MOD TOOL (MICROWAVE OPTICS DESIGN TOOL)

Daniel S. Katz, Andrea Borgioli, Tom Cwik, Chuigang Fu, William A. Imbriale, Vahraz Jamnejad, and Paul L. Springer

Jet Propulsion Laboratory
California Institute of Technology

(contact: Daniel S. Katz, 4800 Oak Grove Drive, MS 168-522, Pasadena, CA, 91104, voice: 818-354-7359, fax: 818-393-3134, e-mail: Daniel.S.Katz@jpl.nasa.gov)

I. Introduction

The Jet Propulsion Laboratory (JPL) is currently designing and building a number of instruments that operate in the microwave and millimeter-wave bands. These include MIRO (Microwave Instrument for the Rosetta Orbiter), MLS (Microwave Limb Sounder), and IMAS (Integrated Multispectral Atmospheric Sounder). These instruments must be designed and built to meet key design criteria (e.g., beamwidth, gain, pointing) obtained from the scientific goals for the instrument. These criteria are frequently functions of the operating environment (both thermal and mechanical). To design and build instruments which meet these criteria, it is essential to be able to model the instrument in its environments.

Currently, a number of modeling tools exist. Commonly used tools at JPL include: FEMAP (meshing), NASTRAN (structural modeling), TRASYS and SINDA (thermal modeling), MACOS/IMOS (optical modeling), and POPO (physical optics modeling). Each of these tools is used by an analyst, who models the instrument in one discipline. The analyst then provides the results of this modeling to another analyst, who continues the overall modeling in another discipline.

There is a large reengineering task in place at JPL to automate and speed-up the structural and thermal modeling disciplines, which does not include MOD Tool. The focus of MOD Tool (and of this paper) is in the fields unique to microwave and millimeter-wave instrument design. These include initial design and analysis of the instrument without thermal or structural loads, the automation of the transfer of this design to a high-end CAD tool, and the analysis of the structurally deformed instrument (due to structural and/or thermal loads).

MOD Tool is a distributed tool, with a database of design information residing on a server, physical optics analysis being performed on a variety of supercomputer platforms, and a graphical user interface (GUI) residing on the user's desktop computer. The MOD Tool client is being developed using Tcl/Tk, which allows the user to work on a choice of platforms (PC, Mac, or Unix) after downloading the Tcl/Tk binary, which is readily available on the web. The MOD Tool server is written using Expect, and it resides on a Sun workstation. Client/server communications are performed over a socket, where upon a connection from a client to the server, the server spawns a child which is be dedicated to communicating with that client. The server communicates with other machines, such as supercomputers using expect with the username and password being provided by the user on the client.

II. User Interface

The initial MOD Tool screen requires the user to provide a username and password, and to select an old or new design on which to work. The client communicates with the server to check this data, and to reach the area of the database containing this design. At this point, the main MOD Tool screen comes up, allowing the user to work in one of six modes.

1. Design Mode

This mode allows the user to load, modify, and save a design. A design is primarily a set of physical optical elements (reflectors), but it also includes two non-physical elements, the location of a feed, and the location for outputs to be gathered. Reflectors are designed as conic sections cut with an arbitrarily-defined oval cutting cylinder. These conic sections may be either flat plates, paraboloids, ellipsoids, or hyperboloids. A feed location is defined by a point at the feed aperture, a direction in which the feed is pointed, and the major and minor radii of the aperture. An output system is defined by a point and a direction. The design information is shown in two forms: a table and a graphic window. The table presents coordinates of the points that determine the object (for a paraboloid, these would include the vertex, the focus, a point on the paraboloid and a point off the paraboloid to determine the axis of the cutting cylinder, and the major and minor radii of the cylinder) and other information that might be useful to the designer (again for a paraboloid, this includes the distance from the vertex to the focus). The graphics window shows this objects determined from this data in 2-D slices, and allows the user to move points on the screen. Changes in either form are reflected in the other (if a graphic point is moved, the table is updated to reflect the new data, and vice versa). Our current plans for the graphics frame in the design mode are to allow the user to work on any Cartesian plane (for example, the x-y plane at z=1.4 mm). The designs are examined and modified by the GUI client, but the data is loaded from and saved to the server's database.

2. Prescription Mode

This mode allows the user to specify and modify the objects that will be used in a given analysis. The user can choose objects and order a subset of the objects from the current design, or load an old prescription. Objects may be modified in location and/or orientation in order to perform a tolerancing analysis. Once the objects are chosen and possible modified, the user can save the prescription. The prescriptions are examined and modified by the GUI client, but the data is loaded from and saved to the server's database. See Figure 1 for an example of this mode.

3. Geometric Optics Analysis Mode

This mode allows the user to perform geometric optics calculations on the prescription and design that have been previously specified. This is done by converting the design/prescription information into a MACOS input file, and running MACOS under control of the GUI. The conversion is performed by the server, which also runs MACOS. Results are transferred to and displayed by the client. This analysis is quite fast, and may be used for initial design work.

4. Physical Optics Analysis Mode

Physical Optics (PO) is a much more precise analysis method, which is used when the objects being analyzed are relatively small (size is in terms of wavelengths). This mode allows the user to set up and launch PO runs, which generally are fairly time consuming and are done on a supercomputer. Currently, the PO code used at JPL has been ported to the Cray J90, and HP SPP-2000, and Beowulf systems. Starting a PO job on a supercomputer involves the server converting the design and prescription information to a format the PO code understands, the client obtaining a username and password for the supercomputer from the user, and the server transferring the files to the supercomputer and starting the job. Additionally, the PO code can be used to analyze surfaces that have been structurally or thermally deformed by the server using mesh and load file which have been submitted (using the following two modes), calculating the coefficients of a bipolynomial surface for the deformation of each surface, and transferring this information to the supercomputer. In order to do this, the server strips the information about each surface from the mesh and load files, and uses MATLAB to calculate the desired coefficients.

5. Submitting a Mesh Mode

This mode is used by the structural engineer. Once he has created a mesh, either based directly on the design, or based on a CAD model which was built from the design, he uses this mode to submit it from the client to the server, and store it in the database associated with the design. MOD Tool currently uses FEMAP neutral files for meshes, which are unit-independent. Therefore, the structural engineer must also provide information about the units that were used in the mesh.

6. Submitting a Load Mode

This mode is also used by the structural engineer. Once a mesh has been created, it is normally deformed by either structural or thermal loading. MOD Tool currently accepts NASTRAN .f06 files that contain the deformation of each node of the mesh. This information is used by the PO code, as described in the Physical Optics Analysis mode.

III. Conclusions and Future work

MOD Tool is being built in conjunction with MIRO. The motivation for this is both to help the MIRO project by providing the analyses required in a timely manner as well as to help MOD Tool by providing a real instrument as a test case. The figures in this extended abstract all come from MIRO. The close involvement of the two projects has been very valuable, and while the measure of success for MIRO will not be seen for many years until it is flying and sending back data, the measures of success for MOD Tool will be seen sooner, as MOD Tool itself is (hopefully) used and supported by more flight projects, and as the pieces that make up MOD Tool are used by other software projects.

Future needs that MOD Tool is designed to be able to meet include using analysis of a metrology input (the instrument as-built), and design optimization. As MOD Tool develops, these capabilities will be added, as will others that are have not yet been identified. MOD Tool has been designed as a framework to which many things can be added. It already includes the use of Tcl/Tk code on multiple platforms, Perl scripts, Fortran programs, and MATLAB code on a workstation, as well as Expect to allow use of networked supercomputers with proper accounting. One of the considerations in this work was developing as little as possible, and thus, reusing previously developed tools and components as much as possible. Figure 2 shows the overall MOD Tool process, which clearly contains many tools that can be used independently.

IV. Acknowledgment

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

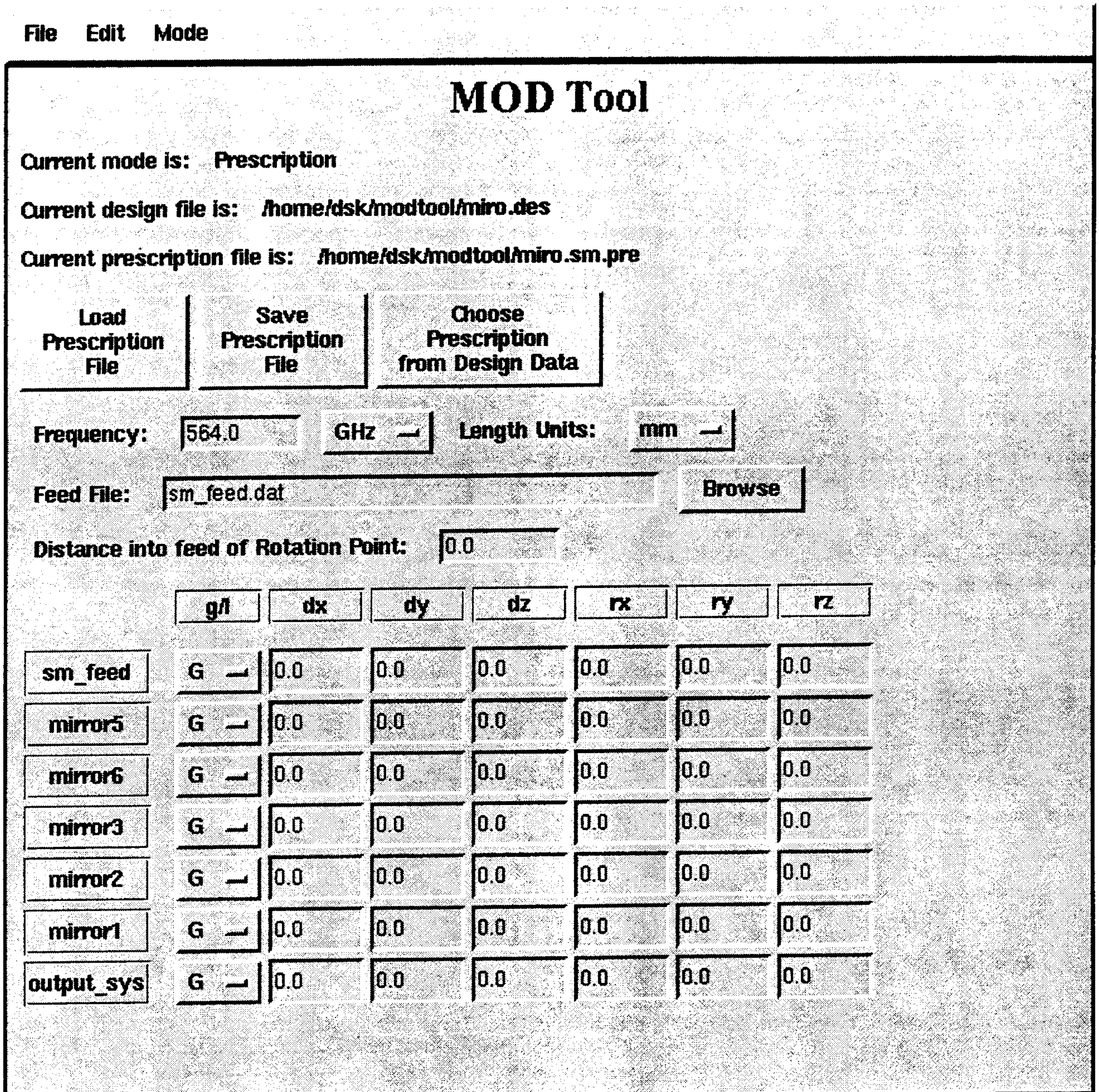


Figure 1. A sample screen captured from MOD Tool showing a prescription being edited.

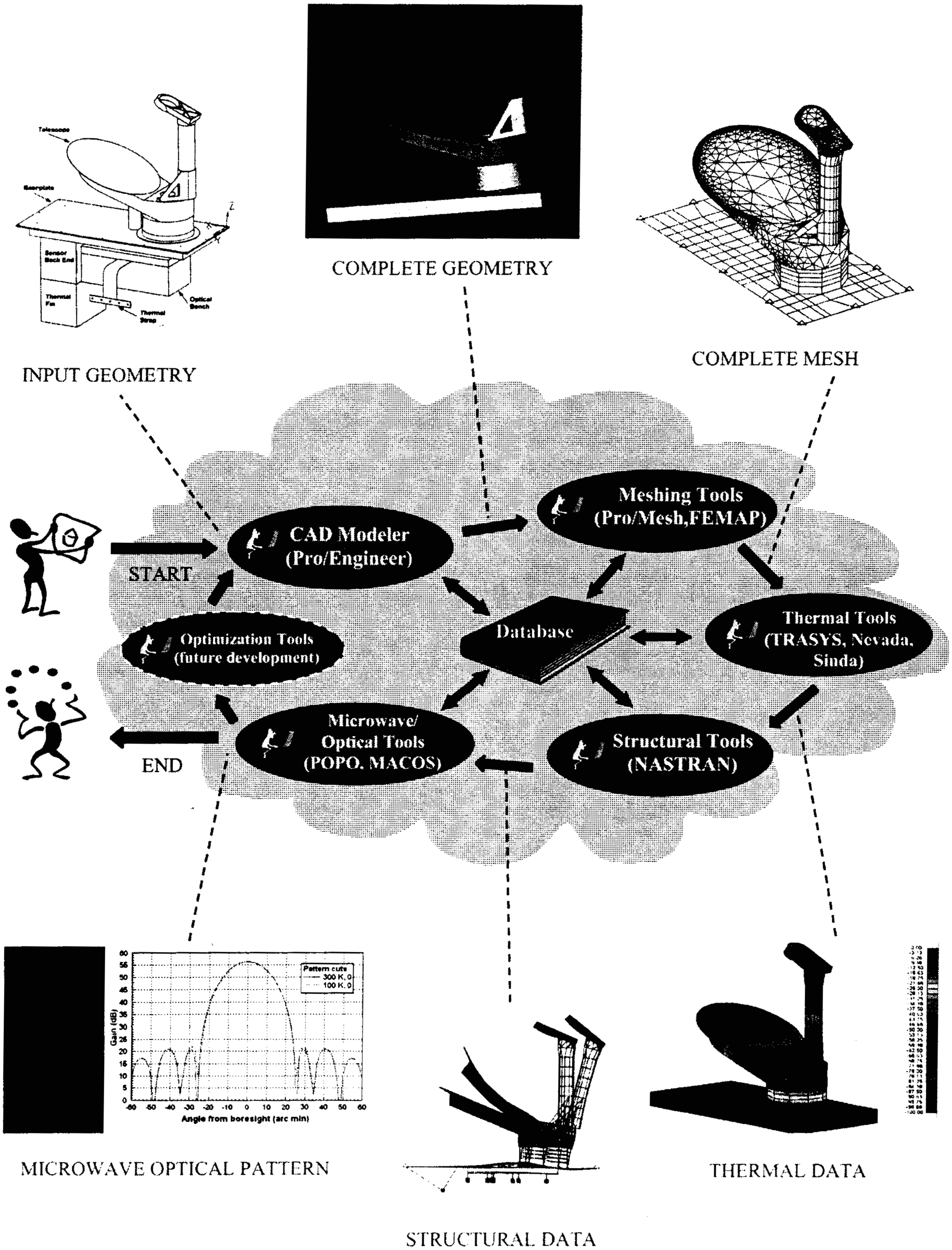


Figure 2. Conceptual process to be performed by MOD Tool

MOD Tool
(Millimeter-Wave Optics Tool)

Jet Propulsion Laboratory

California Institute of Technology

High Performance Computing Group:

Daniel S. Katz, Andrea Borgioli, Tom Cwik, Chuigang Fu, Paul L. Springer

Communications Ground Systems Section:

William A. Imbriale, Vahraz Jamnejad

JPL

Motivation

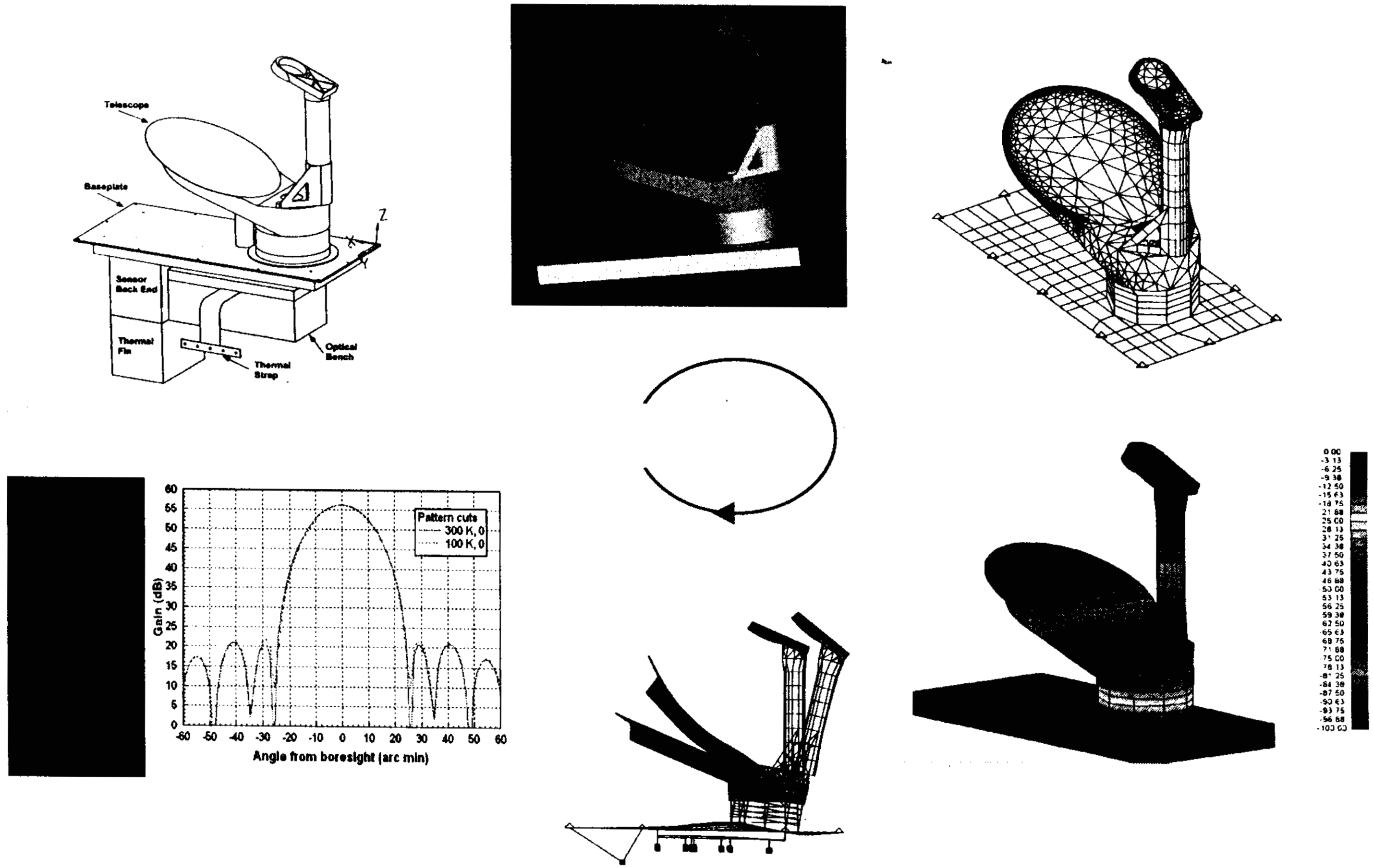
- **JPL designs and builds microwave and millimeter wave instruments**
  - MIRO - Microwave Instrument for the Rosetta Orbiter
  - MLS - Microwave Limb Sounder
  - IMAS - Integrated Multispectral Atmospheric Sounder
- **Instruments must meet design criteria obtained from science mission goals**
  - Beamwidth
  - Pointing
  - Gain
- **Design criteria are functions of operating environment**
  - Thermal
  - Mechanical
- **To design and build instruments which meet these criteria, we must model instruments in their environments**

Current Tools and Methods

- **Disciplines/Tools:**
  - CAD
    - *Pro/Engineer*
    - *CV*
    - *IDEAS*
  - Meshing
    - *FEMAP*
  - Structural
    - *NASTRAN*
  - Thermal
    - *TRASYS*
    - *SINDA*
  - Wave Propagation/Optics
    - *MACOS*
    - *POPO*

- **Each tool is used by an analyst, who works in one discipline**
- **When one analyst completes a model, it is passed to the next**
- **JPL would like to speed up this process**
- **Through the reengineering program, efforts are being made to improve the first four disciplines**
- **MOD Tool focuses on the last discipline, and how it is connected to the others**

MOD Tool Concept



MOD Tool Goals

- **Develop new code only if necessary**
  - Try to use/reuse:
    - *Commercial applications*
    - *Previously developed JPL applications*
  - Try to develop code so that it can be used for other future projects
    - *Use standard packages/tools*
    - *Always think beyond this project*
- **Ensure users on multiple platforms can easily use tool**
  - Client written in Tcl/Tk
    - *Freely available over the web*
    - *Available for Unix, PC, Mac*
  - Server also written in Tcl/Tk, but on a specific type of machine
    - *Uses compiled C code using Unix libraries for user authentication*
    - *Uses Expect for control of other processes on same machine*
    - *Uses Expect for file transfer and control of processes on other machines*
  - Sockets are used between the client and the server

MOD Tool User Interface

- **User starts client code on user's machine**
  - User must enter username and password
  - User must select existing or new design on which to work
- **Client opens socket to server**
  - Server validates username and password
  - Server creates local work directory
- **User may then work in one of six modes:**
  - Design
  - Prescription
  - Geometric Optics Analysis
  - Physical Optics Analysis
  - Submitting a Mesh
  - Submitting a Load

Design Mode

- **Allows the user to load, modify, save a design**
- **Designs are stored on the server, but editing is performed on the client**
- **A design is the description of a set of optical elements**
  - Physical Elements:
    - *Geometric shapes (paraboloids, hyperboloids, ellipsoids, flat plates)*
    - *cut by an oval cutting cylinder*
  - Non-physical elements:
    - *Feed location*
    - *Output system definition*
  - Elements are defined by coordinates of nodes, and distances
    - *A paraboloid is defined by a vertex, a focus, a point on the paraboloid where the cutting cylinder intersects the paraboloid, another point that defines the cutting cylinder, and the major and minor radii of the cutting cylinder*
  - The elements are shown graphically, and the values which control the elements are shown as a table
    - *Either the data in the table or the graphical data may be varied*
    - *The two displays are linked - changes in one are reflected in the other*

JPL

Prescription Mode

HPCC NASA

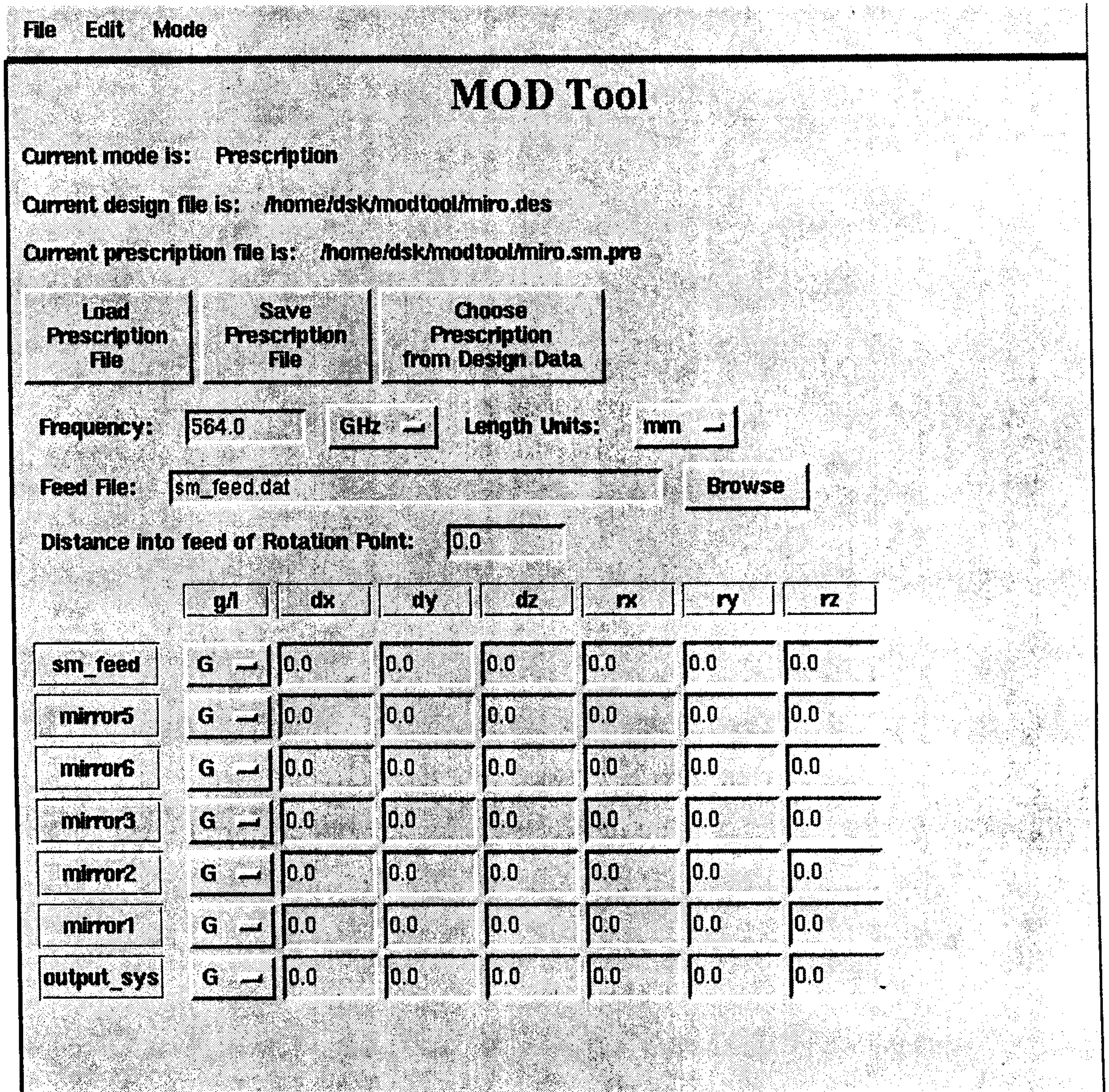
- **Allows the user to load, modify, save a prescription**
- **Prescriptions are stored on the server, but editing is performed on the client**
- **A prescription an ordered list of a subset of the optical elements from the design**
  - Determines which elements should be analyzed and in what order
  - Needed because instruments can have multiple logical paths
    - *Instrument will be used at multiple frequencies*
    - *Instrument has multiple modes of operation*
- **Elements may be modified in location or orientation from the base elements as stored in the design**
  - Used for tolerancing analysis

JPL

Prescription Mode

HPCC NASA

- A screen capture of a prescription file being edited



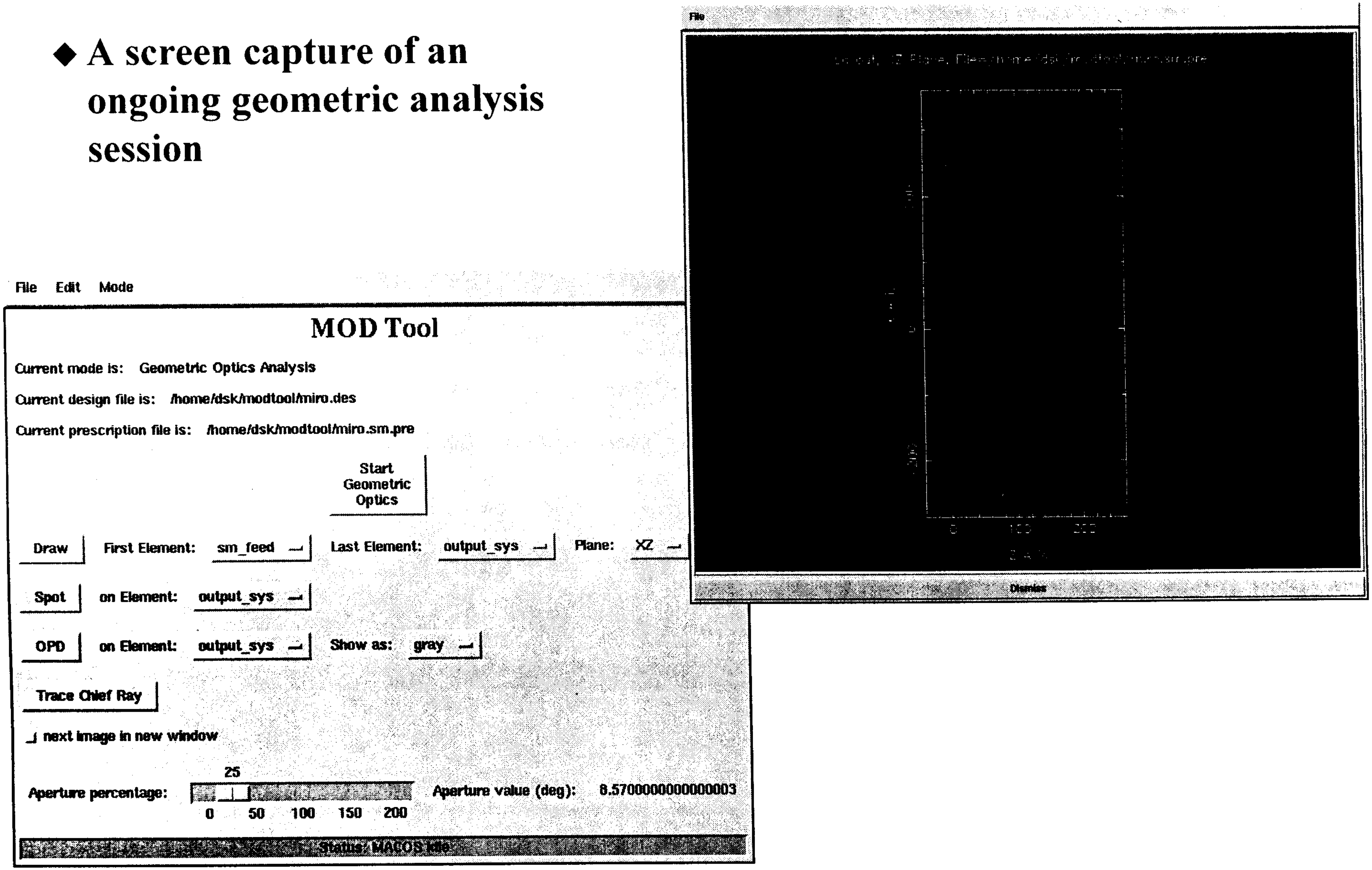
Geometric Optics Analysis Mode

- **Allows the user to perform geometric optics analysis on a design and prescription that have been previously specified**
- **These files are converted to a MACOS input set**
  - MACOS provides computationally efficient general ray-trace, differential ray-trace, and diffraction calculation capabilities
  - Developed at JPL, starting in 1989
  - Commercial and U.S. Government versions are available
- **The files must be on the server, which does the conversion, and runs MACOS**
  - Only the general ray-tracing capabilities of MACOS are currently supported through MOD Tool
- **The user may control and select the outputs from MACOS on the client**
- **The results are also displayed on the client**

JPL

Geometric Optics Analysis Mode

- A screen capture of an ongoing geometric analysis session



Daniel.S.Katz@jpl.nasa.gov - High Performance Computing Group - Imaging and Spectrometry System Technology Section

Physical Optics Analysis Mode

- **Physical Optics (PO)**
  - More accurate and slower than Geometric Optics
  - Used for mirrors that are relatively small (numbers of wavelengths)
- **POPO code**
  - Developed at JPL over a long period of time (1971 - present)
  - Versions exist for PCs, workstations, Cray J90, T3D, Beowulf
- **Can analyze surfaces from design**
- **Can also add deformations from load file before analyzing**
  - Done using MATLAB to read mesh and load data, and to calculate coefficients for a bipolynomial surface approximating the deformation
  - MATLAB is run on the server, where the design, prescription, mesh and load files are stored
- **POPO code is run on supercomputers**
  - Using username and password for that machine supplied by user
  - Ensures correct accounting and time-charging
  - Expect is used between the server and the supercomputer

JPL

Submitting a Mesh Mode

HPCC NASA

- **Used by structural engineer**
  - The person who creates the structural analysis mesh
- **Submits a mesh of an instrument from a client to the server**
  - Currently, meshes must be FEMAP neutral files
  - One layer is used for the elements that make up each optical element
  - Additional layers (supports, struts, other mechanical parts) are ignored by MOD Tool
- **Additional information about the mesh should be supplied**
  - Unit data is essential, as FEMAP files are unit-independent
  - Other information is optional, such as comments about the mesh
- **This data is used to validate that the mesh and the design are the same geometry**
- **It is also used with the load data for physical optics analysis of loaded instruments**

Submitting a Load Mode

- **Used by structural engineer**
  - The person who creates the structural analysis mesh
- **Submits deformations of a mesh of an instrument from a client to the server**
  - Currently, loads must be NASTRAN .f06 files
  - Important data: displacement of nodes on optical surfaces
  - Other data is ignored by MOD Tool
- **Deformations come from structural or thermal loads**
  - How the instrument will change from the original design when it is in its operating environment
- **The load data must be logically tied to a set of mesh data**
- **This data is used in the physical optics analysis, with the corresponding mesh data**

Current Status and Future Work

- MOD Tool framework exists, as do the prescription, geometric optics, and part of the design mode
- Additional work is needed in the design mode
- All the underlying parts of the submit a mesh, submit a load, and physical optics modes exists, but they have not yet been incorporated into MOD Tool
- Ideas for future additions:
  - Implementing optimization
    - *first semi - automatic, then fully automatic*
  - Adding ranges of deformations to prescription elements to compute a sensitivity matrix more automatically
  - Working with a metrology input to analyze the as-built instrument

JPL

Conclusions

- **MOD Tool is being developed as MIRO is designed**
  - Design uses MOD Tool pieces, but not overall tool
  - MOD Tool requirements are being partially determined by MIRO
  - All previous figures are from MIRO
- **MOD Tool success will be measured in the next year**
  - Do other flight projects want to use it (and pay for additional development)?
  - Do the pieces and ideas get re-used in other software tools?