Colorado State University
Computer Science Department

A Future without MDE

Robert B. France
Dept. of Computer Science
Colorado State University
USA
france@cs.colostate.edu

The recent past

- The Infant years (70's to early 90's): Models as documentation/communication artifacts
  - a.k.a the Computer Aided Software Engineering (CASE) generation
- The Tween years (from mid-90's): Models as artifacts for generating implementation and deployment artifacts
  - Raising the level of abstraction
  - Exemplified by OMG's work on MDA/MDD
  - Emphasis on separation of platform independent concerns from platform specific concerns

Is raising the level of abstraction the answer?

- Each successful attempt at raising the level of abstraction level triggers concerted effort to develop even more complex software
  - New programming and computer-based technologies give rise to new software opportunities
  - Result is a new generation of software that gives rise to a new breed of software development problems
- Complexity of technology-based abstraction layers leads to reliance on expert software developers
- Eventually, complexity overwhelms even the experts
- Need for raising the "level of abstraction" is painfully apparent – a "crisis" is declared …

The nature of the software development *crisis* evolves

Where are we going?

- Domain-specific development tools
  - A MDE framework provides concepts and tools that developers can use to build domain-specific development environment
- Models@run.time
  - Leveraging software requirements/design models at runtime
  - Generating adaptation analysis and reasoning models
- Compositional multi-view development of software
  - Separation of aspects/views
- Model-based development and anlysis of software product lines

Next steps

- The teenage years: MDE frameworks that support creation of DSML editors and some code generation facilities

Getting there

- Requires deep understanding of modeling phenomena
  - Understanding can only be gained through development of solutions, costly experimentation, and systematic accumulation and examination of modeling and software development experience
  - Requires working on “real-world” problems
    - Clouds
    - Energy management
    - Cyberphysical systems
    - High-performance computing
    - …

Accelerating MDE research

- Need facilities for collecting, analyzing and sharing modeling experience
  - A number of initiatives are taking form: PlanetMDE, ZoooM, Open Models Initiative, REMODD

MDE challenges

- Modeling technologies
  - Providing support for creating and using appropriate abstractions
  - Lightweight modeling tools that support exploratory, “agile” requirements and design analysis
- Multi-view separation of concerns
  - Providing support for modeling and analyzing views possibly expressed in different languages

The abstraction challenge

Finding the “right” abstractions

Separating concerns: balancing multiple, interdependent development concerns (e.g., availability, performance, survivability, fault tolerance, and security).

MDE vision?

The creation and use of models should become an integral part of the engineering of complex software, thus making the "Model-Driven" in MDE label superfluous

Conclusion

"It is easier to perceive error than to find truth, for the former lies on the surface and is easily seen, while the latter lies in the depth, where few are willing to search for it."

Johann Wolfgang von Goethe