Ten organising principles for coupling in multiphysics and multiscale models

J. Walter Larson[1]

(Received 1 September 2006; revised 7 January 2009)

Abstract

Computational science faces new challenges posed by multiphysics and multiscale, or more generally put, *coupled* models. These systems are composites formed from separate subsystem models that interact via data exchanges. These data dependencies pose a *coupling problem*, and on distributed memory computers, a *parallel coupling problem*. This article presents a definition of terms and a set of organising principles for the coupling and parallel coupling problems. It is meant as a first step towards creating a theory of coupled models. These principles are then employed in a case study of a coupled climate model and offer remarkable insight into its structure.

Contents

http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/138 gives this article, © Austral. Mathematical Soc. 2009. Published February 20, 2009. ISSN 1446-8735. (Print two pages per sheet of paper.)

1 Introduction

Computational science is becoming more ambitious by moving beyond the traditional approach of simulating individual isolated subsystems, towards integrated systems having numerous mutually interacting components. Two distinct types of these composite models are emerging: *multiphysics* models, which violate the frequent modelling assumption that the system under study does not interact with the outside world; and *multiscale* models, which violate an often imposed notion that phenomena prevalent on disparate spatiotemporal scales do not interact. The main driver for this change of approach has been the advent of high performance computing, and in particular, message passing parallel computing (also known as distributed memory parallelism) on commodity microprocessor based clusters.

A classic example of a multiphysics model is a climate system model, comprising an atmospheric general circulation model (GCM), an ocean GCM, a

fully dynamic sea ice model, and a land surface model [2]. Other multiphysics modelling problems can be found in the fields of controlled thermonuclear fusion, space weather, reactive flow, modelling of rocket engines, fluid-structure interaction, materials science, and groundwater hydrology.

Numerical weather prediction provides a leading example of a multiscale application in forecast models that allow multiple, nested, and interacting computational domains possessing different spatiotemporal discretisations and sub-gridscale process parameterisations [9]. Examples of multiscale systems abound in science and engineering in the fields of plasma physics, climate and weather, biology, hydrology, and materials science.

Multiscale and multiphysics models are *coupled models*, a term adopted from climate modelling that describes well the importance of model-to-model interaction in these systems. Some software technology has been developed to support coupling, with many application specific ad hoc solutions, or in some cases slightly more general domain specific packages. More generic coupling infrastructure packages exist [8, 7]. Coupling and parallel coupling form a computational science problem in need of a precise definition and some theoretical foundations. This article is an early attempt to construct a vocabulary for describing coupling and parallel coupling, and to state some organising principles. Taken together they are not yet a rigorous theoretical framework, but rather a set of heuristic notions whose explicatory power will be demonstrated in Section 4.

2 Define terms and state the problem

A *coupled model* $\mathcal{M}$ consists of N constituent[1] models, or simply *constituents*, that collectively model a complex system through their evolution and mutual interactions.

A constituent $\mathcal{C}_i$, $i = 1, \ldots, N$, is characterised by a *model* M_i that solves its equations of evolution on its *domain* Γ_i to calculate its *state* U_i. The state is updated using the current model state and a set of *input variables* V_i. *Output variables* W_i are computed from U_i. The sets V_i and W_i comprise the data connections of $\mathcal{C}_i$ to the outside world, and are defined on the *boundary domain* $\partial\Gamma_i$ (or subset thereof).

Thus, a constituent $\mathcal{C}_i$ is specified as $\mathcal{C}_i \equiv \{M_i, U_i, V_i, W_i, \Gamma_i, \partial\Gamma_i\}$.

Two constituents $\mathcal{C}_i$ and $\mathcal{C}_j$ are *coupled* if and only if

1. $\Gamma_i \cap \Gamma_j \neq \emptyset$ and
2. (a) $W_j \cap V_i \neq \emptyset$ and/or $V_j \cap W_i \neq \emptyset$ or
   (b) the inputs V_i (V_j) can be computed from the outputs W_j (W_i).

That is, two models are coupled if their domains intersect and some of the outputs of one model serve as some of the inputs to the other. Coupling between $\mathcal{C}_i$ and $\mathcal{C}_j$ occurs on the *overlap domain* $\Omega_{ij} \equiv \Gamma_i \cap \Gamma_j$. The overlap domains together form the boundary domain of a constituent. That is,

$$\partial\Gamma_i = \bigcup_{\substack{j=1 \\ i \neq j}}^{N} \Omega_{ij} . \tag{1}$$

[1] Many authors use the term *component* to label the individual parts of a coupled model. Component based software engineering is now emerging as a key software technology for these systems, and a software 'component' is not necessarily the same as a model 'component.' For this reason, I use the term 'constituent' instead of 'component'.

Couplings are transformations $\mathcal{T}_{ij} : (W_j, \Omega_{ij}) \to (V_i, \Omega_{ij})$ and $\mathcal{T}_{ji} : (W_i, \Omega_{ij}) \to (V_j, \Omega_{ij})$ that deliver inputs to $\mathcal{C}_i$ and $\mathcal{C}_j$, respectively.

The above definitions collectively specify a coupled model $\mathcal{M}$ as $\mathcal{M} \equiv \{\mathcal{C}_1, \ldots, \mathcal{C}_N, \mathcal{T}\}$, where $\mathcal{T} \equiv \{\mathcal{T}_{ij}, i = 1, \ldots, N, j = 1, \ldots, N, i \neq j\}$ is the set of all the inter-constituent coupling transformations.

In building a coupled model on a traditional uniprocessor (von Neumann) computer architecture, one must surmount the following obstacle

Problem 1 (Coupling Problem) *Given* N *models executing in mutual interaction, create a working coupled model.*

3 The organising principles

3.1 Aspects of the coupling problem

The coupling problem (CP) has an immediate organising principle in terms of process decomposition.

Organising Principle 1 *The coupling problem is factorable into challenges:*

1. *coupled model architecture;*
2. *data processing in aid of coupling; and*
3. *software environment.*

This article focuses on the architectural and algorithmic aspects of the CP, which correspond to the first two aspects identified in this organising principle. Software environment encompasses issues such as programming language interoperability and software build strategies to arrive at a working executable from source code, and is beyond the scope of this discussion.

3.2 Types of couplings

The definition of coupling stated in Section 2 is refined further by considering how output W_j from a source constituent $\mathcal{C}_j$ is related both to its own state U_j and the input V_i and state U_i of a destination constituent $\mathcal{C}_i$. *Explicit coupling* occurs on Ω_{ij} if and only if

1. $\mathsf{W}_i \cap \mathsf{U}_i = \emptyset$ and $\mathsf{V}_j \cap \mathsf{U}_j = \emptyset$ and
2. $\mathsf{W}_j \cap \mathsf{U}_j = \emptyset$ and $\mathsf{V}_i \cap \mathsf{U}_i = \emptyset$.

Implicit coupling occurs on Ω_{ij} if and only if

1. $\mathsf{W}_i \cap \mathsf{U}_i \neq \emptyset$ and $\mathsf{V}_j \cap \mathsf{U}_j \neq \emptyset$ and/or
2. $\mathsf{W}_j \cap \mathsf{U}_j \neq \emptyset$ and $\mathsf{V}_i \cap \mathsf{U}_i \neq \emptyset$.

Explicit coupling constitutes one way data delivery between source and destination constituents, for example boundary conditions or interfacial fluxes exchanged in a coupled climate model. Implicit coupling is indicative of simultaneous, two way state-to-state coupling, for example self-consistent computation of electromagnetic fields for core-edge coupling in tokamak simulation [1]. Implicit coupling between $\mathcal{C}_i$ and $\mathcal{C}_j$ begs the question of whether they might be better implemented as a single constituent. That said, it is often desirable to implement implicitly coupled models in separate constituents to isolate functionality, which allows model substitution, and makes better use of specialist knowledge.

Organising Principle 2 *To the coupled model builder, explicit coupling is preferable because it is easier to implement. Implicit coupling can impose a requirement for repeated iterative execution of constituents to achieve self-consistent state solutions.*

For the remainder of the discussion in this article, our primary focus will be on explicit coupling. Many of the ideas presented here can be applied directly to implicit coupling. Others, where noted, require modification to suit implicit coupling, and this is an area for future work.

3.3 Data dependencies

The overall complexity of a coupled system is a function of N, the number of constituents. Coupling complexity, however, is proportional to the number of interconstituent data dependencies K. At first blush, one assumes that $\mathsf{K} \leq \mathsf{N}^2 - \mathsf{N}$, but there may exist multiple connections (that is, more than two) between any constituent pair $\{\mathcal{C}_\mathsf{i}, \mathcal{C}_\mathsf{j}\}$, and thus it is possible that $\mathsf{K} > \mathsf{N}^2 - \mathsf{N}$.

Graph theory [4] is an useful tool for describing systems with interdependent processes, yielding a fundamental organising principle for the CP.

Organising Principle 3 *A coupled model $\mathcal{M}$ can be represented as a directed graph $\mathcal{G}$, and this digraph is connected.*

In this graph theoretic picture of coupled models, the constituents and their data dependencies are represented as vertices and edges, respectively (Figure 1). A coupled model's associated digraph $\mathcal{G}$ is connected because were it not, $\mathcal{G}$ would then consist of two or more separate graphs, implying $\mathcal{M}$ could be separated into two or more independent coupled systems. A dependency of vertex A on output from vertex B is expressed by an edge pointing from B to A. Each vertex's associated model updates its state using its time history and its inputs. Self-dependence on state could be signified by a *loop* on each vertex. The convention here is to exclude self-dependency loops.

The *in- (out-) valency* of a vertex is equal to the number of incoming (outgoing) data connections to (from) the corresponding constituent in the coupled system. If a vertex has only incoming (outgoing) edges, it is called a *sink* (*source*), and its associated constituent *can be run off-line*. If $\mathcal{C}_\mathsf{i}$ is associated with a source, it can be run off-line and its time history fed to the rest of $\mathcal{M}$ at a later time. If $\mathcal{C}_\mathsf{i}$ is associated with a sink, the rest of $\mathcal{M}$ can be run first, and its time history subsequently fed to $\mathcal{C}_\mathsf{i}$.

In a digraph, there are five possible distinct connectivity relationships between any two vertices, and each of these corresponds to a different data

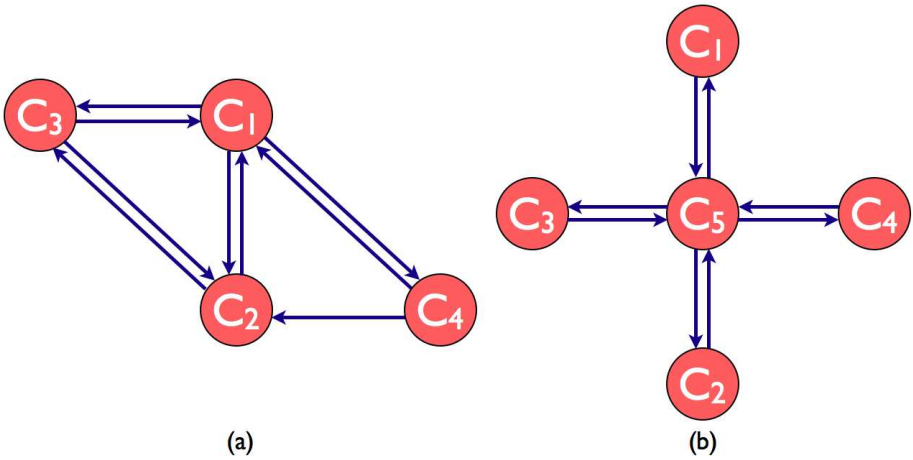

FIGURE 1: Directed graphs for coupled systems with (a) four constituents, and (b) five constituents.

dependency relationship between $\mathcal{C}_i$ and $\mathcal{C}_j$: $\mathcal{C}_i$ receives (delivers) direct input (output) from (to) $\mathcal{C}_j$ (*direct coupling*); $\mathcal{C}_i$ does not receive (deliver) data directly from (to) $\mathcal{C}_j$, but instead via a *path* through a series of one or more intermediate constituents (*indirect coupling*); and $\mathcal{C}_i$ and $\mathcal{C}_j$ have no path connecting them, making them *decoupled*[2].

The connectivity of a parallel coupled model $\mathcal{M}$ is the list of direct intercomponent interactions, expressed as the *adjacency matrix* $\mathbf{A}$ of $\mathcal{G}$. System wide connectivity is proportional to the number of interconstituent connections K in $\mathcal{M}$, which is the sum of the off-diagonal elements of $\mathbf{A}$

$$K = \sum_{\substack{i=1 \\ i\neq j}}^{N} \sum_{j=1}^{N} A_{ij} \,. \tag{2}$$

A constituent's connectivity to the rest of $\mathcal{M}$ is the sum of its in- and out-

[2]Note that for a connected graph $\mathcal{G}$ two vertices A and B are not connected by a path if and only if they are both sources or both sinks

valencies in $\mathcal{G}$

$$K_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} A_{ij} + \sum_{\substack{k=1 \\ k\neq i}}^{N} A_{ki}\,. \tag{3}$$

Constituents are likely to exchange multiple fields in an interaction, and their associated coupling burden can be included by *weighting* the edges of $\mathcal{G}$ accordingly. Edge weighting can also include estimates of the computational complexity of the transformations $\mathcal{T}_{ij}$. Edge weighting is highly application specific; it will be discussed briefly in the case study in Section 4.

This picture of model connectivity works well for systems with purely explicit coupling. Extending the model to include implicit coupling can be accomplished by replacing a directed graph with a *mixed graph*. Implicit coupling between constituents $\mathcal{C}_i$ and $\mathcal{C}_j$ is represented by an *undirected edge*. The adjacency matrix $\mathbf{A}$ is replaced with two adjacency matrices $\mathbf{A}$ and $\mathbf{B}$, representing the number of directed and undirected edges, respectively. The relations (2) and (3) are duplicated to enumerate implicit and explicit couplings separately. Thus far, we have considered two party interactions between constituent pairs $\{\mathcal{C}_i, \mathcal{C}_j\}$, which is sufficient for explicit coupling. In principle, implicit coupling may occur between three or more constituents. Multi-party implicit couplings may be represented by extending the mixed graph to a *hypergraph* in which an edge may connect three or more vertices.

3.4 Time evolution

Coupled models evolve by solving their constituents' model equations, a process in which interconstituent data exchanges play a key role.

Organising Principle 4 *The time evolution of a constituent $\mathcal{C}_i$ can be separated into* ***inter-coupling time intervals*** *during which the constituent solves its equations of evolution using solely its current state U_i and its inputs V_i, and* ***coupling events*** *during which coupling interactions occur.*

An important feature of interconstituent coupling is the relative time stepping of their respective state updates, outputs and inputs. Consider a coupling event occurring at time t^c_{ij} that transforms output W_j from $\mathcal{C}_j$ to input V_i for $\mathcal{C}_i$. Two possible strategies for data delivery are possible. *Instantaneous* data delivery is the transfer of one or more distinct values of W_j from $\mathcal{C}_j$ using values of W_j computed at or near the coupling time t^c_{ij}; for example, a timestep that is nearest in time to t^c_{ij}, either immediately before ($t^c_{j-} < t^c_{ij}$) or immediately after ($t^c_{j+} > t^c_{ij}$) the coupling time. *Integrated* data delivery is the provision of output W_j from $\mathcal{C}_j$ that is integrated with respect to time as $\mathcal{C}_j$ evolves over a time window bracketing or adjacent to the coupling event time t^c_{ij}. Integrated data delivery includes time integrals of flux and time averages of state fields, with fluxes applied incrementally within the destination constituent. One is likely to have time lags between times at which output is prepared and the times at which it is consumed. The inherent time lags that must be tolerated to implement coupling have to be chosen carefully.

Organising Principle 5 *Coupling events between any two constituents can occur either following a* ***schedule*** *with the coupling event times known a priori, or in a potentially nonperiodic and unpredictable fashion* ***triggered by some threshold****.*

In some situations, coupling is sufficiently regular that one can define a *coupling cycle* and, within it, a *coupling frequency* for each interconstituent exchange. Coupling frequency is determined by the timescales over which the constituents evolve significantly. In practice, coupling frequencies and time lags are chosen based on intuition and experimentation to satisfy criteria of numerical stability and quality of model solution with respect to observational data.

Frequently, the terms *loose coupling* and *tight coupling* appear in the computational science literature. There is no rigorous quantitative definition of loose and tight coupling. Through analysis of couplings occurring on a priori schedules or by gathering statistics on the frequency of threshold driven cou-

pling events, one can create criteria for defining loose and tight coupling. This can be accomplished by comparing typical inter-coupling interval times Δt^c_{ij} and Δt^c_{ji} with typical constituent timesteps Δt_i and Δt_j.

Organising Principle 6 *Let constituents $\mathcal{C}_i$ and $\mathcal{C}_j$ be coupled and have coupling events that occur on average with a frequency*

$$\nu = \frac{\sup\{\Delta t_i, \Delta t_j\}}{\inf\{\Delta t^c_{ij}, \Delta t^c_{ji}\}} \,. \tag{4}$$

*For a chosen threshold value $\nu = \nu^*$, one defines $\mathcal{C}_i$ and $\mathcal{C}_j$ to be **loosely coupled** if $\nu < \nu^*$, and **tightly coupled** if $\nu \geq \nu^*$.*

In practice, $\nu \in (0, 1]$ as $\nu > 1$ implies coupling interactions that are more frequent than a constituent's timestepping state updates. Systems with implicit coupling will generally have $\nu = 1$ and are thus tightly coupled.

3.5 Domain overlap

Domain overlap can range in severity from the simplest case of a lower dimensional interface (Figure 2(a)) to partial collocation (Figure 2(b)) to complete collocation ($\Gamma_i = \Gamma_j$). In principle, three or more domains can intersect, forming *higher order overlap domains*; a kth order overlap domain $\Omega_{r_1,\ldots,r_{k+1}}$ results if $k+1 \leq N$ domains overlap

$$\Omega_{r_1,\ldots,r_{k+1}} \equiv \bigcap_{i=1}^{k+1} \Gamma_{r_i} \,, \quad r_1 \neq r_2 \neq \cdots \neq r_{k+1} \,. \tag{5}$$

Figure 2(c) shows three domains Γ_i, Γ_j, and Γ_k sharing such a domain $\Omega_{ijk} = \Omega_{ij} \cap \Omega_{jk}$. In this configuration, two way *merging* of two constituents for subsequent input to a third constituent is required on Ω_{ijk} if and only if one or more of the following conditions are met:

1. $W_i \cap W_j \cap V_k \neq \emptyset$ or $\mathcal{T}_{ki}(W_i, \Omega_{ik}) \cap \mathcal{T}_{kj}(W_j, \Omega_{jk}) \neq \emptyset$, or

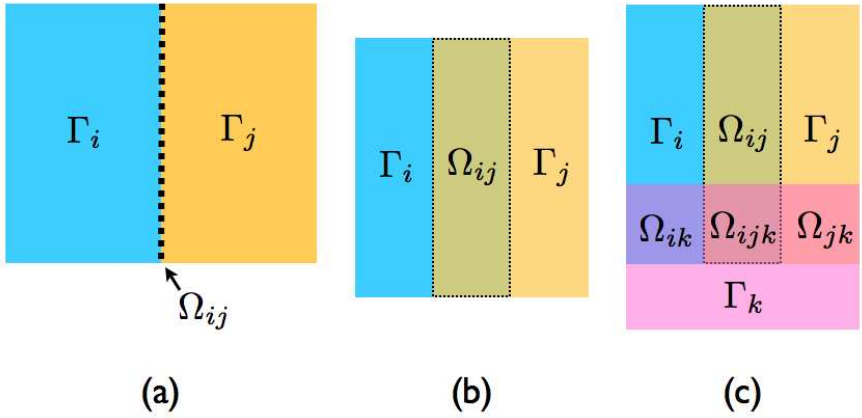

FIGURE 2: Examples of two dimensional overlap domain configurations: (a) a one dimensional interface, (b) partial collocation, and (c) three intersecting domains with multiple first order and one second order overlap domains.

2. $W_j \cap W_k \cap V_i \neq \emptyset$ or $\mathcal{T}_{ij}(W_j, \Omega_{ij}) \cap \mathcal{T}_{ik}(W_k, \Omega_{ik}) \neq \emptyset$, or
3. $W_k \cap W_i \cap V_j \neq \emptyset$ or $\mathcal{T}_{jk}(W_k, \Omega_{jk}) \cap \mathcal{T}_{ji}(W_i, \Omega_{ij}) \neq \emptyset$.

This definition can be generalised to k-way merging on (k+1)th order overlap domains.

Organising Principle 7 *Coupling complexity can be quantified in terms of the number of active overlap domains, their orders, and relative volumes.*

In a coupled model with N constituents, there are potentially $N(N-1)/2$ active first order overlap domains. The potential number of active Kth order overlap domains ($K \leq N-1$) is given by the binomial coefficient $\binom{N}{K+1}$.

In practice, $\Gamma_i \subset \mathbb{R}^D$ is of finite extent and is discretised into a countable finite set of elements. Let $\Delta_i(\Gamma_i)$ be the set of grid points on which $\mathcal{C}_i$ computes its state U_i. Let $\mathcal{N}(\Delta_i(\Gamma_i))$ and $\mathcal{N}(\Delta_i(\partial\Gamma_i))$ be the number of elements in the discretisations of Γ_i and $\partial\Gamma_i$, respectively. The domain surface-to-volume

ratio σ_i for $\mathcal{C}_i$ is

$$\sigma_i = \frac{\mathcal{N}(\Delta_i(\partial\Gamma_i))}{\mathcal{N}(\Delta_i(\Gamma_i))}\,. \tag{6}$$

The range of values for the surface-to-volume ratio are $\sigma_i \in (0, 1]$.

In principle, each constituent has its own distinct domain discretisation, and $\mathcal{C}_i$ and $\mathcal{C}_j$ have differing discretisations of Ω_{ij}—namely $\Delta_i(\Omega_{ij})$ and $\Delta_j(\Omega_{ij})$, respectively. The ratio of the volumes of the discretisations of Ω_{ij} versus Γ_i and Γ_j define the *domain collocation fraction*

$$\rho_{ij} = \frac{\mathcal{N}(\Delta_i(\Omega_{ij})) + \mathcal{N}(\Delta_j(\Omega_{ij}))}{\mathcal{N}(\Delta_i(\Gamma_i)) + \mathcal{N}(\Delta_j(\Gamma_j))}\,. \tag{7}$$

The range of values for the domain collocation fraction are $\rho_{ij} \in (0, 1]$: $\rho_{ij} = 1$ signifies complete domain collocation; whereas $\rho_{ij} \to 0$ for lower dimensional interfacial coupling.

In a computer implementation of $\mathcal{M}$, σ_i and ρ_{ij} are estimators of the relative memory burden due to coupling $\mathcal{C}_i$ to the rest of $\mathcal{M}$, and interconstituent coupling for the pair $\{\mathcal{C}_i, \mathcal{C}_j\}$, respectively. High values of σ_i and ρ_{ij} may steer a coupled model designer away from coupling mechanisms reliant on data copies to minimise intrusion into model source code.

3.6 Coupling transformations

Explicitly coupled constituents $\mathcal{C}_i$ and $\mathcal{C}_j$ exchange data through the transformations $\mathcal{T}_{ij} : (W_j, \Omega_{ij}) \to (V_i, \Omega_{ij})$ and $\mathcal{T}_{ji} : (W_i, \Omega_{ij}) \to (V_j, \Omega_{ij})$. The coupling transformation $\mathcal{T}_{ji}$ ($\mathcal{T}_{ij}$) includes natural law field variable transformations $\mathcal{F}_{ji}$ ($\mathcal{F}_{ij}$), and if $\mathcal{C}_i$ and $\mathcal{C}_j$ use differing spatial discretisations $\Delta_i(\Omega_{ij})$ and $\Delta_j(\Omega_{ij})$, a mesh transformation $\mathcal{G}_{ji}$ ($\mathcal{G}_{ij}$). In many cases, the coupling transformation $\mathcal{T}_{ij}$ is factorable into its respective natural law and mesh transformation components.

Organising Principle 8 *In many cases, the coupling transformation* $\mathcal{T}_{ij}$ *is expressible as a composition of operations for* ***field transformation*** $\mathcal{F}_{ij}$ *and* ***mesh transformation*** $\mathcal{G}_{ij}$*, with* $\mathcal{T}_{ij} = \mathcal{F}_{ij} \circ \mathcal{G}_{ij}$ *or* $\mathcal{T}_{ij} = \mathcal{G}_{ij} \circ \mathcal{F}_{ij}$*. The ordering of these transformations is up to the coupled model developer, and is a source of model coupling uncertainty.*

In the presence of nonlinear field and/or mesh transformations, the ordering of $\mathcal{G}_{ij}$ and $\mathcal{F}_{ij}$ will affect the value of V_i. For example, $\mathcal{C}_j$ might provide a black body temperature T among its outputs and $\mathcal{C}_i$ might require as its input a black body radiative flux P. In this case, the variable transformation is the Stefan–Boltzmann law $P = \sigma T^4$. Additionally, $\mathcal{C}_i$ and $\mathcal{C}_j$ may place their data on different discretisations of Ω_{ij}, requiring a mesh transformation $\mathcal{G}_{ij} : W_j(\zeta) \to V_i(\zeta')$ with $\zeta \in \Delta_j(\Omega_{ij})$ and $\zeta' \in \Delta_i(\Omega_{ij})$.

Merging complicates the set of choices one faces in ordering operations and associated uncertainties. A two way merge $\mathcal{M}_{ijk} = \mathcal{M}_{ijk}(\mathcal{T}_{ij}, \mathcal{T}_{ik})$ combines shared output from $\mathcal{T}_{ij}$ and $\mathcal{T}_{ik}$ to create resulting input for $\mathcal{C}_i$, presenting more choices regarding order operation. This problem is even more pronounced for higher order merging operations.

3.7 Coupling overhead

Coupling overhead is assessed by analysing the system's *load matrix* $\mathbf{L}$. The off-diagonal elements L_{ij} are the cost of performing the transformations $\mathcal{T}_{ij} : (W_j, \Omega_{ij}) \to (V_i, \Omega_{ij})$, and the diagonal elements L_{ii} are the cost of evolving the constituents $\mathcal{C}_i$ in decoupled mode. These costs are typically defined in terms of computer resources (for example, CPU time).

The elements of $\mathbf{L}$ can be used to compute a number of execution cost metrics. The *system wide decoupled simulation cost is* $R_D = \mathrm{Tr}(\mathbf{L})$. The coupling cost experienced by $\mathcal{C}_i$ is

$$R_i = \frac{1}{2}\sum_{j=1}^{N}(L_{ij} + L_{ji}), \quad j \neq i\,. \tag{8}$$

The factor of $\frac{1}{2}$ is present to apportion the cost of coupling constituents $\mathcal{C}_i$ and $\mathcal{C}_j$ between them. The *system wide coupling cost* to $\mathcal{M}$ is $R = \sum_{i=1}^{N} R_i$. The *total coupled simulation cost* is thus $R_T = R + R_D$.

Organising Principle 9 *Coupling overhead can be quantified in terms of the ratio of coupling cost to total simulation cost.*

The *coupling overhead* Q_i imposed on $\mathcal{C}_i$ is

$$Q_i = \frac{R_i}{L_{ii} + R_i}. \tag{9}$$

The *interconstituent coupling overhead* Q_{ij} between $\mathcal{C}_i$ and $\mathcal{C}_j$ is

$$Q_{ij} = Q_{ji} = \frac{L_{ij} + L_{ji}}{L_{ii} + L_{jj} + L_{ij} + L_{ji}}. \tag{10}$$

The *total coupling overhead* Q is

$$Q = \frac{R}{R_T}. \tag{11}$$

3.8 Implications of distributed memory parallelism

The chief impetus for model coupling is the computational capacity created by the message passing parallel programming model. On such platforms, the lack of a global address space poses a different coupling problem.

Problem 2 (parallel coupling problem) *Given* N *models that employ distributed memory parallelism and executing in mutual interaction, create a working and scalable parallel coupled model.*

Organising Principle 10 *The parallel coupling problem (*PCP*) is a superset of the challenges posed by the coupling problem. The definitions and organising principles stated thus far apply equally well to the* CP *and* PCP.

Distributed memory increases coupled model architectural complexity by introducing *concurrency*. In the CP, the constituents and their couplings can execute only in turn as an event loop (*serial composition*). In the PCP, concurrency allows another strategy called *parallel composition* [5] in which the global processor pool is partitioned into *cohorts*, one for each constituent. This allows the constituents to execute simultaneously. Parallel composition has the advantage in minimising processor idle time through cohort sizing based on its respective constituent's parallel scaling behaviour. The disadvantage of parallel composition is that coupling becomes sensitive to synchronisation between independently running parallel models, making performance tuning notoriously difficult. Serial and parallel composition strategies can be combined (*hybrid composition*). Cohorts may intersect partially (*overlapping composition*)—a strategy that is useful for systems with implicit coupling [1].

Data processing operations for the PCP are *parallel operations*. This requires the description of *distributed data* (that is, the coupling fields and their associated meshes). The coupling transformations $\mathcal{T}_{ij}$ are now message passing parallel operations, and in addition to computation they will likely be required to perform parallel data *transfer* and/or *redistribution*. Thus, the factored representations of $\mathcal{T}_{ij}$ discussed in Section 3.6 must include an additional operator $\mathcal{H}_{ij}$ that performs this parallel data transfer. In addition to the concern of *how* to implement parallel transformations, the possibility of concurrency adds a concern of *where* (that is, on which cohort) to execute coupling transformations to best achieve system wide load balance and maximize overall throughput for $\mathcal{M}$.

The coupling complexity metrics developed in this article all need extension to better support parallel programming—a topic for further study.

4 Case study: coupled climate modelling

The Community Climate System Model (CCSM) is a coupled climate model with five constituents

$$\{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5\} = \{\text{Atmosphere, Ocean, Sea Ice, Land, Coupler}\}.$$

The physical components model the Earth's atmosphere, oceans, cryosphere, and biosphere. The coupler is an intermediary through which the physical constituents exchange fields, performs the appropriate transformations $\mathcal{T}$, and acts as an overall coordinator for the system's evolution. Collins et al. [2] give further details regarding CCSM, and Craig [3, 6] its coupling infrastructure.

The atmosphere and ocean domains Γ_1 and Γ_2 are the Earth's global atmosphere and oceans, respectively. The standard atmosphere mesh resolution is $\approx 2.8°$ in latitude and longitude, with 26 vertical levels, yielding a $64 \times 128 \times 26$ grid. The standard ocean mesh configuration has resolution under $1°$ in latitude and longitude, with 40 vertical levels, yielding a $384 \times 320 \times 40$ grid. The sea ice domain Γ_3 comprises the regions of the world's oceans where sea ice is present, and is discretised spatially using the ocean's horizontal grid. The land model's domain Γ_4 is the portion of the Earth's surface covered by land, and discretised spatially using the atmosphere's horizontal grid. The coupler's domain Γ_5 constitutes all of the first and higher order overlap domains between $\{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4\}$; the coupler is by design aware of all of the models' respective discretisations of these regions. The standard timesteps for CCSM's atmosphere and ocean models are 20 minutes. The default timestep for the sea ice model is one hour. The land model does not have a timestep per se as it computes responses to outside forcing when combined with its internal state (for example, soil moisture and temperature). In CCSM the land model is invoked hourly. The coupler's timestep is one hour. Thus, $\{\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4, \Delta t_5\} = \{20\,\text{min}, 20\,\text{min}, 1\,\text{hr}, 1\,\text{hr}, 1\,\text{hr}\}$

In CCSM, all model couplings are explicit, with models handing off output data for subsequent transformation for use as inputs to other system

TABLE 1: Number of Fields Exchanged with the CCSM3.0 Coupler

Number of Fields		Atmosphere	Ocean	sea ice	Land
Sent to Coupler:					
	States	9	6	8	8
	Fluxes	10	2	13	7
Received from Coupler:					
	States	10	3	13	7
	Fluxes	6	15	8	9

components—a commonly used approach in coupled climate modelling.

Figure 1(a) shows data dependencies between the non-coupler constituents in CCSM; evaluating expressions (2) and (3) yield $\mathsf{K} = 9$ and $\{\mathsf{K}_1, \mathsf{K}_2, \mathsf{K}_3, \mathsf{K}_4\} = \{6, 5, 4, 3\}$, respectively. Introduction of the coupler $\mathcal{C}_5$ transforms CCSM into a hub-and-spokes system (Figure 1(b)); evaluating expressions (2) and (3) yield $\mathsf{K} = 8$ and $\{\mathsf{K}_1, \mathsf{K}_2, \mathsf{K}_3, \mathsf{K}_4, \mathsf{K}_5\} = \{2, 2, 2, 2, 8\}$, respectively. The numbers of state and flux fields exchanged with the coupler are summarised in Table 1. For the sake of brevity, we have not listed separately the numbers of fields under exchange between physical components were CCSM implemented without a coupler. Suffice to say there is much redundancy among these fields. Introduction of the coupler reduces significantly the number of send/receive operations each model has to perform.

Coupling occurs on a schedule and is cyclic, with a model day as the coupling cycle period and $\{\Delta\mathsf{t}^{\mathsf{c}}_{15}, \Delta\mathsf{t}^{\mathsf{c}}_{25}, \Delta\mathsf{t}^{\mathsf{c}}_{35}, \Delta\mathsf{t}^{\mathsf{c}}_{45}\} = \{1\,\mathrm{hr}, 24\,\mathrm{hr}, 1\,\mathrm{hr}, 1\,\mathrm{hr}\}$. The hourly exchanges between atmosphere, land, and sea ice are based on the requirement for input radiation fluxes by the atmosphere's radiative transfer package, and because the atmosphere evolves significantly on this timescale. The ocean surface evolves on a slower timescale, and trial and error has arrived at one day for the ocean's coupling frequency. The sea ice and land models send and receive instantaneous values, the ocean and atmosphere send and receive time integrated data. In all cases, there is a time lag between model input and state under update. Based on model timestep and

coupling interval times, $\{\nu_{15}, \nu_{25}, \nu_{35}, \nu_{45}\} = \{\frac{1}{3}, \frac{1}{72}, 1, 1\}$, making the ocean by far the most loosely coupled element of the system, followed in order of increasing tightness by the atmosphere, land, and sea ice models.

The first order overlap domains in the model coincide with the earth's surface. For example, Ω_{12} and Ω_{13} are the portions of the Earth's surface covered by ocean and land, respectively. Along coastlines and in regions where sea ice is present, the interface to the atmosphere will see input of the same fields from multiple entities. Thus higher order overlap domains exist on which merging is required. Examples of second order overlap domains include: Ω_{123}, which constitutes ocean areas in which sea ice is present, requiring weighted merging of surface-atmosphere fluxes based on fractional ice cover; and Ω_{124}, comprising the world's ocean coastlines, requiring weighted merging of surface-atmosphere fluxes based on the respective land/ocean cover. Small third order overlap domains exist; for example, Ω_{1234} constitutes coastal regions in which sea ice is present.

All of the models have effectively three dimensional domains[3], minimising collocation (for example, $\sigma_1 \approx 0.038$, $\sigma_2 \approx 0.025$, and $\rho_{12} \approx 0.037$). Thus, they are amenable to a data-copy-reliant coupling strategy.

The data transformations in CCSM require computation of fluxes and their interpolation between various grids. The strategy used in CCSM is to compute fluxes on the highest resolution grid (that is, the ocean), which requires in some cases interpolation of state variables preceding flux calculation (atmosphere-to-ocean), and in others flux calculation preceding interpolation. In all cases, global spatial integrals surface-atmosphere fluxes are conserved through rescaling using ratios of the integrals computed on the source and destination grids, respectively.

CCSM employs message passing parallelism, and is implemented using a parallel composition. This allows the combination of codes with differing scaling behaviour [8, Figure 4]. The chief performance challenge posed by CCSM is

[3]Ice thickness class and land use type plus soil layer are the additional degrees of freedom in the sea ice and land models, respectively.

the existence of intermittent delays caused by one constituent awaiting data from another, but this is largely solved by shifting these delays from the models to idle time in the coupler, which is highly efficient on a small cohort and highly scalable.

Based on timing data, the system wide coupling overhead in CCSM is low with $Q < 0.05$. In CCSM the ocean and atmosphere are the most computationally intensive parts of the system, and run on significantly larger processor pools than the land, sea ice, and coupler; thus, the object of load balance in CCSM is to minimise the probability of idling in the ocean and atmosphere models caused by waiting for input from the coupler.

5 Conclusions

A heuristic set of definitions and organising principles for coupled models has been stated. This work is a first step towards a more comprehensive theoretical framework for the CP and PCP. Each of the principles stated here provides a glimpse of a rich vein in need of exploration. This conceptual framework has been applied successfully to describe the architecture of a coupled climate model. Future work will refine these principles and expand them to encompass the complexities of parallel coupling. It is hoped that the resulting theory will guide the development of future coupled systems.

Acknowledgments I thank Robert Jacob, Michael Tobis, Jace Mogill, Timothy Mattson, and John Cary for their advice and feedback. This work was funded primarily by the US Department of Energy (DOE) through its Scientific Discovery through Advanced Computing Program. Argonne National Laboratory is operated for the DOE by UChicago Argonne LLC under contract DE-AC02-06CH11357.

References

[1] J. R. Cary, J. Candy, R. H. Cohen, S. Krasheninnikov, D. C. McCune, D. J. Estep, J. Larson, A. D. Malony, P. H. Worley, J. A. Carlsson, A. H. Hakim, P. Hamill, S. Kruger, S. Muzsala, A. Pletzer, S. Shasharina, D. Wade-Stein, N. Wang, L. McInnes, T. Wildey, T. Casper, L. Diachin, T. Epperly, T. D. Rognlien, M. R. Fahey, J. A. Kuehn, A. Morris, S. Shende, E. Feibush, G. W. Hammett, K. Indireshkumar, C. Ludescher, L. Randerson, D. Stotler, A. Yu Pigarov, P. Bonoli, C. S. Chang, D. A. D'Ippolito, P. Colella, D. E. Keyes, R. Bramley, and J. R. Myra. Introducing facets, the framework application for core-edge transport simulations. *Journal of Physics Conference Series*, 78:0120086, 2007. C1095, C1105

[2] W. D. Collins, C. M. Bitz, M. L. Blackmon, G. B . Bonan, C. S. Bretherton, J. A. Carton, P. Chang, S. C. Doney, J. J. Hack, T. B. Henderson, J. T. Kiehl, W. G. Large, D. S. McKenna, B. D. Santer, and R. D. Smith. The community climate system model: CCSM3. *Journal of Climate*, 19(11):2122–2143, 2006. C1092, C1106

[3] Anthony P. Craig, Brian Kaufmann, Robert Jacob, Tom Bettge, Jay Larson, Everest Ong, Chris Ding, and Helen He. cpl6: The new extensible high-performance parallel coupler for the community climate system model. *Int. J. High Perf. Comp. App.*, 19(3):309–327, 2005. C1106

[4] R. Diestel. *Graph Theory*. Springer, New York, third edition, 2006. C1096

[5] Ian Foster. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison Wesley, Reading, Massachusetts, 1995. C1105

[6] CCSM Software Engineering Group. CCSM research tools: CCSM3.0 coupler v6.0 documentation.

http://www.ccsm.ucar.edu/models/ccsm3.0/cpl6/, 2004. C1106

[7] W. Joppich, M. Kurschner, and the MpCCI Team. MpCCI—a Tool for the Simulation of Coupled Applications. *Concurrency and Computation: Practice and Experience*, 18(2):183–192, 2006. doi:10.1002/cpe.913 C1092

[8] Jay Larson, Robert Jacob, and Everest Ong. The model coupling toolkit: A new Fortran90 toolkit for building multi-physics parallel coupled models. *Int. J. High Perf. Comp. App.*, 19(3):277–292, 2005. C1092, C1108

[9] WRF Development Team. Weather research and forecasting (WRF) model web site. http://wrf-model.org/, 2006. C1092

Author address

1. **J. Walter Larson**, ANU Supercomputer Facility, The Australian National University, Canberra, Australia; Mathematics & Computer Science Division, Argonne National Laboratory, Argonne, Illinois, USA; Computation Institute, University of Chicago, Chicago, IL, USA.