

CS 846: Model-Based Software Engineering Winter 2012

Jo Atlee

David R. Cheriton School of Computer Science
University of Waterloo

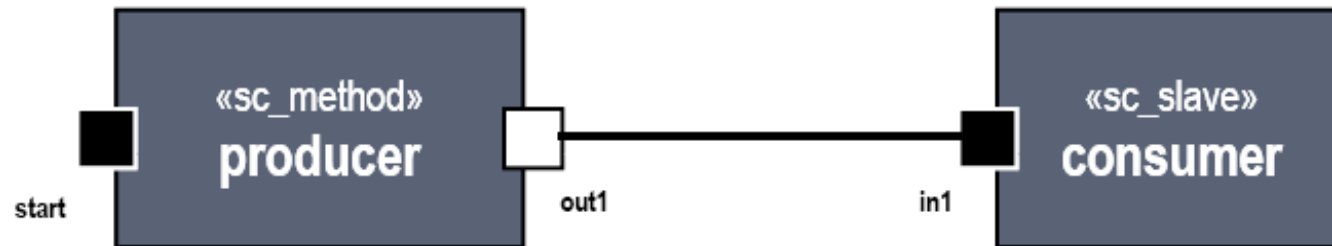
Organizational Meeting

<http://se.uwaterloo.ca/~jmatlee/teaching/846>

```
SC_MODULE(producer)
{
  sc_outmaster<int> out1;
  sc_in<bool> start; // kick-start
  void generate_data ()
  {
    for(int i =0; i <10; i++) {
      out1 =i ; //to invoke slave;}
    }
  SC_CTOR(producer)
  {
    SC_METHOD(generate_data);
    sensitive << start;});
  SC_MODULE(consumer)
  {
    sc_inslave<int> in1;
    int sum; // state variable
    void accumulate (){
      sum += in1;
    }
  }
}
```

```
SC_CTOR(consumer)
{
  SC_SLAVE(accumulate, in1);
  sum = 0; // initialize
};
SC_MODULE(top) // container
{
  producer *A1;
  consumer *B1;
  sc_link_mp<int> link1;
  SC_CTOR(top)
  {
    A1 = new producer("A1");
    A1.out1(link1);
    B1 = new consumer("B1");
    B1.in1(link1);});
}
```

Can you see what this software does?



Can you see it now?

Bill Gates on the Topic

"Modeling is the future ...

And the promise here is that you write a lot less code, that you have a model of the business process ...

So, modeling is pretty magic stuff, whether it's management problems or business customization problems or work-flow problems, visual modeling ...

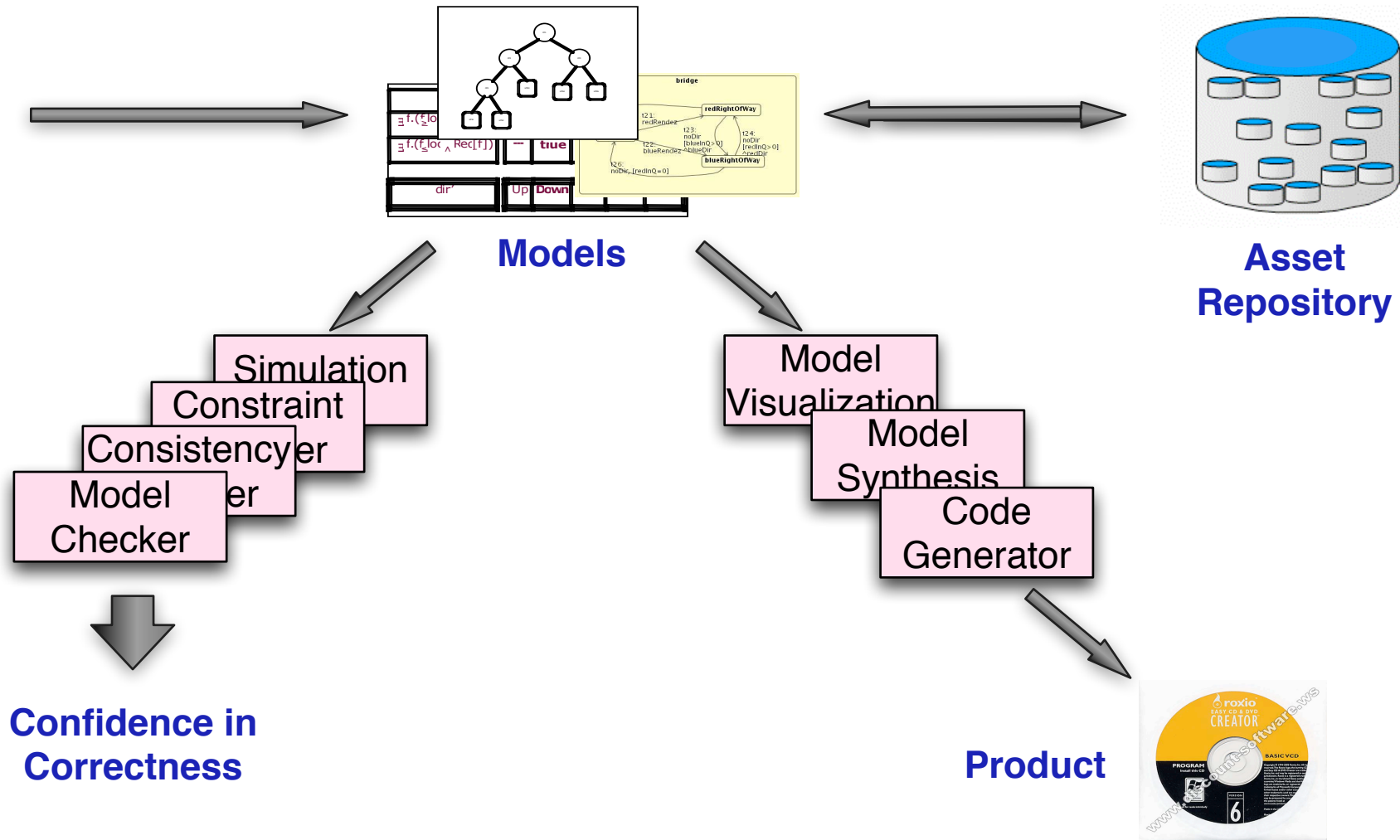
It's probably the biggest thing going on ..."

Bill Gates. "What is Bill Gate Thinking? Interview", eWeek.com, 3/30/2004

[Example from J. Dingel]

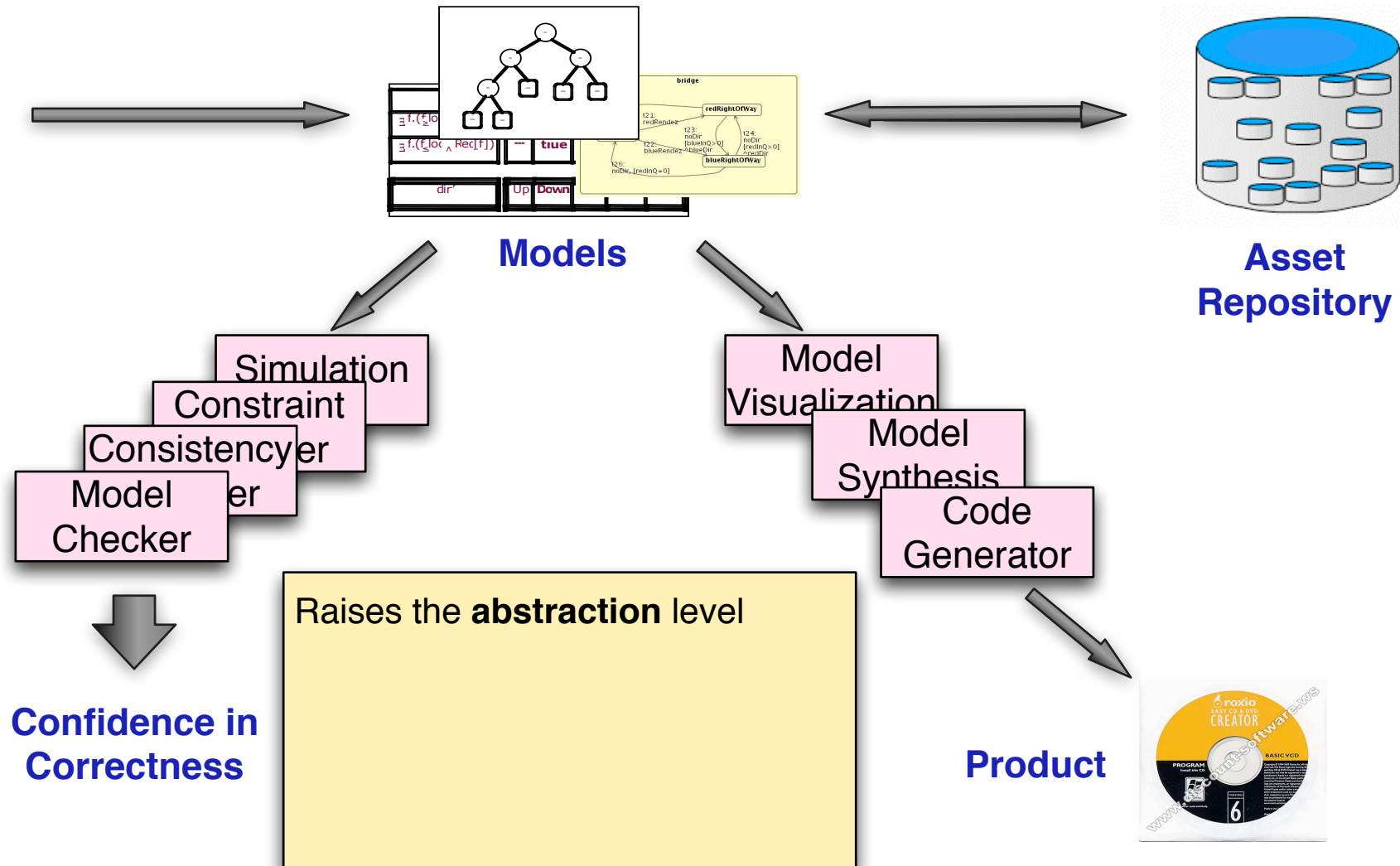
Model-Based Software Engineering

An approach to software development in which the focus and primary artifacts of development are models (vs programs)



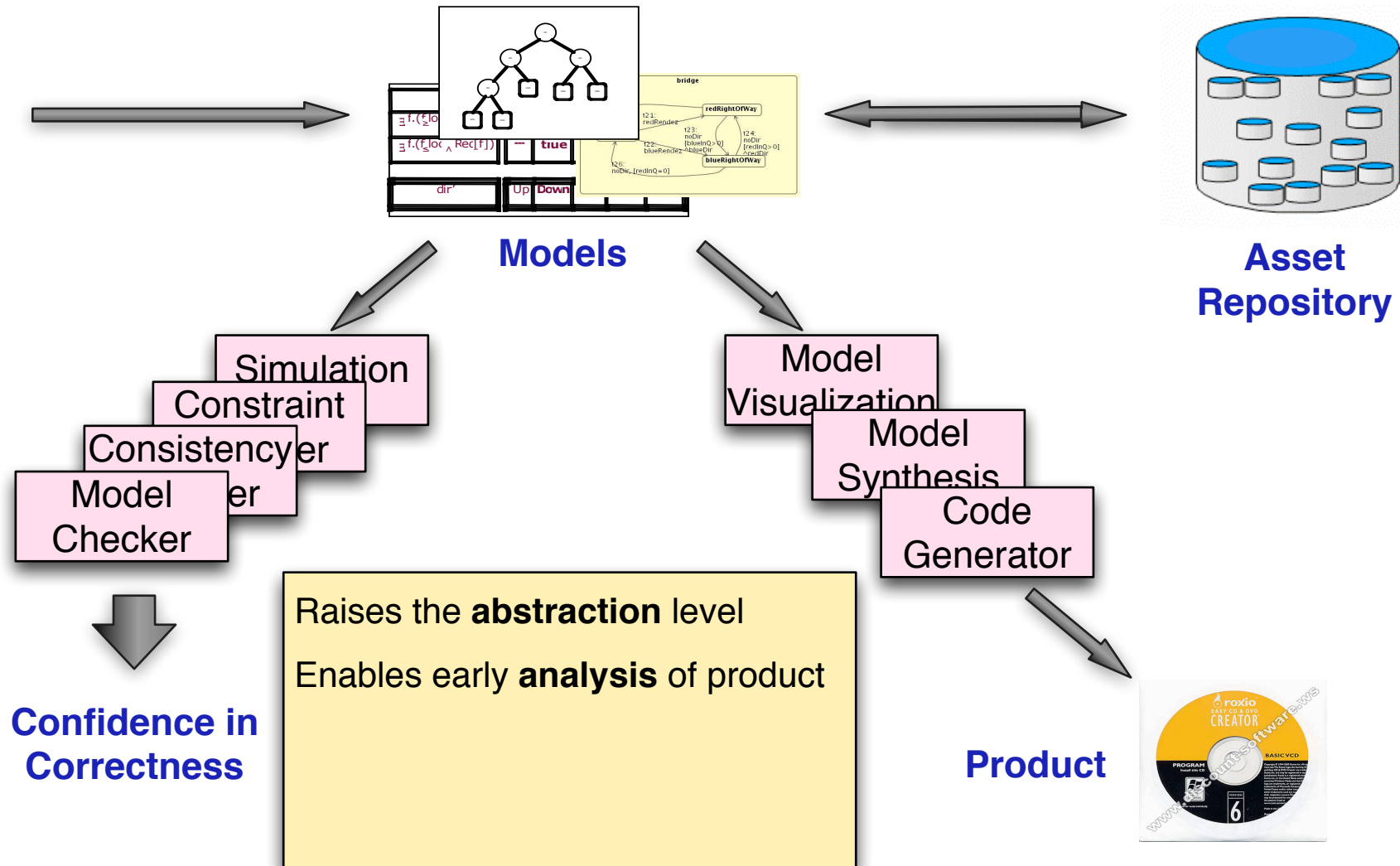
Model-Based Software Engineering

An approach to software development in which the focus and primary artifacts of development are models (vs programs)



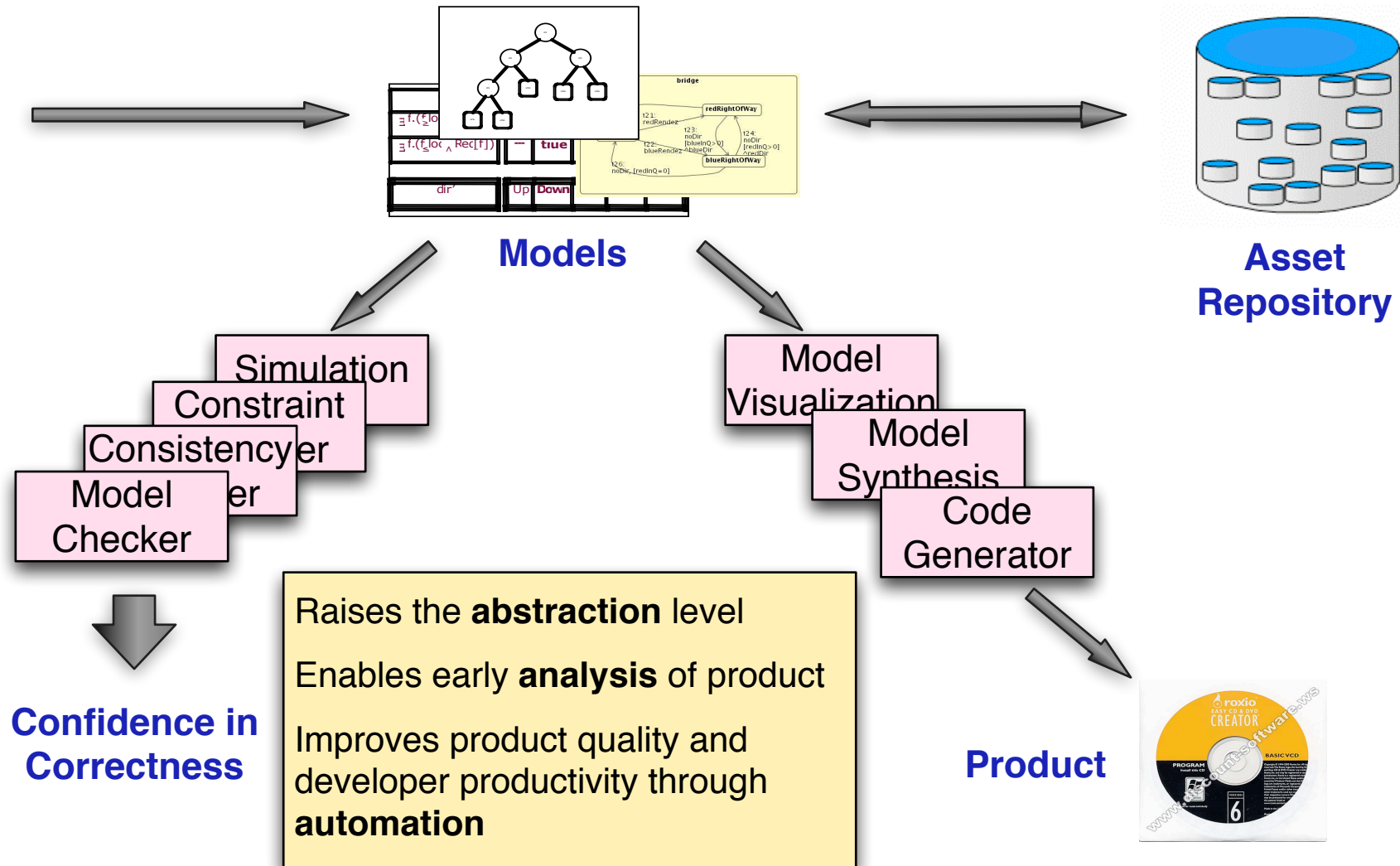
Model-Based Software Engineering

An approach to software development in which the focus and primary artifacts of development are models (vs programs)



Model-Based Software Engineering

An approach to software development in which the focus and primary artifacts of development are models (vs programs)



Sampling of Embedded Software Developed Using MDD

Automated doors, Base Station, Billing (In Telephone Switches), Broadband Access, Gateway, Camera, Car Audio, Convertible roof controller, Control Systems, DSL, Elevators, Embedded Control, GPS, Engine Monitoring, Entertainment, Fault Management, Military Data/Voice Communications, Missile Systems, Executable Architecture (Simulation), DNA Sequencing, Industrial Laser Control, Karaoke, Media Gateway, Modeling Of Software Architectures, Medical Devices, Military And Aerospace, Mobile Phone (GSM/3G), Modem, Automated Concrete Mixing Factory, Private Branch Exchange (PBX), Operations And Maintenance, Optical Switching, Industrial Robot, Phone, Radio Network Controller, Routing, Operational Logic, Security and fire monitoring systems, Surgical Robot, Surveillance Systems, Testing And Instrumentation Equipment, Train Control, Train to Signal box Communications, Voice Over IP, Wafer Processing, Wireless Phone

Several Similar Terms

Model Driven Development (MDD)

- the general notion that we can construct a model of a software system and transform it into software

Model Driven Architecture (MDA)

- the developer creates a software model that abstracts away the program's **execution platform** (e.g., the Web, CORBA, ,NET)
- tools can generate an implementation for a specific platform automatically

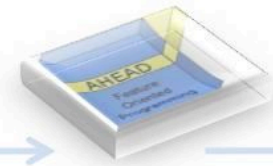
Model Driven Engineering (MDE)

- the developer creates a model in terms of the user's domain, abstracting away **software-technology concepts** (e.g., algorithms, execution platform, programming language)
- tools generate an implementation automatically

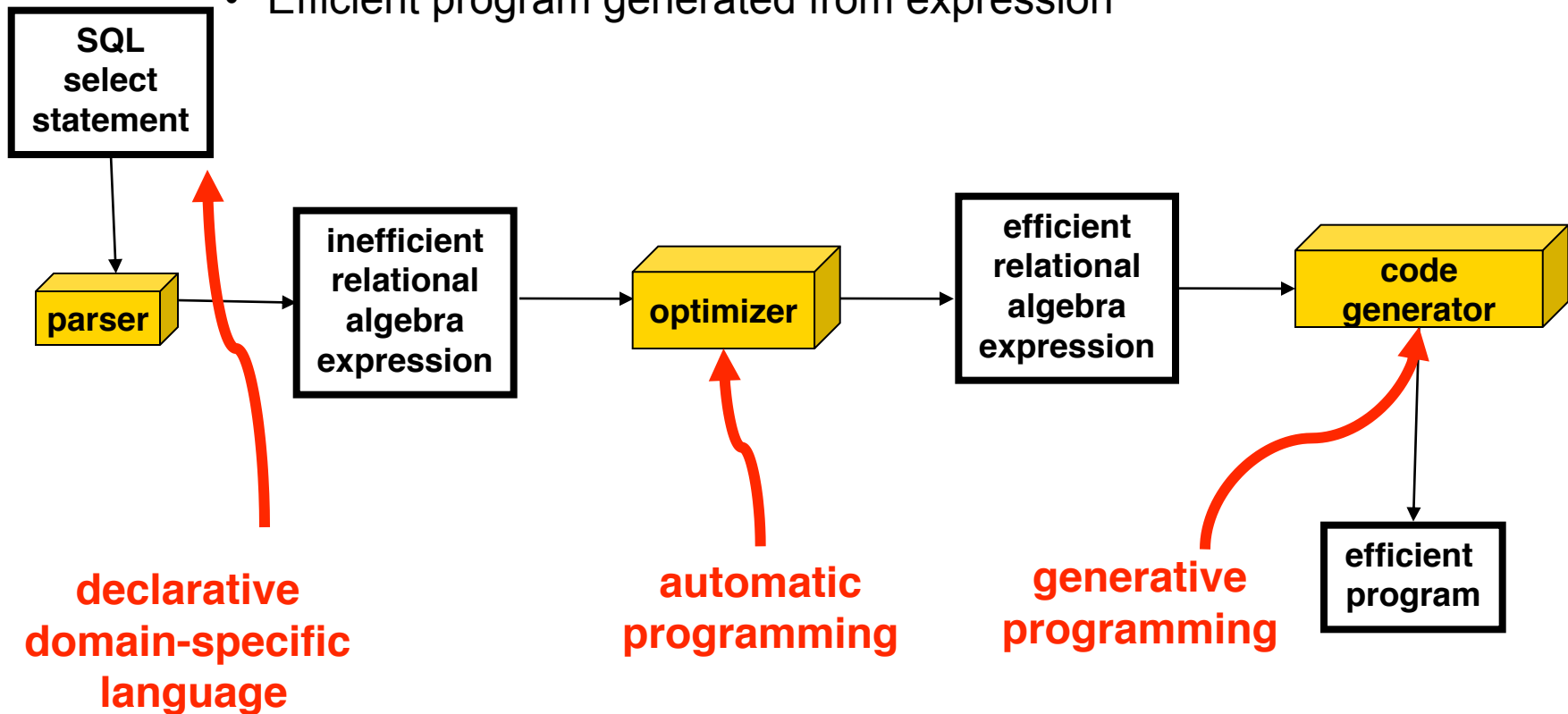
Model Based Software Engineering (MBSE)

- An approach to software development in which the focus and primary artifacts of development are models (vs programs)

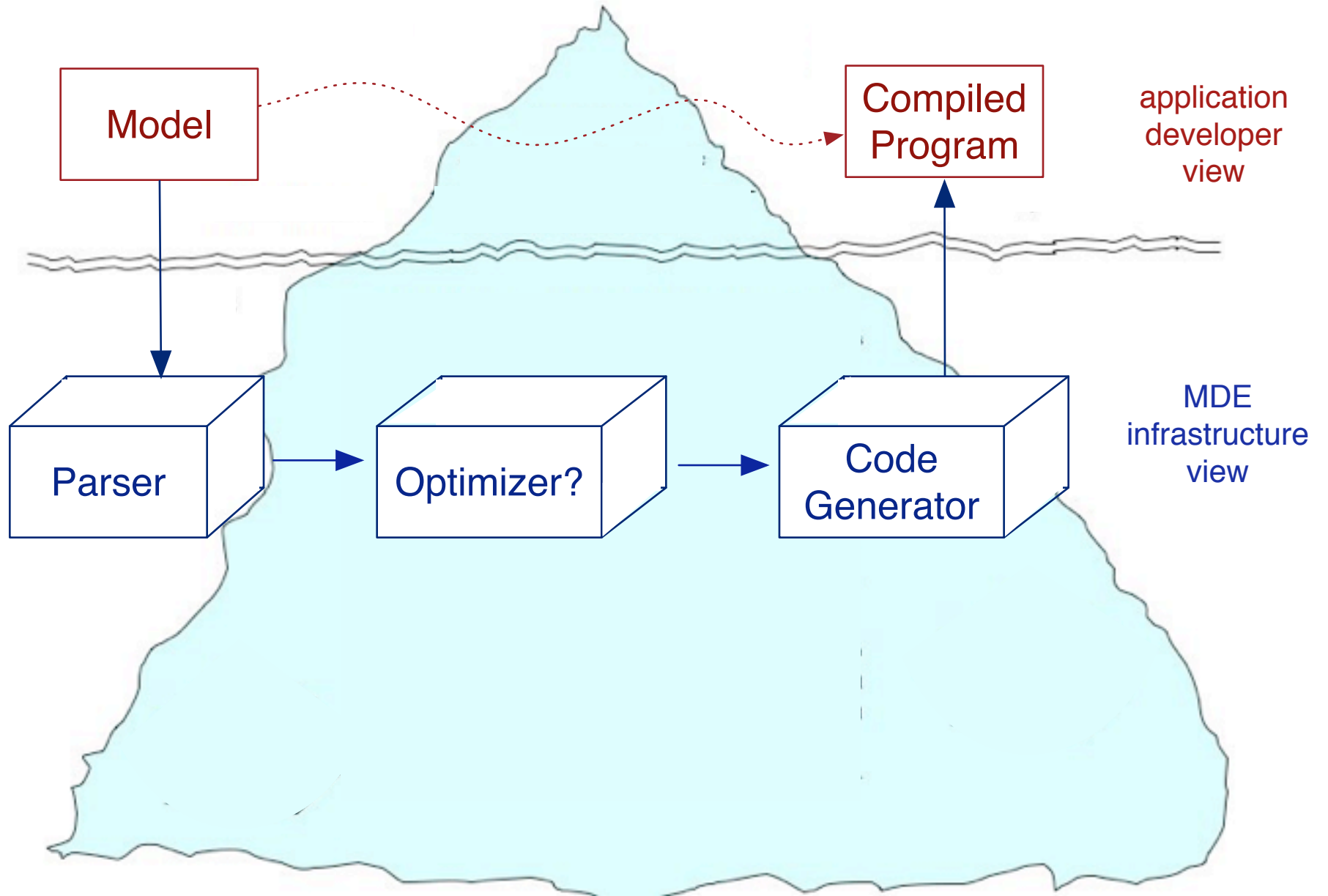
Relational Query Optimization (RQO)



- Declarative query is mapped to an relational algebra expression
- Each expression represents a unique program
- Expression is optimized using algebraic identities
- Efficient program generated from expression



Enabling MDE



Course Topics

Modelling Notations

Software models, domain-specific notations, meta-modelling

Model Management

Relationships between models, model operations (e.g., merge, match, slice, diff)

Analysis and Verification

Consistency and completeness, simulation, constraint solving, model checking

Model Transformations

Code generation, generative programming, model-to-model transformations, abstraction/refinement, model synthesis, model visualization

Assumed Background

Undergraduate course in software engineering

- Software development activities (e.g., requirements, design, testing)
- Modularity, information hiding
- Software modelling (e.g., UML)
- Sets, functions, relations, mathematical logic

Workload and Evaluation

CS 846 is a seminar course that will cover roughly 3 research papers per week.

Workload

- Course readings
- Class participation: 10%
- Paper presentations (2-3): 25%
- Paper reviews (5-7): 15%
- Term project: 50%
research problem or implementation project

Paper Presentations

Presentations:

- ~30 papers to be presented by students, up to three presentations per week
- Normally 50 minutes per paper: 25 minute presentation, followed by presenter-led discussion
- Evaluated by the class and me (form is on course web page).
 - 65% by the instructor
 - 35% by your classmates

Reviews:

- Plan on reviewing 5-7 papers
- Review form is on the course web page

Project

Types of projects

- work on an open research problem (individual)
- perform a case study using MBSE tools (teams of 1-2)

See course Web site for details

Project timeline

- 10 Feb: 1 page project proposals due
- 2 April: project presentations in class
- 9 April: project papers/reports due

Reading List

Reading list and schedule are on-line:

<http://se.uwaterloo.ca/~jmatlee/teaching/846>

- most paper links lead to ACM, IEEE, or Springer web pages from which the paper can be retrieved (from on-campus machines)
- I'm willing to consider alternative papers, if you have suggestions.

Schedule

Week	Date	Topic	Presenter
1	Jan 9	Organizational Meeting	Jo Atlee
2	Jan 16	Introduction to Software Modelling Kühne, T. What is a Model? in <i>Proceedings of Dagstuhl Seminar 04101 on Language Engineering for Model-Driven Development</i> , March 2005 —	Jo Atlee
		Fowler, M. UML Distilled, 3ed , Addison-Wesley, 2004. (Click on Online resource). — Hutchinson, J., Whittle, J., Rouncefield, M., Kristoffersen, S. Empirical assessment of MDE in industry . In <i>Proceedings of the International Conference on Software Engineering (ICSE)</i> , 2011, pp. 471-480.	
3	Jan 23	Modelling Languages	
4	Jan 30	Domain Specific Languages	
5	Feb 6	Creating Domain Specific Languages	
6	Feb 13	<i>To be rescheduled</i> Model Management	
	Feb 20	<i>Reading Week</i>	
7	Feb 27	Model Analysis and Verification	
8	Mar 5	Other Modelling Tools	
9	Mar 12	Model Transformations	
10	Mar 19	Generative Technologies	
11	Mar 26	Model-Based Testing	
12	Apr 2	Project Presentations	

Next Steps

Send e-mail by **Friday 13 January** to

`jmatlee@uwaterloo.ca`

SUBJECT: CS846

Message body should include

- your name
- your preferred e-mail address
- your research area
- titles of papers from the reading list that you would prefer to present or review. Choose up to 10 papers and prioritize your choices

I will try to have paper assignments on-line by next Monday.