

人工衛星システム開発への ZIPC の適用

NEC 東芝スペースシステム株式会社
搭載機器開発部 エキスパートエンジニア

三好 弘晃

宇宙開発事業団
高度情報化推進部 主任開発部員

三好 寛

1. 要約

本論文では、人工衛星システム開発を例に取り、ZIPC を大規模コンカレントシステム開発に適用した事例を紹介する。ここで紹介する方式は、従来紙による情報伝達に過ぎなかったシステムインテグレータと部品設計者との間に、シミュレーション可能な情報伝達手段を提供するものであり、近年浸透してきた³次元設計手法による熱機械インテグレーションと同様、機能インテグレーションの分野でも高度情報化手法による設計品質の向上、並びに開発の低コスト・短納期化に大きく貢献すると期待できる。

2. はじめに

2.1 人工衛星システム開発プロセスの概要

人工衛星は、宇宙環境において通信や観測などの任務（ミッション）を果たすために限られたサイズ（ロケットで運べる外形・重量、太陽電池で発生できる電力など）内に実現される無人システムである。

電子機器に悪影響を与える宇宙放射線や広い温度範囲、あるいは地上から宇宙空間への輸送中に受ける激しい振動・衝撃など、一般に宇宙環境は非常に過酷であるため、人工衛星にはミッション遂行に直接的な役割を果たす中継器やカメラなどのミッション機器だけでなく、それらを過酷な環境から保護・維持する「衛星バスサブシステム」も搭載される。（表1参照）

表1 典型的な衛星バスサブシステム

役割	バスサブシステム名称
衛星の姿勢、位置を決定・制御する	姿勢軌道制御系、推進系
動力源となる電気を作り、貯め、配分する	太陽電池系、電源系
衛星の状態を地上から遠隔監視制御する	テレメトリ・コマンド系、電気計装系
衛星の形状を保持し、温度を制御する	熱制御系、構体系

ミッション機器はその性格上衛星毎にコンフィギュレーションが変わることが

多いため、ミッション機器とそれを衛星バスと統合する衛星システムとは前述した表1の8つのサブシステムインタフェースを媒介とするコンカレント開発である。

設計条件が常に変化しているコンカレント開発では、設計変更の波及範囲を明確化し追跡することが重要である。宇宙業界では、この目的のためインタフェース管理図面(ICD)と呼ばれる文書がツールとして用いられてきた。

ICDには部品の外形形状、質量、発熱量などの熱機械的特性や、部品の機能を外部から利用するための監視制御手段や動作モード遷移情報など、多岐に渡るインタフェース条件が記載されている。各設計者は、ICDを通じて自身の設計変更を相手方に伝達し、変更に伴うインタフェース不整合防止に努める。

システムインテグレータと部品開発者は、初期設計段階において各部品の機能要求に加え、ICDを互いの設計条件として制定し、以降設計作業を通じてICDを維持する。製造フェーズでは、試験を通じて現品が要求機能とICDに適合していることを検証しながらシステムを組立てる。

しかし、このプロセスでも最初に制定したICDや機能配分の誤りは検証できないため、従来の衛星開発ではフライトモデルを製造する前に試作モデルを製造し、ICDや機能配分の誤りを検出してきた。

2.2 市場変化に伴う衛星開発上の課題

これまでは、衛星を開発/宇宙実証することを通じ宇宙開発関連要素技術を習得することがシステム開発の目的であった。しかし既に基礎となる要素技術の習得を完了した近年では、システム開発の目的が衛星を利用した高度なサービスを、低価格・短納期で利用者に提供することに变化しつつある。

提供するサービスの高度化は、衛星システムを高機能化・複雑化するが、低価格・短納期化要求は衛星システムの試作試験による設計検証機会を奪っていく。

一方で、非修理系である衛星システム特有の高信頼性要求は従前通り存在するため、設計検証作業の確度も維持する必要がある。

この様な要請から、設計検証の手段を試作試験からシミュレーションに移行していくことが衛星開発プロセス革新として不可欠である。

近年3次元CADの浸透により、熱機械インテグレーションの分野では設計の可視化や設計検証の早期化が図られてきているのに対し、利用者に対するサービス提供内容の早期可視化や、機能配分に関する早期設計検証の中核をなす機能インテグレーションの分野は、適切なモデリングツールの不在から設計革新が遅れている。

従来の衛星開発では、衛星の機能シミュレーションを行うツールとして「衛星

シミュレータ」を整備してきたが、これはあくまでも運用訓練や運用手順の検証用に衛星設計情報に基づき開発完了後に整備されていたため、初期設計段階では設計者が利用できなかった。

それでは上記衛星シミュレータを初期設計段階から利用可能なスケジュールで整備すればよいが、前述した衛星開発のコンカレント性から従来のシミュレータ整備方針ではシミュレータ維持コストがかかりすぎ現実的ではない。

一方で、UML などのモデリング言語を用いたトップダウン設計の適用も考えられるが、機能部品の数にして百数十点、総部品点数にして数百万点に上る衛星では、1人の設計者が把握できる範囲に自ずと限界があり、作業を分担せずに衛星シミュレーションモデルを作成することは困難である。

以上の問題点を解決し、初期設計段階の設計検証ツールとして衛星シミュレータを利用可能にすることが、課題である。

2.3 デジタルICDに基づく衛星シミュレータの整備

上記課題を解決するため、我々は従来から整備をしてきたICDに着目した。

ICDには、部品に対する制御入力であるコマンド、それに対するアンサバックであるテレメトリ、並びにコマンドに対して動作モードがどのように遷移するかを示す状態遷移図など、部品の振る舞いを表すのに十分な情報が既にリストアッ

プされている。しかしこれらは文書情報であり、動作させて検証することができない。

そこで、ICDの様式をコンピュータが理解できるようデジタル化し、そこからシミュレーションモデルを構築する方式を考案した。(図1参照)本方式によると、衛星シミュレータ整備にまつわる問題点を表2に示すように解決できる。

表2 ICDに基づく衛星シミュレータ整備の効果

整備上の問題点	課題に対する効果
整備が遅く設計フェーズで利用できない	ICDに反映された設計結果からモデルを作成できるため初期設計フェーズから利用可能
シミュレータ維持コストがかかりすぎる	従来から作成しているICD作成作業の範囲であり追加作業はない
トップダウン設計の限界	部品のモデル作成は部品開発者間で分担できるため無理はない

本方式実現の鍵となるのは「シミュレーションエンジン」の構築である。技術調査の結果、CATS株式会社の製品であるZIPCの適用が可能と判断した。ZIPCが扱う拡張階層化状態遷移表は、階層構造をなす衛星システムのモデル化に十分な表現力を有する。

次章では、ZIPCを用いた衛星シミュレータの概要を、次々章では同方式に基づくプロトタイプ評価結果についてそれぞれ紹介する。

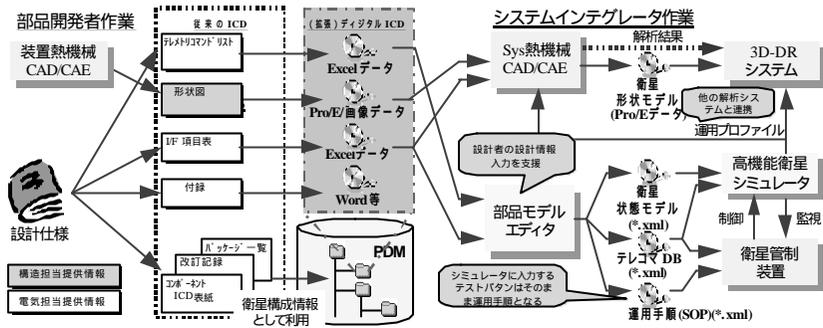


図1 デジタルICD を中心にした衛星機能インテグレーションの革新

3 . 衛星シミュレータの概要

ここでは、図2の衛星シミュレータ機能ブロック図に基づき衛星シミュレータの機能概要を紹介する。衛星シミュレータは、大きく分けて次の4つのブロックから構成される。

(1) ユーザインタフェース (UI) 部

(2) 状態モデル管理部

(3) 衛星シミュレータフレームワーク部

(4) 構成・シナリオ管理部

(1) ~ (3)はシミュレータサーバに、(4)はシミュレータ制御・表示部に、(3)は両者にも実装される。

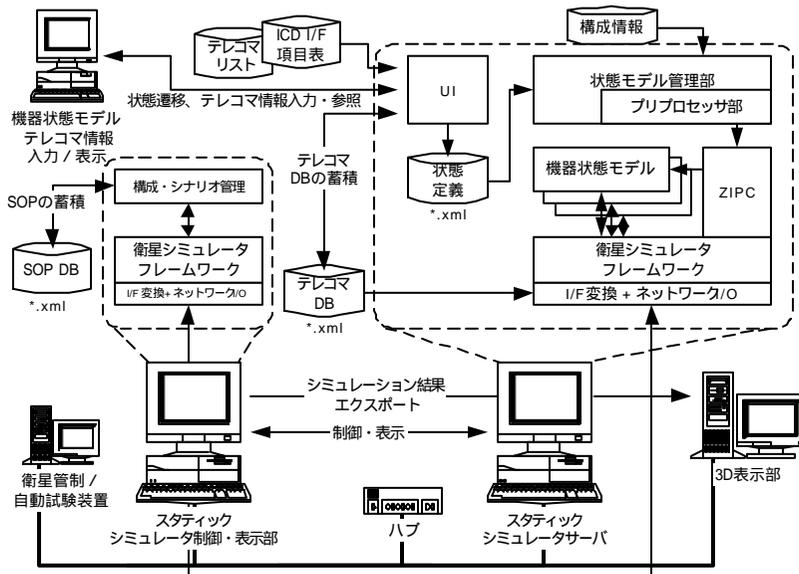


図2 衛星シミュレータ機能ブロック図

3.1 ユーザインタフェース (UI) 部

UI 部は、シミュレーション対象部品のモデル定義を行うための Web インタフェースである。部品モデルの定義には、1) 状態定義、2) イベント/モニタ定義、3) 状態遷移定義及び 4) システム構成ツリー定義 の 4 種の情報が必要である。

状態定義は、部品が有する内部状態の定義である。一般的に、部品は状態毎に消費電力や発熱量などの属性値が変化するため、状態定義と共にこれら属性値の定義も行う。これにより、シミュレーションの結果から衛星全体の消費電力や発熱量の時間推移が簡単に求まるため、衛星全体の電力解析や熱解析に対する入力条件を作成する作業を著しく軽減できる。

イベント/モニタ定義は、部品に対して与えられる制御入力 (コマンド) と、部品の内部状態を観測するために外部に出力されるモニタ情報 (テレメトリ) の定義である。テレメトリ・コマンド情報は、初期設計段階は名称 (略称) により、詳細設計段階は衛星と通信するパケットのビットパターンとなるため、イベント/モニタ定義機能もどちらのインタフェースにも対応できるように設計されている。

状態遷移定義は、既に定義された状態とイベント/モニタ定義の関連づけの定義である。具体的には、コマンド項目毎に前状態と後状態との関係を定

義する。

システム構成ツリー定義は、定義された部品群がシステムをどのように構成しているかの定義である。インテグレーション作業の無駄と重複を省くため、一般的に衛星は機能毎にサブシステムに分割されて構成される。(表 1 参照) またシステムの信頼性・耐故障性を向上させるため、同一機能を有した部品を複数搭載し、冗長構成を取ることも多い。このため、衛星システムを指定した場合にどの部品群から構成されるシミュレーションモデルを作成すればよいかをシミュレータに教える必要がある。

また UI 部は、ここで定義された情報を、部品状態定義、テレメトリコマンド情報 (テレコマ DB) としてセーブするが、モデル作成時に扱いやすくするため、同保存形式は XML ファイルとしている。さらに同情報はシステムインテグレータ・部品開発者間で同期すべき情報であるため、PDM (Product Data Management) 等の版管理ツールを用いて保持すべきである。

なお、従来衛星システムにおける設計資産を有効に活用するため、UI 部は従来の文書形式の ICD (Excel 形式) を読み込み本システムの源泉情報にすることができる。

3.2 状態モデル管理部

状態モデル管理部は、定義された部品状態から、システム/サブシステム状態モデルをコンパイル・生成する機能

ブロックである。

本ブロックの中核に据えるエンジン実装方式のトレードオフ結果を表3に示す。

表3 シミュレーションエンジントレードオフ

	SMC ¹	ZIPC	SSF ²
ベンダ	Open source	CATS	ICS(米国)
方式	状態遷移図	状態遷移表	エキスパートシステム
日本語対応	×		×
信頼性	サポート無し	CASE ツールとして 多数実績	複数宇宙プログラム で採用
判定			

まとめると、ZIPC を選定した理由は次の通りである。

<ZIPC 選定理由>

理由1： ZIPC が扱う拡張階層化状態遷移表は、部品/サブシステム/システムと機能階層をなす衛星システムのモデル化に適している。

理由2： 他方式はモデルの不完全さを設計者が検査しにくい。ZIPC は一旦図を状態遷移表に直すので、状態遷移のものを検査しやすい。

理由3： ZIPC は組込用 CASE ツールとして数々の実績のあるCATS 製品であり、信頼性が高い。

ZIPC を用いたシミュレーションモデル変換フローは図3に示す通りである。

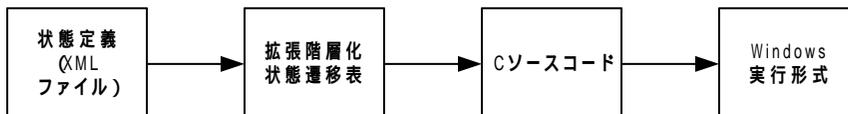


図3 ZIPC による衛星シミュレータモデル生成

3.3 衛星シミュレータフレームワーク部

衛星シミュレータフレームワーク部は、生成されたシミュレーションモデルとイベント/モニタをやりとりし、シミュレーションを実行する機能ブロックである。

シミュレーションエンジンに ZIPC を使用したことを受けて、シミュレーションモデルと外部とのやりとりは、

ZIPC が提供する API を用いて実行される。

UI 部の説明で述べたとおり、モデルとの間のイベント/モニタインターフェースは、設計の進捗に応じて名前からビットパターンへと変化する。このためテレコマ DB に基づく名称とビットパタンの相互変換機能を分離する設計とし、モデルへの入力形式は常に同一となるよう工夫している。(図4参照)

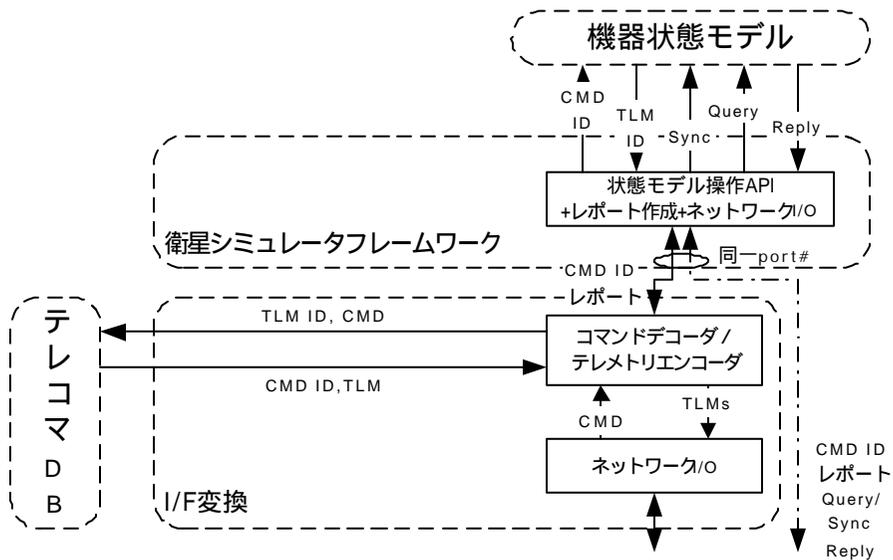


図4 シミュレータフレームワークの構成

3.4 構成・シナリオ管理部

構成・シナリオ管理部は、シミュレータモデルに対して入力するシミュレーションパターン(シナリオ)を生成・維持する機能ブロックである。設計フェーズでは、シナリオは設計者内部情報であるが、将来的には衛星を運用するときの運用手順(宇宙業界ではSOP³と呼ぶ)となるため、構成・シナリオ管理部とシミュレータフレームワークとのインターフェースは衛星管制で定義されているSOP形式に合わせる計画である。

4.ZIPCによるシミュレータプロトタイプ評価結果

これまでに述べた衛星シミュレータ構築のフィジビリティを確認するため、平成13年度4Qにおいて衛星シミュレータプロトタイプの開発・評価を実施

した。ここではその成果を紹介する。

4.1 プロトタイプ開発の目的

プロトタイプは、以下に示すデジタルICDを中心とした衛星シミュレータ構築のイメージアップを目的とし製作された。

- デジタルICDからモデル構築を行う一連の作業のデモンストレーション
- モデルを用いたシミュレーションによる設計検証作業のデモンストレーション

4.2 プロトタイプの構成

図5にプロトタイプの構成図を示す。

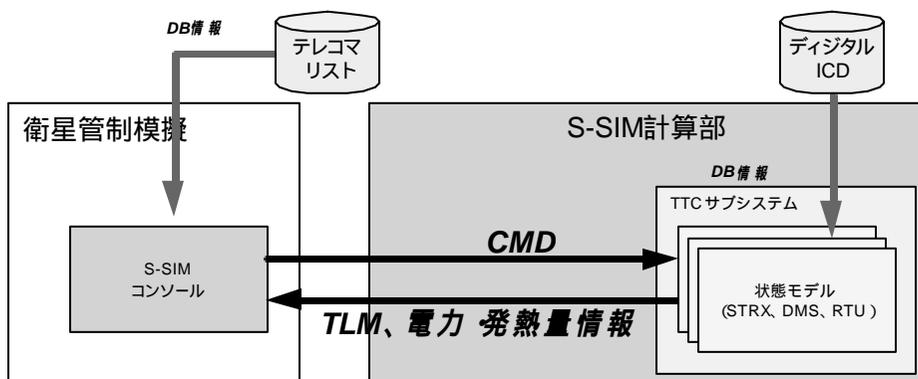


図5 衛星シミュレータプロトタイプ構成図

プロトタイプは、デジタルICD からモデル構築までの一連の作業をデモンストレーションするため、1) 入力インタフェース (UI 部)、2) シミュレーションモデルへの変換 (状態モデル管理部)、3) シミュレーション (衛星シミュレータフレームワーク) の簡易版を備えている。

状態モデル管理部が、複数部品モデルの連携によりサブシステムをモデル化できることを示すため、通信装置 (STRX)、バス制御装置(DMS)、テレメトリコマンドインタフェース装置 (RTU) の 3 種 6 装置から構成される

テレメトリコマンド (TTC) 系をモデル化した。

また他の解析システムとの連携が図れることを示すため、電力解析を題材にとり衛星シミュレータが生成する消費電力の時間推移と電力解析システムが生成する発生電力の時間推移をおなじ画面に表示し、評価する機能も備えている。

4.3 プロトタイプによる試行評価フロー

図6に従い、プロトタイプシステムによる試行評価フローを説明する。

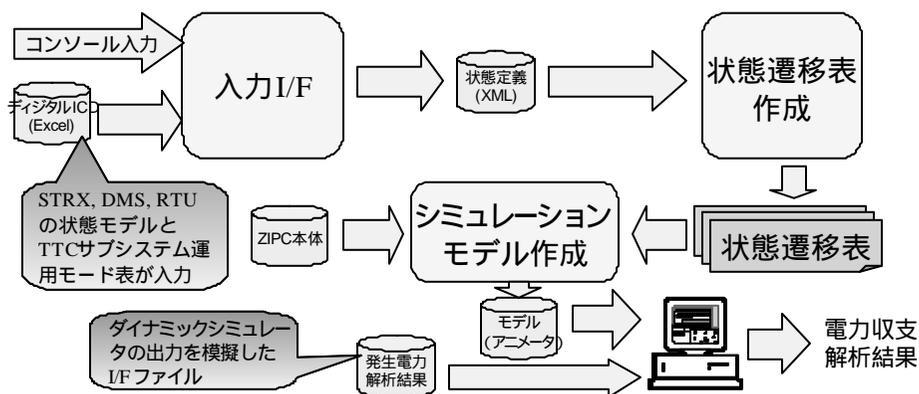


図6 プロトタイプによる試行評価フロー

プロトタイプによる試行評価フローは、以下に示す 3 つのステップから構成される。

- **ステップ1: 文書形式ICD を読み込み、XML 形式デジタルICD に変換**

従来設計資産を生かすため、文書形式(Excel形式)のICD(電力・モード定義表、コマンドリスト、テレメトリリスト、TTCサブシステム運用モード表)を読み込む。文書形式ICDから読み込んだモード定義を“状態”として、電力を“状態の属性”として、コマンドを“イベント”として、テレメトリを“モニタ情報”として、UI上からそれぞれ関連付ける。

全ての入力作業が完了すると、利用者からの指示に従いXML形式デジタルICDとしてセーブする。

- **ステップ2: XML形式ICDからシミュレーションモデルを作成**

デジタルICD作成後、ファイル変換ツール(図7)を用いてシミュレーション用モデルを作成する。本ツールの起動により図3で説明したフローに従いシミュレーションモデルが作成される。中間生成物である状態遷移表を確認したい場合は、ZIPC本体を起動し、その画面から確認する。



図7 ICD モデル変換ツール画面

このインターフェースは、将来運用制約になりうる禁止モード遷移や、予測不能な動作の元となる未定義モード遷移などの早期発見と、是正に大いに貢献する。

- **ステップ3: シミュレーションによる設計検証**

図8にシミュレーション実行画面を示す。

TTCサブシステムへのコマンド(本画面ではコマンド送信ウィンドウでのボタンになっている)によりシミュレーションモデルが動作し、その結果が右の消費電力カーブ及び、左のテレメトリモニタ画面へのアンサバックに反映される(同図では、ボタンのハイライト表示によりON/OFFが表現される)本インターフェースを用いて、設計者が期待するモード遷移が外部からのコマンドにより実現できるかを検証することができる。

この検証のために設計者が入力したシミュレーションパターンは、運用におけるサブシステム運用手順(SOP)の源泉⁴となる。

またシミュレーションの結果得られた消費電力を、電力解析システム出力とあわせて表示するなど、他の解析システムとの連携による高度な設計検証を行うこともできる。

4.4 試行評価結果

プロトタイプによる試行を通じて、デジタル ICD から衛星シミュレータを構築するという方式のフィジビリティが確認できた。具体的な成果は次の通りである。

- (1) 初期設計段階でシミュレーションによる設計検証を行うことにより、「禁止コマンド」や「無意味コマンド」などの抽出が可能となり、運用フェーズに運用制約やリスクを持ち越さない設計の役に立つことが明らかとなった。
- (2) 衛星シミュレータで共通的に使用するデジタルICDのXMLボキャブラリの設計妥当性についても検証することができた。

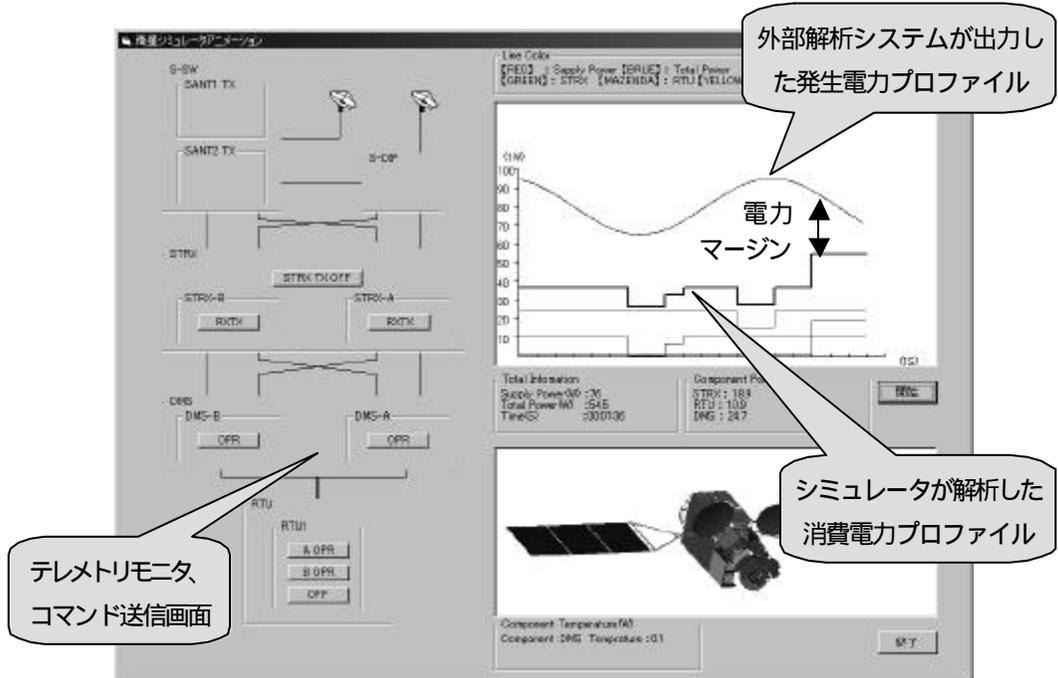


図8 衛星シミュレータプロトタイプシミュレーション画面

5. まとめ

本論文では、人工衛星システム開発における ZIPC の適用を、デジタル ICD から衛星シミュレータを構築する過程を通じて紹介した。

本方式により、従来紙による情報伝達に過ぎなかったシステムインテグレータと部品設計者との間に、デジタルモデルという“検証可能な”情報伝達手段を設けることができる。

これは衛星システム開発に代表される大規模コンカレント開発プロセスにおいてキーとなる上流工程における設計検証手段を提供するものであり、近年浸透してきた³ 次元設計手法による熱機械インテグレーションと同様機能インテグレーションの分野でも高度情報化手法による設計品質の向上、並びに開発の低コスト・短納期化に大きく貢献すると期待できる。

今後は外部からのコマンド入力ではなく外的環境との関係で動的に振る舞いを変える“衛星のダイナミクス”に関するシミュレーション機能も含めて、衛星シミュレータのシステム化を進めていく計画である。

6. 謝辞

本作業は、宇宙開発事業団委託業務契約「超高速インターネット衛星通信網実験システム初期設計」(NASDA-PSPC-27707)のもと実施された。

7. 参考文献

1. 『Extensible Markup Language (XML) 1.0 (Second Edition)』, W3C Recommendation 6 October 2000
2. 『拡張階層化状態遷移表設計手法』, 東銀座出版社
ISBN4-89469-004-7
- 1 State Map Compiler
- 2 Spacecraft Simulator Framework
- 3 Spacecraft Operation Procedures
- 4 運用で使用する SOP には、コマンドの送信間隔やコマンドが正しく動作したことを確認するためのテレメトリの値など情報を付加する必要がある。