

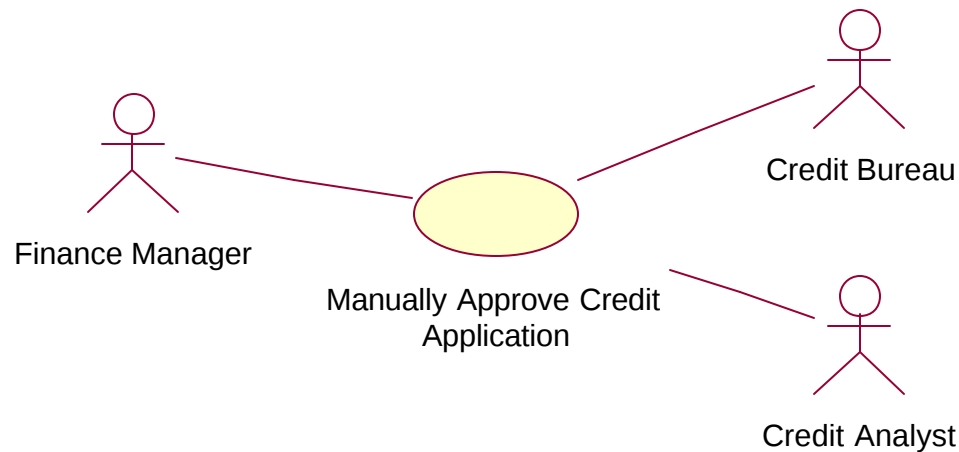
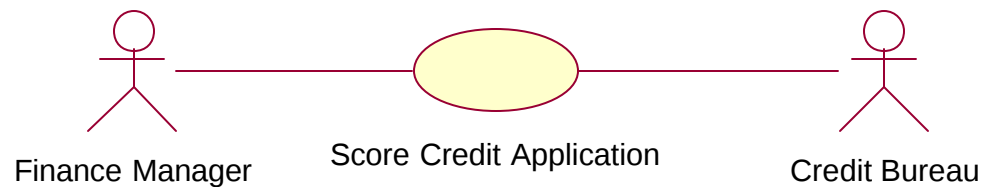
Use Cases? We've Had Them All Along

Marc J. Balcer
ThoughtWorks, LLC



Use Cases...?

- OO Orthodoxy: "Start with use cases"



Use Case Traps

- “Functional decomposition using objects.”
- “Use case:
Object-speak for informal text specification”

It's not hard to see why...

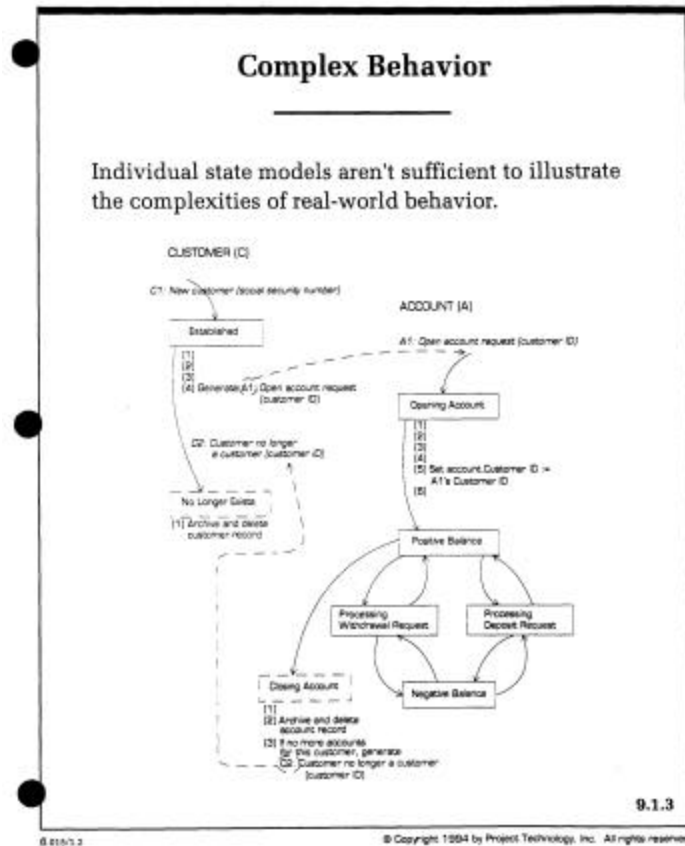
“Submit an application for scoring”

Each time we submit an application for scoring, create a Scoring Request. That's sent to the credit bureau; depending upon how it comes back the customer may or may not be approved; we might get an error and have to resubmit, or if the user doesn't provide a customer ID we may get a “list of similars” and have to pick one name from the list and resubmit it. Each time we submit an application for scoring, create a Scoring Request. That's sent to the credit bureau; depending upon how it comes back the customer may or may not be approved; we might get an error and have to resubmit, or if the user doesn't provide a customer ID we may get a “list of similars” and have to pick one name from the list and resubmit it.

From Objects to States

- Describe behavior
- Ease transition from information modeling to state modeling
- Help figure out layering of responsibility
- Sort in-scope and out-of-scope capabilities

What SMOOA does say...



- Can't just start drawing object lifecycles ad hoc
- Need to plan coordination and control
- Suggest drawing a preliminary OCM

Use Case Process

- Build the OIM
- Define use cases
- Identify conditions
- Partition into choices
- Form scenarios as choice combinations
- Diagram each scenario
- Assemble state models

Build the OIM

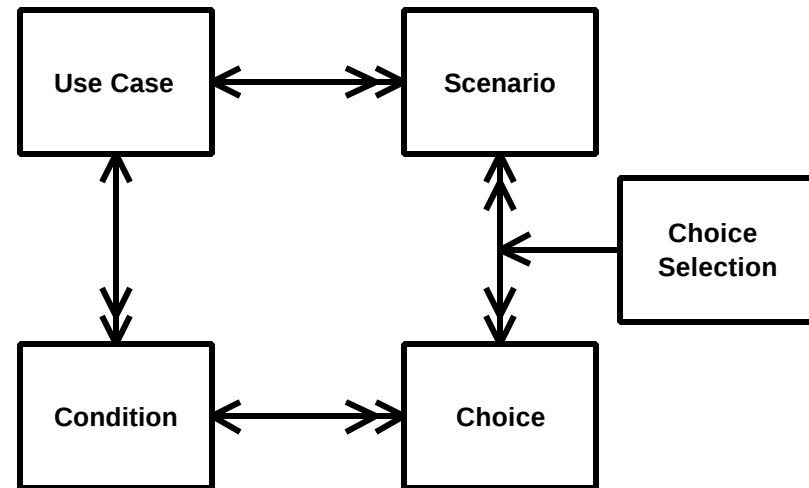
- Need to have objects in place
- OIM does not need to be perfect
- Use case development often identifies additional objects, changed attributes, new relationships

Define Use Cases

- NO INFINITE HIERARCHIES
- Reasonable level = one user operation
 - create a credit application
 - submit an application for scoring

Basic Terms

- Use Case
- Scenario
- Condition
- Choice



Scenarios

- Scenario: one specific situation; a use case subject to particular conditions

“score a credit app that has complete information, where the customer is creditworthy enough to pass, and bureau connections are up.”

Conditions

- Individual factors that affect a use case
- Partitioned into choices

Condition “creditworthiness”

- clean credit
- only 30-day lates
- 60 & 90-day lates
- writeoffs, judgments, or bankruptcy

Condition “credit bureau connection status”

- up
- down

Combine Conditions

- Draw a choice selection from each of the conditions

One use case →	Customer Information	Credit Worthiness	Bureau Connections
	Complete	Clean	Up
	Complete	Clean	Down
	Complete	30 day late	Up
	Missing ID number	Clean	Up
	.	.	.
	.	.	.
	.	.	.

Combinatoric Explosion!

- Few conditions can lead to lots of scenarios
- Need to limit
 - redundant scenarios
 - impossible scenarios

Sources for Conditions

- External entity events
(solicited & unsolicited)
- States of objects
- Attribute values
- Other conditions on external entities

Modeling Scenarios

- How to represent scenarios?
- Must be tied into the rest of the formalism
- Quick to create; easy to evaluate
- Notations:
 - UML: sequence diagrams
 - Shlaer-Mellor: thread of control chart

Thread of Control Chart

- Presented as a means for representing a simulation
- Similar to UML sequence diagram
- Very useful for illustrating a scenario

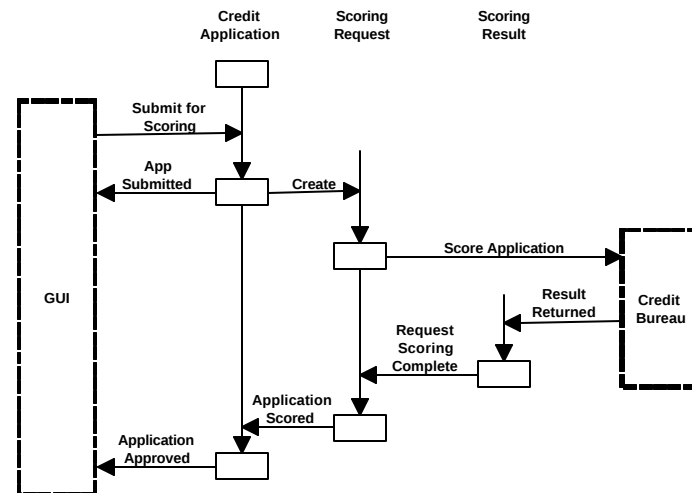
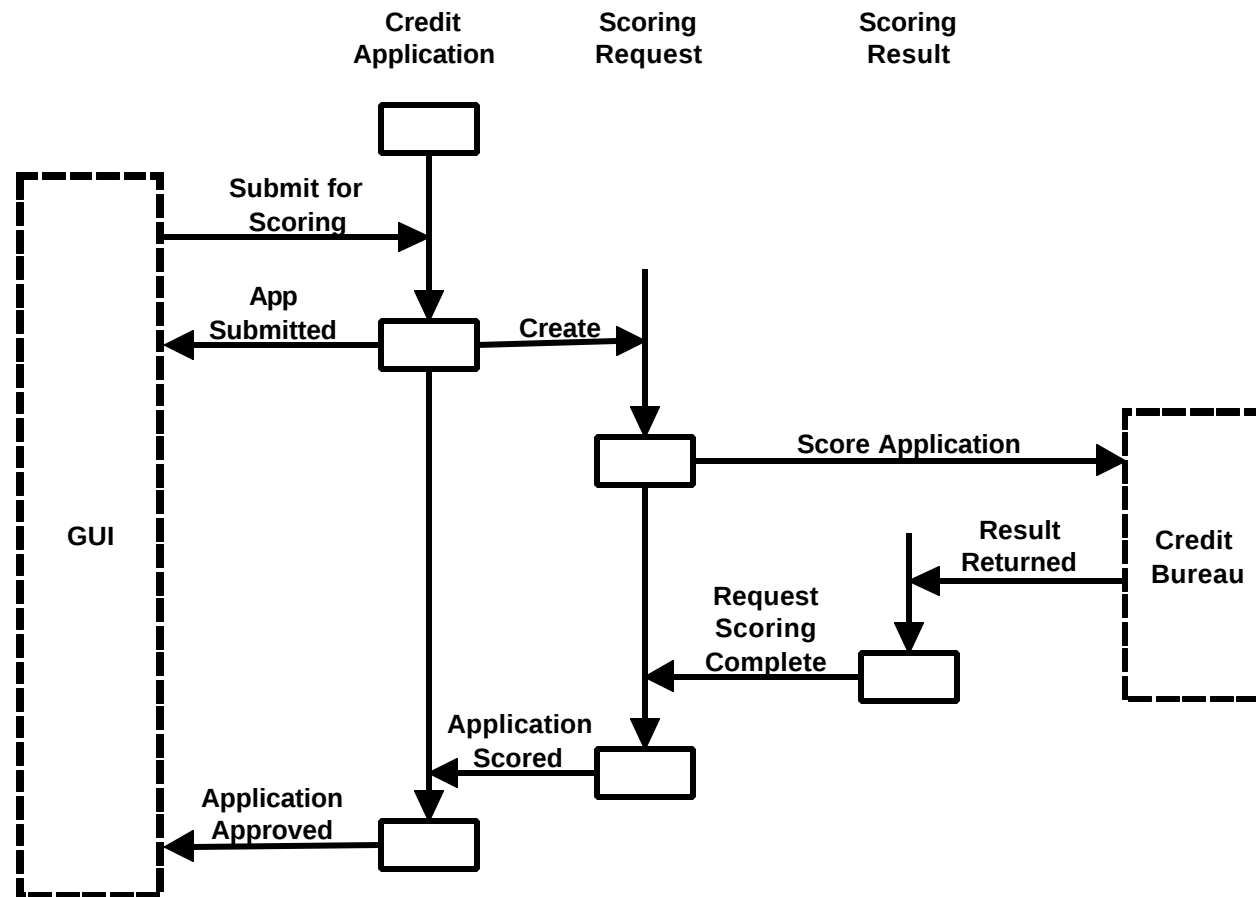


Diagram for "score app"



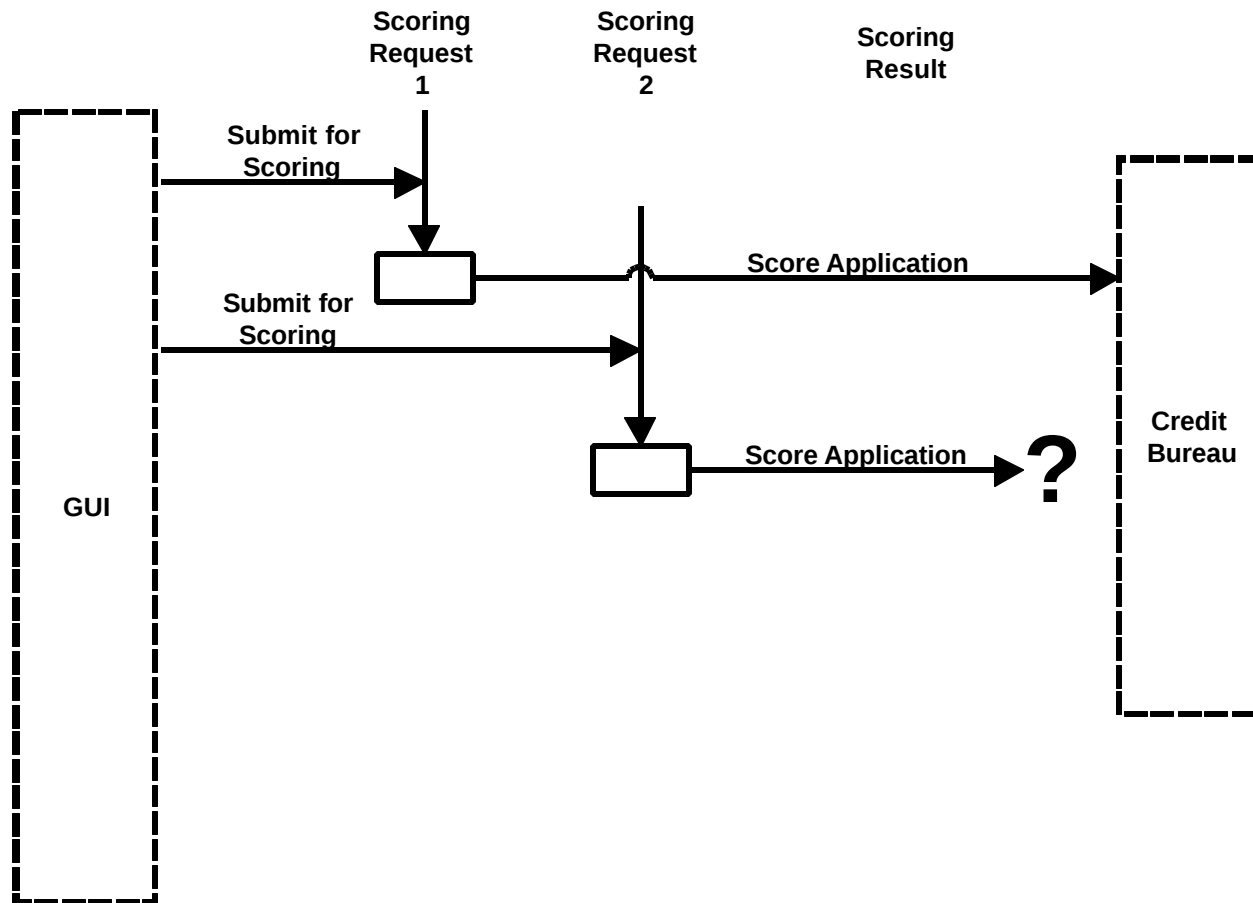
Just Like Simulations

- Pick your instances
- Determine their initial “stages”
- Determine who gets the initial event
- Play through layers of control

Planning Coordination

- Who gets the initial event to score an app?
 - If it just creates the credit scoring request, then we could make several scoring requests for the same app or customer...?

Planning Coordination



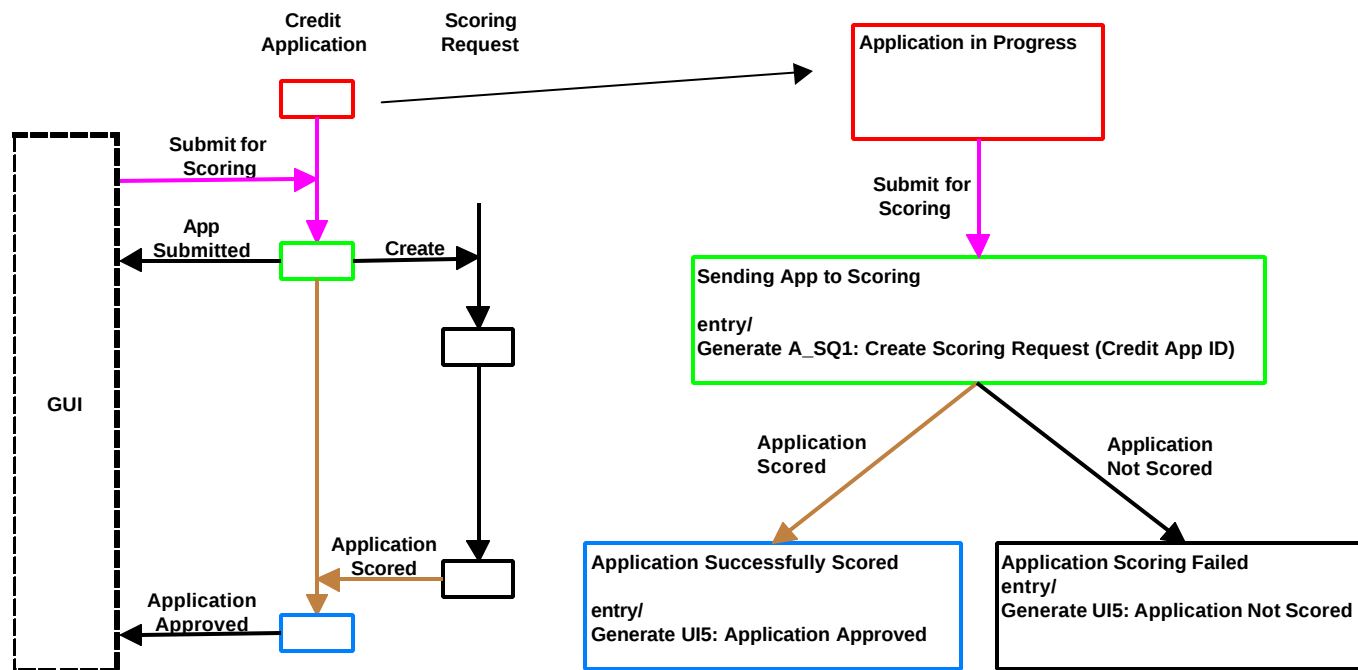
Intended Audience

- Sequence diagrams are engineering documents
- Illustrate use cases for users, clients, management using
 - screen navigation traces
 - cartoons
 - sock puppets...

On To State Models

- Assemble vertical threads into state model fragments
- Assemble an event list and OCM
- Formalize actions
- Look for consistency
 - who gets initial events
 - who “knows” what

On To State Models

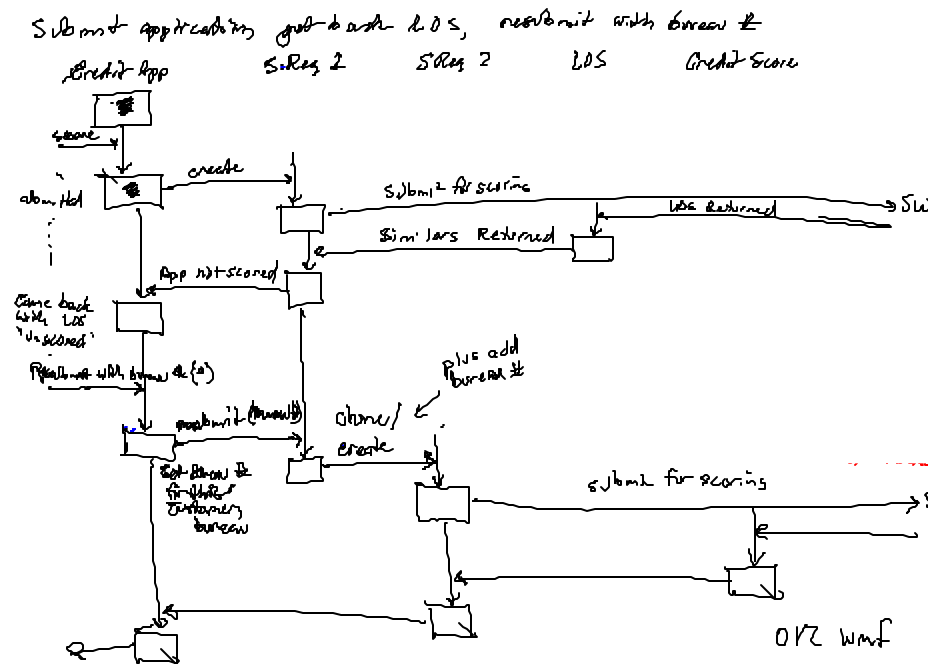


And Beyond...

- Scenarios can be used for testing of models and finished system

Tools - Now

- Scenarios created on a printing whiteboard



Tools - Future

- Tools for forming combinations of conditions to produce scenarios
 - Similar to test-case combination generators
- Integrate scenario-drawing with state modeling tool
 - Capture event list
 - Assemble state model fragments

Results of This Approach

- Identified coordination issues far sooner
- Bridged information & state modeling
- Bounded the “use case” process