



# Surface exchange, Technical development and Validation

Martin Best

JULES first science meeting 28<sup>th</sup> – 29<sup>th</sup> June 2007

- Future plans for surface exchange layout
- Plans for technical development
  - Code structure
  - System structure
- Validation
  - Benchmarking



# Surface Exchange

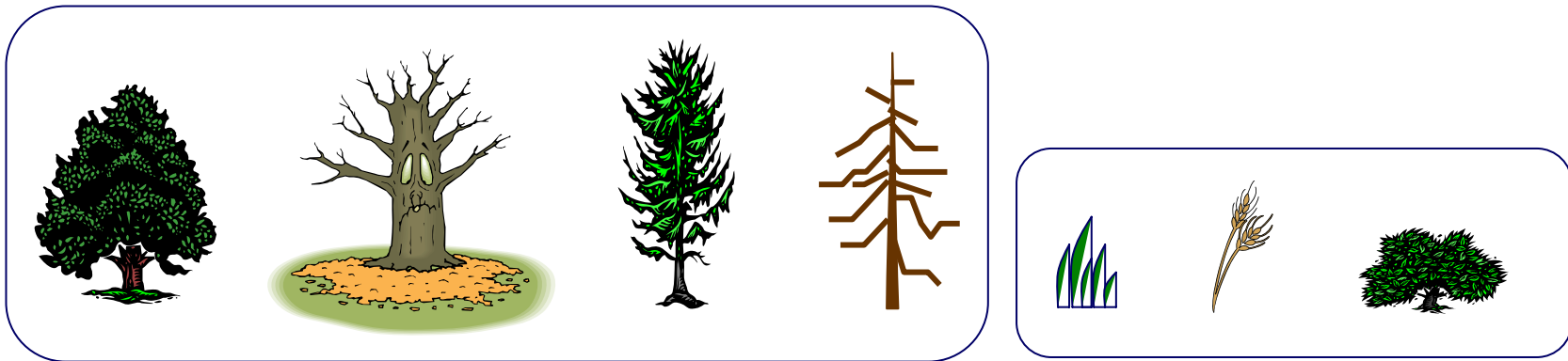
- Currently difficult to understand!!!
- Future structure
  - Fully explicit
    - $H, LE = f_1( T_*, T_1, q_1 )$
  - Penman-Monteith
    - $H, LE, T_* = f_2( T_1, q_1 )$
  - Fully implicit
    - $H, LE = f_3( T_1 = g_1( H ), q_1 = g_2( E ) )$

- **Currently**
  - Different routines for land and sea/sea-ice
  - Mixed up within code making it difficult for development and maintenance
- **Future**
  - Common routines
    - Possibility of additional functionality
  - ..... but in two sections
    - Memory space

# Flexible tiles with elevation bands



- User to specify number and definition of tiles



- Elevation associated with tiles



# Multiple source tiles scheme



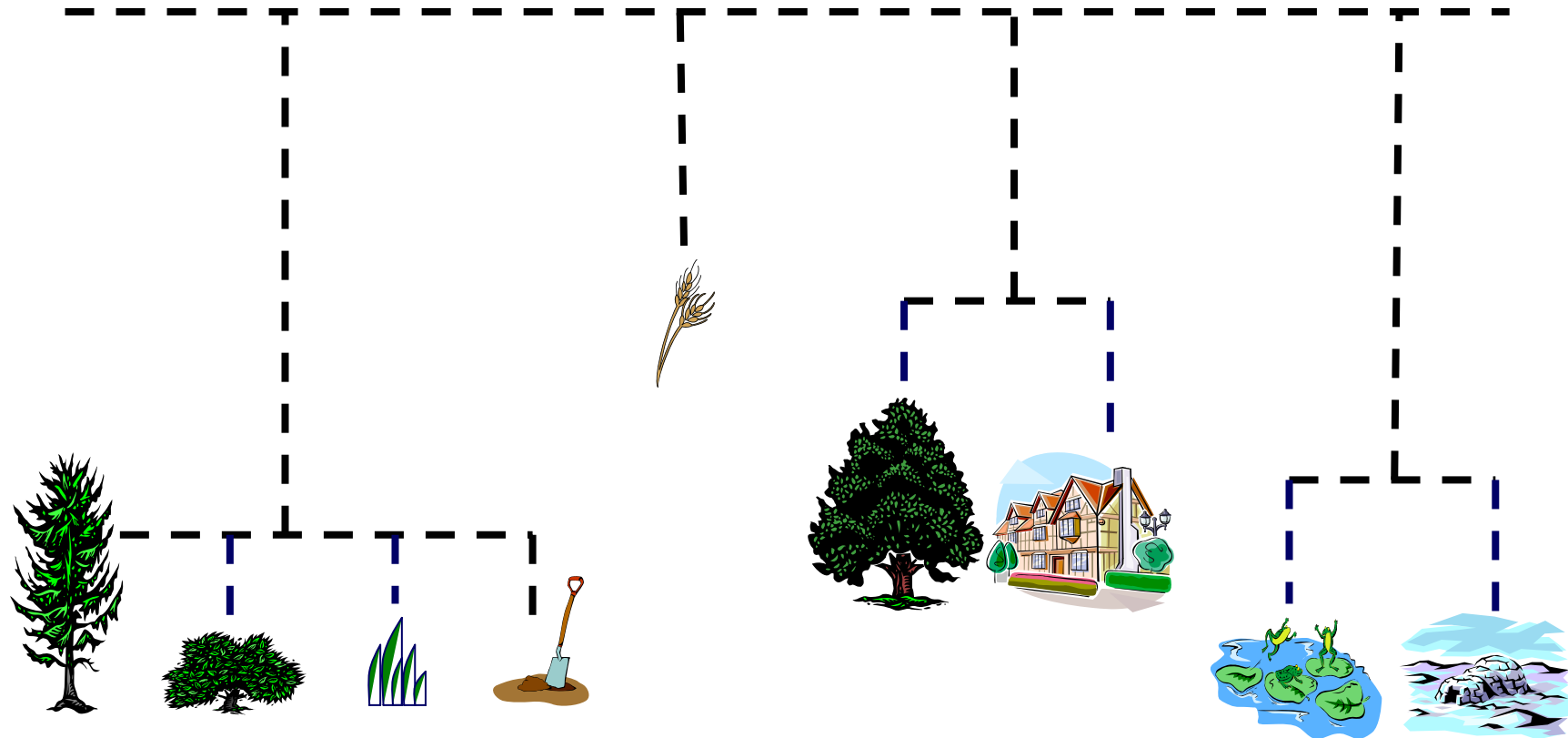
A grid of 10 dashed boxes arranged in two rows of five. The top row is empty. The bottom row contains the following illustrations from left to right: a tall evergreen tree, a small bush, a clump of grass, a shovel stuck in a mound of soil, a single stalk of wheat, a large deciduous tree, a house with a chimney, a pond with two frogs, and a rocky coastline with waves.

# Multiple source tiles scheme





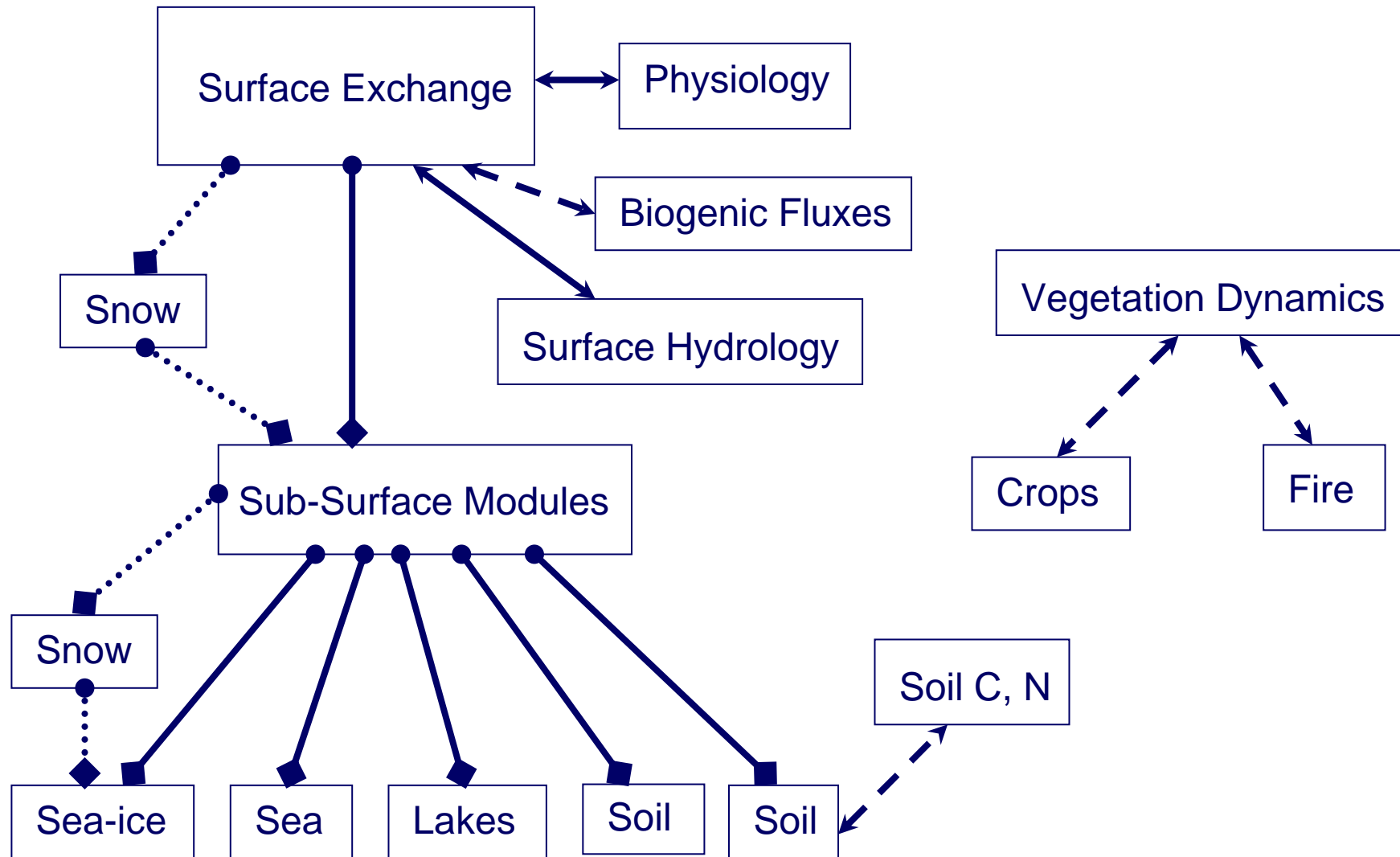
# Multiple source tiles scheme



The background of the slide features a light blue color with several overlapping, wavy, white and light blue patterns that create a sense of movement and depth. The text is centered in a dark blue, bold font.

# Technical Development

# Modular structure



- JULES code within fcm code management system
  - User changes subroutines
  - Help with code consolidation for new versions
  - Used by user community?
- How many code repositories?
  - Single repository outside of UM
    - Executable module called by UM?
    - Subroutines extracted for UM executable build?
  - + mirrored repository inside UM?



**Validation**

- Protocols for new code development
- New functionality introduced by switches
  - Bit comparison for particular model set-ups
- Set of benchmarking data
  - Must achieve acceptable validation against data to be accepted into new version
- What is acceptable?
  - Decisions made by management committee