Support for Model Coupling: An Interface-based Approach

Tom Bulatewicz, Janice Cuny, Computer and Information Sciences, University of Oregon, {tomb,cuny}@cs.uoregon.edu

UNIVERSITY OF OREGON

Abstract

Increasingly scientists need to couple model codes. Often this is a time consuming and error prone process and as of yet there has been little in the way of automated support. Such support may be possible with better representations of the coupling-relevant aspects of a model. Our work seeks to design such a representation and then use it as basis for the specification of model coupling interfaces, a model coupling language, and ultimately a prototype model coupling environment for hydrology. Our goal is to provide infrastructure that will allow scientists to quickly prototype potential couplings.

Why Couple?

The accuracy of models can be increased by

1. accounting for more of the processes of the system, or
2. increasing the area modeled.

In the first case, different models may be coupled; in the second case, multiple instances of the same model may be coupled.

Steps in Coupling

Step 1. Identify and obtain a set of compatible models

Step 2. Specify model interactions and resolve incompatibilities

Step 3. Instrument/modify the model codes and execute the coupled model

Without support for model coupling these steps often require exhaustive understanding of the underlying model codes and significant code rewriting. Our approach will simply the process.

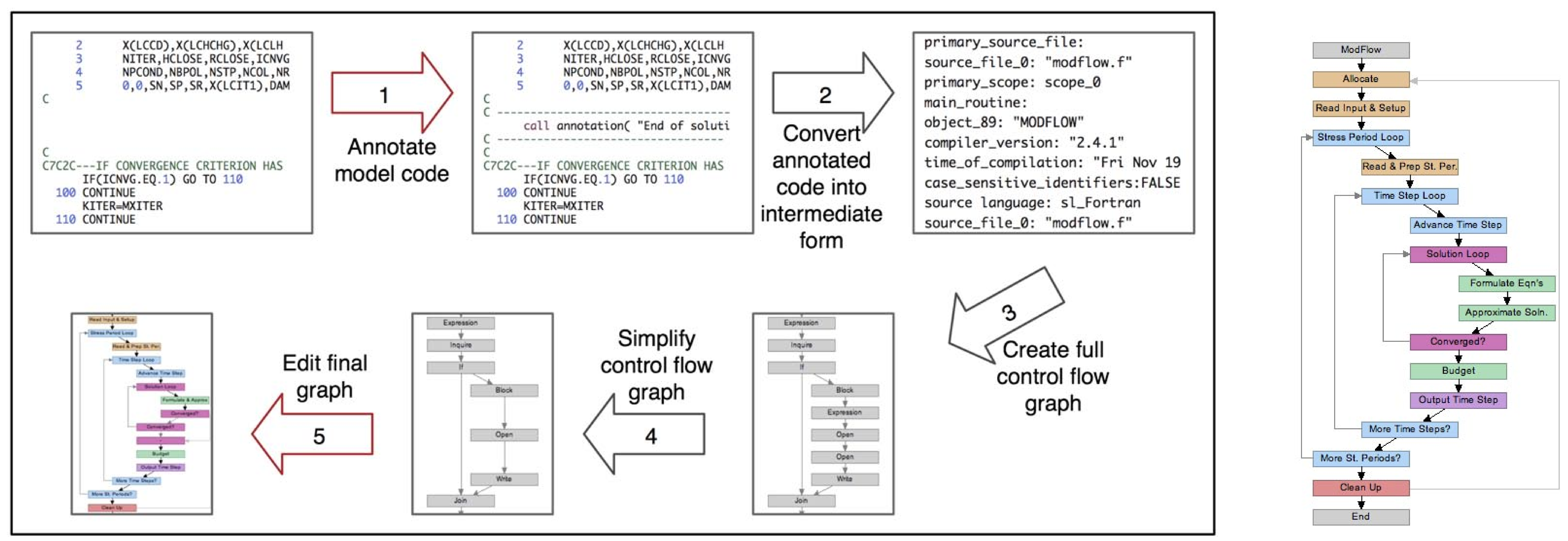
Reference: Bulatewicz, T., J. Cuny, M. Warman. 2004. The Potential Coupling Interface: Metadata for model coupling. Proceedings of the Winter Simulation Conference, Washington D.C.

The Potential Coupling Interface (PCI)

The PCI is a visual representation of a model that describes the possible ways in which the model could be coupled. It supports all 3 coupling steps; it

1. serves as a form of model metadata that aids in determining model compatibility,
2. provides a convenient basis for specifying model interaction, and
3. supports the automatic generation of coupling code.

The PCI is a succinct, visual representation of the coupling-relevant aspects of a model code. It is constructed only once and then reused for all future couplings. The PCI is created with the assistance of a programmer familiar with the source code.



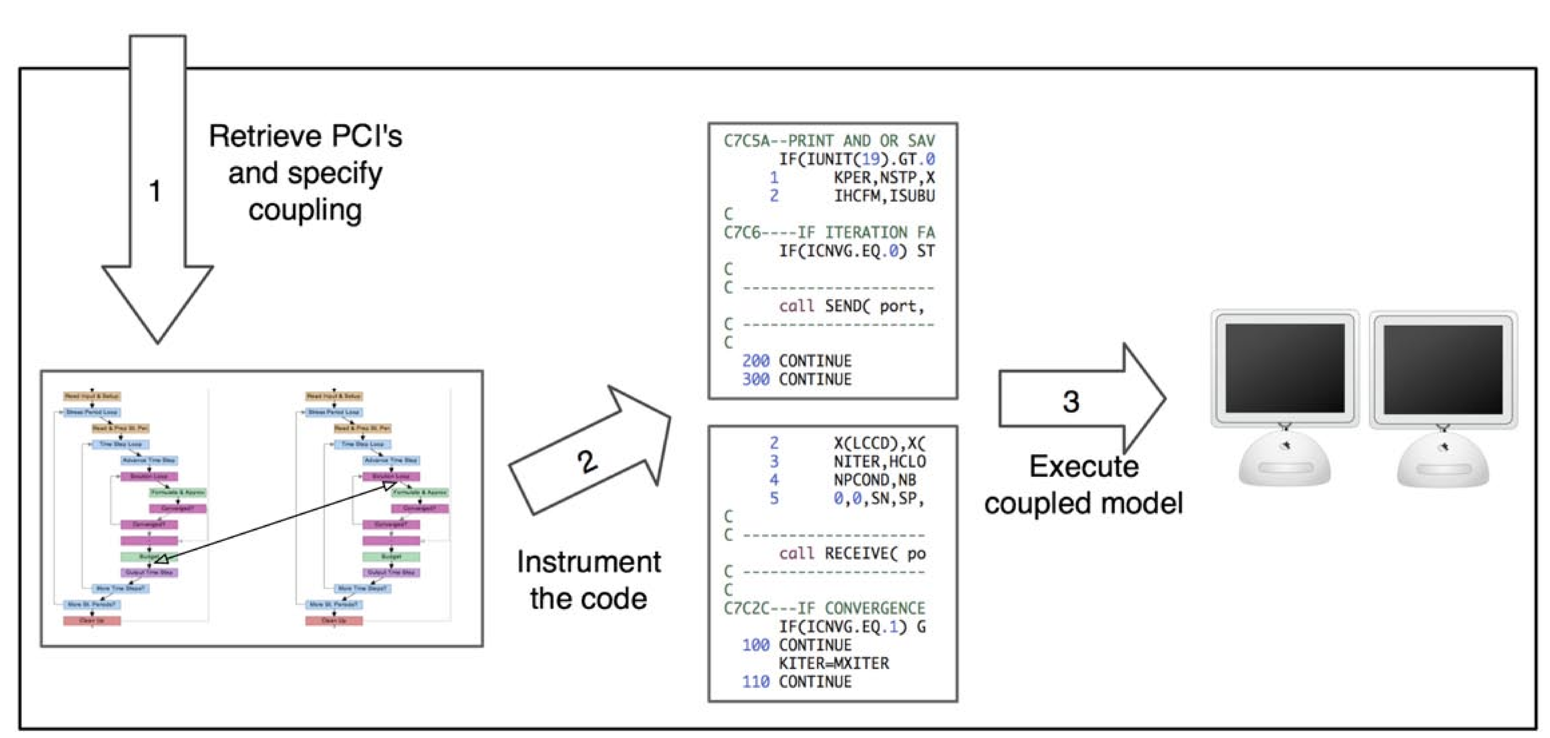
Red arrows indicate programmer intervention; black arrows indicate actions of the underlying software.

The PCI as model metadata

The PCI serves as a form of model metadata used to expedite the task of finding compatible models. It can also be integrated into existing model metadata standards.

The PCI as the basis for complete coupling specifications

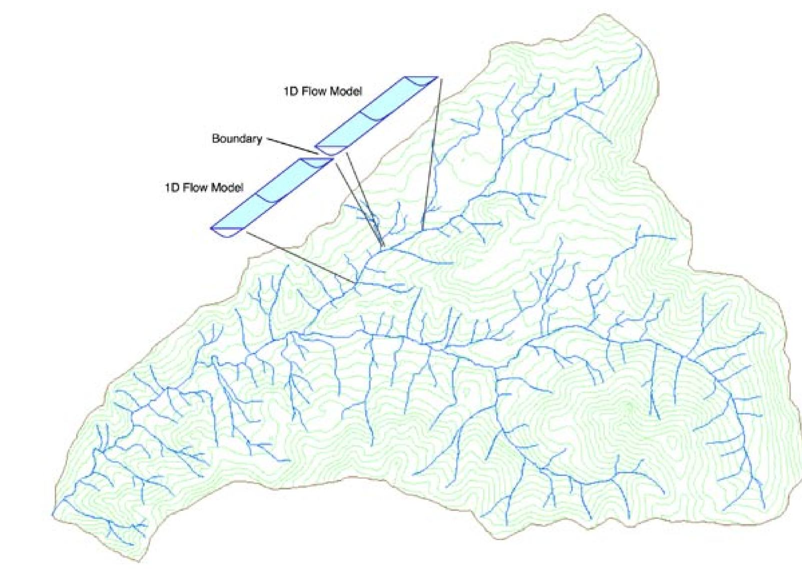
We will also use the PCI to create complete coupling specifications for fast prototyping.



Open Questions

How are the models related conceptually?

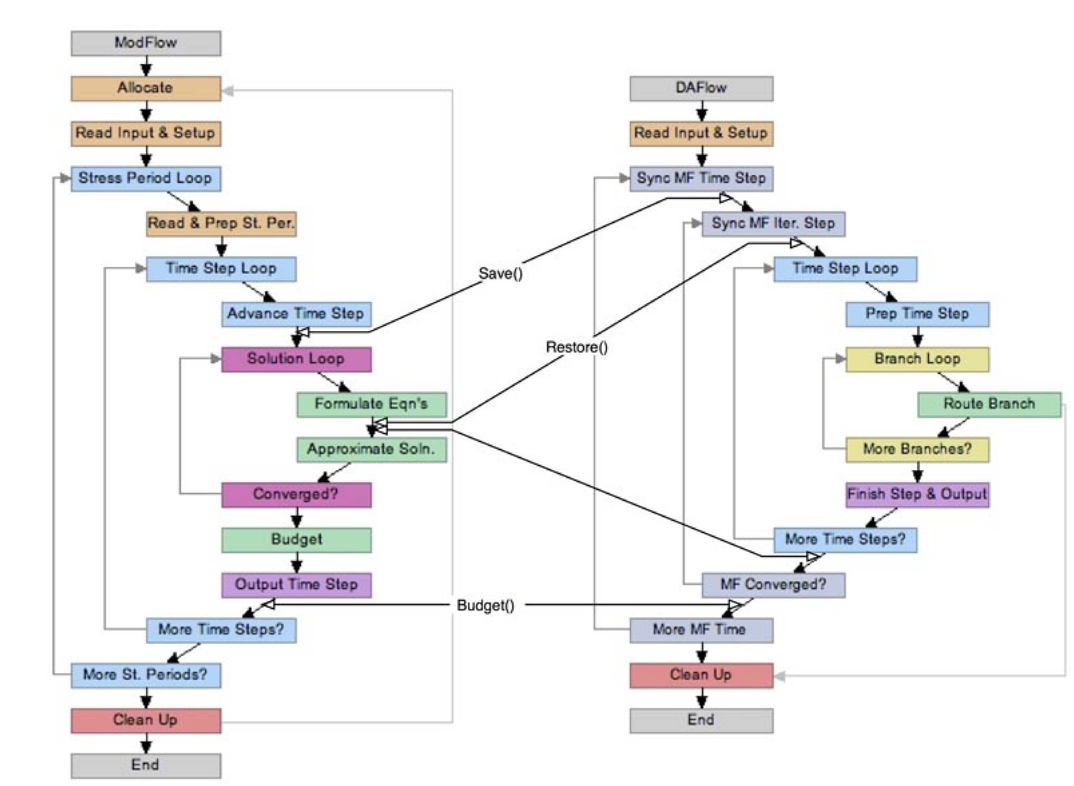
The first step in creating a coupling is to decide how the models will fit together conceptually. For spatially explicit models, this usually involves identifying a common boundary between the models.



We are developing a way to represent these boundaries in the PCI, to support easy matching of model interfaces.

How are boundary interactions carried out by the models?

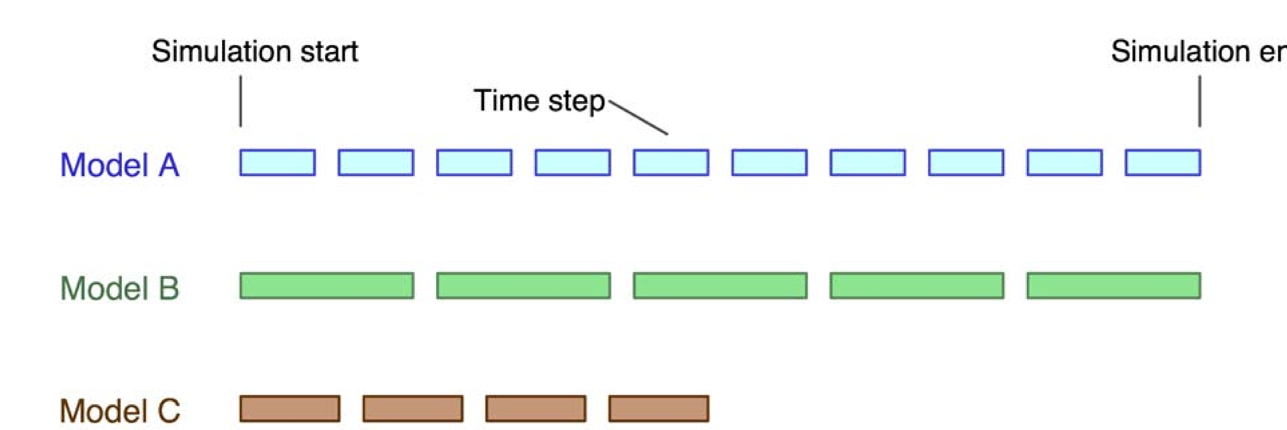
For each boundary, the PCI defines what data is exchanged, and when. The connected PCIs on the right, for example, define the coupling of a ground-water flow to a surface-water flow model. In this case, additional code modules have been added to perform leakage calculations not present in the original models.

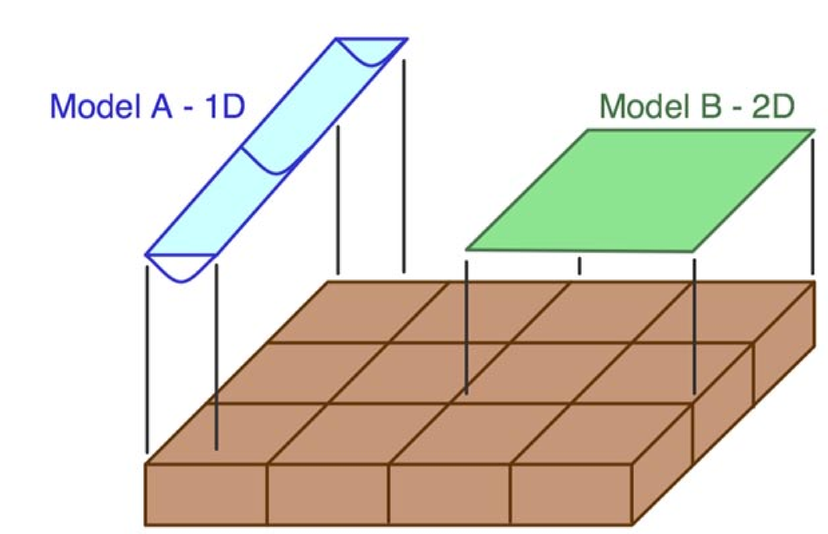


We are developing methods for visually connecting model interfaces to define a coupling.

How are spatial and temporal incompatibilities resolved?

Temporal compatibility depends upon the time step length and the total simulation duration. In the figure here, for example, Models A and B can only interact at the granularity of B's steps, and Model C would need to be executed twice in order to couple to A or B.





Spatial compatibility depends upon the dimensionality of the models, the total area simulated, and the granularity. In the figure here, for example, 1D segments of Model A must interact with the 3D cells of Model C, the 3D cells of Model C must interact with the 2D cells of Model B.

We are developing classifications of the standard forms of these interactions found in hydrology models with the intent of providing assistance, perhaps in the form of templates, for their specification.