

衛星管制ソフトウェア“Birdstar”の海外製衛星への適用

石井俊直*
原内 聡**
小林和貴*

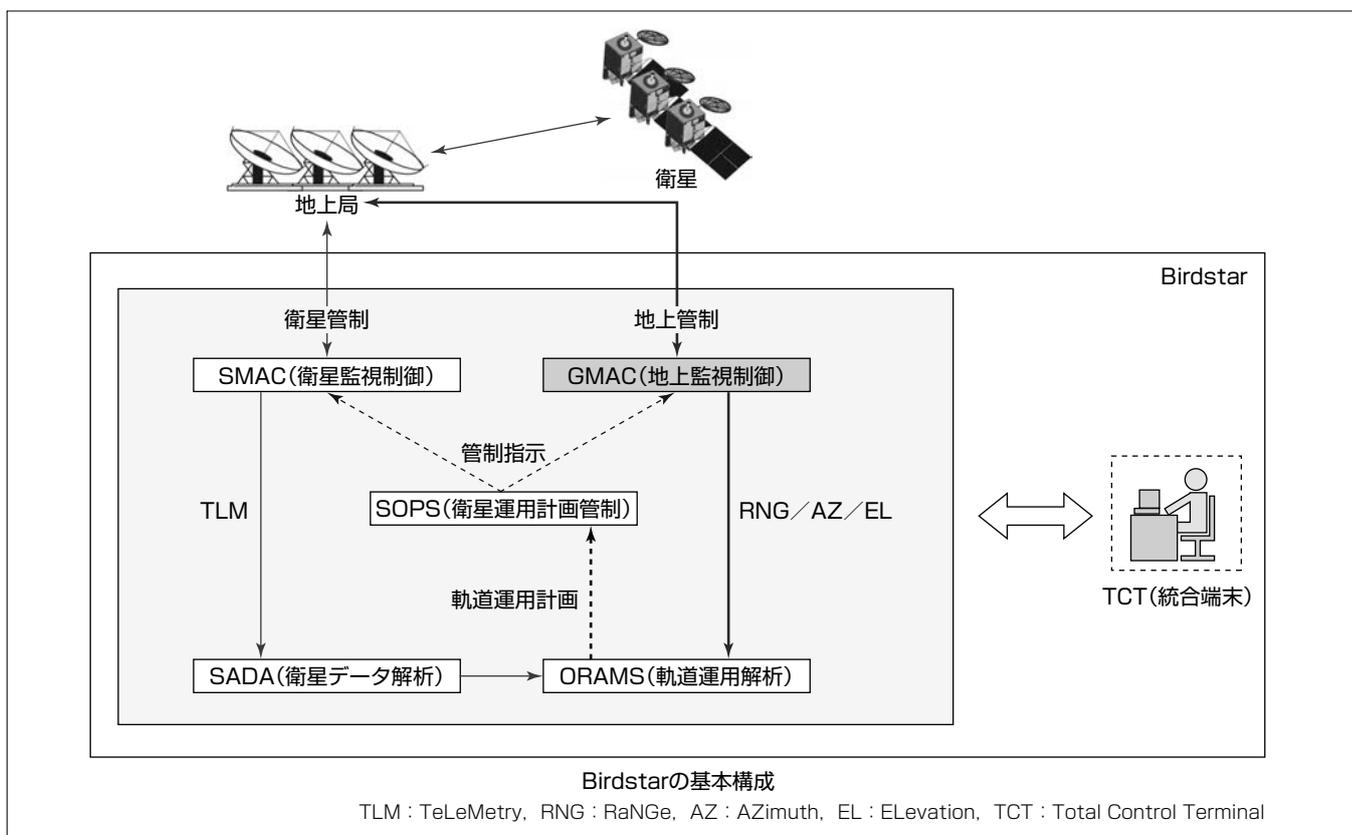
Application of Satellite Control Software "Birdstar" to Foreign-made Satellite

Toshinao Ishii, Satoshi Harauchi, Kazuyoshi Kobayashi

要 旨

衛星管制ソフトウェア“Birdstar”は、これまで気象衛星ひまわり8号・9号に代表される三菱電機製人工衛星の多くに適用してきた。今回、海外製衛星向けの衛星管制システムに適用して初の海外製衛星適用となった。衛星管制ソフトウェアは、テレメトリの送受信指示、運用計画立案、軌道運用計画などの機能を持つ。基本的な機能は変わらないが、衛星ごとにカスタマイズする必要がある。この

案件では初の海外製衛星適用であることに加え、顧客もBirdstarを初めて採用するため、ソフトウェアへの要求が開発初期段階で出そろわず、手戻りも含めた開発量が増大すると予想した。そこで、ソフトウェア開発で一般的になりつつあるアジャイル開発の手法を取り込んで開発を行うことによって、変化を許容して対応した。



衛星管制ソフトウェア“Birdstar”

海外製衛星の管制システムにBirdstarを初適用した。衛星管制に必要な基本機能を適用させるとともに、全サブシステムの英語化とユーザビリティの向上を図った。

1. ま え が き

これまで当社では、当社製人工衛星の地上設備だけを製造してきた。自社製の衛星管制システムを、他社製の人工衛星に適用した実績がなかったが、今回海外衛星メーカー製造の人工衛星の管制システムを受注し、初の海外メーカー製人工衛星に適用することとなった。実績のある気象衛星ひまわりや準天頂衛星など公官庁向けとは異なる民間会社による衛星管制運用向けの機能拡張やユーザビリティ向上を行った。

本稿では、今後Birdstarを拡販していく上でマイルストーンとなるこの開発のプロセスについての分析結果を述べる。

2. システム構成

衛星管制システムは、主に“変復調設備”と“計算機システム”で構成される。変復調設備は、高周波設備(アンテナ)を介して衛星に対して、衛星コマンド送信や衛星テレメトリの受信を実施するために、信号の変復調を実施する設備である。計算機システムは、衛星管制システム用Birdstarのソフトウェア群を用いて、衛星へのコマンド送信指示やテレメトリ受信、スケジュール管理、軌道運用計画を実施するシステムである。システムの各要素は冗長構成を基本とし、障害発生時に衛星管制運用に支障をきたさない構成となっている。

3. Birdstar

Birdstarは、当社が持つ衛星管制ソフトウェアパッケージの標準基盤(総称)である。今回は、コマンド送信、テレメトリ受信、レンジング、スケジュール管理を行うオンライン機能と、データ蓄積・解析、軌道運用計画を行うオフライン機能を備えており、次の4種のソフトウェア(サブシステム)で構成される。

- (1) 衛星監視制御を行うSMAC(Satellite Monitor And Control software)
- (2) 衛星データ解析を行うSADA(SATellite Data Analysis software)
- (3) 衛星運用計画管制を行うSOPS(Satellite Operation Planning & Scheduling software)
- (4) 軌道運用解析を行うORAMS(ORbitAl Mission analysisS software)

Birdstarは、今まで運輸多目的衛星ひまわり7号(2006年打ち上げ)^①を始め、当社製衛星の管制システムの多くに適用してきた実績がある。特長として、衛星運用や計画、解析手順を、シナリオを用いて半自動で実行でき、運用者の操作ミスや操作負担を低減できることが挙げられる。システムはサーバ&クライアント方式を採用し、システム全体を端末1つで操作できる。また全サブシステムの画面・操作に統一感を持たせることで、高いユーザビリティを備えている。

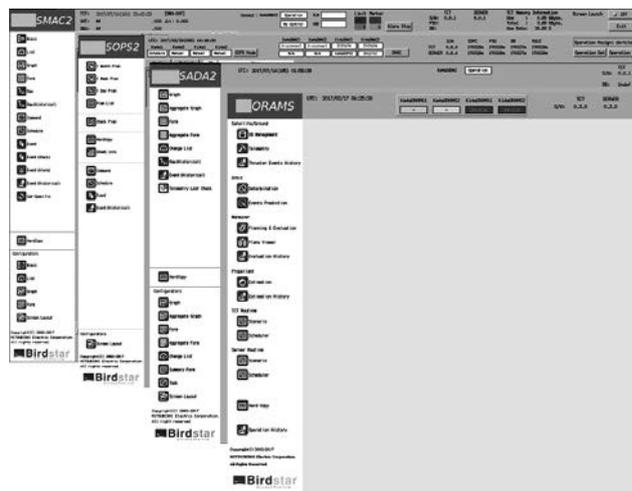


図1. Birdstarの各サブシステムのメイン画面

図1は、Birdstarの各サブシステムのメイン画面を重ねた図である。統一的な構成として、左上にロゴ、上部に各種ステータス表示、左に各種機能を開くアイコンがある。

4. 新機能と開発経緯

本稿では、まず、開発の全体を紹介し、次にシステム開発で大きな課題であった“テレメトリ処理の課題と対策”そして“ユーザビリティの向上”について述べる。理想的なウォーターフォール開発プロセスであれば、開発仕様の特定は要求分析の段階で完了するべきである。しかしながら、このシステムの開発ではそれは困難であった。また、この状況は一般的になっていて、対応策が重要な課題となっているものである。

4.1 開発の全体

この開発は、当社衛星向け機能のベースから、海外メーカー製衛星の管制を行うシステムを、COTS(Commercial Off The Shelf)品である衛星モデム・計算機・ネットワークなどのインテグレーションを含めて構築することであり、受注から納品までのおよそ17か月で行われた。ソフトウェア開発の観点で分類すると次の4つのフェーズに分けることができる。

第一フェーズ(約4か月)では、衛星メーカーから衛星仕様に関する文書が提供され、これを基に作成した衛星管制システム仕様書を作成し、顧客による承認を受けた。ただ、この段階で当社と衛星メーカーが直接会話をするのではなく、ドキュメント及び当社が契約した衛星コンサルタントからの情報だけで衛星管制システムの設計を行った。

第二フェーズ(約6か月)では、衛星メーカーに当社衛星管制システムの仕様の開示・説明を行い、直接お互いの疑問点について明らかにするプロセスが始まった。

第三フェーズ(約4か月)では、衛星メーカーに衛星管制システムをパソコン1台に集約した簡易版を提供し、衛星メーカーで運用リソースの開発が始まった。また、顧客も衛星管制システムを実際に試用する機会が増えていった。

第四フェーズ(約3か月)では顧客施設に設備を構築、運用トレーニングの提供、顧客による運用リソースの開発が開始された。

次に、通常のウォーターフォールモデルでは第一フェーズで決定すべき仕様でありながら、仕様決定できなかった事項について、課題と対策について述べる。

4.2 テレメトリ処理の課題と対策

4.2.1 テレメトリ・コマンド方式

(1) 課題

衛星管制のリアルタイム処理を担うSMACの機能は大きくテレメトリ処理(衛星が送信する衛星の状態を受信して解釈する)と、コマンド処理(制御するコマンドを衛星に届ける)に分けられる。初めての衛星であったため、第一フェーズから次の対応を必要とした。この衛星のコマンド処理は、BirdstarがサポートしているCCSDS(Consultative Committee for Space Data Systems)と呼ばれる標準方式とは異なる方式であり、そこで従来のコマンド処理を流用する段階でコスト及び期間的な課題が発生した。一方、テレメトリ処理はインタフェースこそBirdstarがこれまでサポートしてきたCCSDS方式であるが、様々な点で基本的な考え方が異なっており、早急な対応が課題となった。例えば、衛星の機器を特定するテレメトリのIDがこの衛星では冗長系で同一であった。また衛星内通信経路が複数あり、重複したIDのため地上では衛星機器を判別できなかった。一方でテレメトリID体系が複雑になると運用者の負担を増大することが問題となった。

(2) 対策

地上でのテレメトリID体系に工夫を加えることでこの問題に対応した。例えば衛星メーカーの定義したIDがAB1234の場合、経路を表す番号xを導入して経路ごとに0xAB1234を待機系のID、1xAB1234を運用系のIDなどテレメトリIDについてプレフィックスを付けて細分化した。衛星定義のIDであるAB1234は全ての経路から受信したデータ運用系テレメトリ(1xAB1234)を集約したデータとした。これによって、定常時は、細かな区別を運用者が意識する必要がないので衛星定義IDを使い、経路や系の区別が必要なときは、プレフィックス付きのIDでデータを区別できる。

4.2.2 リアルタイム統計処理

(1) 課題

従来のBirdstarのテレメトリ定義フォーマットには、衛星メーカーが定義する工学値変換(衛星から受信した生データを運用者が理解しやすい形に変換すること)のうち、リアルタイム統計処理を定義するための仕様が含まれていなかった。

(2) 対策

この件に対しては、まず、Birdstarが別途使用している合成テレメトリと呼ぶ種類の定義フォーマット形式でその一部のセマンティクス(解釈)を変更することによってデータ定義インタフェースを実現した。インタフェースが流用

であることで、データ定義を行う衛星メーカーに想定外の追加を受け入れてもらった。

さらに課題であったリアルタイム統計処理の実現は、専用のサーバプロセスを追加することで実現した。Birdstarは基本アーキテクチャがクライアント・サーバ型であり、プロセス間インタフェースがフレームワーク化されているため、こうしたプロセスの追加といった拡張は容易に行えるような設計になっている。

リアルタイム統計処理は、第一フェーズでは想定できなかった機能であったが、設計チームの力が有限である以上、常に起こりえることであり、完璧な想定を第一フェーズで仕上げるよりも、ある程度の想定外を認めて、それに対応してゆく方がトータルのコストが低い場合もある。

4.3 ユーザビリティ向上の課題と対策

第一フェーズではスクリーンキャプチャとその説明などによる仕様確定を行ったが、第二フェーズの終盤から実運用者が使用する機会が徐々に増えた結果、顧客からユーザビリティ改善に関する要望が多く出るようになった。今回の開発では多面的に検討した結果、要望のほぼ全てに何らかの形で対応した。

4.3.1 形・色・音のカスタマイズ

(1) 課題

ユーザビリティ上の要望で最も項目数が多かったのが、グラフィカルユーザーインタフェース(GUI)周りの形・色・音に関するものであった。ユーザーの特徴や経験などによってユーザビリティの高さの形態が変わってくる。

形のカスタマイズで修正が多かったのは、文字表示数に関連するものであった。表示文字は時刻や“O.K.”などの画面ソフトウェアが定義しているものに加え、衛星メーカーが定めるテレメトリ定義やコマンド定義からのものがある。衛星メーカーが定義した表示すべき文字数が想定よりも大幅に多いものであった。

色に関する要望も少なくなかった。例えばイベント画面と呼ばれる情報の表示画面では、イベントの種類や、失敗か成功か又は正常か異常かといったイベントの内容による色分けを細かく行う必要があった。また、音声も運用者がカスタマイズしたいという要望もあった。

(2) 対策

文字表示数に対しては、設計段階で最大20文字としていたものの、最終的に100文字まで表示できるように画面レイアウトを変更した部分もある。

図2に色に対する対策後のユーザーインタフェースのイベント画面の例を示す。情報の種類ごと、同種情報でもその内容に応じて表示色を細かくカスタマイズ可能とした。さらに、画面の色をカスタマイズする仕組みもユーザーに開放した。これに伴い色カスタマイズリソース(設定ファイル)を運用環境に配信するためのツール整備も行った。なお、色のカスタマイズは、図に示したイベント画面だけ

No.	日時	運用管理タイプ	その他	〇〇運用チェック	Normal
3	2017/10/18(水) 09:08:00.640	操作			
4	2017/10/18(水) 14:58:00.416	操作	PT追加	PLANNING TIME_ADJUST	
5	2017/10/18(水) 15:00:00.421	操作	実行開始	計画 PLANNING TIME_ADJUST	
6	2017/10/18(水) 15:00:00.422	判定	OK	STP001(DLTR_1DPE)	STP001 02 asac
7	2017/10/18(水) 15:00:11.811	コマンド	[A]	SHT (R) 1 EPH8789 TIM_CAL 3/79	
8	2017/10/18(水) 15:00:11.837	コマンド	OK	SHT (R) 1 EPH8789 TIM_CAL 3/50	
9	2017/10/18(水) 15:00:14.264	コマンド	OK	SHT (R) 1 EPH8789 TIM_CAL	
10	2017/10/18(水) 15:00:20.200	判定	OK	STP001(DLTR_1DPE) < 125 < STP001 < 125 2 asac	
11	2017/10/18(水) 15:00:26.201	操作	正常終了	計画 PLANNING	

従来の表示



直感的区別を可能にする細かい色分け：
アラーム行は、背景色で正常と異常を区別して強調。その他イベントも、種類と内容によって固有の文字色で表示。

細部まで配色カスタマイズが可能な表示

図2. ユーザーインターフェースのイベント画面例

ではなく、テレメトリプロット(グラフ)などについても同様に運用者によるカスタマイズ可能範囲が大幅に広がった。

管制システムでは、瞬間的に運用者に気づいてもらう必要があること(イベント)には、画面表示に加えて音声を鳴動させるのが一般的である。Birdstarではプリセットとなっている8種類の音声をイベントに関連付けて鳴動する仕組みがある。このシステムでは、色のユーザーカスタマイズ同様、音声ファイルの内容を運用者が変更するためのツールも併せて整備した。

4.3.2 ユーザーマクロ機能の充実

(1) 課題

衛星の制御は個々のコマンド送信とテレメトリの判定を組み合わせた手順である。衛星制御手順は衛星メーカーが指定するが、衛星管制ソフトウェアはこうした手順を実行する仕組みを持っている。Birdstarではコマンドと判定等を論理的に並べたものをPIU(Procedures and Instructions Unit)と呼んでいる。インタープリタで実行処理されるPIUには、コマンドを衛星に送り、それに伴ってテレメトリが変化するのを待って判定し、次に移るといったパターンが多く登場する。従来のPIUでは一定時間(例えば5秒)の待機処理(sleep)を行い、その後テレメトリを判定するという形で記述するのが一般的であった。このやり方で現実の衛星制御手順を全て記述できる。ところが顧客からこのやり方では、判定ロジックを個々に記述する必要があり、処理に必要な時間も長くなってしまったため、もっとスマートな方法を実現してほしいとの要望があった。これは設計者視点では見落としてしまう視点であった。

(2) 対策

テレメトリの判定には“waitMax”というマクロを付けることとなった。この指定によってテレメトリが条件を満たすかど

```
n=0
Block 1
n=n+1
check AB1234==OK
if TRUE break
else
if 5<n break
sleep 1
goto Block1
end
```

従来コード



```
check AB1234==OK waitMax 5
waitMaxを使用したコード
```

図3. PIUのユーザーマクロ機能waitMax

うかの判定を最大で指定された時間だけ待機させる処理が組み込まれる。これによって、判定ロジックの記述が不要となり、しかもテレメトリが所望の値に変化した段階で判定が完了するため処理にかかる時間も短くすることができた(図3)。

これは、カスタマイズの域を超えて要望に取り組むことでより良い新機能を見つけることができた例であった。

5. む す び

本稿では、海外製衛星に適用した衛星管制システムについて述べ、このシステム向けのソフトウェア開発プロセスについての事例について述べた。衛星の仕様は当社衛星とは多くの点で異なっていたが、Birdstarの基本機能と非互換な改造はなく、全体的には従来の機能を活用して新しい衛星に対応するシステムを構築することができた。衛星が変わったことで入れ替えられた機能モジュールはなく、これについては、Birdstarの新規開発当初の基本的な設計の先見性によるところが多い。海外製衛星に初対応できたことは、今後拡販していく上でのマイルストーンとなるだろう。

機能の変更・追加の必要性が開発の中盤以降に特定された事例を述べたが、ユーザビリティ改善としてそれらは現実的には避けられなかった。一方で、開発の後期にも課題が発生することをあらかじめ折り込み、ソフトウェア開発で一般的になりつつあるアジャイル開発の手法を使用し、積極的に課題に対応することで得られた成果もあった。衛星運用の専門家であり経験豊富な顧客が徐々に感じた要求を取り込むことで、今までにない視点から見直すことができ、Birdstarのユーザビリティは確実に向上した。設計審査会での設計どおりに開発されたならばこのような進化はなかった。今回のシステム開発では、顧客の要求変化や追加は対応しなくてはならないという以上に、変化に前向きに対応することがより高い品質、ひいては、顧客の満足につながることを感じた開発であった。

参 考 文 献

- (1) 三菱電機株式会社ホームページ：ひまわり7号(MTSAT-2)
<http://www.mitsubishielectric.co.jp/society/space/satellite/observation/mtsats2.html>