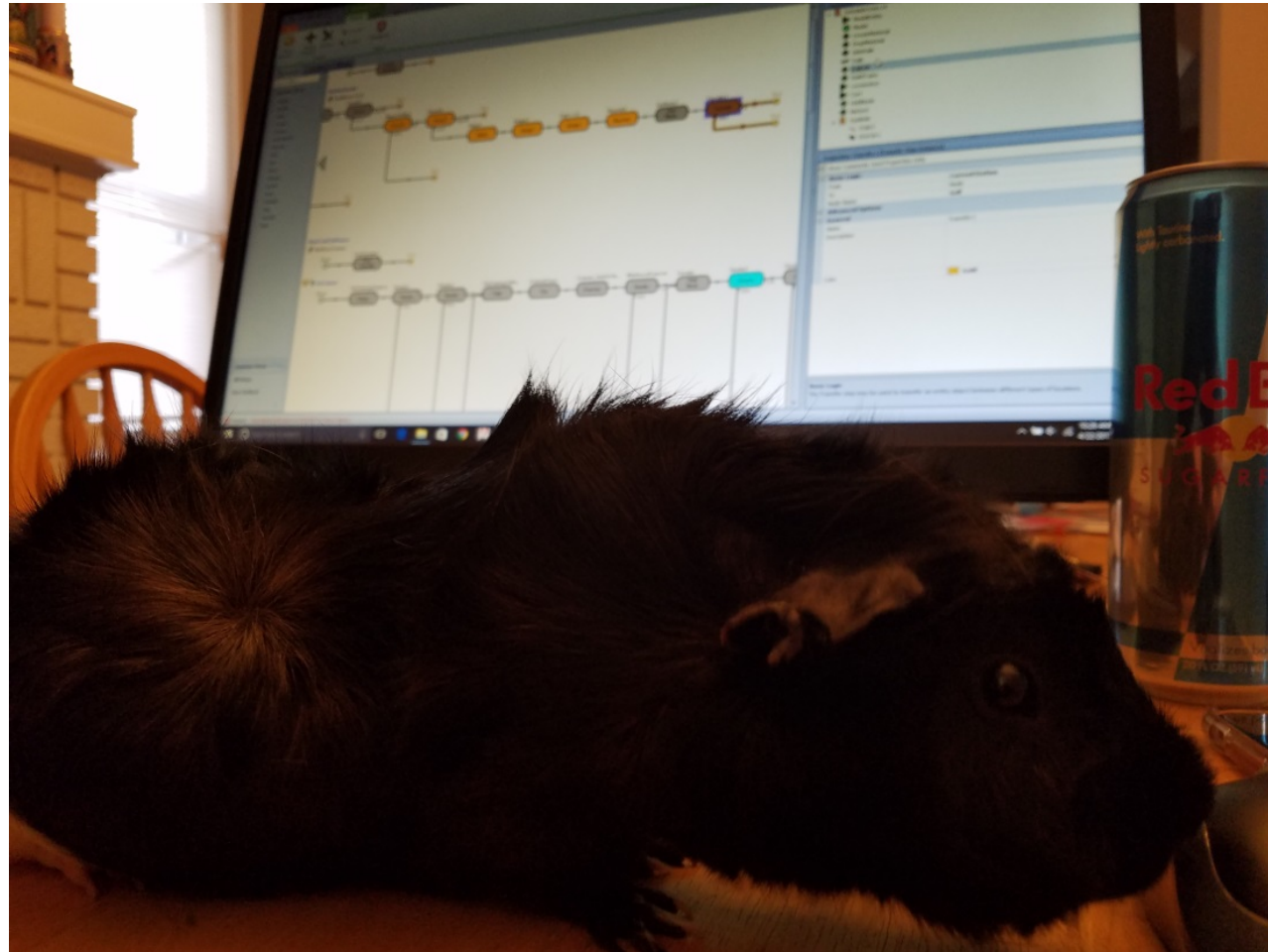


“I think the problem is in the Transfer Step” – Ginny The Simulation Piggie



# Flexible Models in Simio

By Jacob Ingalls

# The Oddest Duck in the Pond

- Education
  - Undergrad – BS in Finance from LeTourneau University (2007)
  - Graduate – MBA from Oklahoma Christian University (2012)
  - 5 year apprenticeship at Diamond Head Associates under Dr. Ricki Ingalls
- 10 Years in the Industry
- Simulation Areas I have Worked
  - Steel
  - Energy
  - Bio Medical
  - Logistics (Rail/OTR/Package Delivery)
  - Material Handling Design

# Industry Experience



Consultant



Business Analyst/Director of Simulation



Simulation and Modeling Specialist IV



Lead Simulation Consultant



Industrial Engineer III

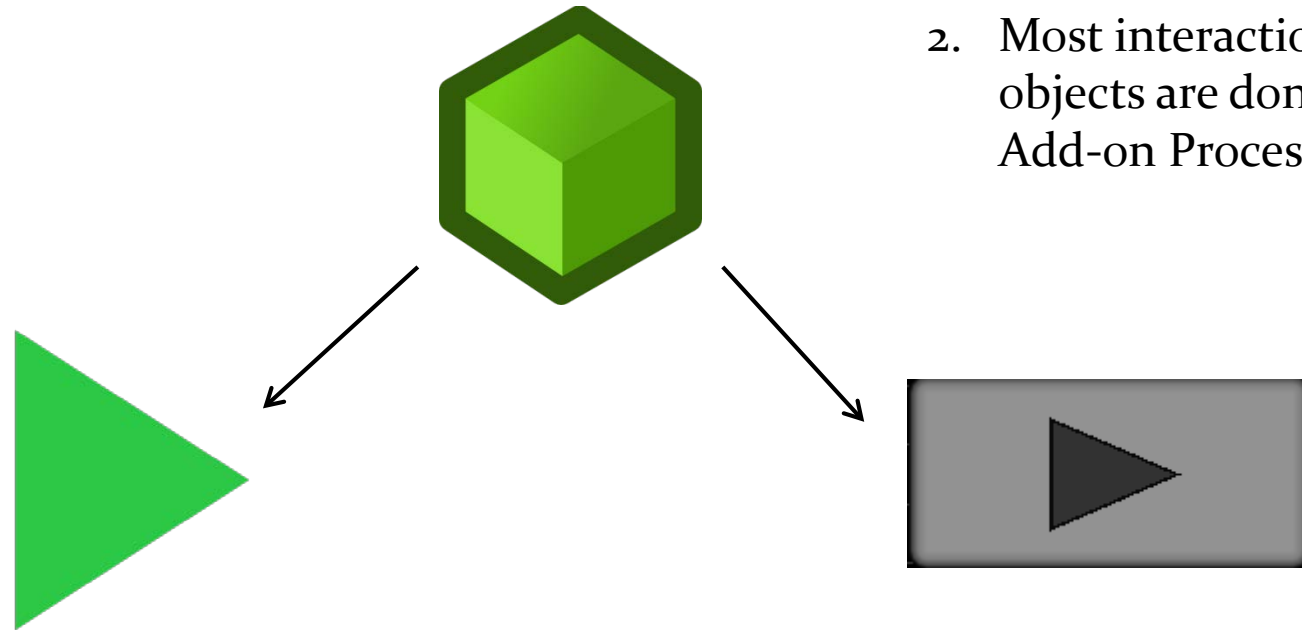
# Disclaimer

- I currently work for FedEx Ground, however all opinions, concepts, etc. are mine and do not reflect those of my employer.
- Any models/concepts discussed today are not reflective of my work at FedEx Ground.

# Typical Models In Simio

1. What are they?
2. What can they do?

# The Typical Simio Model



## Model

1. Most Variables and Process Logic Sits here
2. Most interaction between entities and objects are done through Model Level Add-on Processes

## Entity

1. Cannot Directly update Model Level or Object level Variable
2. Reads information directly from the top level model
3. Is acted upon directly by the top level model

## Object

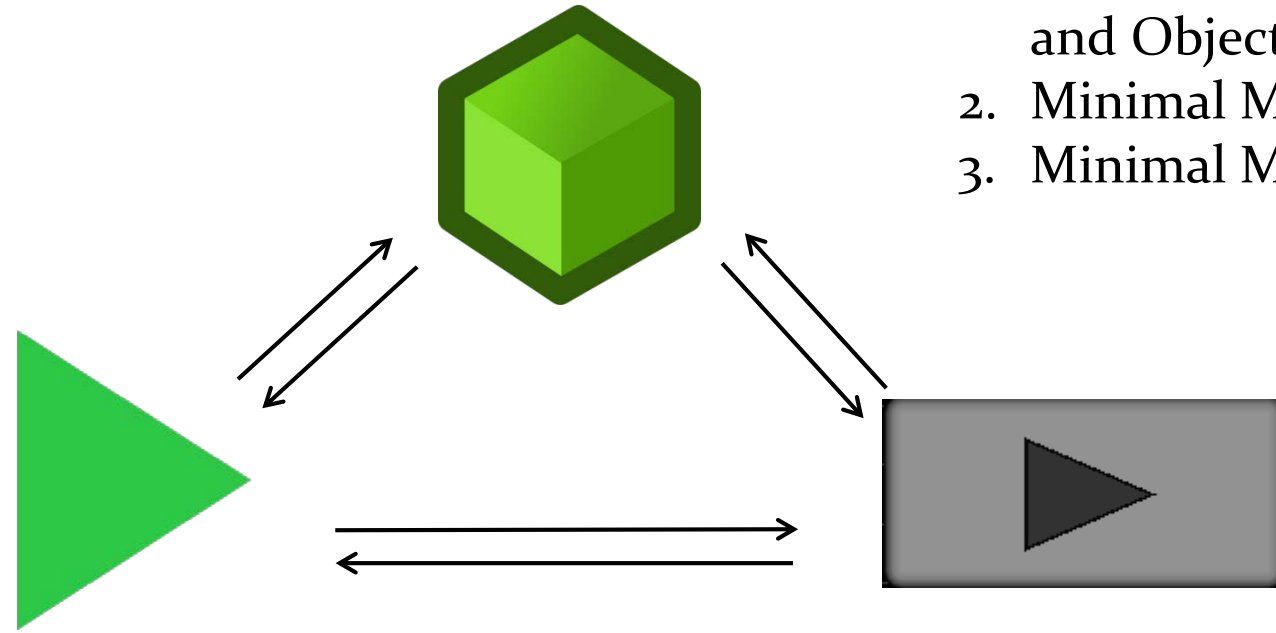
1. Cannot Directly update Model Level or Entity level Variable
2. Reads information directly from the top level model
3. Is acted upon directly by the top level model

# Flexible Models In Simio

1. What are they?
2. What can they do?



# A Flexible Simio Model



## Model

1. Information is passed to Entities and Objects when needed
2. Minimal Model Level Processes
3. Minimal Model Level Variables

## Entity

1. Interact/"Talk" directly to and upon Entities, The Model and Objects
2. Entity will "read" Itself and other objects to determine best course of action
3. Large portion of process will be here

## Object

1. Interact/"Talk" directly to and upon Entities, The Model and Objects
2. Large portion of the processes are here
3. **Easily Scalable**

# “Smart” Entities

- “Smart” Entities and Objects are the key to Flexible model design
- Smart Entities can have the ability to:
  - Create/Destroy entities
  - Transfer entities to and from itself
  - Make decisions regarding its own movement
  - Manipulate object and model level data/variables
  - Execute or Fire processes in other Entities or Objects
  - Perform all the duties of any other object or entity

# “Smart” Objects

- Smart Objects can have the ability to:
  - Create/Destroy entities
  - Transfer entities to and from itself
  - Take control of inbound and outbound links (if node)
  - “Know” what outbound/inbound links are attached to it
  - “Know” what nodes or objects are attached to those links
  - Route Entities based on the state of the model around it
  - Handle traffic flow around itself
  - Manipulate object and model level data
  - Perform all the duties of any other object or entity

# Example Model: An Autonomous Rail System

## Components-

1. The system has multiple entry points that feed a Central Sorting Point
2. The system has multiple exit points
3. Trains are made up of individual cars
4. Locomotives are autonomous.
  1. They can pick and drop trains off as needed
  2. Locomotives are free to decide which Trains to pick up and when.

# Overview of Hypothetical Input Variables

- Total People Coming Into the System
- Rate at which people fill trains
- Number of Locomotives
- Number of Trains
- Time Indexed tables to control flow of passengers into the model and their destinations





Pros/Cons



# How Flexible Modeling Helps Development

- Allows for modular construction
- Once logic is complete, adding new induction or stops is fast and easy
- Nothing additional needs to be done to add transports
- “Code Once/Handle Many” technique reduces development time
- Helps reduce assignment errors
- Allows for more versatile front ends for end users
- Easier debugging
- Reduces “Call Clutter” on the model level
- Objects are free to make changes at any time instead of having to be acted on by another process or event

# How Flexible Modeling Hurts Development

- In-depth Simio knowledge required.
  - To the point of needing to know which branches of steps Simio handles first
- Little Documentation
  - You will be “on an island” if you decided to code this way
- Certain Simio features do not work with this type of programming
  - Some Transfers do not work
  - Cannot read tables in other objects
  - Some work arounds are available in Enterprise edition

# Final Thoughts

- Flexible models are highly useful for:
  - Very large models
  - Transportation models
    - Trains
    - Robots in Facilities
    - Logistics
  - Facility models with many moving pieces
- Flexible modeling is a great tool, but not all jobs require a screw driver

Questions?