

# モデリングツール EA (Enterprise Architect) を活用した運用設計への取り組み

Efforts for operational design utilizing modeling tool EA (Enterprise Architect)

伊藤 毅\* 小浜 光洋\* 井本 綾子\*  
Tsuyoshi Ito, Mitsuhiro Kohama, Ayako Imoto

国立研究開発法人宇宙航空研究開発機構 (JAXA) ではモデルベース開発手法を用いた新たな宇宙機システム設計・検証フレームワークの構築が研究され、「モデルベース開発ハンドブック」としてまとめられている。当社では、その考え方をいち早く取り入れ、火星衛星探査計画プロジェクトの提案段階からモデリングツール EA (Enterprise Architect) を用いた運用設計を進めている。本稿ではその取り組みについて紹介する。

Japan Aerospace Exploration Agency (JAXA) has been researching the construction of a new spacecraft system design and verification framework using model-based development methods, and has compiled it as a “Model-based development handbook”. We have adopted this idea as soon as possible and are proceeding with operational design using the modeling tool EA (Enterprise Architect) from the proposal stage of the Martian Moons Exploration Project. This paper mentions the efforts.

## 1. まえがき

宇宙機システムを運用するためには、軌道に打ち上げられる宇宙機及びデータ取得や制御信号を送る地上システムの開発に加え、それら全体を運営管理する運用作業が必要となる。

これまで日本の宇宙機システム開発は、宇宙機本体の開発が先行し、地上システムや運用は後から宇宙機本体の仕様に合わせて設計や検討が進められるケースが多く見られた。スケジュールに余裕がある場合にはこの開発スタイルでも大きな問題は発生していなかったが、近年、インタフェースの不整合や運用検討の不足などが原因で、軌道上で不具合が発生するケースが少なからず生じている。この要因は、商用衛星をはじめとして開発期間が短くなる傾向が強まり、開発スケジュールに余裕がない中で、宇宙機と地上システムの開発が並行して行われるようになってきたことや、ミッション実現のために高度な運用を求められる傾向が強くなってきたことにあると考えられる。

そのため、宇宙機と地上システムの総合設計の必要性や運用設計の重要性がクローズアップされるようになり、中でも運用設計は宇宙機システム開発初期に宇宙機

や地上システムの仕様を決めるために行う重要な業務であると明確に認識されるようになってきた。

また、月や火星の惑星探査や宇宙デブリの捕獲など、これまでに経験したことのない新しいミッションに対して、早い段階から宇宙機システムの技術課題とその対応策を確認することにより、開発上のリスクを抑制することが必要不可欠となってきている。

こうした背景から JAXA では、技術的なリスクを低減しミッション成功率を高めるために、運用設計から宇宙機システムに必要な機能を効率的に抽出する方法として、UML / SysML を利用したモデルベース開発手法を用いた研究が進められ、「モデルベース開発ハンドブック」としてまとめられている。

我々は三菱電機株式会社鎌倉製作所が JAXA からの委託を受けて開発中の火星衛星探査計画プロジェクトの宇宙機システムの開発に携わっており、新規性の高いプロジェクトの開発方針とマッチするこのモデルベース開発手法の考え方を取り入れて、モデリングツール EA (Enterprise Architect) を用いた運用設計に取り組んでいる。この EA は、モデルベース開発に適したツールであり、各種のモデリングツールの中で運用シナリオから運用手順書への紐づけが容易であり、また導入コストも

廉価である観点から選定に至ったものであり、将来的に運用手順書を自動生成することも視野に検討を進めている。

## 2. モデリングツール EA について

「Enterprise Architect」(エンタープライズアーキテクト)<sup>(1)</sup>(以下、EA とする)は、Sparx Systems 社が開発した、UML (Unified Modeling Language)<sup>(2)</sup>、SysML (Systems Modeling Language)<sup>(3)</sup>などの様々な記法に対応し、業務システム・Web アプリケーション・組込み系ソフトウェアなどの設計開発全体を通して多くの企業で利用されている設計開発支援ツールである。

表 1 に示すように、UML で規定された 14 種類の図、SysML で規定された 9 種類の図の表現に対応する。

この他にもビジネスプロセスモデリング表記法 (BPMN) やデータフロー図 (DFD) など様々な記法に対応し、さらにユーザそれぞれに適したカスタマイズが可能な拡張性を持つ。また、モデルからのソースコード生成機能、モデル内のトレーサビリティ管理機能等、多種多様な機能を有する「広範囲をサポートする実用的モデリングツール」である。

## 3. 運用設計への適用

### 3.1 EA を用いた運用設計概要

モデルベース開発と言うと、その手法は CAD などを用いた仕様の表現、MATLAB / Simulink を用いた

シミュレーションによる解析など多岐に渡るが、これらモデルを用いることで、「開発の上流工程を重視する」「仕様を明確化する」「開発プロセス全体のコミュニケーションを改善する」ことを共通の目的としている。

本稿におけるモデルベース開発では、その中でも UML や SysML 等のモデリング言語を用いた運用設計を行う。UML は、オブジェクト指向のソフトウェア開発に用いられ、データ構造や処理の流れなどを図示するためのモデリング言語である。SysML は UML をハードウェアも含めたシステム開発に適用できるよう拡張したものであり、UML のオブジェクト指向の考え方にに基づき、システム全体を設計するためのモデリング言語である。これらモデリング言語の中から、宇宙機システム開発に適した図を取捨選択し、図 1 に示す表や図を用い運用設計を行った。

運用の一連の流れを、運用フェーズや運用内容により一塊にしたものを運用シナリオと呼ぶ。まず、宇宙機システム全体における運用シナリオの量や各シナリオの階層構造を明確にするために、運用項目一覧表を作成した。

運用項目一覧

表 1 代表的な対応ダイアグラム

UML 構造図	クラス図
	パッケージ図
	オブジェクト図
	コンポジット構造図
	コンポーネント図
UML 振る舞い図	配置図
	プロファイル図
	ユースケース図
	シーケンス図
	コミュニケーション図
SysML	アクティビティ図
	状態遷移図
	タイミング図
	相互作用概要図
	要求図
SysML	ブロック定義図
	内部ブロック図
	アクティビティ図
	パラメトリック図
	シーケンス図
	状態遷移図
	パッケージ図
	ユースケース図

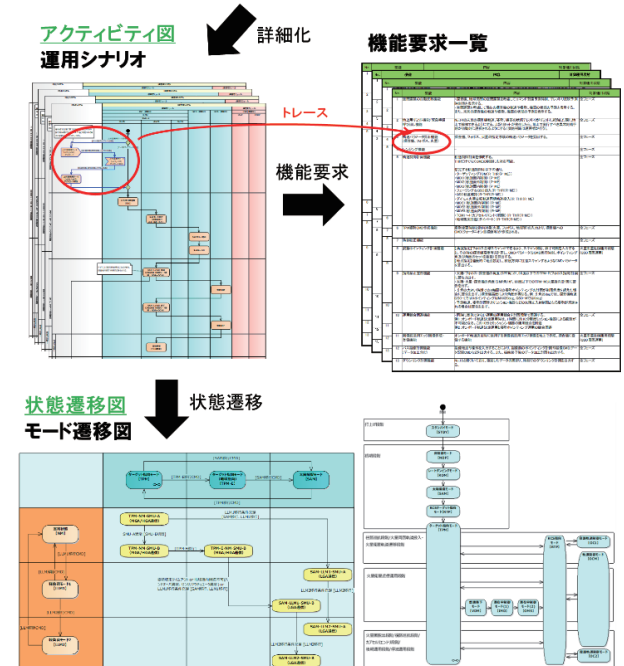


図 1 利用図表相関図

次に、運用項目一覧表を基に個々の運用を深掘りし、詳細な運用シナリオを作成した。宇宙機システムの運用では、時系列に沿って、「誰が」、「何を」、「どの順番で」行い、さらにその際の「データのやり取り（インプット／アウトプット）」を示すことが重要となる。そこで、これらを表示するためにアクティビティ図を用いて「運用フロー」を作成した。一般的に、処理やデータの流れを可視化する場合、シーケンス図を用いる事もあるが、全体的な俯瞰、並行実行の可視化、異常時の復帰ポイントなどが理解しやすいという特徴があるため、アクティビティ図を採用している。

同じく宇宙機システム設計とその運用設計に密接に関係するのが、システムの状態又はモードである。運用シナリオと連携して、モードの定義、モードの遷移条件を明確にする意図で状態遷移図を活用しモード遷移図を作成した。

最後に、作成した運用フローを分析し宇宙機システムに必要な機能を抽出する。その際、複数のシナリオを横通しで見て共通的な機能をまとめるなど、システム、サブシステムとして必要な機能単位でまとめる。抽出された機能要求を可視化するために機能要求一覧を作成した。

これらの運用項目一覧、運用シナリオ、モード遷移図、機能要求一覧を用いて、宇宙機システム運用を設計した。

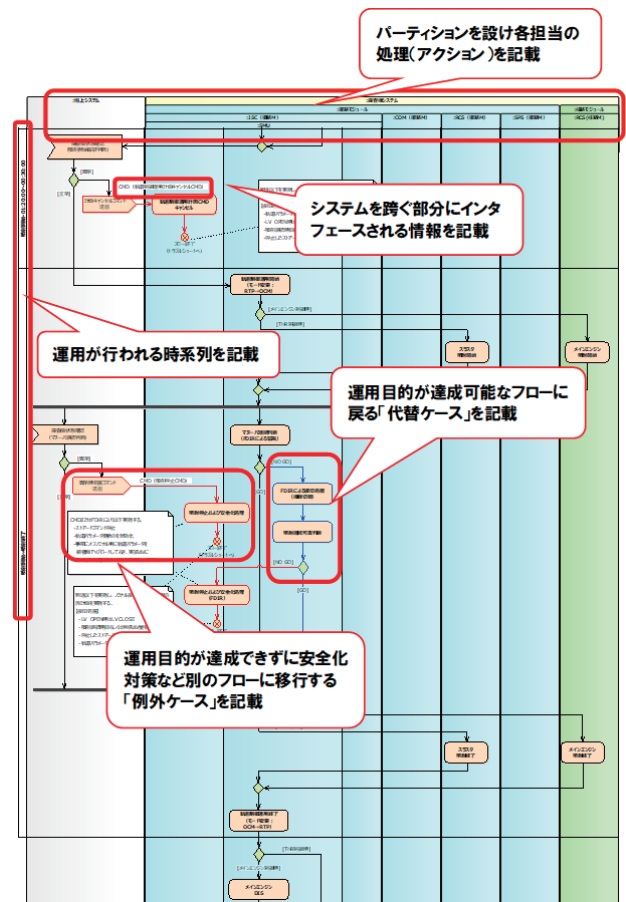


図2 運用フロー例

### 3.2 モデルの作成手法

宇宙機システムの運用設計においてEAを用いることが最も効果的であった運用シナリオ作成について詳しく説明する。

宇宙機システム開発において、機能が動作する流れ、やり取りされるデータ、その順番を知るためには、開発関係者との運用イメージの共有が重要となる。上述の運用シナリオを文章だけで書き表すことも可能ではあるが、開発者が直観的に全体を俯瞰して確認することは非常に困難である。また各個人による解釈の違いから、イメージの共有が正確に行われず可能性がある。そのため、フローチャート型の図面方式として可視化することで、共通の視覚情報として入力し理解を容易にする。

運用のモデルはアクティビティ図を用いた運用フローで示す。運用項目一覧で定義した運用シナリオ、及びインプット、アウトプット情報に基づき、各処理の時系列、データのやり取りを示し可視化する。運用フローには図を区切るためのパーティションを設け、区切られた各レーンにシステムやサブシステム名称を設定し、それぞれの担当する処理（アクション）を明確にした。さらに異常発生時の対応、復帰ポイント、及び安全化処理開始ポイント等、オフノミナルな事態を想定した運用につい

でも表現した。図2に作成した運用フローの一例を示す。

運用フロー作成の際に課題となったのがモデルの記述粒度である。粒度を細かく設定した場合、モデルが大量に生成され、調整すべき相手に関係しない内容が多く記載されることになり、関係者がレビューしきれない事態を生む。逆に粒度を大きく設定した場合、個々のアクションが抽象的になり設計を行う上で必要な情報が欠落する。そこで、開発フェーズごとに必要な内容に合わせ、運用フローの記述粒度を調整した。例えば概念設計フェーズにおいては、図3①に示すように、総合システム的な視点から、地上システム、及び宇宙機システムとの機能配分に着目し、両システムのインタフェースのタイミング、及びインタフェース項目についてのみの表示とした。その後の基本設計フェーズにおいては、図3②に示すように、概念設計時のフローを元に、地上システム及び宇宙機システムの個別システムの中で、内部サブシステムやコンポーネントに詳細化し、それぞれの運用設計を行った。EAを利用することで、上記のような開発フェーズに応じたフローの作成を行う際、運用要素がどのように配分されたか追うことができ、トレーサビリティが担保された。

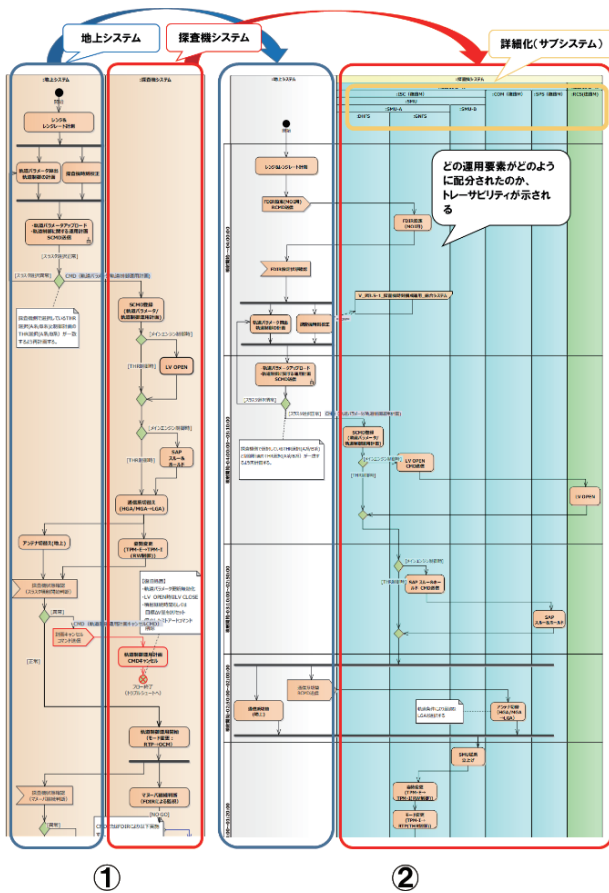


図3 運用フロー詳細化 (①概念設計 ②基本設計)

EA を利用しモデルを作成することで、運用設計を効率的に、かつ効果的に進めることができたが、あくまでモデルの作成は、仕様を明確にするとともに、開発プロセス全体における各開発担当者間の円滑なコミュニケーションを目的としている。そのため、いずれの開発フェーズにおいても、運用シナリオを伝えたい相手とどのような情報を共有するのかをあらかじめ調整した上で、モデルを作成することを心掛けた。

#### 4. 今後の展望

運用設計にモデリング言語を取り入れたことによる成果の一つは、概念設計、基本設計と進む中で常にシステム間の情報伝達を意識することを要求され、設計の詳細化に伴い段階的にやり取りする情報の粒度（情報の種類、条件）を高めることができたことである。これにより運用基本設計の結果を宇宙機システムへの機能要求として提示することができ、宇宙機システム設計の一翼を担うことができた。

次の運用設計のフェーズにおいては、現状の運用設計に宇宙機システムの設計結果を取り込み運用シナリオから運用手順に落とし込むことが主題となる。手法として

は、まず宇宙機システムの設計結果であるコマンド、テレメトリの情報を内部サブシステムやコンポーネントに割り当て、コマンド、テレメトリのやり取りを定義していく。そしてコマンド、テレメトリの情報交換を定義した運用シナリオから特定の場面を取り出すことで運用手順を構築する。これにあたり、モデリングツールであるEA に対して運用設計用アドオンツールを開発し、作業性の向上を図る計画である。

拡張する機能としては、次を想定している。

- (1) テレメトリ、コマンド DB 情報を取り込む機能
- (2) 作成したアクティビティ図のアクションにテレメトリ、コマンドを割り当てる機能
- (3) アクティビティ図から必要なアクション、記載情報を選択して運用手順として出力する機能
- (4) (3)の逆変換として運用手順を読み込みアクティビティ図に取り込む機能

モデリング言語を取り入れたことによる成果の二つ目は、状態遷移を定義することにより、派生手順（主系、従系の手順等）や逆手順（切り替え手順）の作成を情報の欠如なくシステムティックに行えることである。従来の手順作成では派生手順や逆手順は作業者が一つ一つ手作業で作成していた経緯があるが、今後の運用手順の整備にあたりモデリングツールの機能を用いてアクティビティ図と状態遷移の連携を行い、システムティックに変換して手順作成の効率化を図る。

このための具体的な実現手法は検討中であるが、現在想定しているシナリオを次に示す。

- (1) 現状のアクティビティ図の各コンポーネントに対して Class 化を行う
- (2) 各コンポーネント Class に対して状態を定義する
- (3) モード遷移図にて、対象となるコマンド、テレメトリの関連付けを行う
- (4) ステート変換を行い派生手順、逆手順を作成する

ただし非対称な運用手順も多くあるため幾つか簡単な運用パターンでまずは試作し、作業の有用性を検証していく必要がある。

宇宙機システム機器の開発でもモデリング言語を取り入れた設計作業が浸透してきている。将来的にはモデリング言語を介して機器の設計から運用までの一貫的な開発が行えるようになることを期待している。

#### 5. むすび

今回、新しい設計手法として、火星衛星探査計画プロジェクトでのモデリングツール EA を用いた運用設計への取り組みを紹介した。具体的な宇宙機システム開発への適用はまだ始まったばかりであり、今後の展開も含め

スタンダード化に向けての課題は多い。最終的な目標は冒頭でも述べたとおり、技術的なリスクを低減しミッション成功率を高めることである。これまでの宇宙機システム開発で培った経験や知見をベースに、多くの関係者と協力しながら新しい設計手法の確立に向け取り組んでいきたい。

#### 参考文献

- (1) SPARX SYSTEMS : ENTERPRISE ARCHITECT  
<https://www.sparxsystems.jp/>
- (2) Object Management Group : OMG Unified Modeling Language (OMG UML), Version 2.5.1 (2017)
- (3) Object Management Group : OMG Systems Modeling Language (OMG SysML), Version 1.6 (2019)

UML (Unified Modeling Language) / SysML (System Modeling Language) は、OMG (Object Management Group) の登録商標です。

EA (Enterprise Architect) は、Sparx Systems 社の登録商標です。

MATLAB / Simulink は、The MathWorks 社の登録商標です。

#### 執筆者紹介

##### 伊藤 毅

2016年入社。入社以来、宇宙機の運用設計・実運用業務等に従事。

##### 小浜 光洋

2012年入社。入社以来、宇宙機の運用設計・実運用業務等に従事。

##### 井本 綾子

2003年入社。入社以来、宇宙機の運用設計・実運用業務等に従事。