

すばる望遠鏡中間赤外撮像分光装置用制御ソフトウェアの開発

中村京子, 宮田隆志¹, 片堅宏一²

(2001年4月2日受理)

Development of Control Software System for Cooled Mid-Infrared Camera and Spectrometer on the Subaru Telescope

Kyoko NAKAMURA, Takashi MIYATA¹, and Hirokazu KATAZA²

Abstract

We have developed a control software system for the Cooled Mid-Infrared Camera and Spectrometer (COMICS) use at the Subaru telescope which is located at the summit of Mauna Kea in Hawaii island. The Subaru telescope is controlled by several software systems, in which SOSs (Subaru Observation Software system) supervises observations. CSC (Comics Software Control system) realizes series of observations in cooperating with SOSs. CSC provides GUI to control COMICS for developers and observers. We developed CSC using the same machine and operating system as those of SOSs but using different GUI which is based on the X window system (Xtoolkit).

CSC uses several communication methods in order to control equipment and to communicate with SOSs. The RPC protocol is used for communications with outside workstations, however functions such as pipe, signal and events are used for communications with inner processes.

1. はじめに

中間赤外撮像分光装置 COMICS¹⁾ (COoled Mid-Infrared Camera and Spectrometer) は、ハワイ島マウナケア山頂に設置されているすばる望遠鏡²⁾の第一期共同利用観測装置の一つであり、1999年12月より試験観測を開始した。本稿では COMICS の制御ソフトウェア CSC (Comics Software Control system) の開発について述べる。

すばる望遠鏡山頂観測システムの構成を図1に示す。なお図1では本稿に関連するシステムを点線で囲んだ。全体システムをすばる観測所運用システム (Subaru Observatory System) と呼ぶが、この下に、すばる観測ソフトウェアシステム SOSs³⁾ (Subaru Observation Software system) とすばる解析システム、すばるアーカイブシステムの三つが並列で存在している。SOSs は望遠鏡制御システムと観測装置システムとのインタ

フェースを持ち、観測全体を管理するシステムである。特にすばる望遠鏡は複数の観測装置を有するため、観測装置システムに共通のインタフェースを提供し、どの観測装置に対しても変わらない観測制御を実現している。CSC は観測装置システムに相当し、観測装置 COMICS を制御し、また SOSs と共同で一連の観測をも制御する。

CSC は SOSs との連携を重要視し、通信の確実性のために SOSs 側の計算機とほぼ同じ環境で開発した。すなわち、計算機は Sun ワークステーションを使い、OS は Solaris 2.5.1、プログラムは C 言語で記述し、コンパイラは SPARCompiler 4.2 を使った。また GUI は、C 言語および UNIX システムと相性の良い X ウィンドウシステムを採用し、X ツールキットで作成した。

2. ハードウェアシステム構成

2.1 COMICS システムの構成

COMICS のハードウェア構成を図2に示す。COMICS は、8~26 μ m の中間赤外線域撮像分光装置であり、VMEbus に接続した複数枚のボードが直接制御する。VME ボードの種類は、VME ボード計算機 wiw-

¹ 東京大学大学院理学系研究科天文学教育研究センター (Institute of Astronomy, The University of Tokyo)

² 宇宙科学研究所 (The Institute of Space and Astronautical Science)

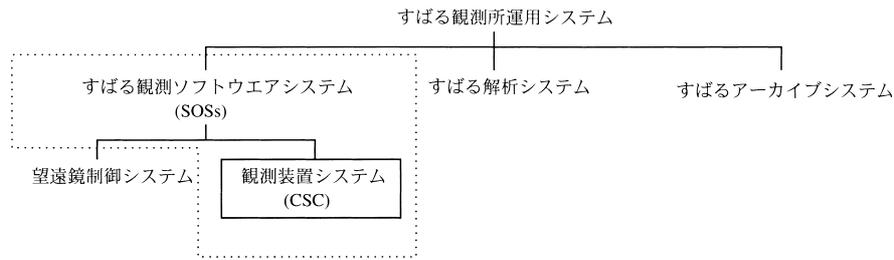


図 1. 山頂観測システムの構成.

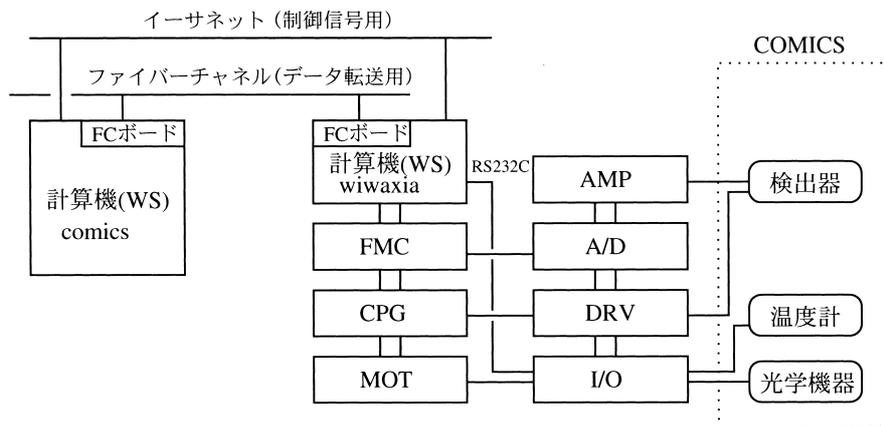


図 2. COMICS のハードウェア構成.

axia (OS は Solaris 2.5.1), 6 つの検出器各々に対応する 6 枚のクロック・パターン・ジェネレータボード (CPG), 撮像/分光各々に対応する 2 枚のフレーム・メモリ制御ボード (FMC), モータ制御ボード (MOT) である. wiwaxia は CPG や MOT を制御し, さらに RS232C を通じて検出器まわりの複数箇所に設置されている温度計も制御する. これらのボードは一つのラックに納められているが, 別にもう一つのラックがあり, それには検出器ドライバーボード (DRV), アンプボード (AMP), A/D 変換ボード (A/D), 入出力ボード (I/O) が接続されている. CPG はクロックを生成し, DRV を通じて検出器を駆動する. 検出器が取得したデータは AMP へ送られ, A/D 変換された後, FMC に蓄えられる. また, MOT は I/O を通じてフィルターやスリット, レンズ, 入射平面鏡, 回折格子の光学系機器を駆動制御する.

ところで, これら 2 台のラックは COMICS 本体の脇に据え付けられており, 望遠鏡のカセグレン焦点部に存在する. したがって観測者が直接操作する計算機は, wiwaxia と同一イーサネット上に存在する計算機 comics (OS は Solaris 2.5.1) になる. CSC は comics と wiwaxia にまたがるソフトウェアシステムで, comics が主となってイーサネットを通じ wiwaxia に制御指令を出す. また両者は共に Ancor 社製 Fibre Channel SBus 266/1062 ボードを実装しており, 検出器で取得した観測データはファイバーチャネルを使って comics へ転送され, さらに山頂システム上のデータベースへ送られ, 格納される.

2.2 山頂観測システムの構成

すばる望遠鏡の観測に関する山頂ネットワークの構成を図 3 に示す. すばる望遠鏡の観測ソフトウェアシステム SOSs は観測制御ネットワーク上にあり, OBS と呼ばれる計算機が全体を管理している. またこのネットワーク上の計算機 OBC は, 主に観測データのデータベースへの転送管理を受け持つ. 望遠鏡制御システムは望遠鏡制御ネットワーク上に位置し, TSC と呼ばれる計算機が管理している. このネットワークは望遠鏡ドームまで伸びており, TSC はこれを通して望遠鏡やドームなどの駆動制御, 温度・湿度や風速などの環境情報の収集を行う. 一方, TSC は観測制御のためのインターフェースを設けており, これにより OBS は望遠鏡制御を行うことができる. 観測装置システムは, OBS から伸びるサブネットワークである観測装置ネットワーク上に存在する. OBCP と呼ばれる計算機がその管理計算機であり, OBE と呼ばれる計算機や観測装置を制御する.

前節で述べた COMICS システムをこの山頂ネットワーク構成に当てはめると, 計算機 comics は OBCP に相当し, VME ボード計算機 wiwaxia は OBE にあたり, CSC は OBCP と OBE に渡るシステムである. 観測の時には, OBCP である comics は OBS と通信を行いつつ OBE である wiwaxia を通じて COMICS 本体を制御し, 観測で取得し comics のハードディスクに蓄えられたデータは, ファイバーチャネルを通して OBC へ, さらには観測所のデータベースへ転送される.

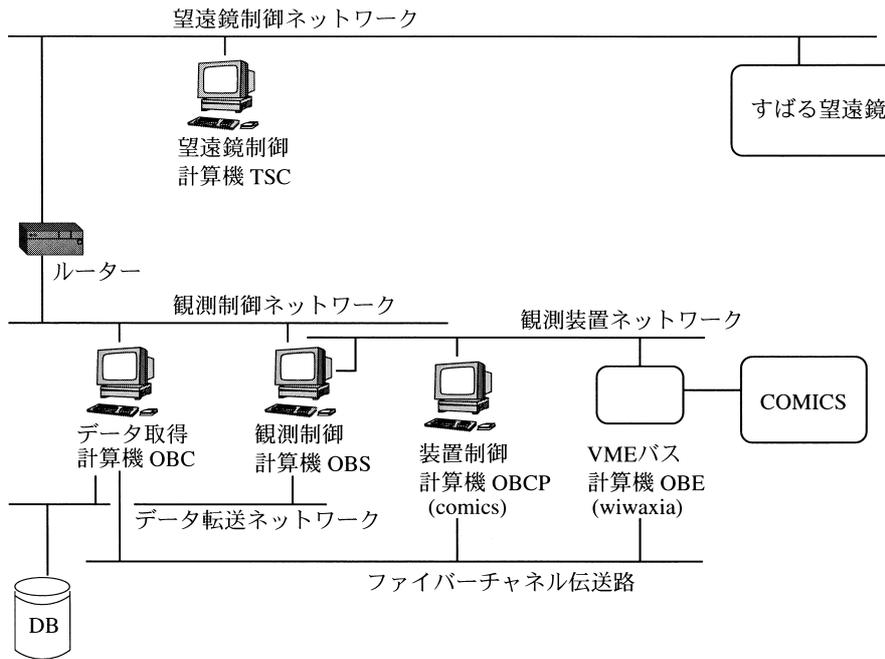


図3. 山頂ネットワークの構成。

3. 制御ソフトウェアに要求される機能

この節では、観測における COMICS のソフトウェア制御に必要な項目、そして SOSs と共同で実現する機能について述べる。

3.1 COMICS 制御項目

観測時に必要な COMICS のソフトウェア制御項目一覧を表 1 に示す。

観測の時に駆動する必要がある光学系機器は、フィルター、スリット、レンズ、入射平面鏡、回折格子の 5 つである。フィルターは前光学系に 2 カ所、撮像系に 1 カ所設置されており、それぞれ複数個からフィルターを選択できるため、それらを適時選択交換できるようにする。スリットやレンズも同様に複数個から選択して交換する。平面鏡の場合は光学系へ正確に光が入るように、入射光の反射角度を変えられるようにする。回折格子は複数個からの選択と交換のほかに、回転角度を調整できるようにする。以上は光学系機器の駆動内容であり、すべて GUI を使って操作できるようにする必要がある。また、機器の交換や調整にあたっては、現在どういう状態であるかの情報も必要のため、現在の状態を読み取り GUI に表示する機能と、変更の都度 GUI に自動表示する機能の二つが必要である。さらにデューワー内の各所に設置されている温度計の情報を、一定間隔で GUI 上に自動表示することも求められた。

一方検出器制御については、COMICS は撮像用に 1 個、分光用に 5 個、計 6 個の CCD 検出器を持っており、撮像/分光で互いに異なるクロックを使うため、クロックパラメータは別個に設定する必要がある。なおクロックの生成は別に作成したスクリプトファイルが行うが、

表 1. COMICS ソフトウェア制御項目一覧。

制御対象	制御内容
光学系機器制御	
前光学系フィルター (2カ所)	現フィルター名表示と交換
撮像系フィルター	現フィルター名表示と交換
スリット	現スリット名表示と交換
レンズ	現レンズ名表示と交換
入射平面鏡	現在角度表示と角度調整
回折格子	現回折格子名表示と交換、および回転角度表示と調整
温度	デューワー内複数箇所の温度表示
検出器クロック制御	
クロックパラメータ	入力および設定変更
スクリプトファイル	選択および実行

複数あるファイルから適切なものを簡単に選択できるようにする。

ところで観測は標高 4,000 メートル強の山頂で行うため、人間の判断能力や作業効率は地上に比べて大幅に低下する。したがって GUI の作成にあたっては、人間の操作ミスを見越した安全対策をとること、そしてわかりやすく単純な操作で済ませるような方式をとる必要がある。

3.2 観測制御システムと共同で実現する機能

COMICS が SOSs と連携し共同で実現する制御機能は、前節の COMICS に対する制御を OBS からの通信を契機としても行えるようにする機能であり、また観測したデータを観測所データベースに登録するための機能である。それらの実現のために 4 つの通信機能が SOSs に

表 2. SOSs が提供する通信機能一覧.

項目	要求元	要求先	内容
ステータス転送機能	CSC	SOSs	観測装置のステータスを送信
ステータスデータ取得機能	CSC	SOSs	望遠鏡ステータスデータの取得要求
取得データ転送機能	CSC	SOSs	観測データのデータベース転送要求
コマンド通信機能	SOSs	CSC	観測装置制御要求

より提供されている。その一覧を表 2 に示す。

「ステータス転送機能」とは、観測装置システムから SOSs へ、観測装置のステータスを送信する機能である。COMICS では CSC が現在稼働中であることを通知するために使用している。SOSs は観測装置システムから一定間隔で送信されるこの信号を監視しており、信号が途絶した場合、観測装置システムが稼働していないと判断し、通信を切断するという仕様になっている。したがって CSC はこの機能を使って、観測の開始から終了まで、一定間隔で連続して信号を送出する必要がある。

「ステータスデータ取得機能」とは、SOSs から望遠鏡側の情報を受け取るための機能で、COMICS では、観測全般に関する情報と、作成する FITS ファイルに与える通し番号の 2 種類を要求する。観測全般に関する情報とは、観測者名・望遠鏡の方向・ロータ角度・温度・湿度・風速などであり、FITS ファイルのヘッダ部に書き込む情報である。COMICS では一夜に一回取得するだけでよい 5 個の情報、観測対象ごとに取得する必要がある 19 個の情報、露出ごとに取得する 13 個の情報の 3 種類に分け、一回に取得する情報量が多くなりすぎないようにしている。一方、作成した FITS ファイルには一意の通し番号がふられるが、この番号は SOSs が管理している。CSC は観測の直前に開始番号を受け取り、FITS ファイル作成の度に数字を加算し、その番号を FITS ファイルのヘッダ部に書き込む。

「取得データ転送機能」は、作成した FITS ファイルを観測所のデータベースに転送する機能である。COMICS の場合、転送の頻度は基本的には一日に一回で、観測データは一旦計算機 comics のハードディスクに格納し、一夜の観測が終了した後でまとめて転送する。本来は一天体観測終了後にデータベースに転送するほうが望ましいが、中間赤外の観測は露出時間が短いゆえに観測制御を頻繁に行う必要がある。そのために時間のかかるデータ転送を観測中に行って、観測効率を下げることは回避したかったため、このような方式をとっている。

「コマンド通信機能」は観測における中心機能であり、このコマンドは望遠鏡制御と観測装置制御の両方へ送られる。コマンドの種類には、装置共通コマンドと装置依存コマンド、さらに一連のコマンドを組合せ、また条件判断も盛り込んで記述できる抽象化コマンドがあり、共

通コマンド以外は観測装置側が装置制御がやりやすいように、作成登録しておくコマンドである。なお、ほかに望遠鏡制御用コマンドもある。これらのコマンドは、OBS から一つずつ実行することもできるし、あらかじめ観測手順書に記述しておき、この手順書によりまとめて実行することもできる。COMICS では観測の効率化のために、望遠鏡の駆動要求から CSC への制御要求までの観測に必要な一連のコマンドを観測手順書に記述し、OBS に登録している。OBS は観測手順書に書かれているコマンドを順に送出し、観測制御を行う。

観測制御の主体は SOSs であり、CSC は OBS から受け取るコマンドの内容に基づいて COMICS コマンドやスクリプトファイルを実行したり、また OBS に対してステータスデータ取得やデータ転送を要求する。ところで観測中に予想外の制御操作が必要となった場合、観測手順書を使う場合にはその登録に若干の時間がかかるため、すぐに COMICS の制御ができない。その時には CSC の GUI によって制御を行う。そしてもちろん、開発者による COMICS の機能試験は、SOSs と接続する必要がないので CSC のみで行う。したがって CSC は、GUI による COMICS の制御と、OBS からの要求による制御とを同じように扱えるよう、そしてどちらの場合も処理の遅延が起らないようにプログラムの構成を考える必要がある。

4. 本システムの開発

4.1 COMICS のソフトウェア構成

COMICS をソフトウェア制御するにあたり、まず最初に装置を直接制御するハードウェア付随ともいべきソフトウェア群 (COMICS コマンド) を作成し、次にクロックを生成して検出器を駆動するスクリプトファイルを作成した。COMICS コマンドやスクリプトファイルは、コマンドラインで実行するものであり、一機能の試験では有用であるが複数機能の試験、さらには観測となると、複数コマンドをキーボードから入力することになってかえって不便である。そこで、COMICS コマンドやスクリプトファイルを使って簡単に COMICS を制御する GUI、そして観測のために SOSs と通信ができる制御ソフトウェアが必要となり、CSC を作成することとなった。CSC と COMICS コマンド、スクリプトファイ

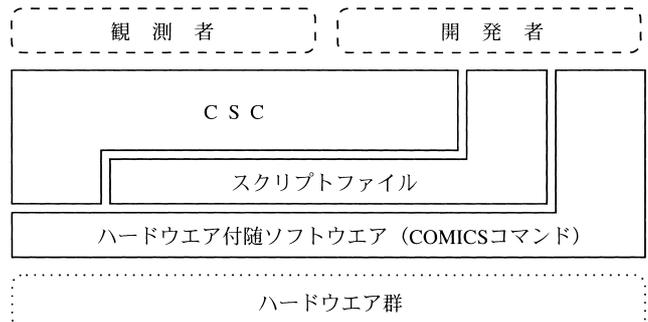


図 4. COMICS ソフトウェアの構成.

ルの関係を図 4 に表す。

COMICS コマンドは、一機能一コマンドのポリシーに従って作成した小プログラムである。したがってこれらコマンドは基本的に通信機能を持たず、ステータスファイルを設定してコマンド間で情報を共有している。CSC は直接このコマンドを実行する場合もあり、スクリプトファイルにより実行する場合もある。

クロックを生成して観測データを取得する一連の流れを記述したものがスクリプトファイルである。スクリプトファイルは、コマンドの実行判断と実行コマンドを perl で記述しており、観測目的によって複数個存在する。スクリプトファイルも SOSs との通信を行わない。なお、COMICS コマンドとスクリプトファイルは開発者が実行するものであり、観測者がこれらを直接実行することはない。

SOSs と通信を取りながら、COMICS コマンドやスクリプトファイルにより COMICS を制御するのが CSC である。CSC の実体は GUI と通信処理制御であり、CSC によって一連の観測が実施できる。GUI では COMICS の制御や状態表示を行い、SOSs との通信制御は内部処理による。

4.2 CSC のプログラム構造

CSC のプログラム構造を図 5 に示す。先述のとおり、COMICS の制御計算機 comics は観測装置システム上の OBCP にあたり、wiwaxia は OBE に相当する。

CSC は機能を切り分ける目的で、いくつかのプロセスを設けその上で専用のプログラムを実行している。まず OBCP 上で主プログラムが実行されるが、RPC 通信のために、サーバプロセスを OBCP と OBE のそれぞれで立ち上げ、また OBS に対する I/F 処理を行うプロセスを OBCP 上に立ち上げる。このように複数プロセスが存在するためにプログラム相互間の通信が重要で、CSC のプログラムは、前処理部で受信した通信内容を調べて処理を決定する判断作業を行い、本処理部で関数を実行し、後処理部で GUI 上への表示、そして必要に応じて他プロセスへの通信を行う、というような三層構造になっている。

主プログラムの本処理部は、GUI 操作と OBS からのコマンド要求による処理を行うにあたって、両者の異同に関わらず COMICS に対しては同じ処理を行うよう、同一の関数を使えるよう工夫した。また、それぞれのプロセス上で実行されるプログラムは、開発がしやすいことと後からの変更作業がやりやすいことを考慮しサブディレクトリに分けて作成したが、実行ファイル作成のための make は一番上のディレクトリで一回実行すればいいようにした。

4.3 CSC の通信機能

CSC は複数の通信手段を使っているが、図 6 はそれらの相関図である。基本的には、計算機間のプロセス間通信では RPC を用い、計算機内のプロセス間通信ではパイプおよびイベント、プロセス内ではイベントを使っている。

まず COMICS 制御における通信を説明する。CSC は OBCP と OBE に渡るシステムであり、COMICS への制御指令は OBCP から OBE の RPC サーバへ、COMICS コマンドから GUI への通知は、OBE から OBCP の RPC サーバへ通信する。

前者の場合、COMICS のステータスファイルからのデータ取得や、ディレクトリからのスクリプトファイル名の取得のように、その処理結果がすぐに返る場合と、スクリプトファイルの実行のように処理に時間がかかるがその間他の処理を実行してはいけない場合は、どちらに対しても RPC 通信は実行終了まで待ち続ける。一方 COMICS コマンドの実行のように、処理に時間がかかるがその実行中に他の処理を行ってもいい場合は、その実行終了まで待たずにすぐに通信を終了し、主プログラムへ制御を返す。本来ならば、処理実行に時間がかかる場合はすぐに主プログラムへ制御を戻し、処理の結果は別に通信する方法で統一するべきだが、処理が終了するまで待ち続けるようにしたのは、重要処理実行中に人間が不注意に他の操作をしないように、との安全対策のためである。

後者の OBE から OBCP の RPC サーバへの通信だが、通信機能のない COMICS コマンドに CSC 通信ライブ

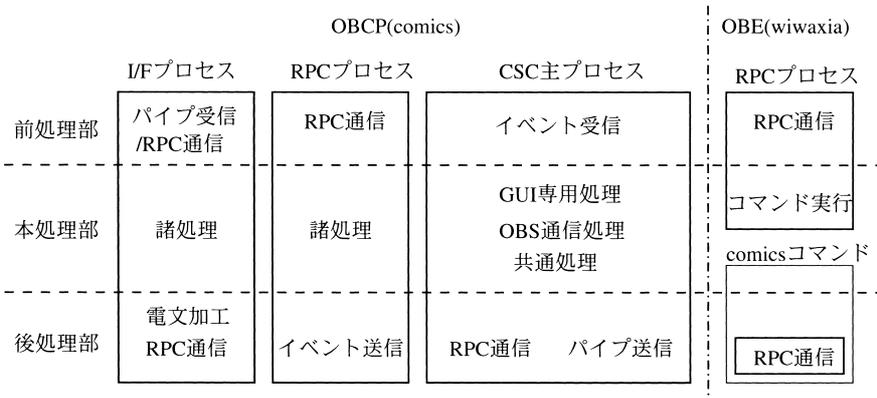


図 5. CSC のプログラム構成.

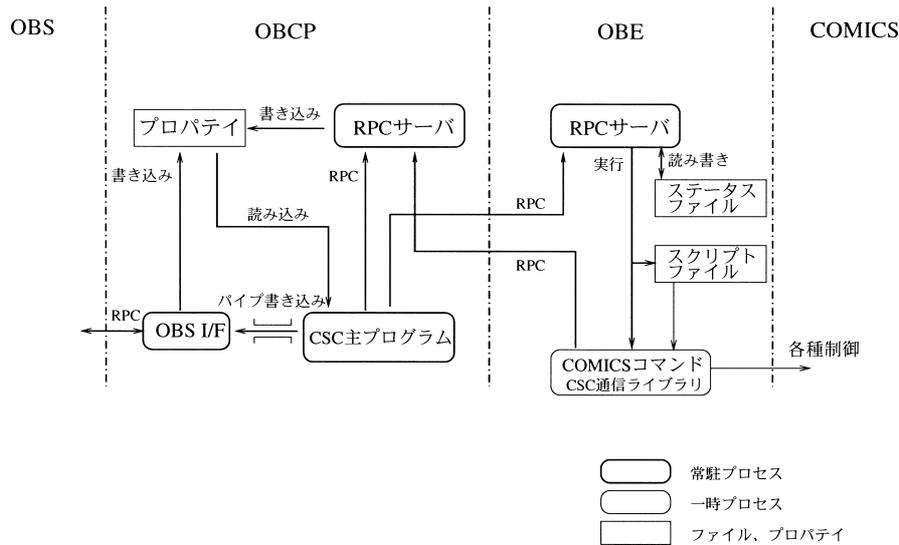


図 6. CSC の通信関連図.

ラリを付加し、これによってコマンド実行の結果を OBCP 上の RPC サーバに通知できるようにしている。

ところで、X にはアプリケーション間の情報共有のためにプロパティという汎用的機構があり、アプリケーションはプロパティにデータを格納したり、他のアプリケーションが格納したデータを取得したりできる。なお任意のプロパティの内容が変更されたとき、イベントが変更を知らせる。OBCP 上の RPC サーバはこの機構を利用して CSC 主プログラムにデータを送信する。すなわち、主プログラムがその変更通知を登録しているプロパティに RPC サーバがデータを格納すると、Property Notify イベントが CSC 主プログラムへ送出され、主プログラムはそのイベントを契機としてプロパティからデータを取得する。この方法は OBS からの通信を受け取る時と同様で、したがって CSC 主プログラムは、イベントだけで OBE と OBS の 2 カ所からの通信を受信でき、また GUI の各操作もイベントが契機となっているため、GUI 操作による COMICS の制御と SOSs からの COMICS 制御要求をまとめて扱えることとなり、処理に一貫性がとれた。

一方 CSC から OBS への通信は、パイプを利用して I/F プロセスへ信号を送り、I/F プロセスは決められた書式に電文を加工して、RPC により OBS へ通信する。

CSC はこのように多くの手段を擁して通信を行うため、一見複雑なシステムに見えるが、GUI を X ツールキットで作っているために主プログラムはイベント駆動型で処理を行う方式で、それにプロセス間通信用に別イベントを付加した形となり、プログラムの構造は単純で明瞭である。主プログラムから I/F プロセスへの通信は、OBCP の RPC サーバは COMICS コマンドや OBS からの通信の受け口専用にしたかったために使えず、そこで簡単に実現できるパイプにした。

4.4 CSC による COMICS 制御

ここでは SOSs と通信しない場合、すなわち GUI による COMICS 制御について述べる。まず GUI を作成する際に注意したことは、操作が単純であること、項目を選びやすいこと、操作ミスを起こしにくいことの三つである。単純な操作の実現のためにボタン操作中心にし、項目を選びやすくするためにポップアップ画面やプルダウンメニューを活用し、操作ミス避けるために重要なボタンは押しやすい時まで隠すことにした。さらに、観測の時には計算機画面上に観測データの表示解析画面や、観測ログを随時書き込むためのエディタ画面など複数の画面が表示される。その中で GUI を立ち上げるので、画面上のスペースを有効に使えるように、必要な情報のみを盛り込んだコンパクトな画面になるように注意した。

次に具体的な制御について述べる。光学系の駆動制御は、機器毎に選択できる項目をポップアップ画面上に一覧表示し、マウス操作だけで選択・設定をする。機器の角度調整のように細かく数字を指定するものについては、数字をキーボードから入力できるエリアを設けた。GUI 上では光学系機器の現状を表示するが、ポップアップ画面で変更した場合、GUI を使わずに COMICS コマンドを直接実行して変更する場合のどちらについても、その変更内容を即座に GUI 上に自動表示するようにした。また COMICS 各所の温度情報についても、一定間隔で GUI 上に自動表示する。

検出器クロックのパラメータ設定は、撮像/分光の各パラメータごとにキーボードから数値を入力できるエリアを設けている。またクロックを生成するためのスクリプトファイルは複数存在するため、その一覧をポップアップ画面で表示し、簡単に選び出せるようにした。なお、パラメータとスクリプトファイルは関連しているため、一度入力したパラメータとスクリプトファイル名はファイルに書き残し、いつでも以前設定した値を読み出

せるようにした。ところでその後、試験観測が進んだ段階でパラメータが数種にパターン化できるようになったため、あらかじめそれらを登録しておき、プルダウンメニューから簡単に選び出せるように変更を加えた。

COMICSの開発段階において、クロックの生成はコマンドボタンを押下して実行する。このボタンは安全のために、パラメータおよびスクリプトファイルを確定した後で初めて使用可能になる。ただこれは開発者がCOMICSの試験の際に使用するボタンであり、観測では基本的にこのボタンを使うことはなく、SOSsからの要求によってボタン押下と同じ処理をCSCが自動で実行する。

ところで、スクリプトファイルの実行でクロックの生成から取得データの並べ替えまでを行い、当初はこの一連の処理終了まで次の制御が待たされていた。クロックの生成は短時間でできるが、データの並べ替えにはかなりの時間がかかる。そしてデータの並べ替えは並列処理ができるため、この間にCSCで制御ができるようになると観測時間の大幅な増加につながる。そこで初期の試験観測の後で変更を加え、クロックの生成が終わった段階で制御をCSCへ戻し、データの並べ替えの処理中に次のクロックの生成、データ加工ができるようにした。ただし計算機全体の処理速度の低下を招かないように、データの加工は最大5プロセスまでとし、これを越える場合には次の加工処理を待たせる。

CSCで行った操作はすべてログファイルに書き残しているが、一晩の観測の全処理を1ファイルにまとめたいため、世界時をもとに一日ごとにファイルを作成し記述するようにしている。またこのファイルには、SOSsから送られた電文内容もそのまま記述し、後からの確認に役立てている。

4.5 CSCによる観測制御システムとの通信制御

SOSsとの通信を行うにあたって注意したことは、確実に通信が行えること、GUIによる処理と統合させること、通信方法を単純にすることの三つである。確実な通信のために、CSCはOBSとほぼ同じ環境の計算機で開発した。処理の統合のためには、イベントを受け取った後の処理で同じ関数を使えるよう、関数を機能ごとに作成した。単純な通信方法のためには、OBSと通信する専用のプロセスを別に立ち上げ、機能分担を明確にした。以下、通信制御の実際について述べる。

まずGUIから制御するのは、SOSsと共に実現する4つの機能それぞれに対応するI/Fプログラムの起動、終了のみである。これらはGUIのプルダウンメニューで、一つずつ、または全部まとめての起動および終了ができる。

「ステータス転送機能」の実現のために、該当I/Fプログラムは起動と同時に、1分間隔で決まったメッセージをOBSに送信する。この通信に対する結果待ちはない。このプログラムは、CSC主プログラムからの通知を

受け終了する。

「ステータスデータ取得機能」の実現のために、該当I/Fプログラムは起動と同時にCSC主プログラムからの要求待ち状態に入る。CSC主プログラムは、OBSからステータス取得の要求コマンドを受けると、このI/Fプログラムへ要求を出し、ここからOBSへステータスデータを要求する。データ取得後、CSC主プログラムはOBE上にあるCOMICSステータスファイルに内容を書き込む。この情報は、FITSファイル作成の時に参照される。

「取得データ転送機能」の実現のために、該当I/Fプログラムは起動と同時にCSC主プログラムからの要求待ち状態に入る。CSC主プログラムは、OBSから取得データ転送の要求コマンドを受けると、作成したFITSファイルの格納場所をこのI/Fプログラムを通してOBSに通知する。OBSはOBCに通知し、OBCが渡された情報に基づいて、FITSファイルをまとめて観測所データベースに転送する。この際、1ファイル転送ごとに転送結果が返るが、たとえ途中で転送が失敗してもそこで処理は中断せず最後まで実行させる。したがって失敗した場合は、人間が確認して別に再度転送することになる。

「コマンド通信機能」の実現のために、該当I/Fプログラムは起動と同時にOBSからの要求待ち状態に入る。OBSからの要求を受けると、CSC主プログラムに受け取った電文内容を送信し、その処理結果を待つ。CSCが処理を行った結果を返すと、I/FプログラムはOBSにその結果を送信する。結果は必ず送信しなければならない。受信したOBSは結果が正常であれば次のコマンドを続けて実行する。観測手順書を使った観測ではこのようにシーケンシャルな制御を行い、同時に複数のコマンドが実行されない作りになっている。

4.6 装置依存コマンド

前節で述べた4つの機能をもとに実現する具体的な

表3. 装置依存コマンド一覧。

コマンド名	内容
OBCP_SCRIPT	COMICSコマンド、一般コマンドの実行
CSC_SEQ	スクリプトファイル実行
FRAME_NO	FITSファイル通番の取得と検査
TEL_STATUS	望遠鏡ステータスデータの取得
FTP_DATA	FITSファイル転送
MIRROR	平面鏡の角度調整
PRE_FILTER	前光学系フィルターの設定
SLIT	スリットの設定
IMG_FILTER	撮像フィルターの設定
IMG_LENS	レンズの設定
GRATING	回折格子の設定と角度調整
SET_EXPPARM	クロックパラメータとスクリプトファイル名の設定

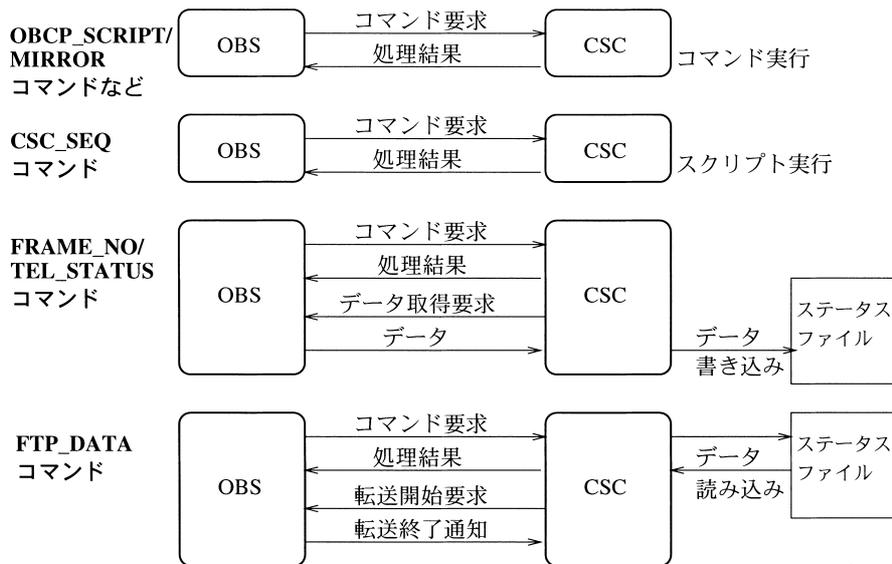


図 7. 装置依存コマンドを契機とした信号の流れ.

制御は、観測手順書に記述する装置依存コマンドに対応する。COMICS の装置依存コマンド一覧を表 3 に、これらコマンドを契機とした信号の流れを図 7 に示す。

これらのコマンドが先の 4 つの通信機能のどれを使って実現されているかを述べる。OBCP_SCRIPT コマンドと CSC_SEQ コマンドは COMICS コマンドやスクリプトファイルを実行するために使うコマンドで、「コマンド通信機能」のみで実現されている。FRAME_NO コマンドと TEL_STATUS コマンドは、OBE 上にあるステータスファイルに望遠鏡側のデータを書き込むために使うコマンドで、「コマンド通信機能」と「ステータスデータ取得機能」を使って実現する。FTP_DATA コマンドは、FITS ファイルを観測所のデータベースに転送するために使うコマンドで、「コマンド通信機能」と「取得データ転送機能」を使って実現する。なお、光学系機器を制御し、検出器を駆動するためのその他のコマンドは、「コマンド通信機能」のみで実現している。

5. 観測時における CSC の使用法

ここでは観測の時の CSC の使用方法について述べる。観測の前に、先に述べた装置依存コマンド、抽象化コマンドや望遠鏡制御コマンドを記述した観測手順書を作成し OBS に登録しておく。ここではすでに観測手順書があり、OBS がその手順書に従ってコマンドを送出できる状態にあるものとして、CSC の操作についてのみ述べる。

5.1 起動と終了

まず OBE 上で RPC サーバを立ち上げ (csc_obsrvr を実行)、続いて OBCP 上で RPC サーバを立ち上げ (csc_obspsvr を実行)、そして最後に CSC 主プログラムを立ち上げる (csc を実行)。すると図 8 のような GUI が現れる。

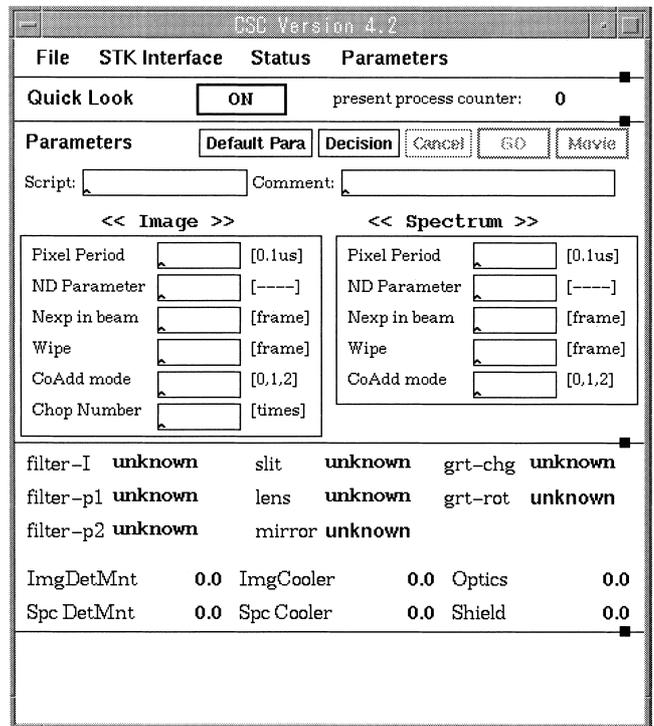


図 8. GUI 起動時の GUI 画面.

次に SOSs との I/F プログラムを立ち上げる。メニューバー項目「STK Interface」から「Exec All」を選択すれば、OBS との全 I/F プログラムが起動される。メニューにはほかに「Exec STAT I/F」や「Exec DAT I/F」などがあり、これらを選択すれば 4 つの機能を別々に起動することができる。

終了は、先のメニューバー項目から「Exit All」を選択して I/F プログラムを終了させ、その後でメニューバー項目「File」から「Quit」を選択して、RPC サーバを含めた CSC 全体を終了させる。

5.2 観測直前準備

観測直前にやることは、光学系機器の状態の検査、温度が適切な範囲にあるかの確認である。光学機器の現状を調べるために、GUIのメニューバー項目「Status」から「Reload All Status」を選択する。これにより、GUI上に撮像フィルター、前光学系フィルター、スリット、レンズ、入射平面鏡、回折格子の現状が表示される。また、温度はその下に自動表示されるので、それらを見て確認する。

光学系機器の状態を変えたい場合、GUIを使って変更操作を行う。GUI上の光学系機器名はコマンドボタンになっており、これを押下するとポップアップ画面が前面に表示される。その画面上に交換できる部品名または角度入力エリアが表示されるので、選択または入力の後、一番下にある「OK」ボタンを押下する。図9はレンズ変更のためのポップアップ画面を表示したところである。

5.3 観測中操作

検出器を稼働させるためにクロックを発生させるが、まずその前にクロックパラメータを設定しスクリプトファイルを指定する。スクリプトファイルとパラメータのいくつかのパターンが登録されているので、メニューバー項目「Parameters」から選択する。すると、GUI上に、スクリプトファイル名、全パラメータが表示確定される。またこれらは手入力でも変更可能である。さらに、以前に指定したスクリプトファイルおよびパラメータ値はファイルに書き込まれており、「Default Para」ボタンを押下して読みだせる。手入力、またはファイルから

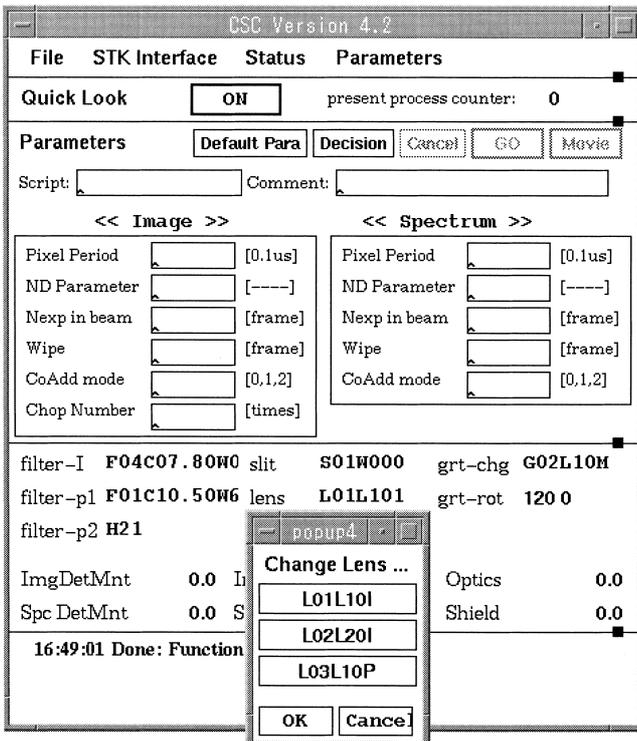


図9. レンズ変更のポップアップ画面。

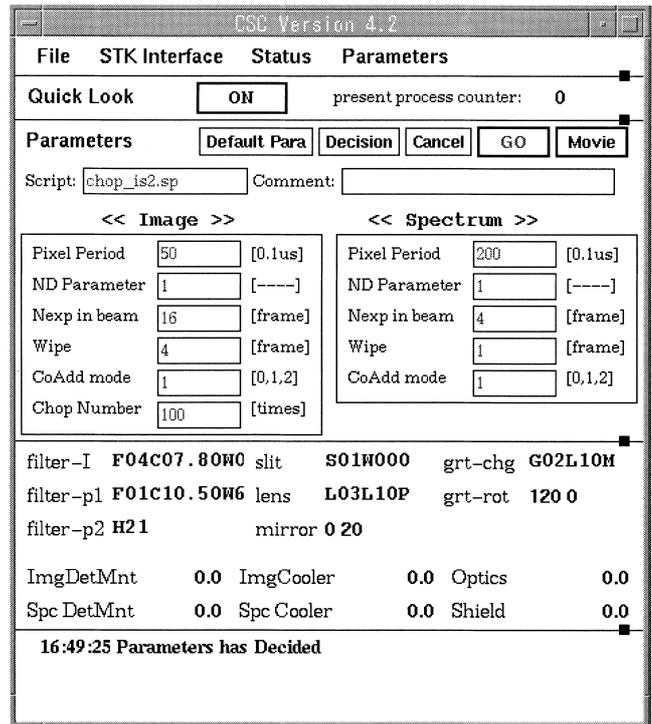


図10. スクリプト実行直前のGUI画面。

値を読みだした場合は、「Decision」ボタンで確定する。確定すると値は赤色に表示変更され、編集ができなくなる。確定した後で再び編集したい場合には「Cancel」ボタンを押せばよい。ところで、現在あるスクリプトファイルを確認したい場合には、メニューバー項目「File」から「Script List」を選択すると一覧がポップアップ画面上に表示される。

スクリプトファイルおよびクロックパラメータを確定したら、次はスクリプトの実行である。図10がスクリプト実行直前の画面になる。開発や機能試験の際には「GO」ボタンを押下すればOBE上に存在するスクリプトファイルが実行されるが、観測の時にはOBSからの要求により実行する。すなわち、パラメータの確定まではGUIで行い、以降は観測手順書に従いOBSがコマンドを順に発行して観測を行っていく。観測で取得したデータは、画像表示簡易解析ソフトウェア xfc⁴⁾で確認する。

ところで試験観測初期にはSOSsによる観測に不慣れであったために、光学系の変更やクロックパラメータの変更、スクリプトファイルの指定はGUI上で行ったが、その結果、観測者はGUIとOBSの制御画面の両方を監視する必要が出、観測が複雑になっていた。現在は抽象化コマンドを整備しており、GUIを使っていた制御がOBSからできるようになっている。GUIはCOMICSの状態表示やOBSとの通信の状態を監視するために、そして突発的な制御が必要な場合に使用されている。

6. まとめ

すばる望遠鏡で観測する場合、観測装置の制御ソフト

ウェアシステムはSOSsとの連携をとることが最重要であり、その上で装置の制御も行うために、CSCではこの二つの機能をどのように統合するかが一番の問題であった。結局はGUIをXツールキットで作成し、OBSとの通信もXのアプリケーション間通信方法を採用したことで、Xの処理で統一することができ、両機能をうまく融合することができた。おかげでCOMICSの開発段階においても、そして観測においても、開発者すなわち観測者にとってとまどいのないシステムを実現できた。ところで試験観測においては、装置依存コマンドや抽象化コマンドを十分に用意することができなかつたために、GUIを使っただけの操作が多かった。GUIは単純で間違いにくい操作の実現という考え方で作成したのだが、そのため4,000メートル級の山頂での観測という苛酷な環境においても、観測者に負荷を与えず、そして確実な制御を行うことができた。

またSOSsは、ツールキットや疑似システムといった、観測装置側の制御システム開発のための補助手段を設けている。CSCは開発の初期段階ではSOSsが提供したツールキットによって、SOSsとの通信の方法を確認した。ただしツールキットはSOSsの受け口を提供するだけのものであり、その中で判断処理などはしていない。ある程度通信処理を作成した段階で、何回かにわたって三鷹にあるSOSsの疑似システムとの接続試験を行った。これはハワイ島の山麓システムとほぼ同等の疑似システムであり、この接続試験により多くの不具合を発見することができた。

COMICSがハワイ島に移送された後では、本格的な通信試験を行った。まず山麓の疑似システムで装置受け入れ試験が行われ、数項目にわたってCSCが要求している機能を満たしているかを確認した。この後、山頂にある本システムでの最終試験を行い、SOSsとCSC間の通信に問題がないことを最終確認し、そして観測に望むことができた。このように慎重に段階をおいて接続試験

ができたことで、CSCは別システムとの通信の連携処理という課題を克服できたものと思う。

謝辞

三鷹およびハワイ観測所、山頂システムにおけるSOSsとの接続試験、そして試験観測時には、小杉城治氏、八木雅文氏をはじめとするすばる計算機グループの方々には様々な支援や助言、協力をしていただいた。そしてCOMICSハワイ移送後のイーサネットやファイバーチャネルなどのネットワーク接続においては、小笠原隆亮氏に非常に尽力をしていただいた。CSCがSOSsと順調に通信を行い、COMICSがすばる望遠鏡で問題なく観測ができるようになったのもこれらの方々のお力のおかげであり、心から感謝している。また同様に、SOSsを作成した富士通の方々にも大変にお世話になった。あわせてここで感謝の意を表したい。

参考文献

- 1) H. Kataza, Y. Okamoto, T. Onaka, S. Takubo, S. Sako, K. Nakamura, T. Miyata, and T. Yamashita: COMICS: The Cooled Mid-Infrared Camera and Spectrometer for the Subaru Telescope, *Proc. SPIE*, **4008**, 132 (2000).
- 2) N. Kaifu: Subaru Telescope, in *Advanced Technology Optical/IR Telescopes VI*, ed. by L. M. Stepp, *Proc. SPIE*, **3352**, 14–22 (1998).
- 3) T. Sasaki, G. Kosugi, J. Noumaru, T. Takata, Y. Mizumoto, R. Ogasawara, Y. Chikada, W. Tanaka and J. Kawai: Observation control system for SUBARU telescope and its user interface, *Proc. SPIE*, **3349**, in press (1998).
- 4) 中村京子, 宮田隆志, 片堅宏一: すばる望遠鏡中間赤外撮像分光装置用画像表示簡易解析システムの開発, 国立天文台報, **5**, 43–49 (2001).