本論文では暗黙知を持つ者である
- 新規参入者 : 技術継承をされる側を示す
- アーキテクチャ : システム(境界)と外界との関係及びシステムを構成する構成要素間関係(の抽象的記述)。アーキテクチャには3つの状態がある。①開発当事者の暗黙知の状態，②形式知の状態，③新規参入者の暗黙知の状態，②，③の状態において技術継承が完了されたとみなすこととする

2.2 工数損失関数による技術継承問題モデル

暗黙知の損失によって害を被るのはプログラムやプロジェクトの技術力である。技術力が低下するとより高いレベルの要求に対処できない，またはシステムの失敗を招く可能性を高める。どのようにして暗黙知の損失による技術力の低下が起こるのかを説明するために以下のように工数損失関数 $L(y)$ を提案する。

$$L(y) = L_A(y) + L_B(y)$$

$$L_A(y) = \frac{A_A}{\Delta^2} y^2, \quad L_B(y) = A_B(\Delta - y)^2$$

$L(y)$: プログラムやプロジェクト(全体)が損失する工数

$L_A(y)$: 暗黙知損失による再開発工数関数

$L_B(y)$: 継承活動に要する工数関数

$y(0 < y < \Delta)$: 継承がなされない暗黙知の数

Δ : 伝達されるべき暗黙知の数

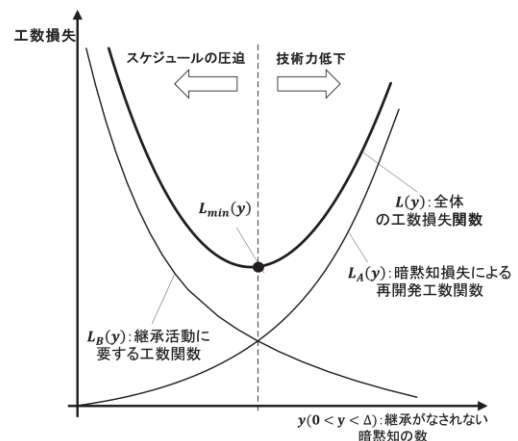


図 1 工数損失関数

A_A : 暗黙知損失による再開発工数係数
 A_B : 技術継承活動に要する工数係数

技術継承問題の重要な論点は伝達されるべき暗黙知をどのように決めるかということである。

また、実践においては y : 継承がなされない暗黙知の数、つまり、どこまで継承するのかの決定が重要となる。図1のようにプログラムやプロジェクト(全体)において y : 継承がなされない暗黙知の数が増えると、そのアーキテクチャを再開発するための工数を要することとなるため全体が技術力低下のリスクを抱えることと成る。また、 y : 継承がなされない暗黙知の数が減る、つまり暗黙知を多く継承しようとする、継承するための工数を要することとなるため、全体のスケジュールが圧迫される。そのため全体が損失する工数の最小値 $L_{min}(y)$ となるような y に技術継承の目標を設定するべきである。

2.3 技術継承を実現する「場」の設計

A_A : 暗黙知損失による再開発工数係数には、個人の技術力などの考慮しにくい影響が含まれていると考えられ、これはプログラムとして制御しにくい。

そのため、プログラムやプロジェクトにおいて暗黙知の損失関数を最小化するためのアプローチとして A_B : 技術継承活動に要する工数係数に着目しアプローチすることが望ましいと考える。

いかに工数をかけずに効率的に、かつ重要な暗黙知を伝達するかという着眼点である。暗黙知を伝達する方法として主に2通り存在する。

- ① 暗黙知→暗黙知 (有識者の暗黙知をOJT, ディスカッション, レクチャー, 口伝などによって伝達する方法)

- ② 暗黙知→形式知→暗黙知 (有識者の暗黙知を文書化, モデル化などして, 他者がそこから知識を得る方法)

このような手段が「場」である。

3 衛星開発プログラムにおけるアーキテクチャと技術継承問題点

3.1 SPROUT プロジェクトにおけるアーキテクチャ：プロジェクトの地図

プロジェクトのアーキテクチャを完全に網羅するためにプロジェクトのライフサイクル全般に渡る境界を明確にする必要がある。

3.1.1 プロジェクトブレイクダウンストラクチャー (PBS)

本研究では、図2に示すPBSのように、プロジェクトの境界を粒度に渡り、明確にした。

- ✓ **SPROUT ミッション (ゴール)** : ミッションのサクセスクリテリアや、外部からの制約などプロジェクトがアウトプットすることとした成果に関するアーキテクチャ
- ✓ **SPROUT 運用システム** : ミッションを達成するために開発・製造された衛星システムや地上局システムに関するアーキテクチャ
- ✓ **SPROUT 開発・製造プロセス** : 運用システムを実現するプロセスに関するアーキテクチャ. ミッションのアーキテクチャを決定するプロセスも含む
- ✓ **SPROUT 運用プロセス** : 運用システムとの相互関係を通して SPROUT ミッション達成に至るまでのプロセスに関するアーキテクチャ
- ✓ **宮崎研衛星開発・運用チーム** : プロジェクトのミッ

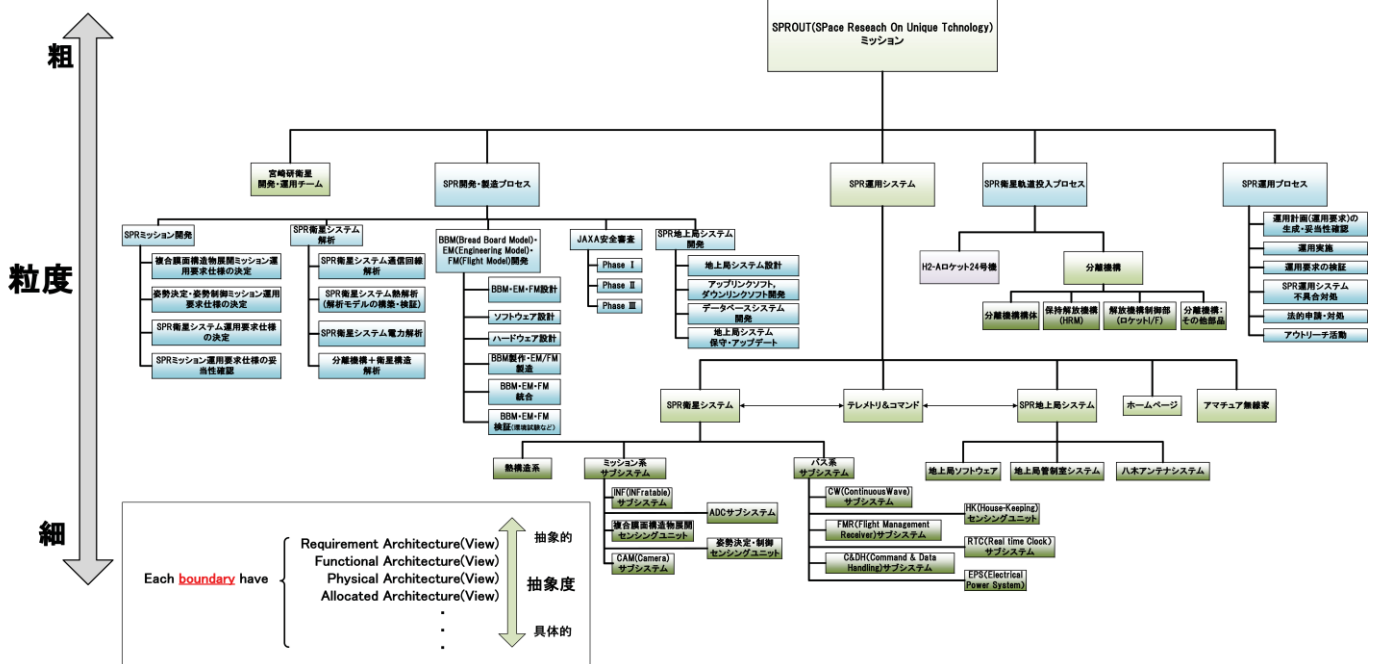


図2 SPROUTプロジェクトブレイクダウンストラクチャー(PBS)

ョンを成功させるためにシステム開発・運用を行なう組織アーキテクチャ

3.1.2 プロジェクトブレイクダウンストラクチャーに(PBS)の粒度と抽象度

プロジェクト内で存在するアーキテクチャは多岐に渡る。PBS では粒度に渡ってどのような境界が存在するかを示したが、1つ1つの境界には抽象度にわたるアーキテクチャが存在する。また、アーキテクチャは関係性(相互作用または相互接続)をもって目的を達成している。そのため、アーキテクチャ同士の関係性もまたアーキテクチャとして存在する。このように粒度と抽象度に渡ってアーキテクチャが継承されることが重要である。

3.2 5年間のSPROUT 衛星プロジェクト運用に関する技術継承問題点

5年間のSPROUT 運用を適切に行うために、開発当事者の暗黙知や衛星の運用の仕方に関する暗黙知を適切に継承することが要求される。しかし、このような暗黙知がSPROUT のミッション・運用を行う次世代に適切に継承されないと、運用中に発生する衛星や地上局の不具合事象に対処できないばかりか、次世代のSPROUT 運用者が開発当事者の間では行っていなかった操作を誤って行ってしまい不具合を誘発するなどといったことが想定される。

また、運用システムに不具合が発生した際にその対処にかかる工数がプロジェクトのスケジュールを圧迫することとなる。さらに、スケジュールの圧迫により不具合対処がなされないと技術力の低下ばかりか、プロジェクト破綻のリスクを背負うこととなる。

3.3 SPROUT プロジェクトから次期衛星開発プロジェクトへの技術継承問題点

この点に関してはシステム仕様に関する技術継承と開発プロセスに関する技術継承の2通りが考えられる。

① システムアーキテクチャに関する技術継承

「SPROUT 衛星システム」のバス部、「SPROUT 地上局システム」や「SPROUT 運用システム」はプロジェクトとして成功した「SEEDS」のアーキテクチャを再利用して設計されている部分がある。既存の成功したシステムのアーキテクチャを再利用することは、信頼性の担保や開発コスト(期間、費用など)の短縮を実現する。

SPROUT 開発ではSEEDS よりも多くの機能を有し開発が複雑になったことからアーキテクチャが開発当事者間の暗黙知となってしまう部分が多々発生し、開発当事者が卒業してしまった場合は次期衛星開発者との直接のやりとりがない限り伝承されない。その結果、次世代開発者は暗黙知となっていたアーキテクチャまたはシステムの再開を要求されることとなる。また、既存のアーキテクチャの要求や振る舞いを適切に理解しないままアーキテクチャの再利用を行うと、次期衛星開発者が既存ア

ーキテクチャ開発当事者間で暗黙知的に仕様化しているアーキテクチャを考慮できず、誤った設計判断をおこなうリスクを高めることとなる。

② 開発プロセスアーキテクチャに関する技術継承

システムアーキテクチャの再利用と同様に、SPROUT プロジェクトで行った、各種レビュー会を実施するプロセスや各種検証に関するプロセス、アクティビティは次期衛星開発プロジェクトに再利用可能なものを含んでいる。特に、検証に関するものにおいては、各種解析、試験の要件や実施手順を再利用することはコストや労力の面から次期衛星開発プロジェクトにとって有益である。

4 技術継承問題に対して改善を繰り返す行なうプロセスアプローチ

プロセスアプローチとは暗黙知の損失関数を最小化するように「場」を設計し、組織単位で繰り返し改善を行なうことである。これを推進するためには組織が取り組む問題設定を「単一の衛星開発プロジェクトの成功」から「継続的な衛星開発プロジェクト(=衛星開発プログラム)の成功」へと拡大して取り組むという全体的な最適化を目指す必要がある。

プロセスアプローチを組織単位で実現するにあたって直面する重要な課題点の一つは何が暗黙知で、どのアーキテクチャを継承するのが効果的なのかを評価することである。現状では開発当事者が技術継承と言われて無作為に継承するアーキテクチャを決めており、それが正しいのかどうか判断することが難しい状況である。

そこでシステム再開発や理解に工数を要するであろうアーキテクチャを相対的に順位づける手法を考案する。

4.1 SPROUT 衛星システムソフトウェア(SW)アーキテクチャのデザインストラクチャーマトリクス(DSM)モデル: SatSW-DSM



図3 SatSW-DSM モデル(1部分抜粋)

PBS ではプロジェクトやシステムを構成する構成要素の相互関係をモデル化することには向いていない。DSM モデルを用いることによって相互関係をモデル化することが容易になる。本研究ではSPROUT 衛星システムのソ

フトウェアアーキテクチャ（C 言語で記述されている）を DSM モデル化した。SPROUT 衛星システムは 12 のサブシステムから構成されており、それぞれのサブシステムはマイクロコンピュータを搭載している。各サブシステムソフトウェア（バス部のみ）をサブルーチンレベルにまで切り分け、主にサブルーチン間のコマンド・データのやりとり、包含関係などの相互関係を図 3 のようにモデル化した。

4.2 モデルの評価

SatSW-DSM において各サブルーチンを中心性指標を用いて評価した。以下に結果を示す。

- ① 次数中心性：各ノードと関係を持つノードの数
- ② 媒介中心性(BC)：そのノードが他の 2 つのノードをつなぐ最短経路上にいる程度

表 1 中心性評価結果（表は BC の上位 5 位を示す）

サブシステム	サブルーチン(ノード)	次数中心性	媒介中心性(BC)
C&DH1	void send_cmd1()	48	30759
C&DH2	int main() case 0x88	3	22106
C&DH1	int main() case 0x88	3	16383
C&DH1	void line_check1()	10	12693
FMR1	void line_check()	5	12584

4.3 結論

この評価にランクインした媒介中心性(BC)が高いサブルーチンは事実、SPROUT 衛星システムアーキテクチャを理解する上で重要である。この評価によって技術継承を行うための指針が明確に示された。

5 今後の展望

5.1 DSM モデルからマルチドメインマトリクス (MDM) モデルへ拡張した評価

SatSW-DSM モデルでは開発の手戻りなどによって課せられた要求、実際の運用の際に発生した不具合に関する要求などをモデル化できていない。これらは工数をかけた結果として現れるので重要である。また、これらを中心性評価に含めることで評価の順位が変わる可能性もある。より適切な評価を行うために全体を粒度、抽象度にわたって完全に網羅するマルチドメインマトリクス(MDM)を PBS に適用しモデル化を行なっていくのが良いと考える。

5.2 抽象度を高めたシステムのモデル化手法開発

フローチャートのような抽象度の高いモデルをアウトプットするようなプロセスは重要である。

例えば、新規参加者が表 1 の void send_cmd1() というサブルーチンが担っている役割を、与えられたプログラムコード情報の中だけから理解せざるを得ない場合を考えると、void send_cmd1()を介する約 3 万通りの経路と 48 通りの接続を精査し帰納的に理解することとなる。この膨大な工数を減らすために有用なのが抽象度を高めたシス

テムのモデルである。要するに何なのか示されているとアーキテクチャの理解が促進されるためである。これを実現する手法として超小型人工衛星開発で実践可能なモデルベースシステムズエンジニアリング手法(MBSE)の開発が望まれる。

5.3 技術継承プロセスアプローチの継続的改善

技術継承問題へのアプローチは、プログラムやプロジェクトで活動する開発当事者や新規参加者の意識が重要である。いかにプロセスアプローチを促進し、継続的に改善を繰り返すか考え実行するためには一人ひとりに組織全体に対する問題意識がなければならない。そのため、システムアーキテクチャの評価を行うことによって継続的になぜ技術継承が問題であるのかを明確に指南されていかなければならない。

6 参考文献

- [1] 石井亮介, 佐原宏典：「超小型衛星のミッション正否に基づく現状把握と将来予測」, 日本航空宇宙学会論文集, Vol.60, No.2, pp.65-73, 2012.
- [2] 中須賀真一, 白坂成功：「「ほどよし信頼性工学」の考え方とその適用～生き残る衛星を目指して～」, UNISEC 2013-001, 2013.
- [3] 井上祥子, 他：「超小型衛星開発の技術継承における新入生教育の試み」, UNISEC Space Takumi Journal for Practical Study of Problem Finding and Solving in Space Systems, Vol.3, No.3, pp.40-62, 2012.
- [4] UNISEC：「UNISEC Space Takumi Journal for Practical Study of Problem Finding and Solving in Space Systems」, Vol.1, 31 March 2011.
- [5] Hideki Uchikawa, Toshiyuki Yasui, Seiko Shirasaka：「Hierarchical Knowledge Transfer and Creation Model」, APCOSEC2013.
- [6] Institute of Electrical and Electronics Engineers (IEEE)：「IEEE Standard for Application and Management of the Systems, Engineering Process, pp.4-17, IEEE 1220-2005 (2005).
- [7] スティーブン・D・エッピンジャー, タイソン・R・ブラウニング著, 西村秀和監訳, 大富浩一, 関研一訳：「デザイン・ストラクチャー・マトリクス DSM 複雑なシステムの可視化とマネジメント」, シリーズ Engineering Systems, 慶應大学出版, 2014 年.
- [8] Mark W. Maier：「The Art of Systems Architecting, Third Edition」, CRC Press, 2009.
- [9] サンフォードフリーデンタール, アランムーア著, 他, 西村秀和監訳, 白坂成功共著, 他, 「システムズモデリング言語 SysML」, 東京電機大学出版局, 2012.