Chemical Transport Modeling Framework(CTMF)

Brice Womack, Megan Damon

10.12.2006

NASA GSFC

Code 610.3

Northrop Grumman / IT / TASC

CTMF Presentation Outline

- CTMF
  - Project Goals
  - Completed to Date
    - Model analysis
    - Prototype implementation
  - Framework Overview
    - CTMF Decomposition
    - CTMF Manager
    - Framework Components
  - Community Issues
  - Next Steps
    - Framework
    - Community component integration
  - Discussion

CTMF Goals

- Increase interoperability and productivity - simpler integration of new science
- Increase maintainability - consolidated code base, standard tests, builds, etc
- Develop/provide common functionality - Met-fields, TP-Core, Fast IO, SM-Gear, etc
- Lower total cost - develop and maintain a standards based (ESMF compliant) infrastructure

Analysis of models

- Analyzed GMI, GEOS-Chem, GOCART and Goddard CTM to:
  - Outline required major functional components
  - Identify primary I/O variables required
  - Understand model's use of OpenMP, MPI and common blocks.
  - Document -
    - component interfaces
    - flow of control between components

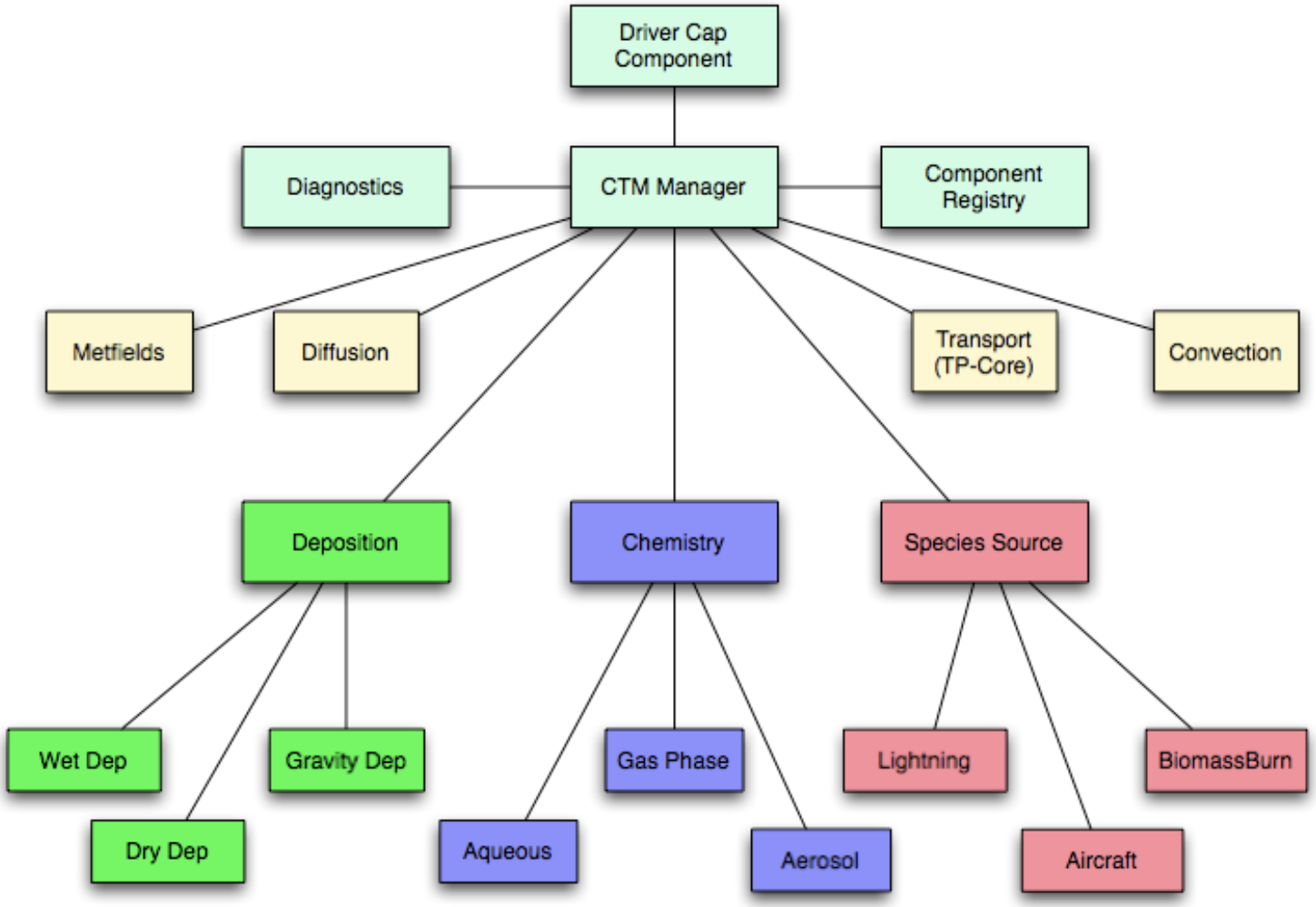
Cross component Analysis

- Analyzed input/output requirements for components from each model application (GMI, GEOS-Chem, GOCART and GSFC-CTM) and created spreadsheet comparing component data input/output interfaces.
- Analyzed model application (GMI, GEOS-Chem, GOCART and GSFC-CTM) implementations and documented:
  - Potential issues likely to be encountered during framework integration.
  - Potential areas of extracting/leveraging common functionality.

Prototype development of CTMF

- Created Sourcemotel project for CTM/ESMF task.
- Began development of supporting unit tests for CTM framework using FUnit.
- Prototyped initial ESMF States and Component interfaces for major components including - Diagnostics, Emissions, Diffusion and Transport components.
- Established CTM framework's basic flow of control, time management and coupling of base CTM components.
- Developed supporting utility functions for component state initialization and testing.
- Explored options of various ESMF and GEOS-Generic functions for supporting CTM Framework functionality.
- Worked on design to allow both standalone execution driven with file based metfields and coupled execution linked to external atmospheric model.
- Developing framework documentation using Protex templates to allow auto-generation of reference API.

CTMF Component Decomposition



CTM Manager is an ESMF component that can be invoked by the CTM driver or by another ESMF model like GEOS-5

CTMF Manager

- The CTM component manager specifies:
  - The experiment configuration(Aero, Trop, Strat, Tracer, etc)
    - Components to be used - dynamically allocated
    - The order of component execution
  - Automatic ESMF data coupling between components
    - Standardized CF naming conventions for data exchange

Automated tests, across multiple HPC systems, are included as an integral part of the frameworks development. Nightly build and tests ensure rapid identification and isolation of bugs introduced during development.

Prototype Components Implemented…

- Chemistry
  - Aqueous
  - Aerosol
  - Gas Phase
- Convection
- Deposition
  - Wet
  - Dry
  - Simple
- Diagnostics
- Diffusion
- MetFields
- Settling
- Species Source
  - Lightning
- Transport
- CTM Manager

Each component builds as dynamic library, uses standard CTM utilities and includes unit tests that validate implementation.

Status of Community Issues

- Handling surface emissions in chemistry solver
  - Approach A: Each component responsible for maintaining specified boundary conditions.
  - Approach B: One component resets boundary conditions - probably good enough. (GMI current situation)
  - Approach C: Boundary conditions reset after each component
  - Approach D: Boundary conditions reset at beginning/end of each time step. This appeared to be the consensus solution.
- Status – Framework will provide reset ability through component couplers in time step flow of control.

Status of Community Issues

- Representation issues
  - Some higher fidelity models will require more complex state than the current 4D array of concentrations.
    - E.g. components inside/outside liquid water, multi-column state information.
    - Primarily an issue for the infrastructure to support data exchange.
  - Status – State representation is relatively generic and current 4D concentration representation will be refactored later if necessary to support new components. Minimal impact is expected.

Status of Community Issues

- FastJx componentization requested
  - Integration of FastJx as a component for use in framework was requested.
- Status – worked with Xin to reorganize FastJx into component init/run/finalize structure. ESMF wrapper still to be added.

Status of Community Issues

- Naming conventions between framework components (Field names)
  - Need standard names to refer to fields/arrays exchanged between components.
  - Recommendation was made to adopt CF conventions where defined and gather community agreement on those not defined by CF.
- Status – Document detailing input/output fields for components from each of the community models was produced.
  - CF conventions were searched and standard names were identified/documented for matching input/output fields.
  - TODO – review document with community to gather agreement for field names not currently defined in CF conventions.

Status of Community Issues

- Chemistry component needs means to express reaction rate and Jacobian function in a manner independent of particular solver.
  - This may require opaque handles from ESMF
- Status – still under consideration/design. Not an issue for current framework activities, will revisit when solver refactoring begins

Status of Community Issues

- Example Documentation for Componentization
  - CTM community would like an example of componentization steps based on GMI work.
- Status – Example based on GMI emissions was provided in April. We will provide an updated version in November.

Next Steps - CTMF Development

- Integrate common core components - TP-Core, Metfields, etc.
- Expand CTM framework flow of control to allow variations required to support additional configurations.
- Develop expanded framework tests to include various conditional flows, system and component configurations.
- Continue development of common utilities and supporting functionality (Asynch IO, etc)
- Integrate MAPL coupling mechanism to support auto regrid, etc.
- Integrate components from CTM Groups GMI, GOCART, GEOS-Chem, Goddard CTM, etc.

Next Steps - Target Dates

- End of October FY06:
  - Functional Framework integrated with ESMF version of GMI emission component(compiled and under test now)
- Jan FY07:
  - GMAO's TP-Core integrated into framework
- April FY07:
  - Demonstrate Tracer run with CTMF's core components
- October FY07:
  - Tested and functional framework with major components from across CTM community integrated (need community support)

Discussion

- Questions/Comments?