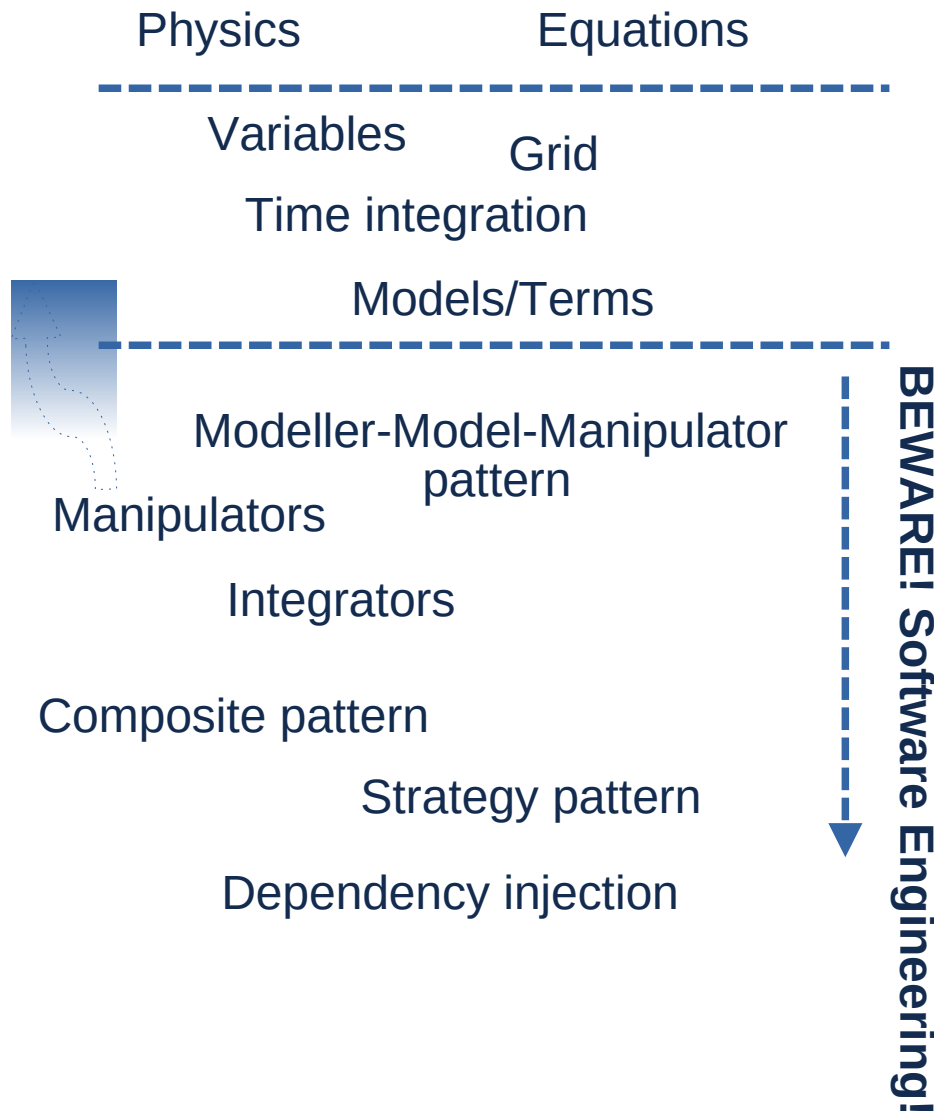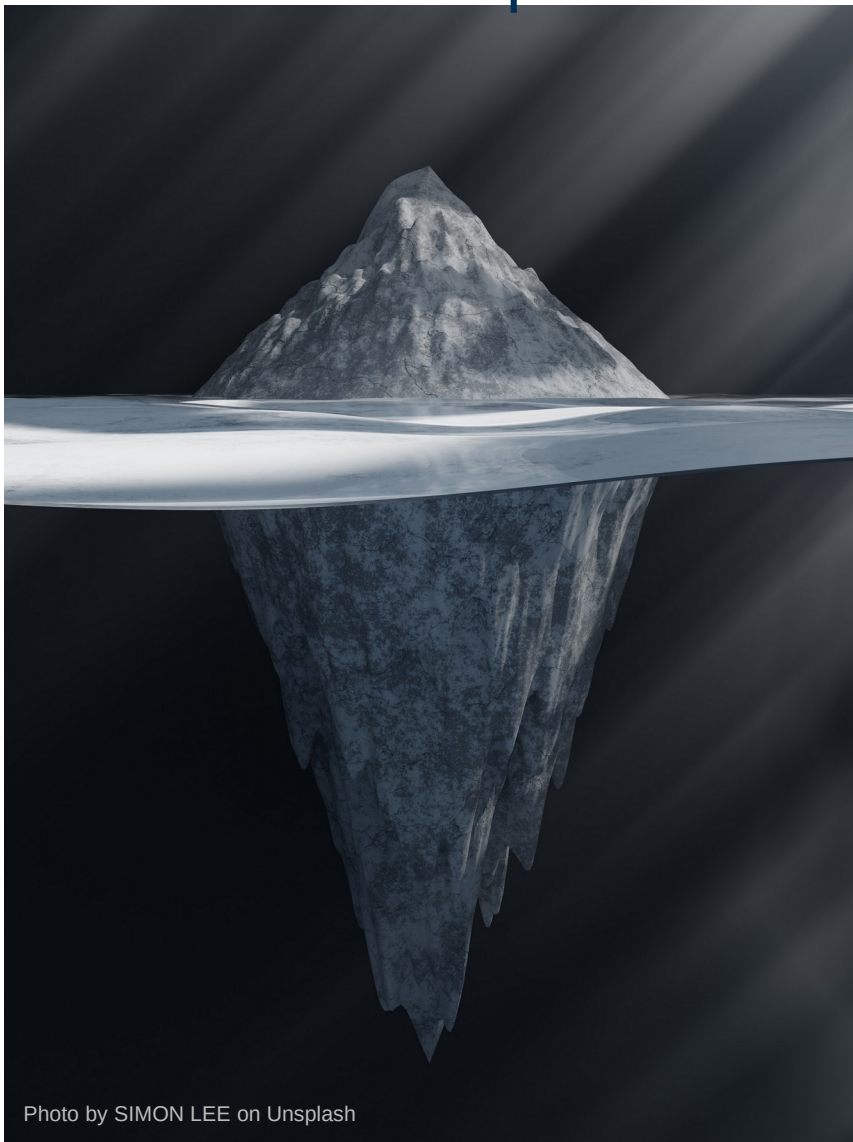UKAEA – Jan 30-31 2024

ReMKiT1D Workshop January 2024
Deep-level concepts in ReMKiT1D

Shallow- vs Deep-level concepts

UK Atomic Energy Authority

Physics   Equations

- - - - - - - - - - - - - - - - - - - -

Variables
                    Grid
        Time integration

                Models/Terms

- - - - - - - - - - - - - - - - - - - -

        Modeller-Model-Manipulator
                    pattern
    Manipulators

            Integrators

Composite pattern

                Strategy pattern

        Dependency injection

BEWARE! Software Engineering!

Photo by SIMON LEE on Unsplash

Modeller-Model-Manipulator pattern

Let's revisit the implied flow of our shallow version of ReMKiT1D

1) Define our grid and partition it between MPI processes

2) Define some variables with set properties that live on that grid

3) Build models/terms that use the above variables and grid

4) Define the time integration scheme used to evolved the variables using the models

5) The integration step:

   1) Calculate any derived variables need in evaluating our models

   2) Evaluate each of our models in turn – these will need access to our variables, and might need to know something about our grid

   3) Make sure that whatever these models do is usable by our integration scheme

   4) Update the variables with the integration result

Can we formalise and abstract some of this structure?

Modeller-Model-Manipulator pattern

The 3 classes of objects in our abstracted design:

Class	Responsibilities	Relationships
Modeller	Owns the variables Communicates variables between MPI processes Centralises the integration call	Contains the Models and Manipulators to which it delegates the physics and variable modification details, respectively
Model	Knows how to evaluate the terms in various physics equations	Contains various terms and data, to which it delegates the individual contributions to the equations
Manipulator	Talks to the Modeller in order to update variables	It is contained in the Modeller, through which it can make requests

Modeller-Model-Manipulator pattern

Variables and the grid

Manipulators

Time integration

Models/Terms
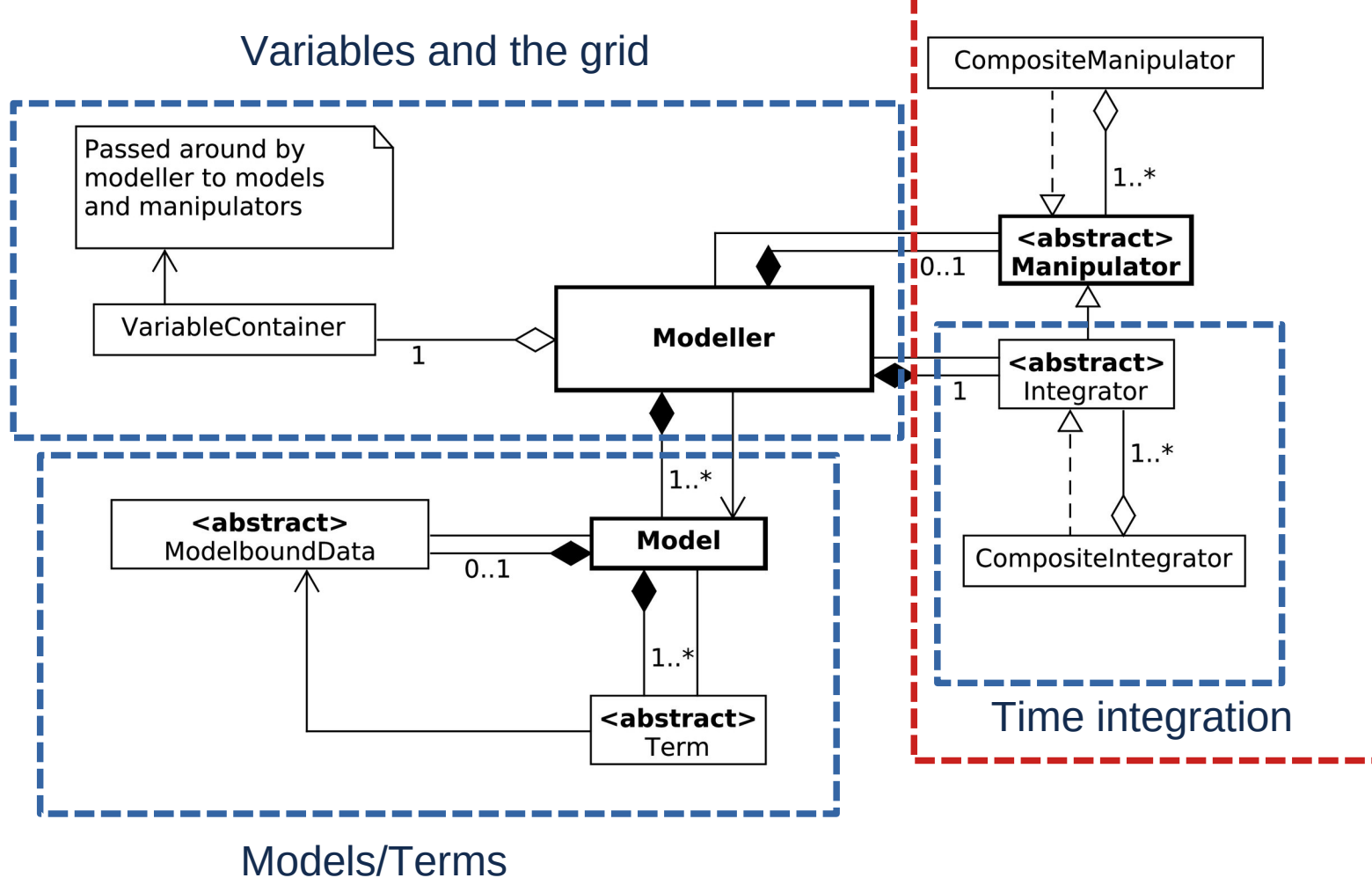
Manipulators

Integrators are Manipulators!

Other examples of Manipulators are data-access Manipulators:

Evaluator:	Extractor:
Manipulator object which can evaluate a terms in a model and store the evaluation in a variable	Manipulator object which copies the value of a modelbound variable into a derived variable for output

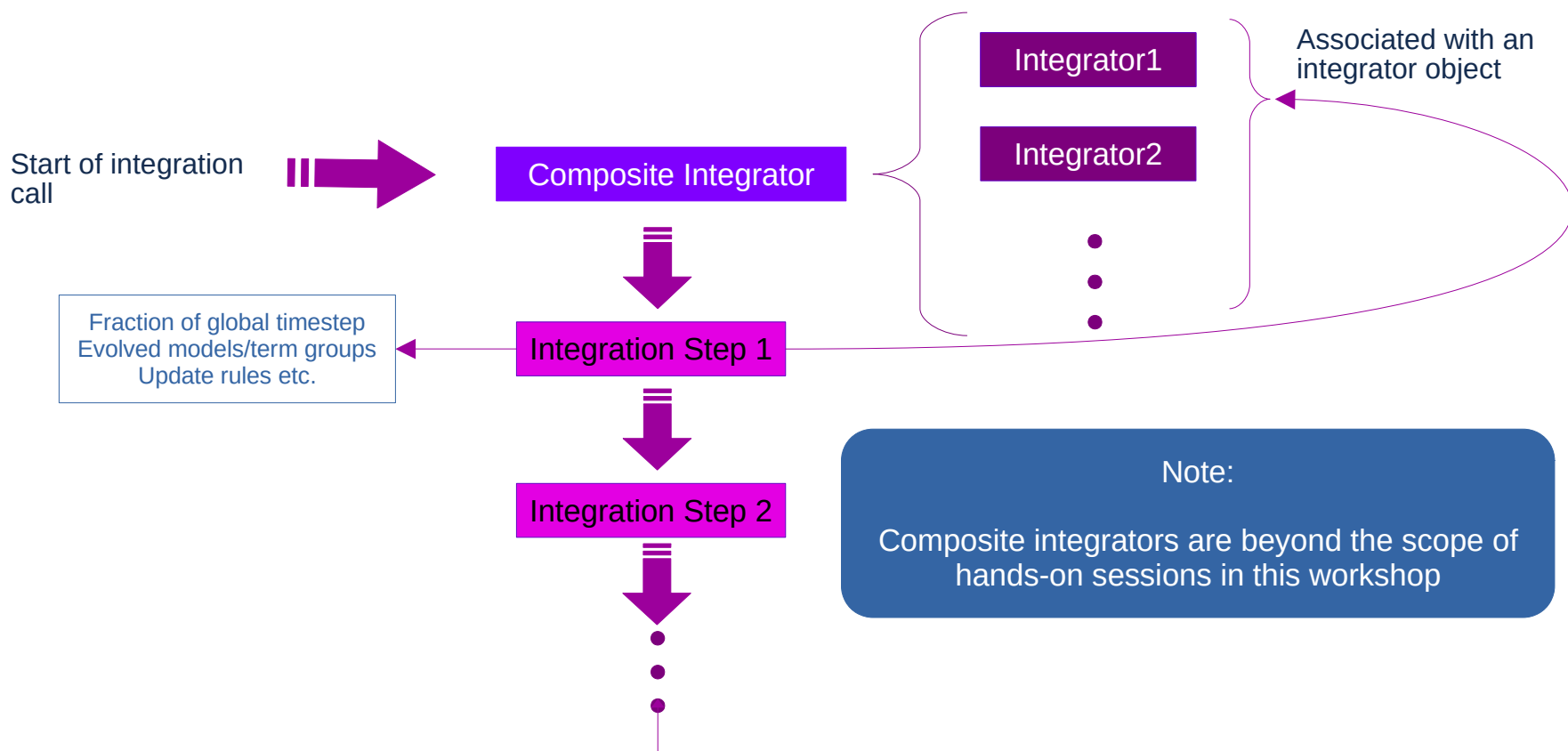Evaluators can be added for all terms evolving a particular variable automatically

Next hands-on session

Integrators briefly revisited

Both integrators and manipulators in general can be composed

Integrator composition represents multiple integration steps in an algorithm – think operator splitting, implicit-explicit methods, etc.

Associated with an integrator object

Integrator1

Integrator2

Start of integration call

Composite Integrator

Fraction of global timestep
Evolved models/term groups
Update rules etc.

Integration Step 1

Integration Step 2

Note:

Composite integrators are beyond the scope of hands-on sessions in this workshop

Hands-on session