NASA/GSFC Contract NASW-99032

Photochemical Phenomenology Model for the New Millenium

Annual Report, 20 August 2000

Dr. James Bishop and J. Scott Evans

ABSTRACT

This project, entitled "Photochemical Phenomenology Model for the New Millennium" (NASW-99032), tackles the problem of conversion of validated *a priori* physics-based modeling capabilities ("legacy" computer codes) to application-oriented software for use in science and science-support activities. The modeling capabilities of specific interest are those relevant to the analysis and interpretation of planetary atmosphere observations, with particular focus on the atmospheric remote sensing data to be acquired by the Composite Infrared Spectrometer (CIRS) instrument on the CASSINI spacecraft during its Jupiter flyby and its orbital tour of the Saturnian system. Initial implementations of the software package under development, named the Photochemical Phenomenology Modeling Tool (PPMT), are aimed at construction and evaluation of photochemical transport models that execute rapidly for use in mission planning and data analysis activities. Overall, the project has followed the development outline given in the original proposal, and the Year 1 design and architecture goals have been met. Specific accomplishments and the difficulties encountered are summarized below. Most of the effort has gone into complete definition of the PPMT interfaces within the context of today's IT arena: adoption and adherence to the CORBA Component Model (CCM) has yielded a solid architecture basis, and CORBA-related issues (services, specification options, deployment plans, *etc*) have been largely resolved. Implementation goals have been redirected somewhat so as to be more relevant to the upcoming CASSINI flyby of Jupiter, with the focus now being more on data analysis and remote sensing retrieval applications than on multidimensional transport modeling capabilities.

1. *YEAR 1 ACCOMPLISHMENTS*

From the original proposal:

> "Current [photochemical transport] models are procedural driven codes that are awkward to adapt to new photochemical schemes and, in the case of the more comprehensive codes, are too bulky in execution to permit rapid turn-around in data-model comparisons and analysis. We expect that the main utility to NASA programs of the [Photochemical Phenomenology Modeling Tool (PPMT)] will be in the creation of photochemistry models that execute rapidly for use in mission planning and data analysis activities. In particular, it will be used in place of GRIFFIN to generate photochemical models of the Saturn stratosphere and the atmosphere of Titan for CASSINI/CIRS planning and testing simulations and in eventual data analysis..."

1.1 *Photochemistry Phenomenology Algorithms*

A detailed functional breakdown of algorithms for modeling the phenomena of direct interest in the initial implementations has been carried out using three independent "legacy" source codes: the GRIFFIN model of P.N. Romani (*e.g.*, Romani *et al.* [1993]), the F1D model of J. Bishop and J. Roberts [Bishop *et al.*, 1994], and the CHEM1D model of M.E. Summers (*e.g.*, Summers and Strobel [1996]). (We are especially grateful to Dr. Mike Summers for providing us with his source codes without hesitation.) All are in FORTRAN, and the core photochemical transport solution algorithm in each is based on an extended version of the classical Newton-Raphson technique (indeed, the similarity between the Romani and Summers solution schemes is somewhat surprising). As described in the original proposal, the Bishop/Roberts F1D approach provides the flexibility needed in the PPMT and the sets of algorithms within F1D are good analogues for many of the needed PPMT methods; breakdown of the Summers' CHEM1D code has been valuable, however, owing to its procedures for handling a wider variety of boundary conditions and its inclusion of algorithms for physical processes (*e.g.*, photodissociation by MUV diffuse radiation fields) not found in the current GRIFFIN or F1D codes. A full description of the Newton-Raphson algorithms as implemented in PPMT will be provided in the documentation accompanying the first "official" version to be delivered to GSFC/Code 693 at the end of the 2nd quarter of Year 2.

1.1.1 *Language Considerations*

Although the performance of C++ executable code is generally believed to be superior to that of interpreted code such as Java, it is also accepted that the time-to-market is shorter with Java development than with C++ development. Therefore it has been important for us to weigh the advantages and disadvantages of choosing one language over another for our application(s). In the 1st quarter report (Year 1) we stated that the core architecture and most if not all of the photochemistry algorithms would be implemented in C++ and that this decision was based primarily on performance considerations. Given that we have since elected to utilize the CCM as the core architecture for the PPMT, we have been forced to reevaluate this decision. CCM is still undergoing final revision within the standardization process at the Object Management Group (OMG) and third-party implementations of the CCM are not yet available (Java implementations named ZEN and OpenORB will be released to the public sometime in the next few months). In view of the consequential time and resource limitations, we decided that it would be prudent to sacrifice some performance in favor of an accelerated development. Therefore, we have chosen to implement most if not all of the PPMT architecture using the Java language (although using CORBA we can certainly implement specific components of the system in C++ if performance becomes severely reduced). We have conducted informal performance comparisons of several numerically intensive applications written in both C++ and Java. The performance tests showed that Java and C++ perform equally well on a Windows/Intel platform when performing matrix inversions and Singular Value Decomposition using matrices ranging from 3×3 to 1000×1000 floating point values. We will continue to gauge the performance of the PPMT as implementation proceeds.

1.2 *Use Case Model*

In order to develop a Domain Model for the PPMT we first constructed a Use Case Model and corresponding Use Case Specifications (*i.e.*, template documents describing a basic flow of events needed to satisfy a Use Case). The Use Case Model was developed using Rational Rose and incorporates both user interface and photochemical phenomenology requirements. Top-level use case diagrams from the Rational Rose Use Case Model are shown in Figures 1–5. A sample

Use Case Specification document written for one of the PPMT use cases is available in the 2nd quarter report. The Use Case Specifications are based on the Rational Unified Process (RUP) Use Case Specification template available from http://www.rational.com/. The Use Case Model will be available for download in the second year following deployment of the first "official" version of the PPMT system.

1.3 *PPMT Design & Architecture*

We have completed an analysis and design of the PPMT architecture using the Use Case Model and Use Case Specifications discussed above. The PPMT design derives from the CORBA Component Model (CCM). The CCM is discussed in more detail below; class diagrams (Figs 6 – 26), sequence diagrams (Figs 27,28) and a deployment diagram (Fig 29) illustrating the more important aspects of the PPMT design are provided at the end of this report along with an assessment of the overall design and planned implementations. Note that although some information has been suppressed from the diagrams for the sake of clarity, the diagrams represent actual Java code that was generated from a CORBA Interface Definition Language (IDL) compiler using a complete description of the system expressed in the CORBA IDL. Some interfaces have been left undefined because they pose little technical risk and are not required for the initial implementation testing of the PPMT system. At the present time we have implemented approximately 25% of the generated Java code that is required for the first release version of the PPMT.

1.3.1 *CORBA Component Model (CCM)*

The goal of the PPMT project is to construct a state-of-the-art distributed, heterogeneous (*i.e.*, multi-language and multi-platform), component-based architecture that provides flexible and extensible connectivity and legacy integration. To achieve this goal, we are deriving the PPMT architecture from the CCM because the CCM architecture provides mechanisms to connect distributed, heterogeneous "components" via generic, non-proprietary interfaces that facilitate dynamic system aggregation and integration/fusion of legacy systems. The CCM also provides a complete Application Programmer Interface (API) for event handling, security, persistent state, and transactions. Figures 6—11 at the end of this document provide class diagrams that show the key behavior of the CCM generic interfaces that we have implemented in Java (see "Language Considerations" above for a justification of using Java to implement the CCM). Note that the CCM specification is not yet a standard but is currently undergoing final revision; a ratified specification is expected in the first quarter of CY 2001. Since the CCM specification has not yet been accepted as a standard, there are no commercial or open source implementations of the specification available. The CCM implementation we have produced is cursory and will be replaced with a fully compliant (open source) implementation as soon as one exists. We have taken great efforts to ensure that our interpretation of the CCM specification (OMG TC Document ptc/99-10-04) is compliant (via private communication with OMG members and CPI's recent membership in the OMG) and hence we expect few problems in porting the system to a compliant CCM implementation. The interested reader should visit http://www.omg.org for more detailed information on CORBA and the CCM.

1.4 *CORBA Specifications, Deployment Issues, etc*

1.4.1 *CORBA Services*

The CORBA Common Object Services specification (version 12/09/98) defines behavior and functionality for commonly used services such as security, persistence, name resolution, transaction, *etc.* In developing the PPMT architecture we have chosen to utilize the Interoperable Naming Service (INS), Notification Service, Telecom Logging Service, and Persistent State Service (PSS). The services currently being utilized (except the PSS) are provided free for non-commercial use by Object Oriented Concepts (http://www.ooc.com). The PSS we have adopted is available from Intalio (http://www.intalio.com), also free for non-commercial use.

1.4.2 *ORB Benchmarks*

Given that the PPMT core architecture is distributed using CORBA technologies, one decision we had to make is the choice of the third-party implementation of the CORBA specifications. We evaluated several third-party implementations (details were presented in the 2nd quarter report). The evaluated implementations are:

- ORBacus OB (C++) by Object Oriented Concepts (http://www.ooc.com)
- ORBacus JOB (Java) by Object Oriented Concepts (http://www.ooc.com)
- The ACE ORB (TAO) (http://www.cs.wustl.edu/~schmidt/TAO.html)
- MICO under development for GNU http://www.mico.org/.
- JavaORB by Intalio (http://www.intalio.com)

Based on our evaluations of the listed products, we expect to use the products available from OOC. Since both Washington University and Intalio are expected to release open source CCM compliant ORBs within the next few months, named ZEN and OpenORB respectively, we will be considering a port of the PPMT implementation to one or both of these CCM compliant ORBs once a stable release is available.

1.4.3 *Deployment Issues*

The PPMT consists of a platform-independent, distributed, client/server-based architecture that subscribes to the Open Source Initiative (see http://www.opensource.org). In order to support this initiative, the PPMT will be made available from a host web site under the restrictions outlined in the GNU General Public License (GPL) (see http://www.gnu.org/copyleft.gpl.html). The deployment platform(s) will provide the capability to execute the PPMT as well as download some or all of the PPMT source code and binaries including a client, servers, relational database(s), and documentation. The execution of the PPMT on the host web site is expected to be multi-platform in order to support multi-threaded load balancing as well as concurrent processing for computationally intensive scenarios. The initial target platforms are Windows/Intel, Linux/Intel, and IRIX/Silicon Graphics. A stable software compilation environment has been constructed and is currently operational on both Windows NT 4.0/Intel and IRIX/Silicon Graphics. We will be testing the compilation environment on Linux/Intel during the 1st quarter of the Year 2. All of the software and accompanying documentation is under configuration management using the Perforce client/server configuration management tool (see http://www.perforce.com).

1.5 *User Interface Design & Implementation*

The design and implementation of the GUI has proceeded steadily although more slowly than the phenomenology aspect of the PPMT. The design and implementation of the GUI has been guided primarily by the Use Case Specifications that were written during the 1st and 2nd quarters. The technologies we are currently using for the PPMT GUI are HTML, XML, and Java. These particular technologies have been selected so that the PPMT GUI can be executed through a browser or as a standalone Java application. As implementation proceeds we will ensure that the PPMT GUI functions as both an application and an applet. The PPMT GUI is being developed with the Forte for Java Integrated Development Environment (IDE) provided without license or cost to the general public by Sun Microsystems (see http://www.sun.com/forte/ffj/).

2. *YEAR 2 GOALS & TASKS*

2.1 *Application Goals & Tasks*

- Complete development of initial implementations (1-D photochemical transport modeling of planetary stratospheres using empirical atmosphere models and the Newton-Raphson algorithm) (1^{st} & 2^{nd} quarters)

- Validation and testing of initial implementations via detailed analysis of CASSINI/CIRS Jupiter flyby data (*e.g.*, CIRS/JUPITER Far-IR Composition Study data, CIRS/Jupiter Mid-IR Composition Map data) (2^{nd} & 3^{rd} quarters)

- Delivery of a completed, validated "official" PPMT Jupiter version to NASA/GSFC Code 693 (end of 2^{nd} quarter)

- Construction (*i.e.*, recoding of legacy algorithms) and implementation of photon transport algorithms, for correct evaluation of stratospheric MUV photodissociation, for calculation of molecular rovibronic emissions (*e.g.*, ethane & acetylene), and for calculation of emissions from chemically-produced metastable species (3^{rd} & 4^{th} quarters)

- Begin extension of photochemical modeling methods to thermospheric/ionospheric region and addition of methods for calculating corresponding emissions at UV and visible wavelengths (3^{rd} & 4^{th} quarters)

- Delivery of the next "official" PPMT version including basic photon transport algorithms (MUV stratospheric photodissociation and IR molecular band spectral radiances) to NASA/GSFC Code 693 (end of 4^{th} quarter)

2.2 *Software Development Goals & Tasks*

- Complete implementation of Graphical User Interface (GUI) for the PPMT that satisfies the Use Cases needed to support detailed analysis of the CASSINI/CIRS Jupiter flyby data sets

- Incorporate type definitions and use of the CORBA Persistent State Service (PSS) to enable archiving of phenomenology inputs as well as post-processing outputs

- Incorporate use of XML to maintain disconnect between the PPMT GUI and the PPMT server and for standardized configuration of CCM component "Homes"

- Establish test suite for automated component and integration tests of the PPMT

- Develop web pages for deployment platform that provides access to GUI, PPMT servers, compiled middleware products (ORB and CORBA Services), Application Programmers Interface (API), and documentation

- Port PPMT to open source CCM implementation (when available)

- Port PPMT to Linux/Intel platform

- Implement security measures in the PPMT by incorporating use of the CORBA Security Service and the CORBA Firewall specification. These security measures are needed primarily by users and developers that work within a network that is protected by a firewall or IP packet filter.

3. *ASSESSMENT OF WORK TO DATE*

The following topics are discussed:

- Development direction for implementations and applications in Years 2 & 3, which has shifted from that outlined in the original proposal, in which the Year 2 effort (1st year of the software system construction phase) was centered on finalizing the implementation of an F1D-based algorithm and the addition of alternate solution approaches/algorithms, leading to inclusion of 2-D coupled photochemistry and flow-field modeling applications

- Rational for selection of CCM to define the basic architecture, from the perspective of software development

3.1 *Year 2 & Year 3 Directions*

Over the past year, work on the PPMT has proceeded slowly. While there is the obvious desire to lay out an architecture for an implementation sequence that will eventually embrace all of the relevant phenomenology issues, the initial steps must focus on constructing the components, methods and GUI capabilities needed for applications of immediate relevance and deciding on the corresponding implementations will be has taken a lot of time and thought. With hindsight, it is now recognized that there have been two major obstacles to drafting the basic design and architecture:

1. the continuing rapid evolution of CORBA-related issues, and
2. the argumentative issue as to what the development direction should be during the next two years.

The issues and development pressures created by rapidly evolving segments within the general IT arena (distributed interface specification and management, programming language developments, security issues, *etc*) have been identified and our handling of them have been described in previous quarterly reports and elsewhere in this document. The focus of this discussion is on giving a clear description of the development direction choices from an applications-and-use perspective and on why we have selected the direction we will follow in the next two years.

Within the context of this project, there are two complementary approaches to atmospheric data modeling and analysis based on *a priori* physics-based algorithms: dynamics-based modeling and modeling of observables. Dynamics-based modeling here refers to the continuing development of modeling and simulation codes that attempt to encompass all the basic physical processes underlying the overall structure and multi-scaled (temporal, spatial, compositional) variations within an atmosphere: horizontal transport, vertical transport, photochemistry, *etc.* A large number of studies have been published to date illustrating various perspectives and degrees of comprehensiveness along these lines (as noted in the original proposal); examples include

- 2D photochemistry with eddy mixing and advective transport (*e.g.*, Dire [1997])
- thermospheric general circulation models (*e.g.*, Bougher *et al.* [1999])

These are valuable for in-depth analyses of the couplings among the underlying physical processes and their cumulative impacts (*e.g.*, diurnal, seasonal & solar cycle variations), for comparative studies of atmospheric composition variations, for determining the limits on modeling capabilities related to uncertainties in various inputs, simplifying assumptions and algorithmic approximations, *etc.* However, such models can be very computationally intensive and are not appropriate when attempting to infer atmospheric conditions directly from observational data.

The complementary approach centers on using observational data to *infer* current atmospheric conditions through the use of *a priori* algorithms focused on the microphysical and transport processes giving rise to the observables, *e.g.*, collisional excitations, chemical reactions, radiative transport. This is the approach taken when utilizing remote sensing data (*e.g.*, IR emissions from stratospheric regions, FUV emissions from upper atmospheric regions) to construct (or retrieve) an empirical "picture" or "snapshot" of the atmosphere at the locations and times of the measurements, particularly when such data are acquired in large volumes or in an automated measurement sequence (subject to the obvious requirement that the microphysical processes and parameters are well understood and the associated algorithms have been validated, and the available observables provide adequate information to reliably infer the environmental parameters/atmospheric conditions of interest, *e.g.*, vertical profiles of atmospheric temperature and abundances of tracer species). The differences between the two approaches were not acknowledged in most earlier studies of outer planet atmospheres (including the Voyager flybys). The relevant data sets typically consisted of small numbers of measurements and were viewed as "representative" of mean conditions. Earth-based data in particular were effectively hemispheric averages, with perhaps partial resolution of latitudinal or solar-zenith angle variations. Analyses of these datasets with 1-D photochemical transport models aimed at providing constraints on "mean" atmospheric parameters (*e.g.*, minor species mixing ratio profiles) within the limitations imposed by either limited computing capabilities, limited instrument capabilities, or both. In some cases the older data sets were subjected to dynamics-based modeling studies, pointing out the limitations of 1-D modeling and perhaps getting some insight into horizontal transport conditions; however, the main focus in such studies has generally been the development of the multi-dimensional transport algorithms themselves.

Despite the great increases in computer capabilities, the value, diversity and shear volume of remote sensing data that can be collected by imaging instruments with high spectral resolutions force the recognition of the differences between the two approaches; use of computationally intensive dynamics-based modeling codes to analyze large volumes of remote sensing data acquired, say, by an orbiter with continuous limb-scanning or disk-imaging instrumentation, is both impractical *and* inappropriate. Rather, the thrust of the analysis is to retrieve an empirical quantitative model of the atmosphere in its current state, as described above. A good analogy here is offered by meteorology forecasting, where large volumes of the most current data recorded (both conventional and remote sensing) are rapidly analyzed via retrieval algorithms and assimilated to generate as complete a "picture" of current atmosphere (troposphere) conditions as possible, which is then used to predict or forecast future development.

This has been taken as the primary thrust of this project – i.e., the principal aim is to assemble (or construct) and implement physics-based photochemistry and remote sensing retrieval algorithms that return atmospheric structure constraints or parameters on the basis of available data, in effect converting measurements of a particular location within a particular time interval to environmental parameters. In the original proposal, the aim of having the design & architecture encompass both remote sensing-related and dynamical-modeling related algorithms and use cases was discussed, however the potential for implementation conflicts was not recognized. Thus, planned development for the 2nd & 3rd years of the project now aims more toward the incorporation of radiative modeling algorithms in the PPMT rather than the inclusion of multidimensional (horizontal) transport modeling algorithms. It is hoped that it will be possible to construct methods for at least 2-D dynamical (advection) modeling (McMillan [1992], Dire [1997]) during the 3rd year, which will require expansion (revision) of the overall design and architecture. There is also the obvious consideration that the initial implementations will be focused on the Cassini Jupiter flyby data, for which a 1-D approach to the photochemical modeling of observables (including higher order hydrocarbons such as methyl acetylene and benzene and nonhydrocarbons like H_2S and CO_2) is suitable; multidimensional transport modeling capabilities will not be called for until Titan measurements begin in 2005. The high spectral resolution and good spatial resolution of the CASSINI/CIRS instrument in the Jupiter flyby measurement sequences (*e.g.*, the CIRS/Jupiter Mid-IR Composition Map and the CIRS/Jupiter Far-IR Composition Study) will yield not only latitudinal profiles (if measured) of species heretofore unresolved but also complete maps of stratospheric pressure-temperature profiles and far better determination of the vertical profiles of the major hydrocarbons (*e.g.*, C_2H_2, C_2H_4, C_2H_6) than have been possible to date.

In addition to CIRS hydrocarbon analysis and retrieval applications, we are also very interested in extending the PPMT to include physical observables at deeper and shallower (higher altitude) pressure regions; the former includes CIRS data acquired at far-IR wavelengths (*e.g.*, H_2S) and CASSINI/VIMS data, while the latter pertains to the thermospheric and ionospheric data to be acquired by CASSINI/UVIS. Each of these anticipated sets of use cases, *i.e.*, use cases requiring handling of radiative transport and/or ionospheric/thermospheric modeling capabilities, are encapsulated in the attached class diagrams (Figs 12—24). These diagrams illustrate the architecture that has been laid out (in the course of much argument) so as to be equally suitable for electron impact and ionospheric chemistry phenomena as for stratospheric FUV-MUV photochemistry. Photon transport is important in both regions, so it has been included on an equal footing with electron and molecular species transport.

3.2 *Selection of CCM*

Although "requirements creep" and redirection of development are common and unavoidable, it is not trivial to design a software architecture that is entirely resilient to change. We have stated that our goal has been, and still is, to develop a component-based architecture that is flexible, extensible, and amenable to legacy integration. Achieving this goal is significantly more difficult than simply adding a few more Use Cases to the model. Experience has shown that not all Object-Oriented software architectures are responsive to all degrees of new requirements. Experience has also shown that attempting to capture all possible requirements in a grandiose Use Case analysis will usually result in "analysis paralysis", an intractable architecture, and no source code. Hence, it is necessary to take great care when deciding how many Use Cases are "enough", which Use Cases involve the greatest technical risks, and how to utilize development iterations to avoid "stuck" analysis and to achieve flexibility, extensibility, and the Holy Grail of reusability.

Our conscious decision to base the PPMT architecture on the CORBA Component Model is a first step towards satisfying our self-imposed architectural requirements. It is not the case, however, that a CCM-based architecture is immediately useable and reusable. The CCM only serves as the framework for more specialized systems. To guide us in our extension of the CCM we have employed numerous software patterns that have been identified as established solutions for specific software problems. For example, we have employed the Quantity Pattern [Fowler, 1997] to encapsulate physical quantities, units of measure, conversion algorithms, and geometrical objects. We have also made extensive use of association classes [Booch *et al.*, 1999] to model physical interactions such as gravitation, magnetism, reaction phenomenology, scattering phenomenology, and impact phenomenology. Finally, we have incorporated use of the Iterator pattern [Gamma *et al.*, 1995] as a general approach for modeling *n*-dimensional scalar and vector fields. By combining the flexibility of the CCM with well-documented solutions offered by software patterns, we can ensure that the PPMT is both useable and reusable.

In order to accommodate Use Cases for radiative modeling as well as multidimensional transport modeling we have attempted to generalize the PPMT architecture so as to not exclude any aforementioned functionality that may become of greater importance in the 2^{nd} or 3^{rd} year. More specifically, we have designed the iterator-based scalar and vector fields to allow for *n*-dimensional field representation in the event that time and resources permit us to implement dynamics-based modeling. Likewise, we have included interfaces specific to particle impact and scattering in order to facilitate inclusion of radiative and energetic particle transport. Note that the heterogeneous, distributed nature of CORBA and the CCM in particular will enable us to integrate legacy components into the PPMT in a plug-n-play fashion.

REFERENCES

Bishop, J., *Flexible code metastructures for photochemical modeling (final technical report)*, NASA SBIR Contract NAS5-38012 (Phase I), 15 July 1994. *See also*: Bishop, J., J. Roberts, P. Marsden, and C. Biczel, *A prototype flexible photochemical modeling code for atmospheric science applications: Science notes, user's guide and reference materials*, attachment to final technical report, 15 July 1994.

Booch, G., J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 1999.

Bougher, S.W., S. Engel, R.G. Roble, and B. Foster, Comparative terrestrial planet thermospheres 2. Solar cycle variation of global structure and winds at equinox, *J. Geophys. Res.* **104**: 16591-16612, 1999.

Dire, J.R., *A two-dimensional photochemical transport model for the stratosphere of Titan*, Ph.D. dissertation, Johns Hopkins Univ., 1997.

Fowler, M., *Analysis Patterns*, Addison Wesley, 1997.

Gamma, E., R. Helm, R. Johnson, and J. Vissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley, 1995.

McMillan, W.W., *Revelations of a stratospheric circulation: The dynamical transport of hydrocarbons in the stratosphere of Uranus*, Ph.D. dissertation, Johns Hopkins Univ., 1992.

Romani, P.N., J. Bishop, B. Bezard, and S. Atreya, Methane photochemistry on Neptune: Ethane and acetylene mixing ratios and haze production, *Icarus* **106**: 442-463, 1993.

Summers, M.E., and D.F. Strobel, Photochemistry and vertical transport in Io's atmosphere and ionosphere, *Icarus* **120**: 290-316, 1996.

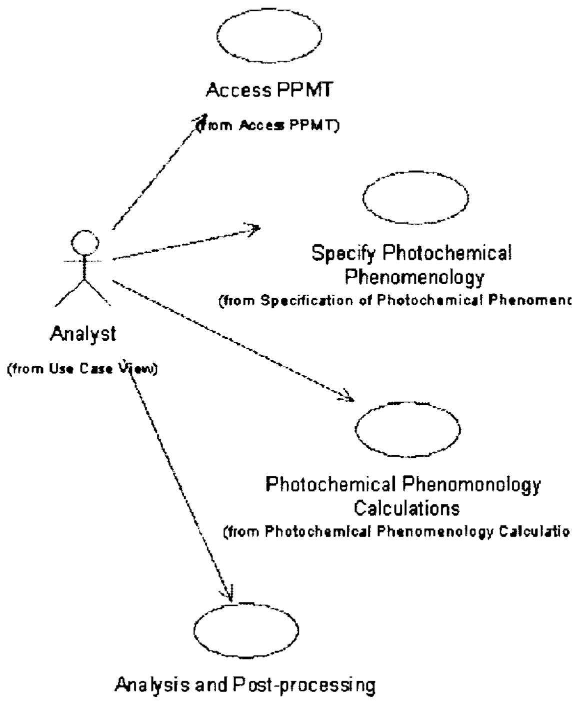

Figure 1 Main Use Case diagram from the PPMT Use Case Model.

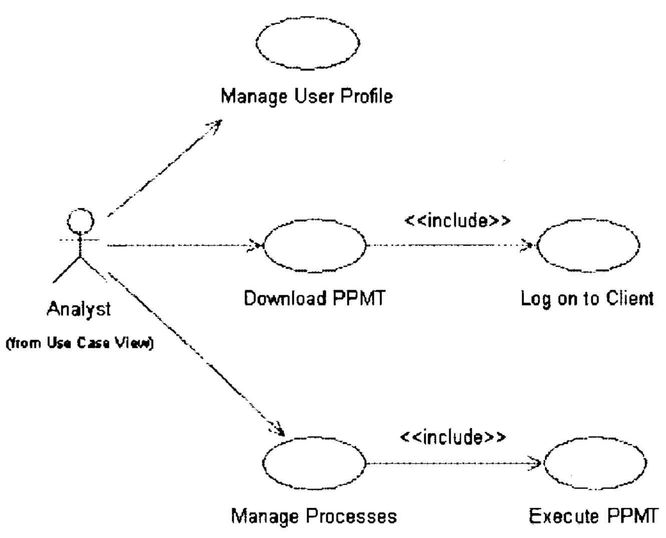

Figure 2 Access PPMT Use Case diagram.

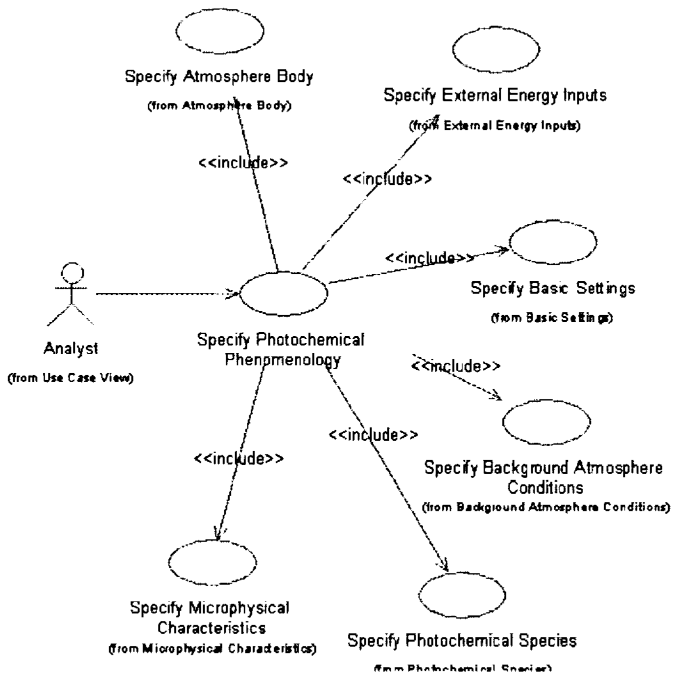

Figure 3 Specify Photochemical Phenomenology Use Case diagram.

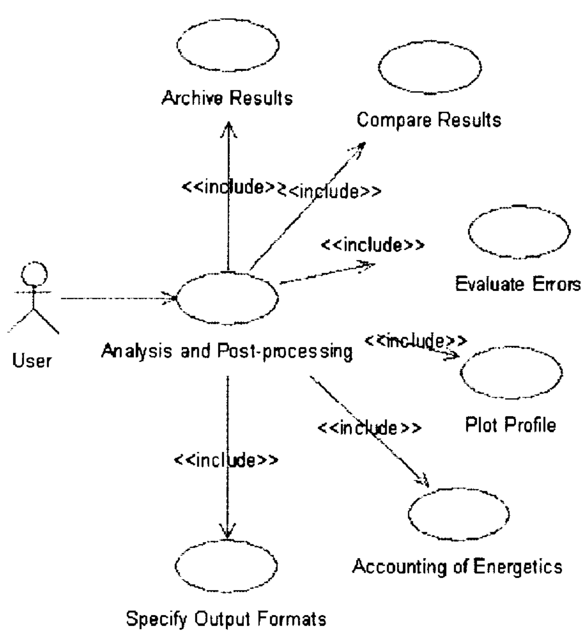

Figure 4 Analysis and Post-processing Use Case diagram.

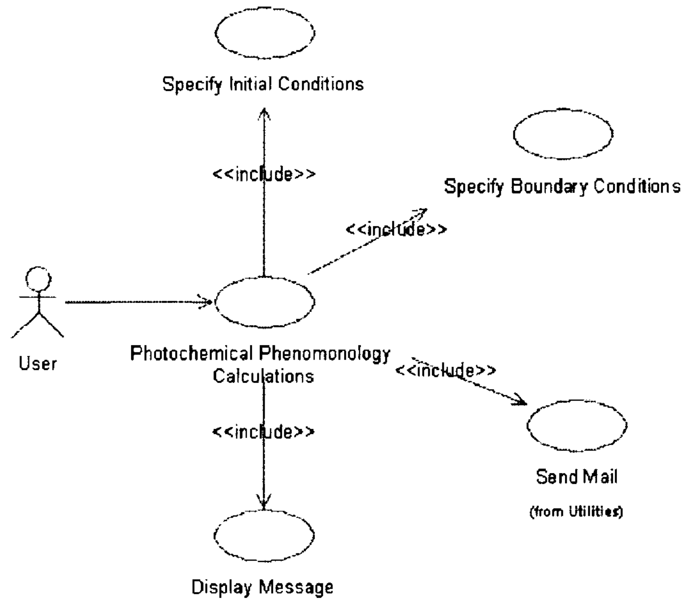

Figure 5 Photochemical Phenomenology Calculations Use Case diagram.

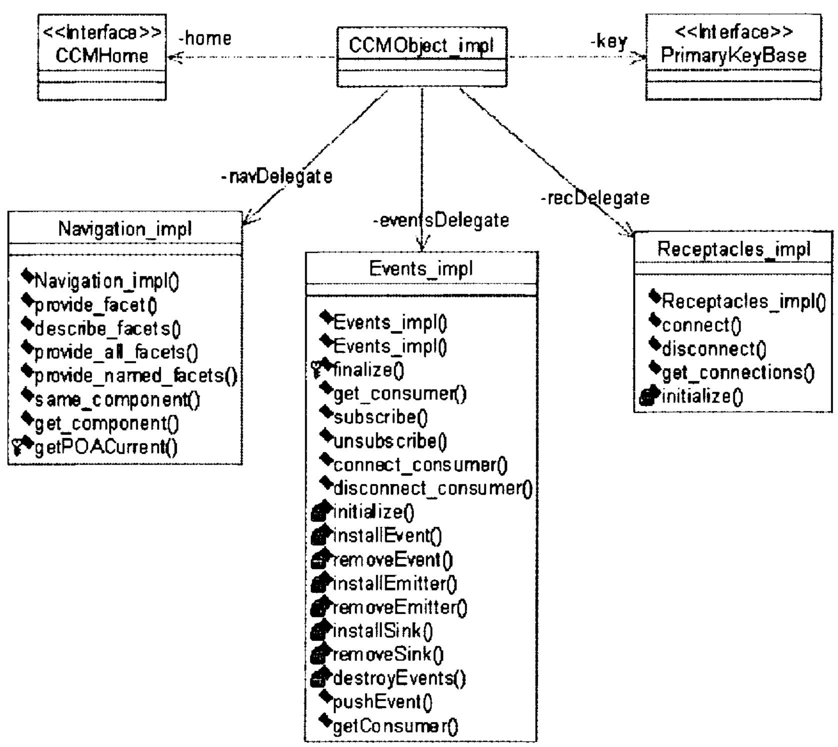

Figure 6 Main class diagram of the classes required to implement the CCMObject interface.

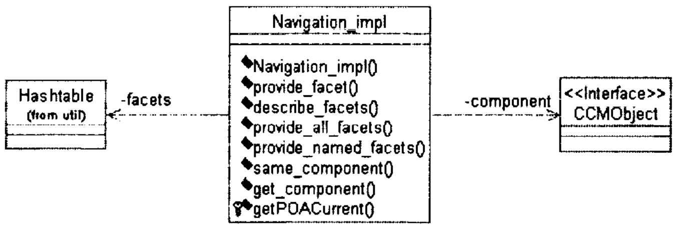

Figure 7 Details of the implementation class for the CCM Navigation interface.

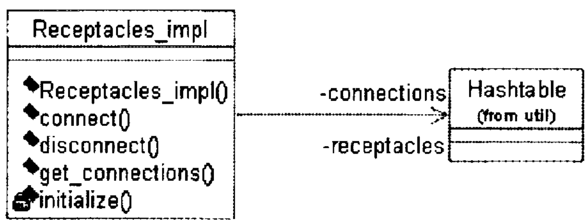

Figure 8 Details of the implementation class for the CCM Receptacles interface.

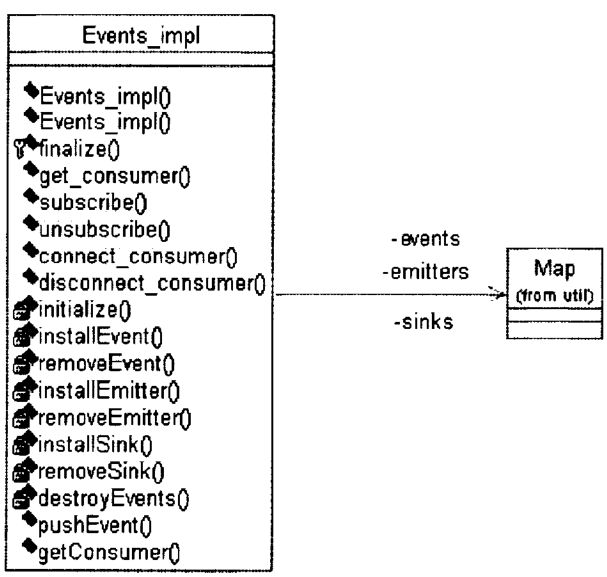

Figure 9 Details of the implementation class for the CCM Events interface.

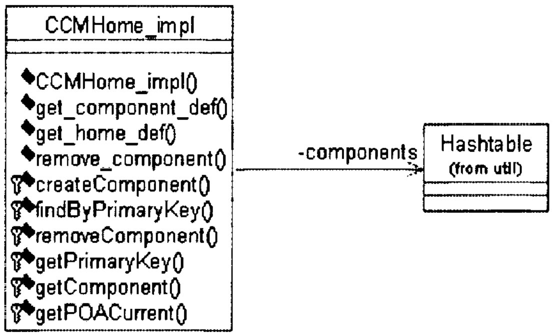

Figure 10 Details of the implementation class for the CCMHome interface.

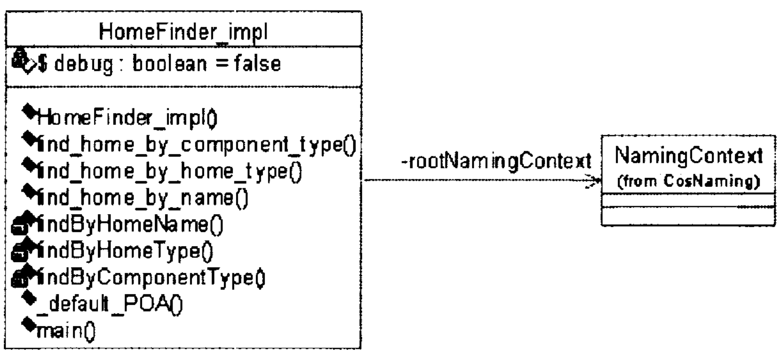

Figure 11 Details of the implementation class for the CCM HomeFinder interface.

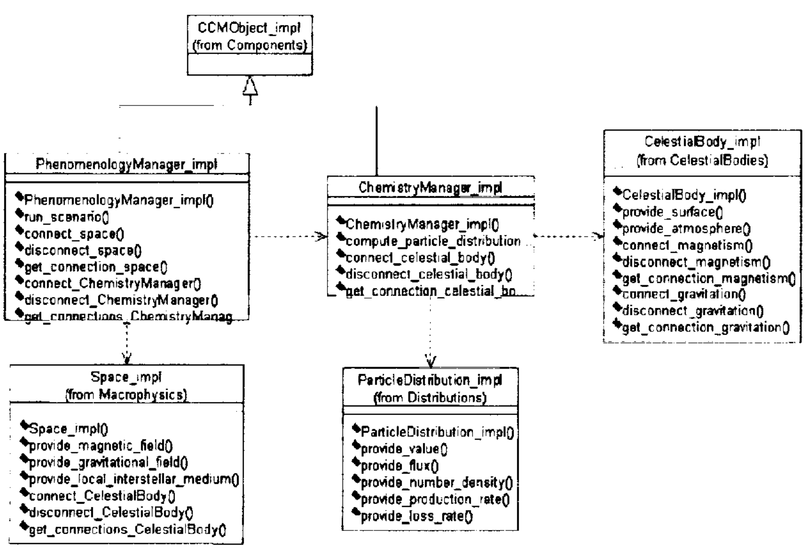

Figure 12 Class diagram showing the main controllers of the PPMT architecture.

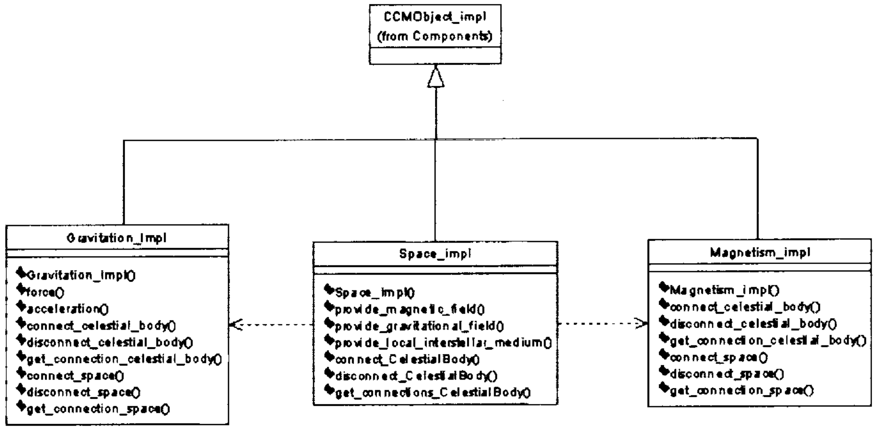

Figure 13 Class diagram showing the container class for macrophysical fields.

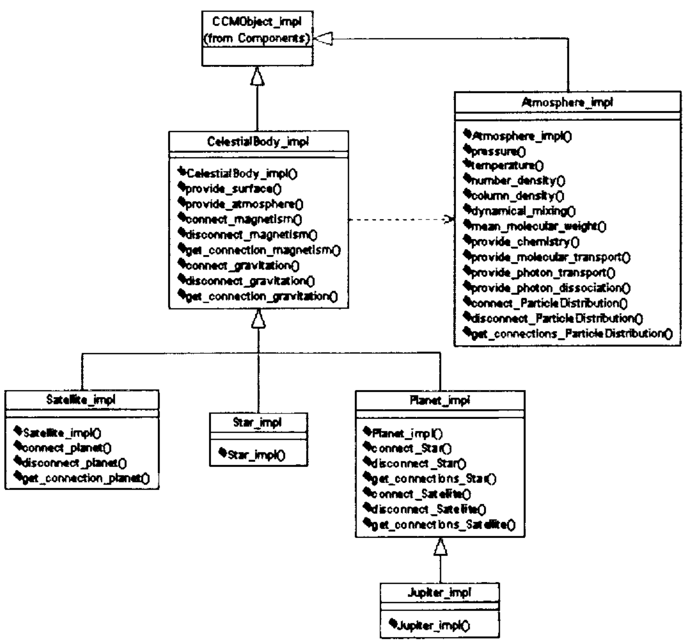

Figure 14 Class diagram showing specialized Celestial Body interfaces.

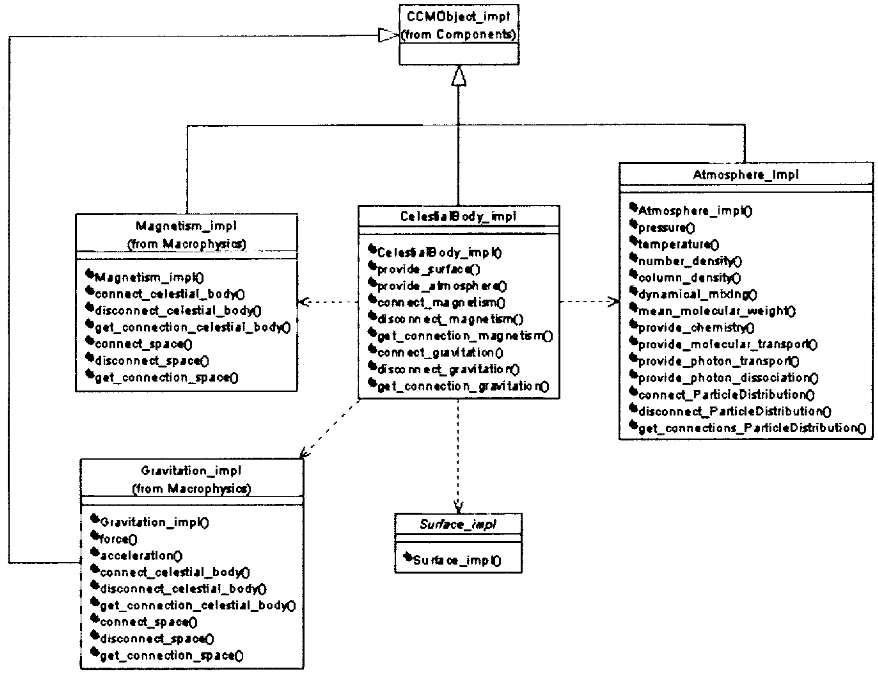

Figure 15 Class diagram showing associations between CelestialBody and related interfaces.

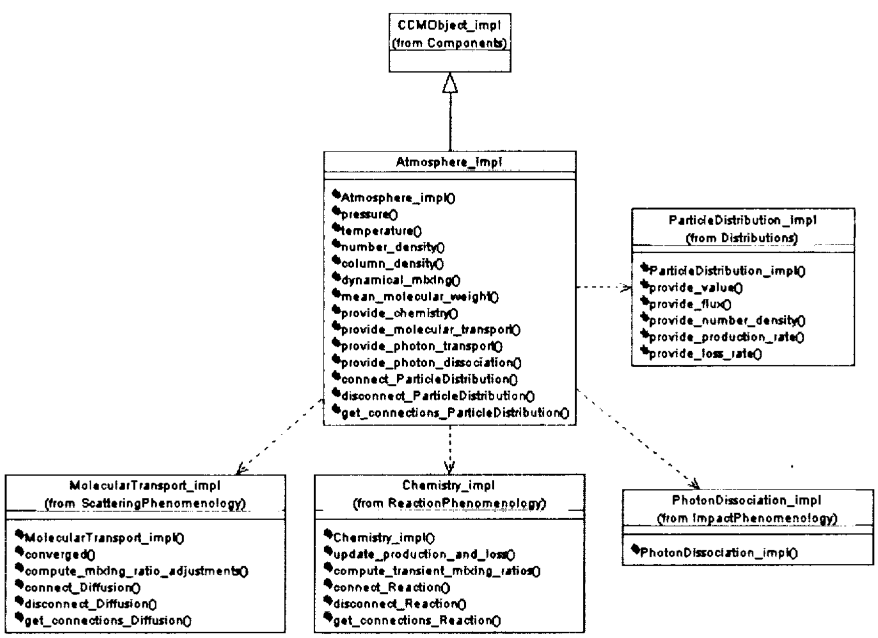

Figure 16 Class diagram showing details of Atmosphere interface.

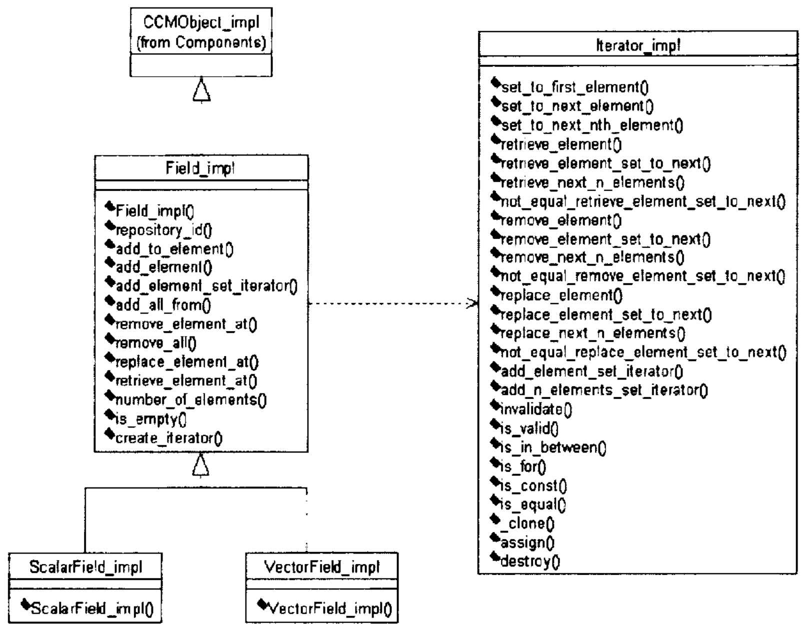

Figure 17 Class diagram showing details of generic field interfaces.

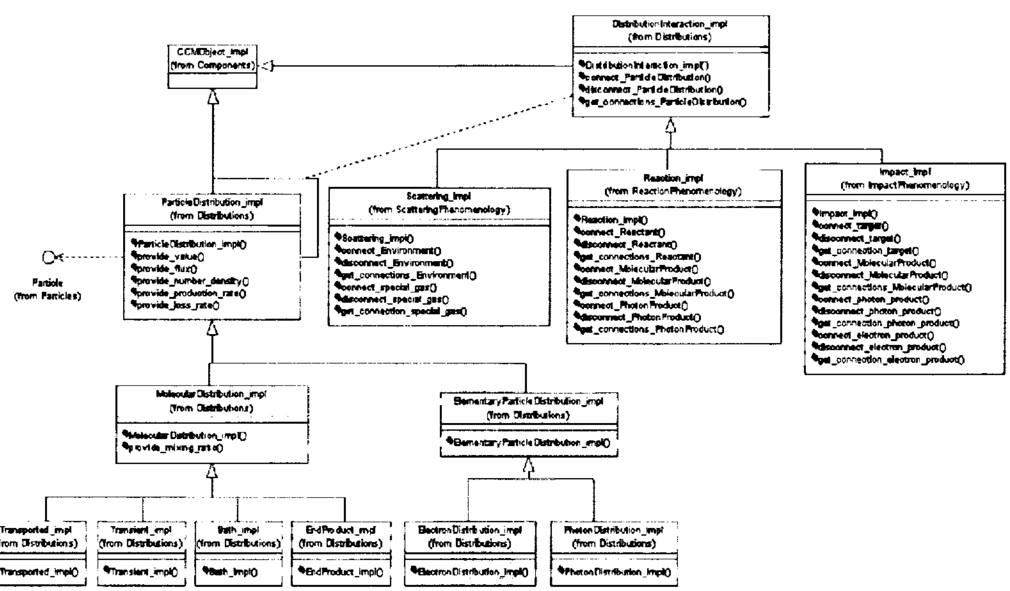

Figure 18 Class diagram showing kinetic distribution function interfaces.

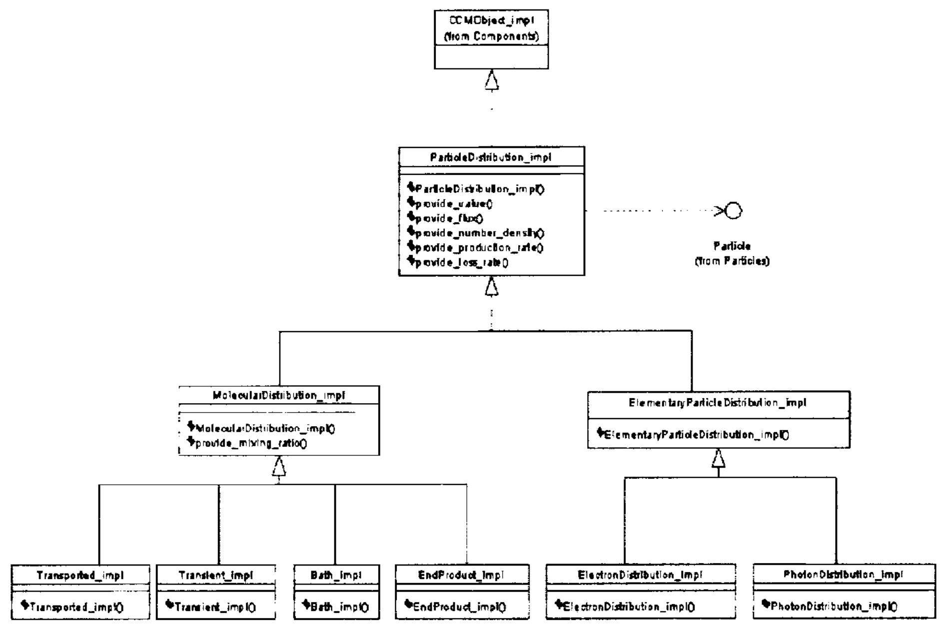

Figure 19 Class diagram showing specialized kinetic distribution function interfaces.

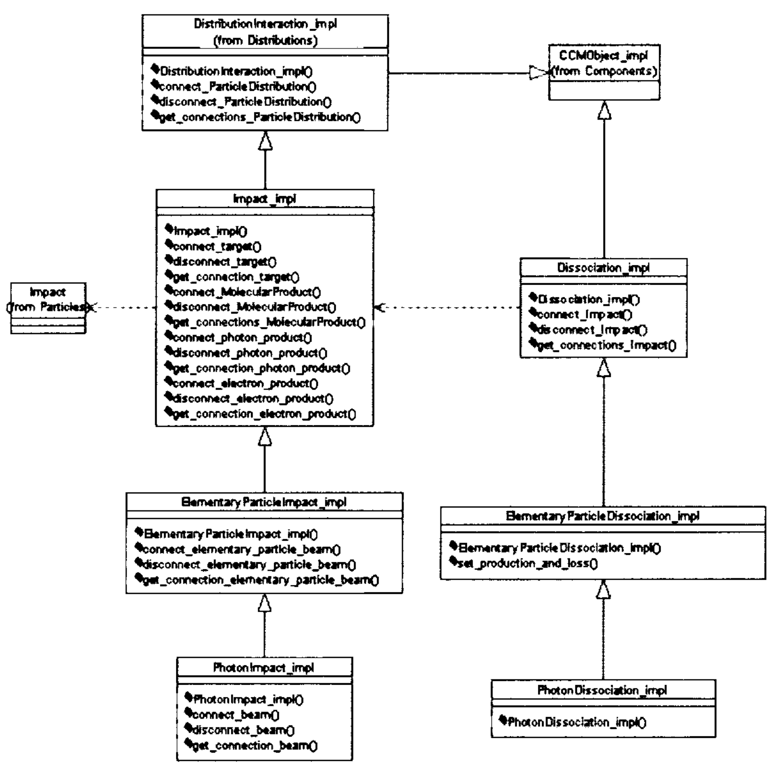

Figure 20 Class diagram showing impact phenomenology interfaces.

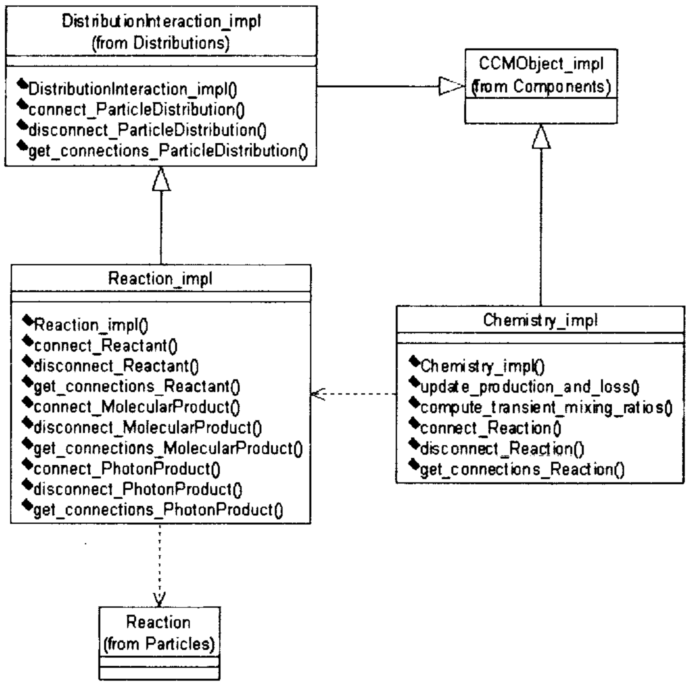

Figure 21 Class diagram showing reaction phenomenology interfaces.

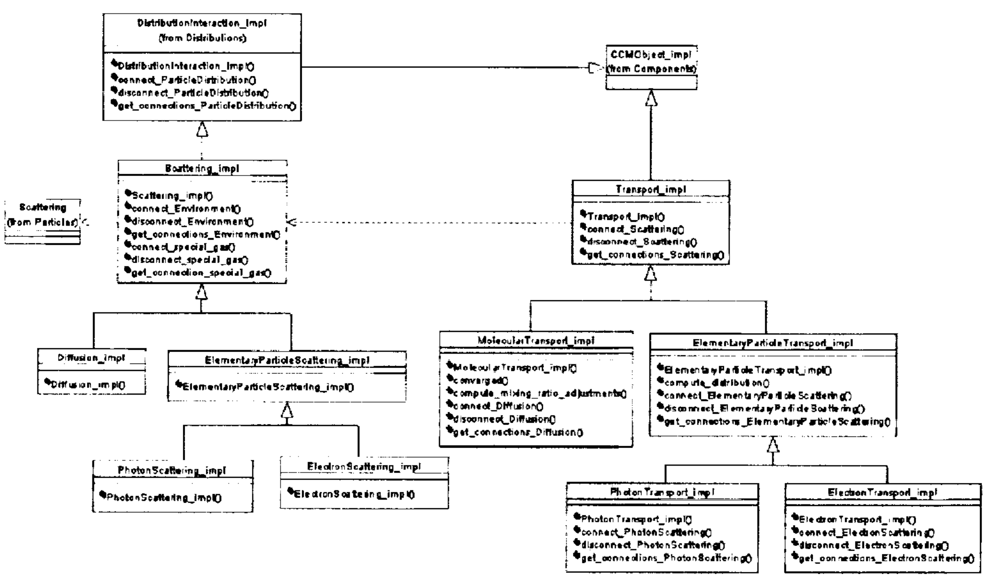

Figure 22 Class diagram showing scattering phenomenology interfaces.

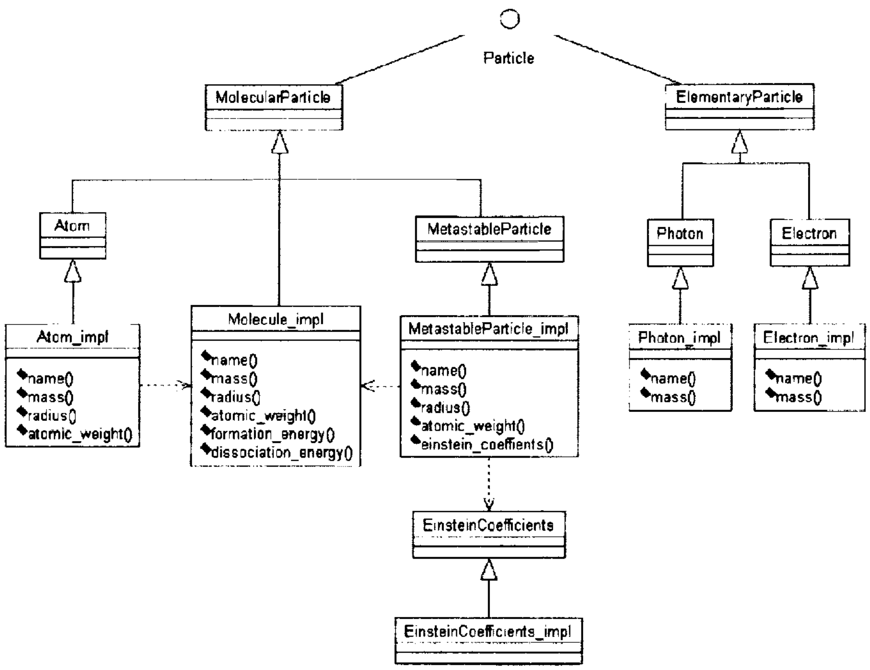

Figure 23 Class diagram showing all supported particle types and their attributes.

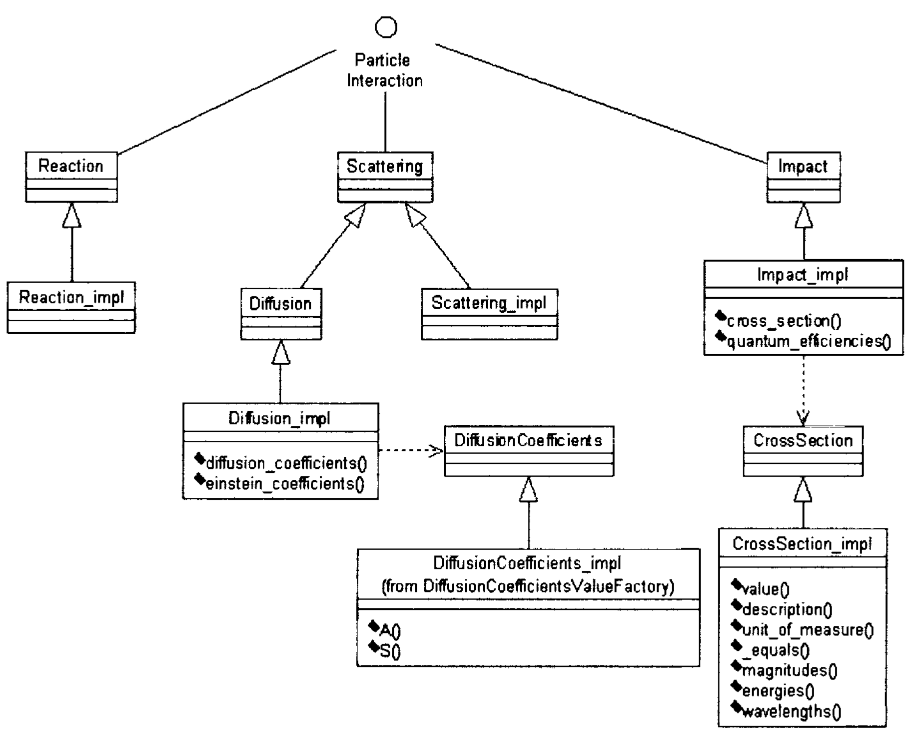

Figure 24 Class diagram showing all supported particle interaction types and their attributes.

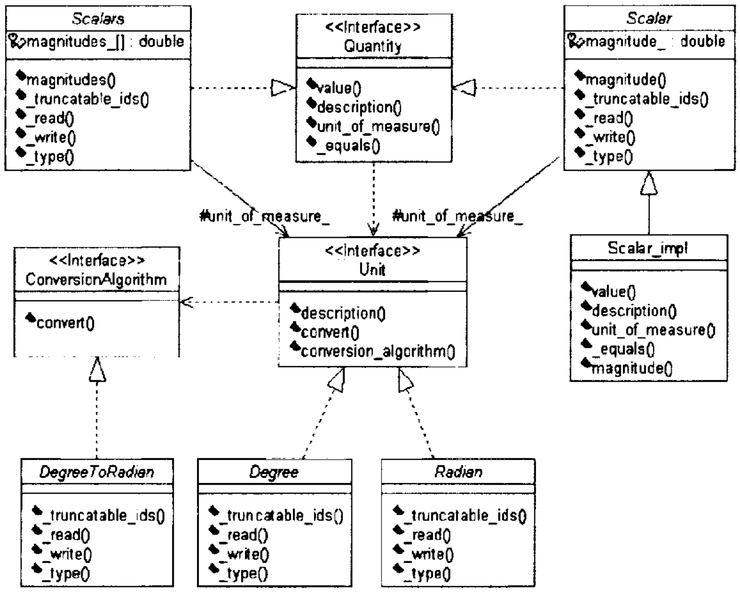

Figure 25 Class diagram showing interfaces for the Quantity pattern.

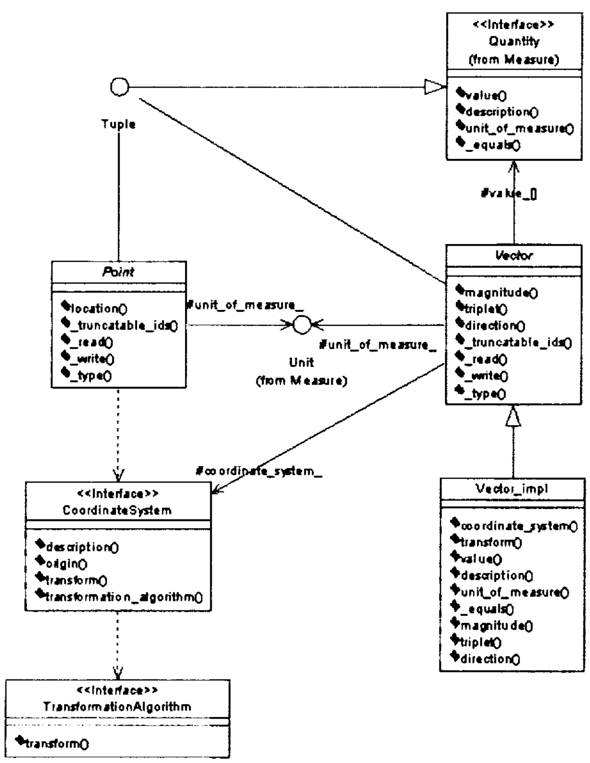

Figure 26 Class diagram showing use of the Quantity pattern for geometrical interfaces.

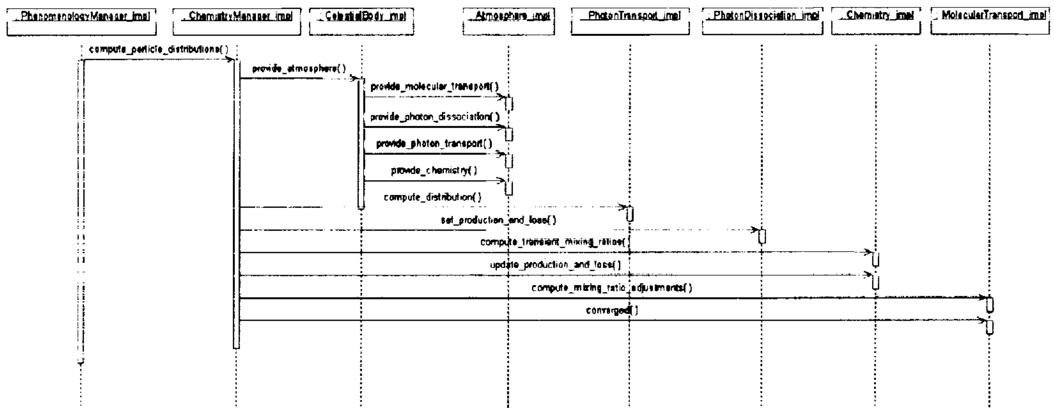

Figure 27 Sequence diagram for the Photochemical Phenomenology Calculations Use Case.

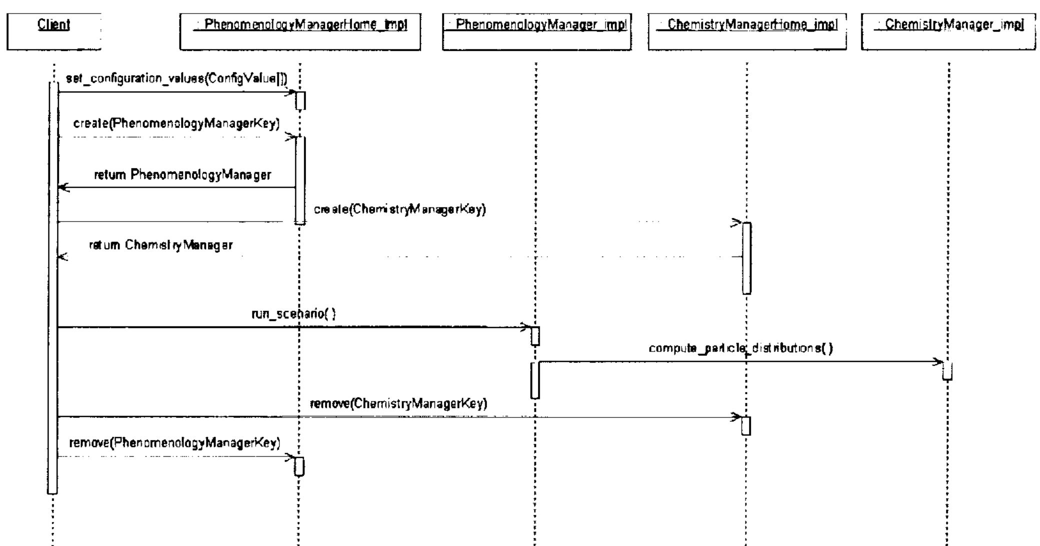

Figure 28 Sequence diagram for the Execute PPMT Use Case.

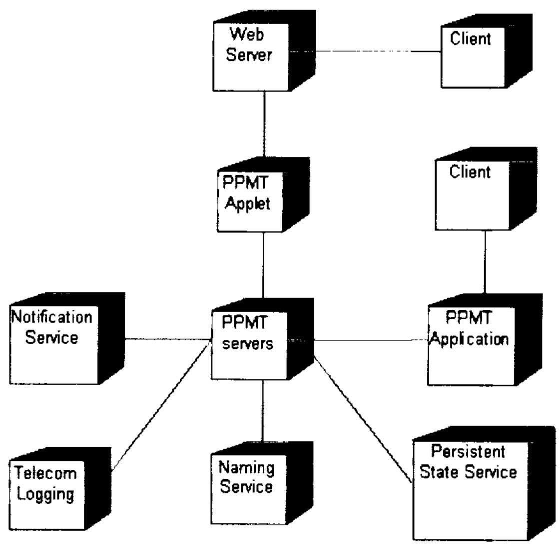

Figure 29 Deployment diagram showing several deployment configurations and client accessibility.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 18 August 2000	3. REPORT TYPE AND DATES COVERED June 2000- August 2000 – Annual Report (8/99-8/00)

4. TITLE AND SUBTITLE
Photochemical Phenomenology Model for the New Millennium

5. FUNDING NUMBERS
NASW-99032

6. AUTHORS
Dr. James Bishop and J. Scott Evans

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computational Physics, Inc.
2750 Prosperity Avenue, Suite 600
Fairfax, VA 22031

8. PERFORMING ORGANIZATION REPORT NUMBER
5108-4

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NASA
Goddard Space Flight Center

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
4

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This project tackles the problem of conversion of validated *a priori* physics-based modeling capabilities, specifically those relevant to the analysis and interpretation of planetary atmosphere observations, to application-oriented software for use in science and science-support activities. The software package under development, named the Photochemical Phenomenology Modeling Tool (PPMT), has particular focus on the atmospheric remote sensing data to be acquired by the CIRS instrument during the CASSINI Jupiter flyby and orbital tour of the Saturnian system. Overall, the project has followed the development outline given in the original proposal, and the Year 1 design and architecture goals have been met. Specific accomplishments and the difficulties encountered are summarized in this report. Most of the effort has gone into complete definition of the PPMT interfaces within the context of today's IT arena: adoption and adherence to the CORBA Component Model (CCM) has yielded a solid architecture basis, and CORBA-related issues (services, specification options, deployment plans, *etc*) have been largely resolved. Implementation goals have been redirected somewhat so as to be more relevant to the upcoming CASSINI flyby of Jupiter, with the focus now being more on data analysis and remote sensing retrieval applications.

14. SUBJECT TERMS Photochemical Phenomenology Modeling Tools (PPMT)	15. NUMBER OF PAGES 23
	16. PRICE CODE $139,449

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	SAR