Introduction to Object Oriented Modeling

R K Joshi

IIT Bombay

Modeling: Why?

Modeling: Why?

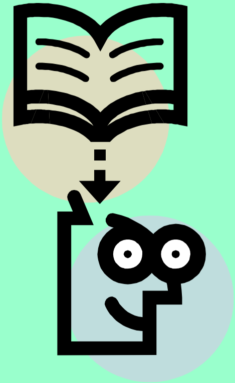
Concepts to Realization

Modeling: Why?

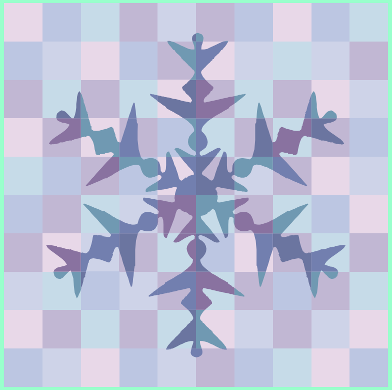
Realization to Conceptual Understanding

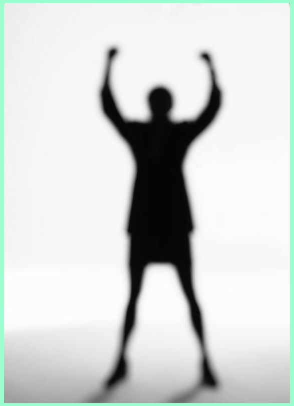
Modeling: What?

Need to Model Different aspects of/for the software to be built



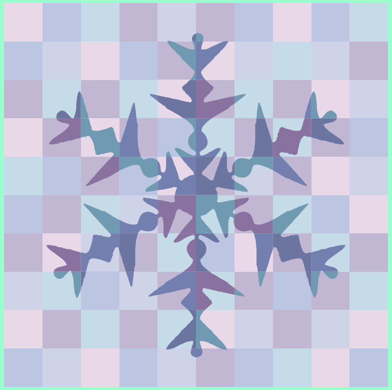
Modeling: What?

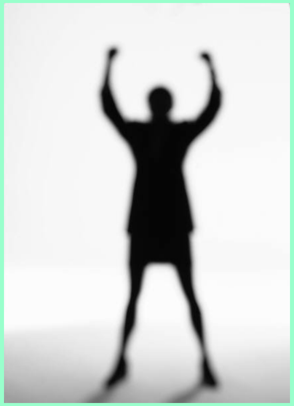




Conceptual
ideas

Modeling: What?

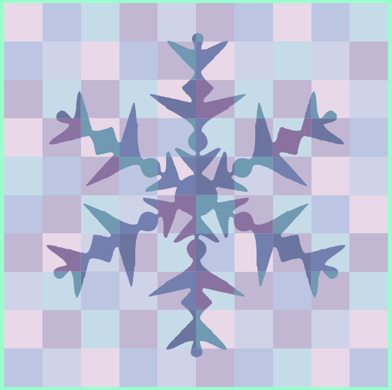


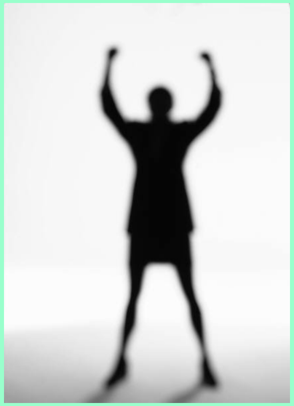


Modeling: What?

Conceptual ideas

End goals



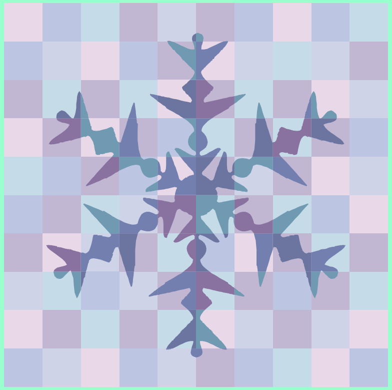


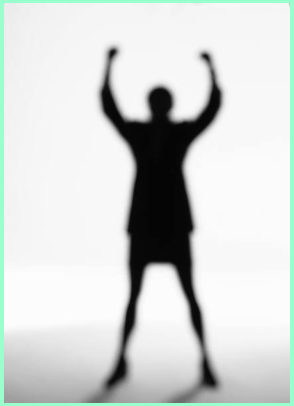
Modeling: What?

Conceptual ideas

End goals

User interfaces





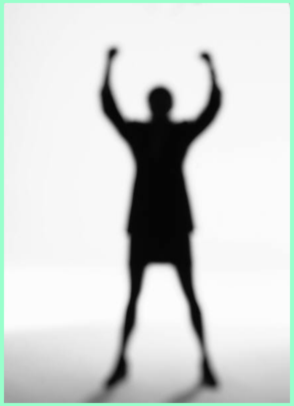
Modeling: What?

Conceptual ideas

End goals

User interfaces

Product Implementation Models



Modeling: What?

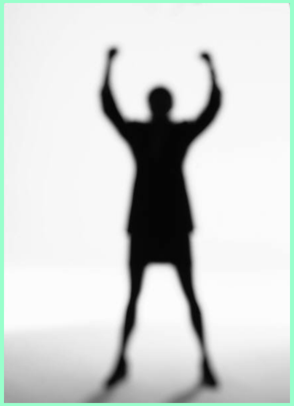
Conceptual ideas

End goals

User interfaces

Product Implementation Models

Deployment Models, N/W, Architecture



Modeling: What?

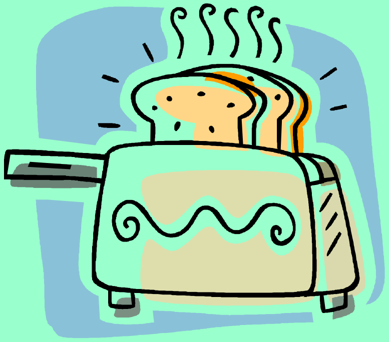
Conceptual ideas

End goals

User interfaces

Product Implementation Models

Deployment Models, N/W, Architecture



Man month

Modeling: What?

Conceptual ideas

End goals

User interfaces

Product Implementation Models

Deployment Models, N/W, Architecture

Man month

Financial Model

Modeling: What?

Conceptual ideas

End goals

User interfaces

Product Implementation Models

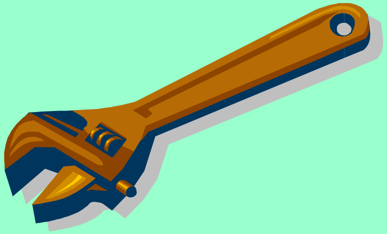
Deployment Models, N/W, Architecture

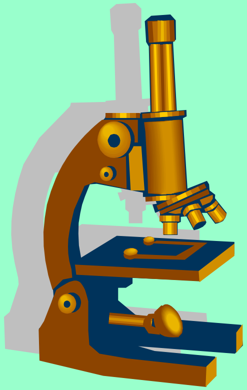
Man month

The software development process

Financial Model

Modeling

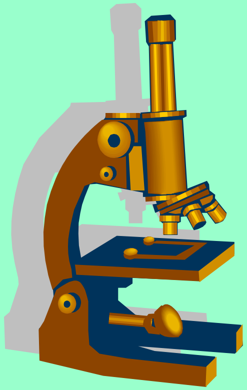




How?

Modeling

- Modeling tools



How?

Modeling

Modeling tools

Microscopic views
&
Iterative Refinement

How?

Modeling

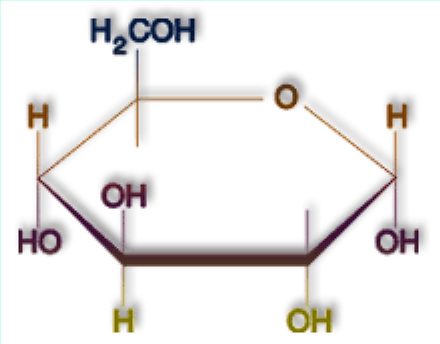
Modeling tools

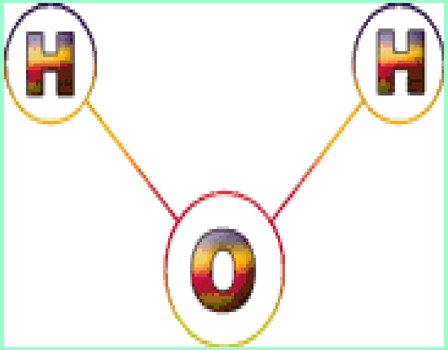
Microscopic views
&
Iterative Refinement

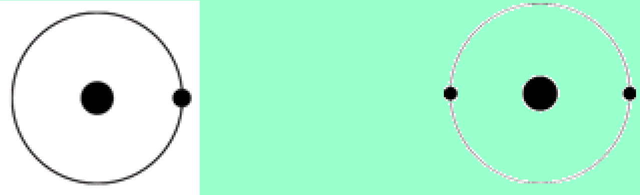
Documentation

How?

Product Models







Structure

Dynamics

Static Modeling

Furniture?

Air ducts?

Rooms?

Water pipelines

Building?

Electrical Wiring?

Flats?

what aspects to model?

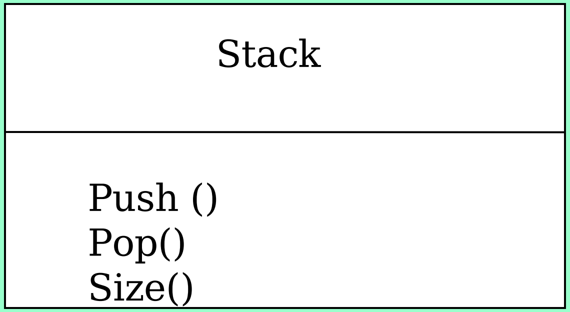
Models throughout the Lifecycle

- Per class
- Per object
- Collaborating classes
- Collaborating objects
- Collection of classes (source code)
- Location of classes, modules ..

Structural (Static) Modeling

- Class diagrams
- Object Diagrams
- Deployment Diagram
- File directory hierarchy
- Component Package diagrams
- Entity Relationship

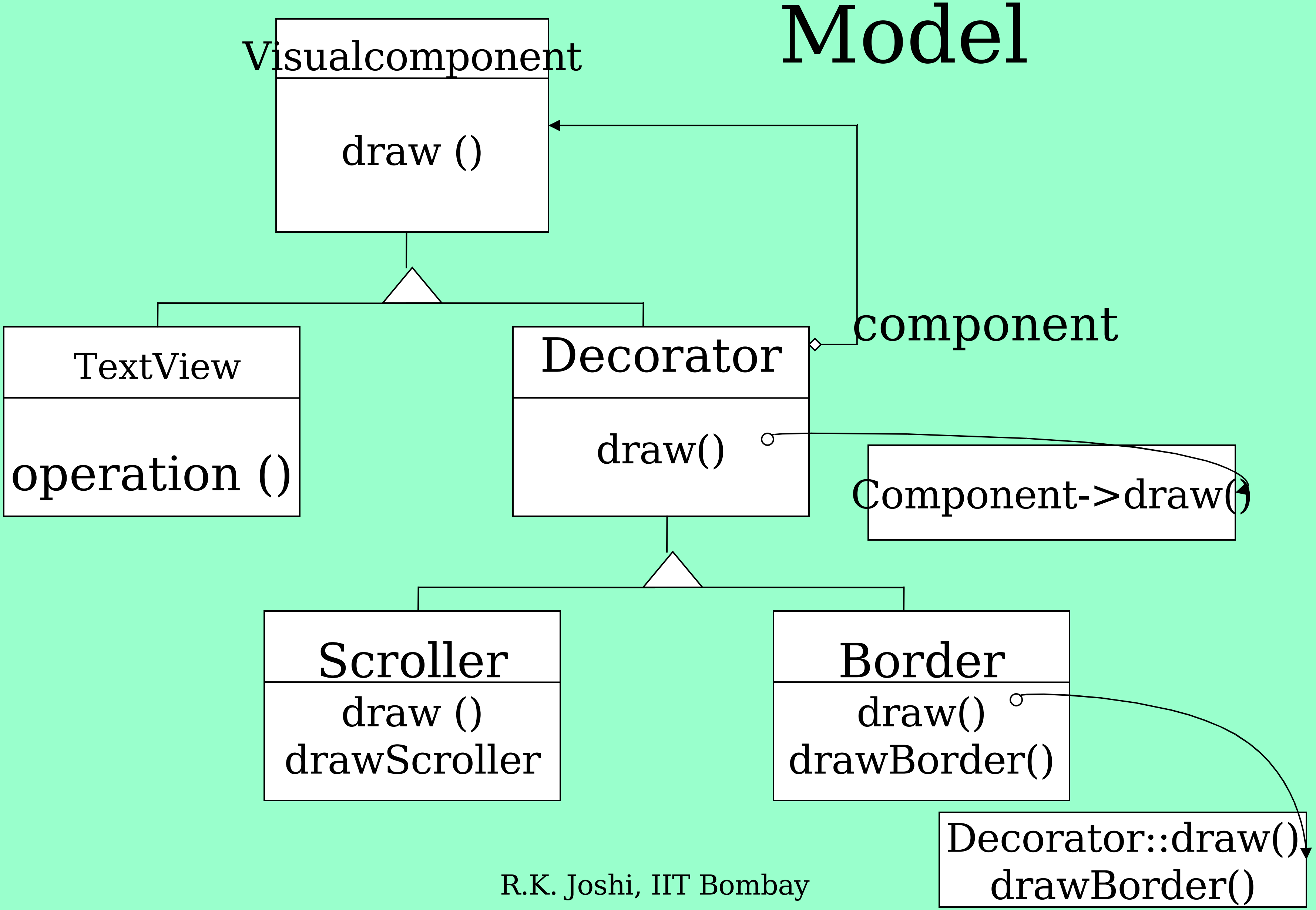
A Static Model



Stack

Push ()
Pop()
Size()

Another Static Model



Visualcomponent

draw ()

TextView

operation ()

Decorator

draw()

component

Component->draw()

Scroller

draw ()

drawScroller

Border

draw()

drawBorder()

Decorator::draw()

drawBorder()

Dynamic Modeling

what aspects to model?

Dynamic Modeling

How does the group collaborate?

what aspects to model?

Dynamic Modeling

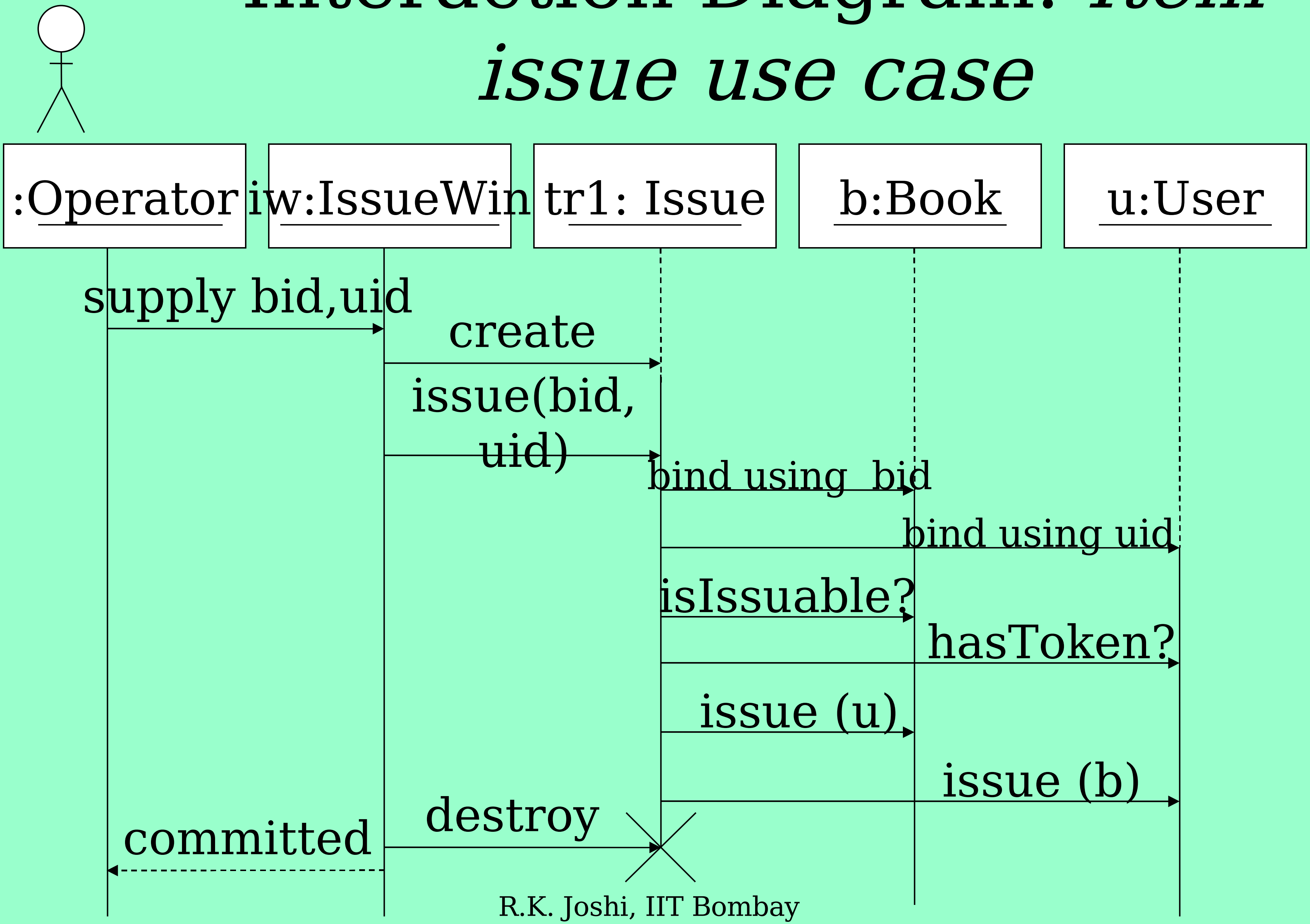
How does the group collaborate?

How does the individual respond?

what aspects to model?

What happens amongst the objects?

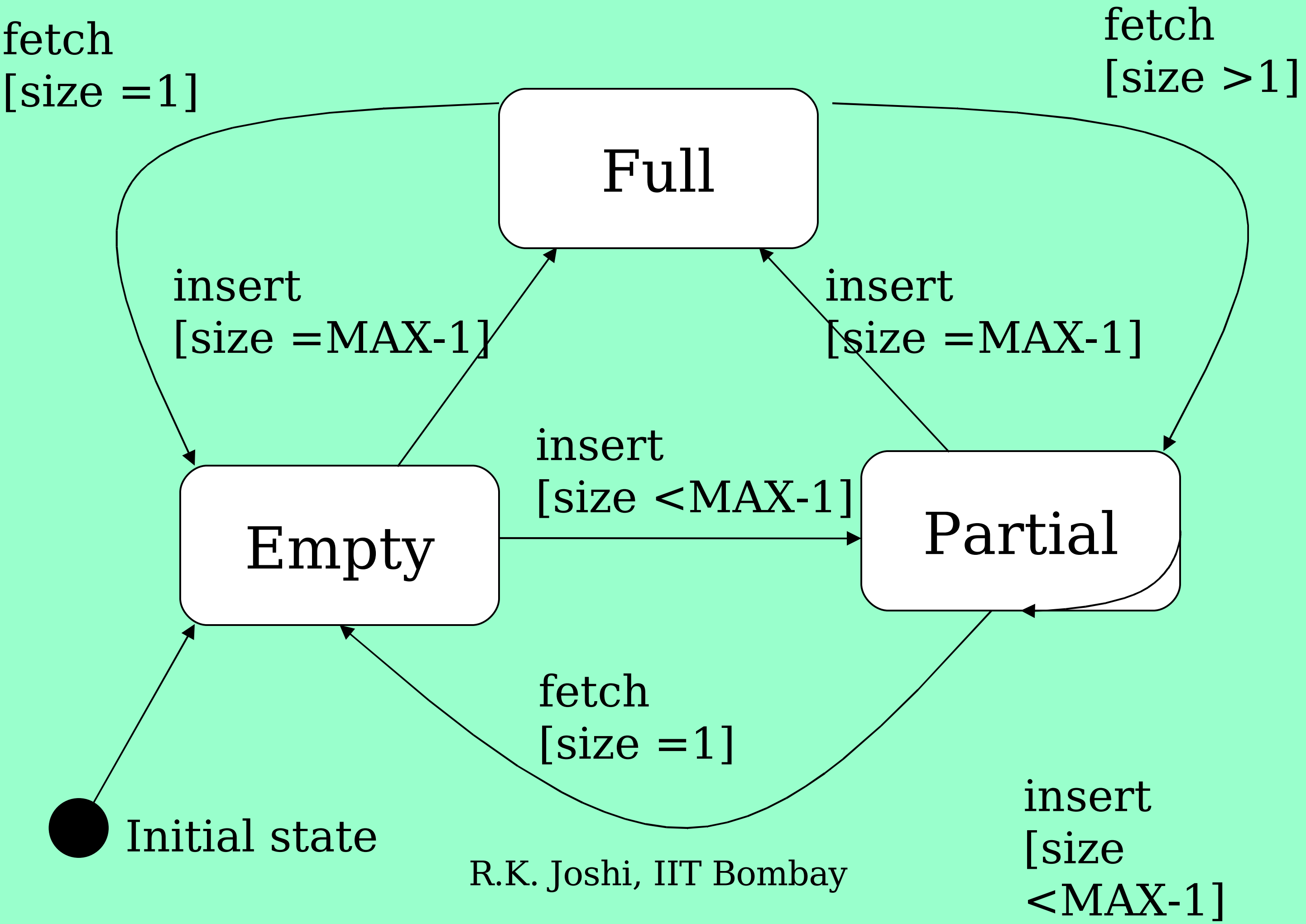
Interaction Diagram: *Item issue use case*



R.K. Joshi, IIT Bombay

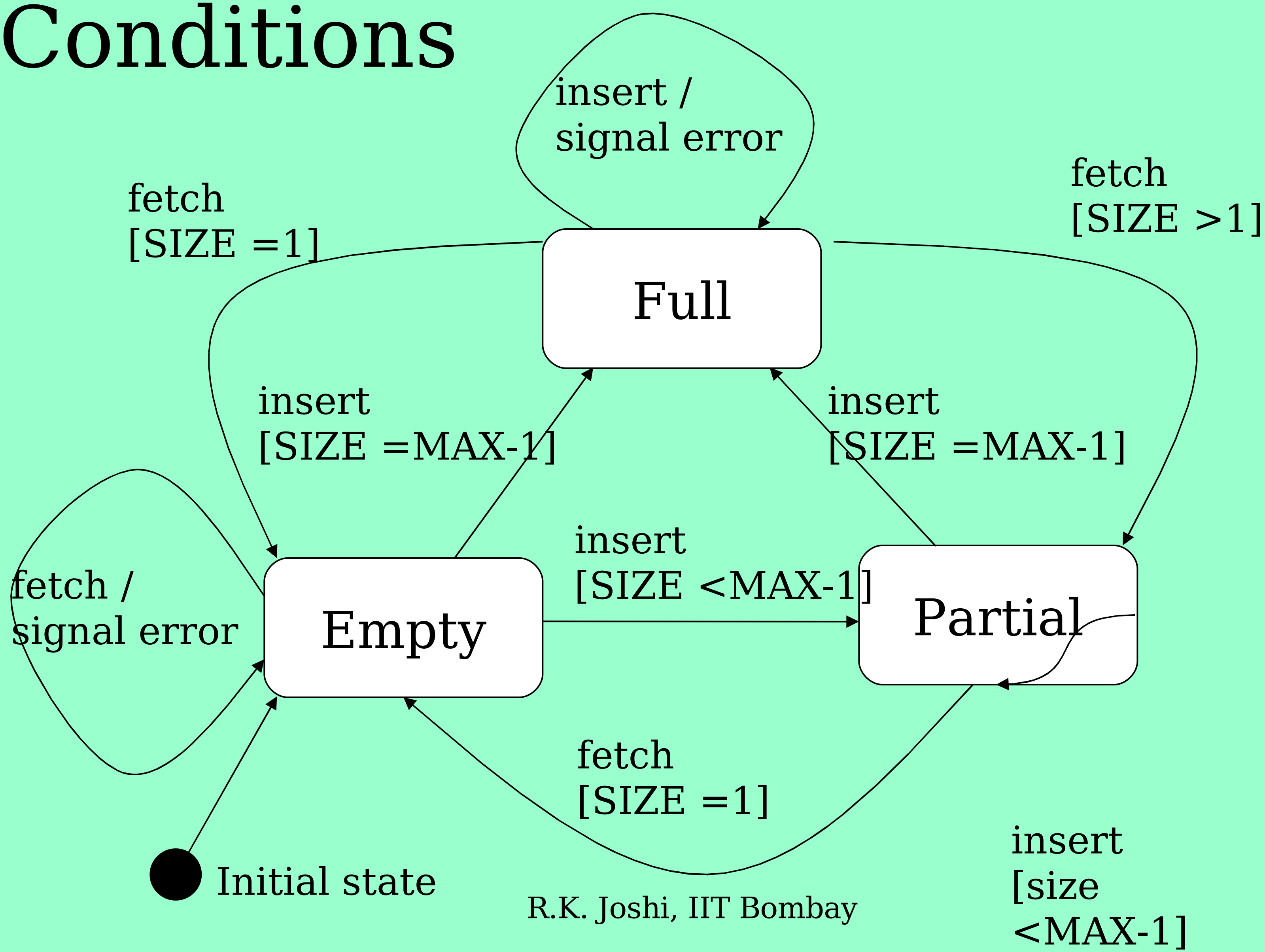
What happens inside an object?

A State Machine



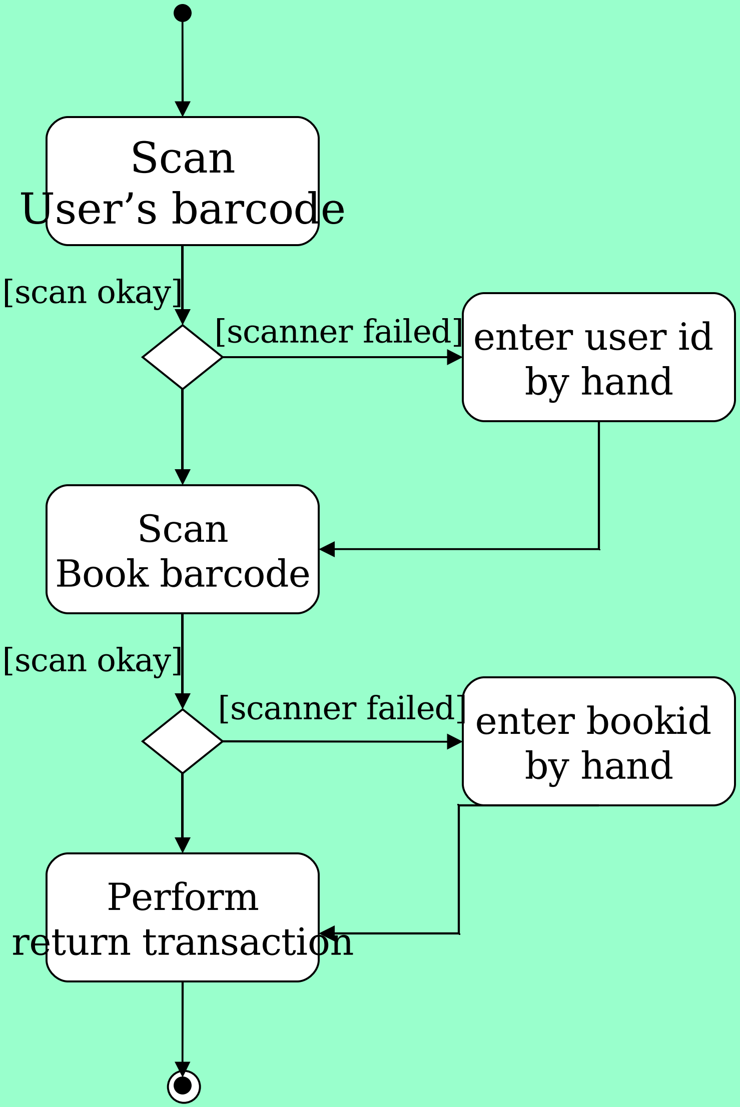
R.K. Joshi, IIT Bombay

Including Error Conditions



How are various interactions/activities linked with each other?

Activity Diagram



How much can you model?

- Can the full system be specified during modeling?
- What other modeling languages exist?
- Can the full system be generated from the model?
- Benefits of Modelchecking
- Modeling language vs. implementation language

Code Generation

- Class structures
- Basic Relations
- Basic constructions
- Helper Code
- Service Orientation
  - Intermediate formats and Platform dependent code generation techniques

Traceability

- Models
  - Seamlessness
- Moving from early conceptualization to construction and eventually into delivery and maintenance
- Changes reflected back

Summary

- Model ---> Implementation
- Implementation --> Model
- Forward engineering
  - Specify/model first and then go on to build
- Many facets of a system --> many models
- Consistency between models
- Traceability to Implementation
- Automatic Code Generation