Break-out “Modelling”

- software modelling – design issues
- driving design by risk mitigation
  - what could lead CSDMS to fail?
    - ➔ how to avoid it?

Break-out “Modelling”

- Risk 1: getting models into reposity: low threshold to participate but how to ensure long term reliability? in general or specific area of use
  - Pyramid picture from donated to quality assured: workgroup vetting?
  - SourceForge: activity measures, compliancy, # unit test
  - comes with (analytical) tests (also good for checking compiler dependency)
  - multiple icons on the website (code, compl, tests, …)
  - literature
  - room for comments by users (like comments on hotels)
  - forum

Break-out “Modelling”

- Risk 2: meta data for “plug & play”
  - OpenMI: get information from components
  - bad experience with visual modelling; “what is ice?”
  - first get models in, solve issues as come along
- Risk 3: CCA toolchain dependencies → complex for development
  - only Ccafeine GUI at end user
  - CCA toolchain only needed for specific request
  - Dan: what if I have a new model? Scott: server account

Break-out “Modelling”

- Risk 4: transition from listing models to simulation hosting → membership requirement
  - “Build server” for download configurations
  - low/no threshold for membership
  - still code available → possible benefit for hardwiring
- Risk 5: computer dependency → working offline
  - climate modelling: slight change in way of working
- Risk 6: Misapplication vs guiding users to make “good” connections
  - always possible

Break-out “Modelling”

- Risk 7: dependency on other developments (OpenMI Java, CCA)
  - waiting for critical bug fixes
- Risk 8: quantity conservation while interpolation
  - work on good re-mapping tools
  - use models on similar grids
  - parallel remapping (look at ESMF,MCT)
- Risk 9: increasing data size
  - porting from PC to HPC
  - grid indexing, scaling “N^2” issues, distributed memory
  - opportunity for additional funding (peta scale)
  - TeraGrid, Grass

Break-out “Modelling”

- Risk 10: no transition to HPC
  - CCA support HPC; but requires gradual rewriting component by component
- Risk 11: cloud computing (Amazon EC2, Eucalyptus) and GPU
  - migration away from supercomputing
- Risk 12: black box of CSDMS/CCA communication layer
  - Mozilla: tried introducing COM object model (slow) then classes (concurrency problems) → firefox
  - ESRI: modular, object oriented, black box
  - request different degrees of diagnostic output
  - transparent code, e.g. option to easily write netcdf output

Break-out “Modelling”

- Working top-down: wrap whole model and split only as needed
- Prevent doubling code
- Swappability: test against each other’s code helps finding bugs
- Target audience: not point-and-click, but fairly experienced modellers
- Request for something like “support@csdms.org”

Break-out “Modelling”

Prioritize models

- overview of processes
- identify components (“1d flow”, “2d flow”, “turb. closure”)

Implementation path for contributor

- IRF
- thinking about HPC (SC’09)

Anticipated needs

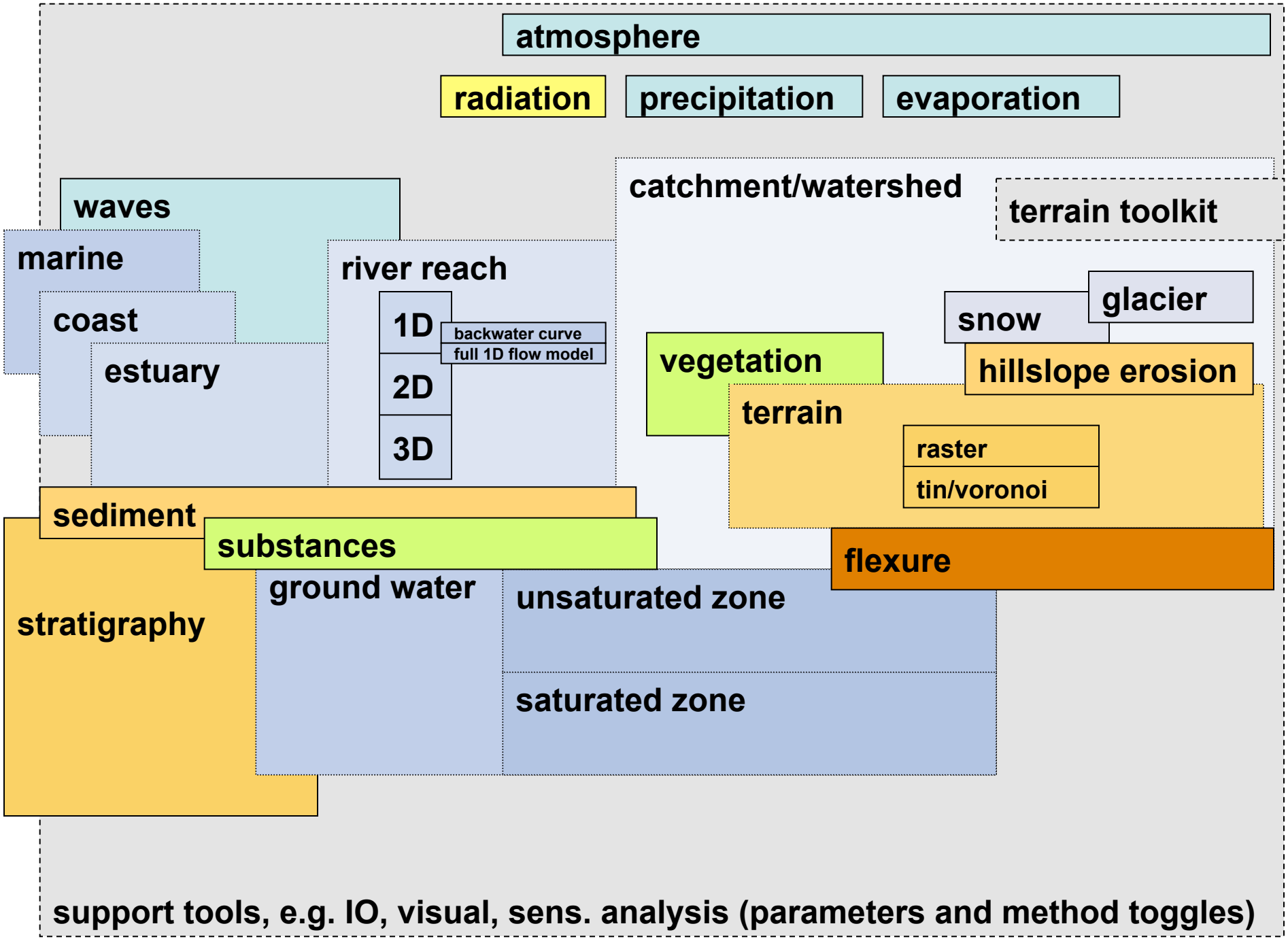
- coding camps
- stick figures in manual

Break-out “Modelling”

Toolkits (either as lib within components; or as separate component)

- e.g. terrain tools (slope, aspect, curvature)
- HPC supported
- from framework for coupling models to one supporting construction of models
- drainiage area / watershed tools
- soil heterogeneity → process tools / data tools
- collect feedback using wiki on good open source toolkits out there; start with recent book (ref. Scott)

- process tools
- terrain tools
- hydraulics tools
- basin tools


atmosphere
radiation
precipitation
evaporation
waves
marine
coast
estuary
river reach
1D
backwater curve
full 1D flow model
2D
3D
catchment/watershed
terrain toolkit
glacier
snow
vegetation
hillslope erosion
terrain
raster
tin/voronoi
sediment
substances
flexure
ground water
unsaturated zone
stratigraphy
saturated zone
support tools, e.g. IO, visual, sens. analysis (parameters and method toggles)