

A Suite of Domain-Specific Visual Languages For Scientific Software Application Modelling

Mohamed, John, Richard,
David, Willem & Owen

SWIN
BUR
NE

SWINBURNE
UNIVERSITY OF
TECHNOLOGY

Swinburne
▶ think forward



Outline

- Problem / Motivation
- Approach
- Example Usage
- Future work



Problem

- Scientists are increasingly developing complex software for data analysis
- Most are not trained programmers
- Many are using complex software platforms and techniques e.g. distributed & parallel programming, GPUs, etc - that are hard for experienced CS grads to do
- Approaches to address range from packaged software (Lack flexibility), DSLs (Also Flexibiity/Domain issues), programming patterns and toolkits (still complex)
- Still lack high-level, human-centric support; still really hard to develop high quality software for scientific apps



Motivation – how scientists design their applications...

$$\mathbf{F}_i = \sum_j^{N_m} \mathbf{F}_{ij}(\mathbf{r}_i, \mathbf{r}_j) - \nabla_i \left(-\frac{1}{2} \sum_j^{N_m} \mathbf{E}_j^0 \cdot \boldsymbol{\mu}_j^{\text{ind}} \right) + \sum_j^{N_m} \sum_k^{N_m} \mathbf{F}_{ijk}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k),$$

Part 1 Initialization:

Part 1.1 Assign number of particles, energy, volume, etc.
 Assign coordinates (r_i) at time (t) = 0.
 Assign molecules an initial velocity (v_i).
 Scale v_i consistent with the energy of the en
 Assign any other $(\partial r^n / \partial t^n)_i$ values ($n > 2$)

Part 1.2 Initial force calculations:
 Calculate forces (f_i) on each atom from all
 Calculate acceleration (a_i) for each atom.

Part 2 Simulation Process:

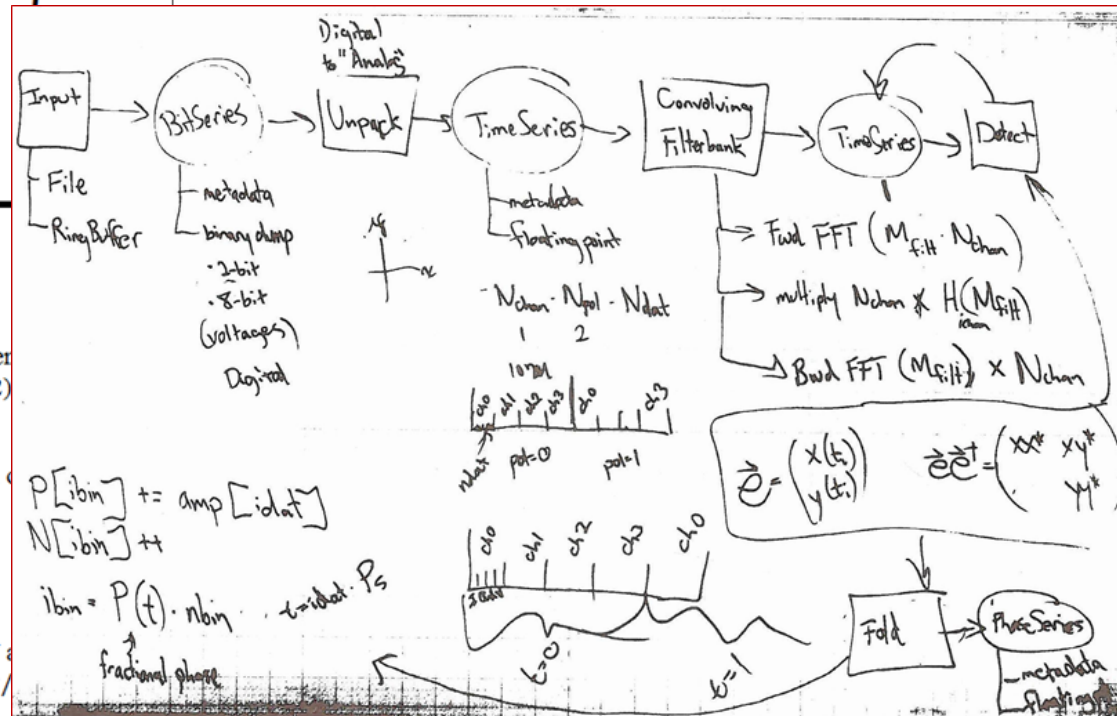
loop

Part 2.1 Integrate equations of motion:

Part 2.1.1 Calculate f_i on each atom from all other j

Part 2.1.2 Apply integrator to update r_b , a_i , v_b ($\partial r^n / \partial t^n$)

Part 2.2 if ($t > t_{\text{Equilibration}}$)
 Accumulate averages.
 else if ($\text{mod}(n, \text{scalingInterval}) = 0$)

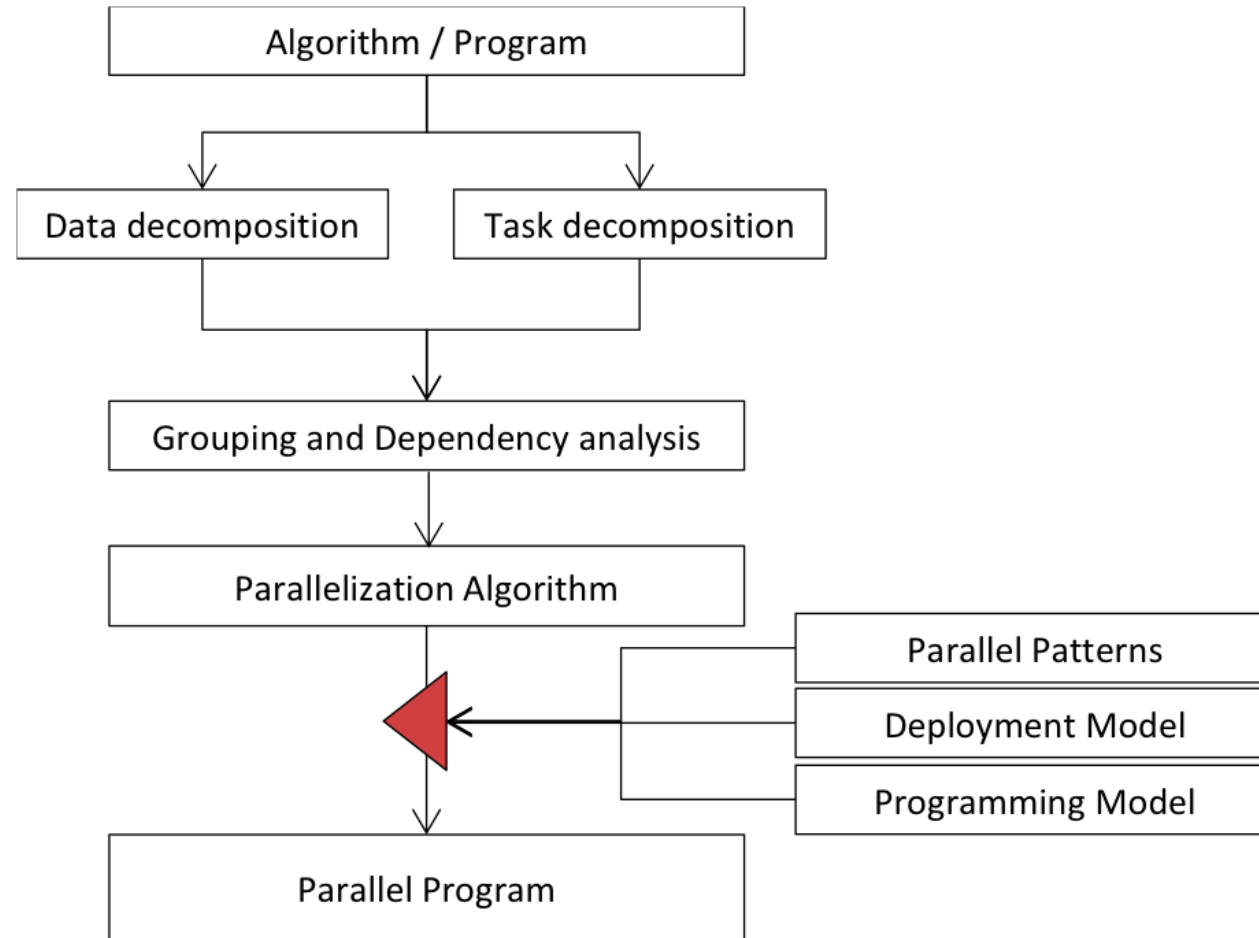




Approach

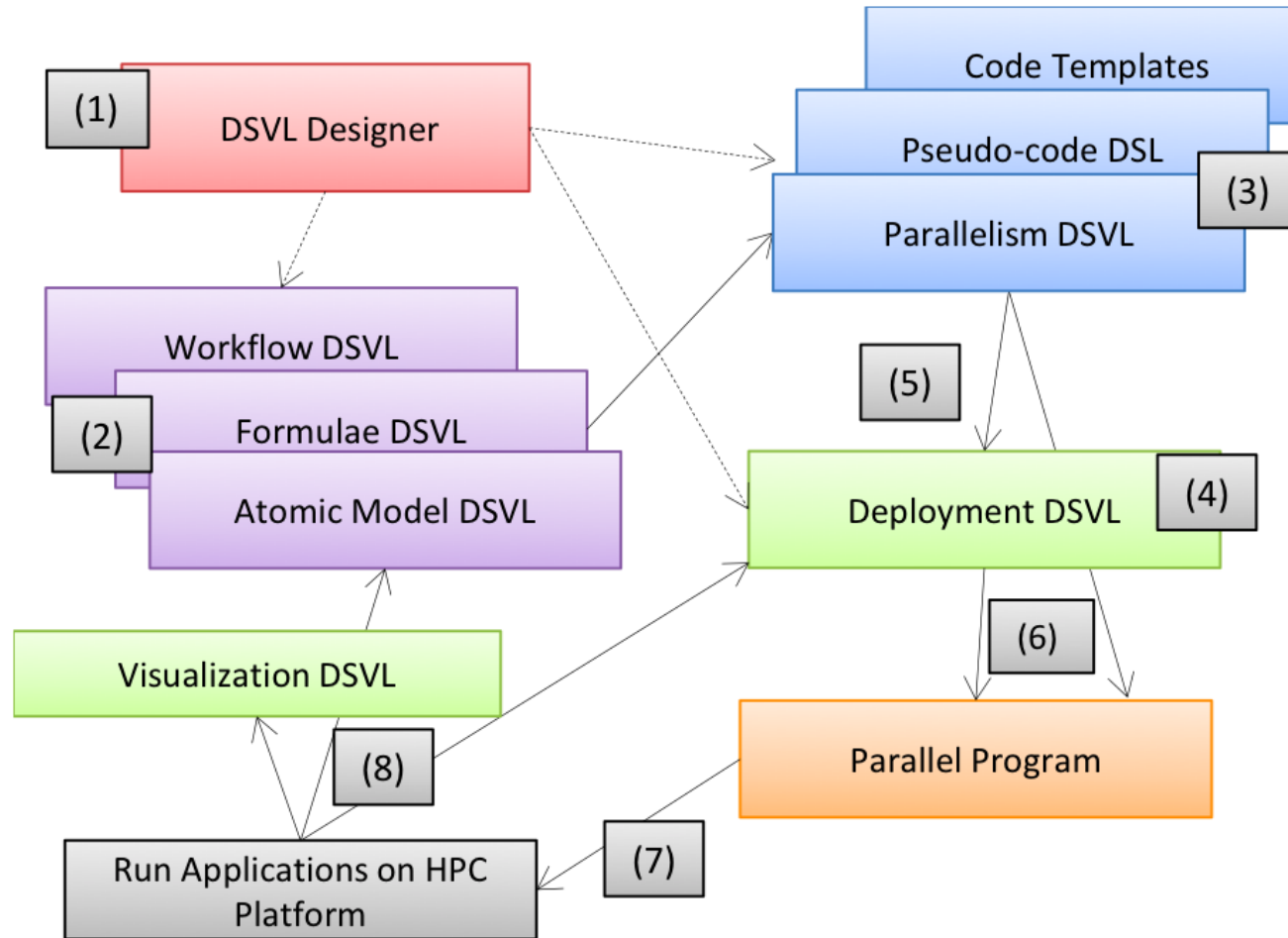
- Support scientists – and developers! – to model their applications at multiple levels of abstraction – domain right down to detailed C/C++/GPU kernel code
- Use set of user-defined and reusable DSLs to model
- Provide web-based environment including DSLs designer, coding, debugging, linking DSL views
- Provide semi-automated support for generating lower-level models, generate code/code annotations, reverse-engineer higher-level models from (existing) code

Development approach to support





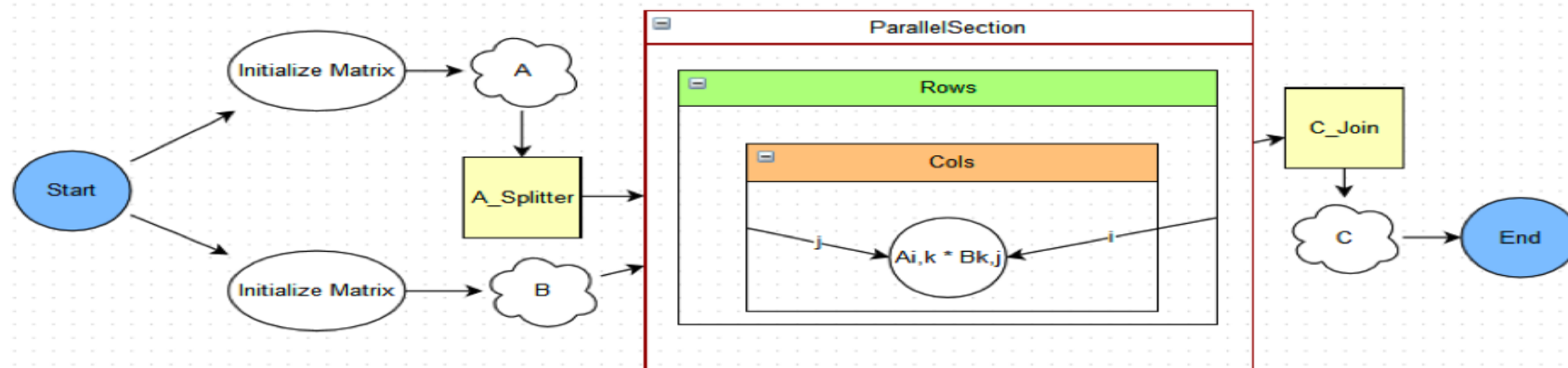
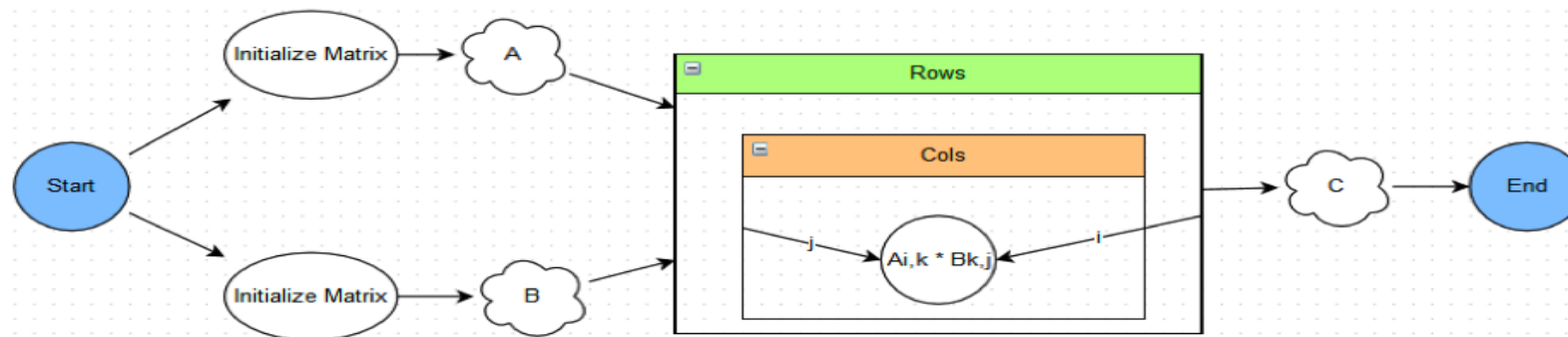
Approach





Example

$$C = A \times B, C_{i,j} = \text{Sum}(A_{i,k} * B_{k,j})$$





Web-based development tool

DS/ML Designer

Molecular Simulation

Magnitic Reasonanse Imaging

General

UML

Properties

Text Editor

```

1 Part 1.2 Initial force calculations:
2   Calculate forces (fi) on
3   Calculate acceleration (
4
5 Part 2 Simulation Process:
6   loop
7 Part 2.1 Integrate equation
8 Part 2.1.1 Calculate fi on ea
9 Part 2.1.2 Apply integrator t
10 Part 2.2 if (t > tEquilib

```

Simulation Loop

Atoms Loop

CalculateForces

Calculate a(i),v(i), r(i)

MaskConv

Diffing

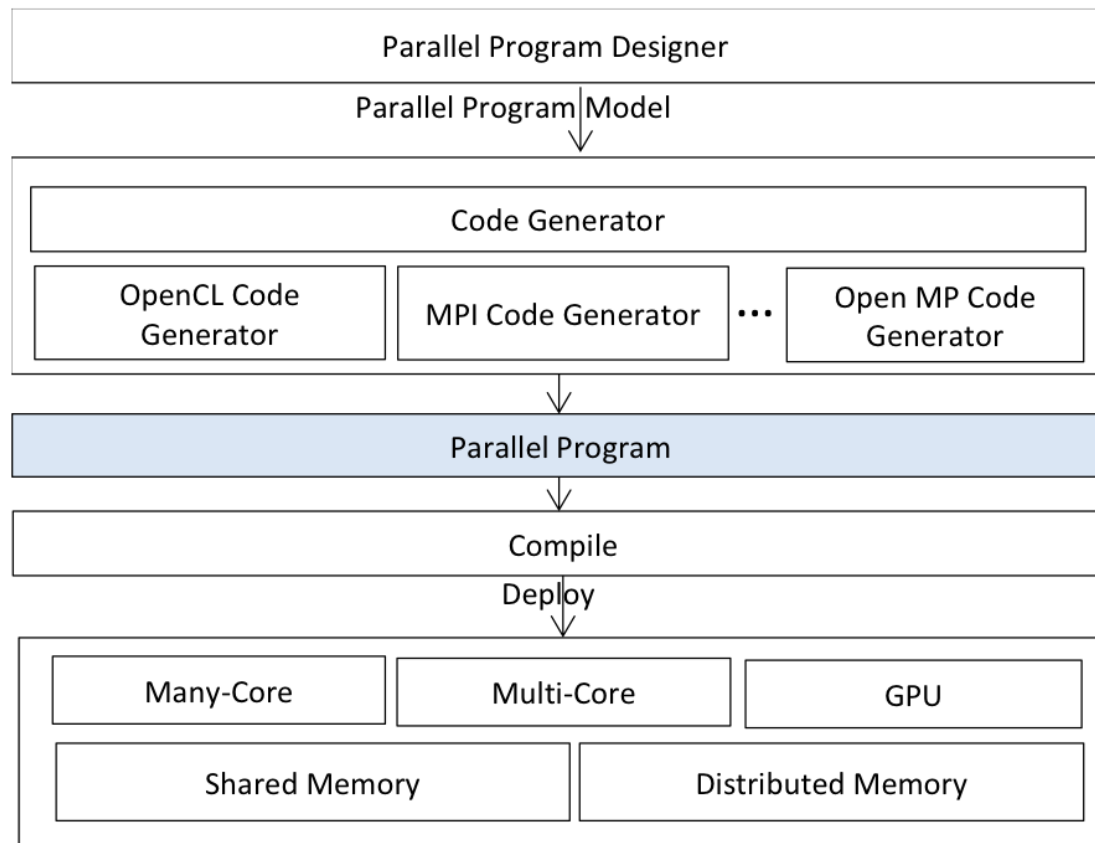
A < Threshold

end

SCIENCE | TECHNOLOGY | INNOVATION | BUSINESS



Tool





Evaluation

- Define set of DSLs to model domain (molecular simulation and signal processing for radio telescopes), architecture (GPU and MPI-based CPU), parallelism, processing models (MapReduce etc)
- Model hand-implemented programs at multiple levels of abstraction including links between models
- Generate C and OpenCL code
- Reverse-engineer (parts of) models from C code
- 3 scientists validating approach iteratively



Summary & Future work

- Integrated web-based development environment for GPU-based (and other) scientific applications
- Flexible DSL designer with pre-packaged DSLs and user-defined DSLs
- Semi-automated roundtrip engineering support: model->code-> model
- Working on:
 - Code generation/reverse engineering – C/C++, OpenCL
 - Patterns and critics to guide users, analyse models/code
 - Visualisation of running GPU code in models

Questions?

Mohamed Almorsy
malmorsy@swin.edu.au

