

# *Domain-Specific Programming*

Christian Lengauer

Department of Informatics and Mathematics



# What is a Domain?

- A set of people (a user community)
  - high-performance computing
  - embedded systems
  - web computing
  - networking
- A set of techniques
  - loop parallelization
  - divide-and-conquer
  - dynamic programming
  - finite-element methods
- A set of programs (product line)
  - device drivers
  - network protocols
  - form generators
  - booking systems

# Ways of Domain-Specific Programming

- **Domain-specific language and programming environment**
  - + complete control over language and implementation
  - no reuse of general-purpose technology
- **Domain-specific program library in a general-purpose language**
  - + almost exclusive use of general-purpose technology
  - the language implementation knows nothing about the domain
    - cryptic error messages
    - no domain-specific optimizations in the language implementation

→ “active” libraries via metaprogramming
- **Domain-specific structures embedded in a general-purpose language**
  - preprocess domain-specific features
  - reuse general-purpose technology
  - separates domain-specific and general-purpose technology
  - domain-specific optimization?
- **Software product lines**
  - completely automatic program generation

# Observations

- The more domain-specific languages the better
- It will be easy to spot the language that is right for you
- The more specific a language,
  - the more powerful its compiler can be
  - the more easily building its compiler should be
- Domain-specific languages can be but need not be commercial successes
- Smaller domains should be independent of commercial interests!

→ IFIP Working Group 2.11 on Program Generation