New trends, technologies and tools in Modeling and Simulation

Dr. David A. Cook
Department of Computer Science
Stephen F. Austin State University
Nacogdoches, TX 75962

cookda@sfasu.edu

What is Simulation?

- *Simulation* – very broad term – methods and applications to imitate or mimic real systems, usually via computer
- Applies in many fields, industries
- Very popular, powerful way to save time and money.
- When used effectively, speeds up “model to reality” by allowing visualization and validity testing of the model

Where is simulation used?

- Manufacturing facility
- Banks
- Airport operations (passengers, security, crews, baggage)
- Transportation/logistics/distribution operation
- Hospital facilities (ERs, operating room, admissions)
- Computer networks
- Freeways
- Medical and Surgical Training
- Fast-food restaurants, supermarkets
- Theme park
- Emergency-response system
- Shipping ports, berths
- Military combat, logistics

Why use simulation?

- Study system – measure, improve, design, control
  - Maybe just investigate changes to actual system
    - Advantage — unquestionably looking at the right thing
  - But often impossible in reality with actual system
    - System doesn’t exist
    - Would be disruptive, expensive, dangerous
    - Would result in loss of lives

Using Computers to Simulate

- General-purpose languages (C, C++, C#, Java, Matlab, FORTRAN, others)
  - Tedious, low-level, error-prone
  - But, almost complete flexibility
- Support packages for general-purpose languages
  - Subroutines for list processing, bookkeeping, time advance
  - Widely distributed, widely modified
- Spreadsheets
  - Usually static models (only *very* simple dynamic models)
  - Financial scenarios, distribution sampling, SQC
  - Examples in Chapter 2 (one static, one dynamic)
  - Add-ins are available (@RISK, Crystal Ball)

Using Computers to Simulate (cont'd.)

- Simulation languages
  - GPSS, SLX,
  - Popular, some still in use
  - Learning curve for features, effective use, syntax
- High-level simulators
  - Very easy, graphical interface
  - Domain-restricted (manufacturing, communications)
  - Limited flexibility — need to make sure model is valid

When Simulations are Used

- Use of simulation has evolved with hardware, software
- Early years (1950s – 1960s)
  - Very expensive, specialized tool
  - Required big computers, special training
  - Mostly in FORTRAN (or even Assembler)
  - Processing cost as high as $1000/hour for a sub-PC level machine

When Simulations are Used (cont'd.)

- Formative years (1970s – early 1980s)
  - Computers got faster, cheaper
  - Value of simulation more widely recognized
  - Simulation software improved, but still languages to be learned, typed, batch processed
  - Often used to clean up "disasters" in auto, aerospace industries
    - Car plant; heavy demand for certain model
    - Line underperforming
    - Simulated, problem identified
    - But demand had dried up — simulation was too late

When Simulations are Used (cont'd.)

- Recent past (late 1980s – mid 2000s)
  - Microcomputer power
  - Software expanded into GUIs, animation
  - Wider acceptance across more areas
    - Traditional manufacturing applications
    - Services
    - Health care
    - "Business processes"
  - Still mostly in large firms
  - Simulation is often part of "specs"

The Search:: 1955 - 60

- 1955 – 1960: Fortran was King, General Simulation Program was envisioned. Fortran based (with reusable functions)
- 1960s: GPSS (IBM, queueing models). Also SIMSCRIPT and SIMSCRIPT II (Rand Corp. and USAF). GASP (Algol and Fortran), and SIMULA (mostly Europe.
- 1970s – GPSS/H, GASP IV, SIMULA
  - Attempt to simplify the modeling process
  - Program generators – severe limitations

Next Leap Forward – the 1980s

- Movement to mini and PC computers
- SLAM II (descendant of GASP)
  - 3 world views
    - Event, Network, Continuous
- SIMAN (descendant of GASP)
  - General Modeling + Block Diagrams
  - 1^{st} first major language - PC & MS-DOS
  - Fortran functions w/ Fortran programming

1980s – Present Integrated Environments

- Growth on PC's
- Simulation Environments
  - GUI
  - Animation
  - Data analyzers

The Future (NOW)

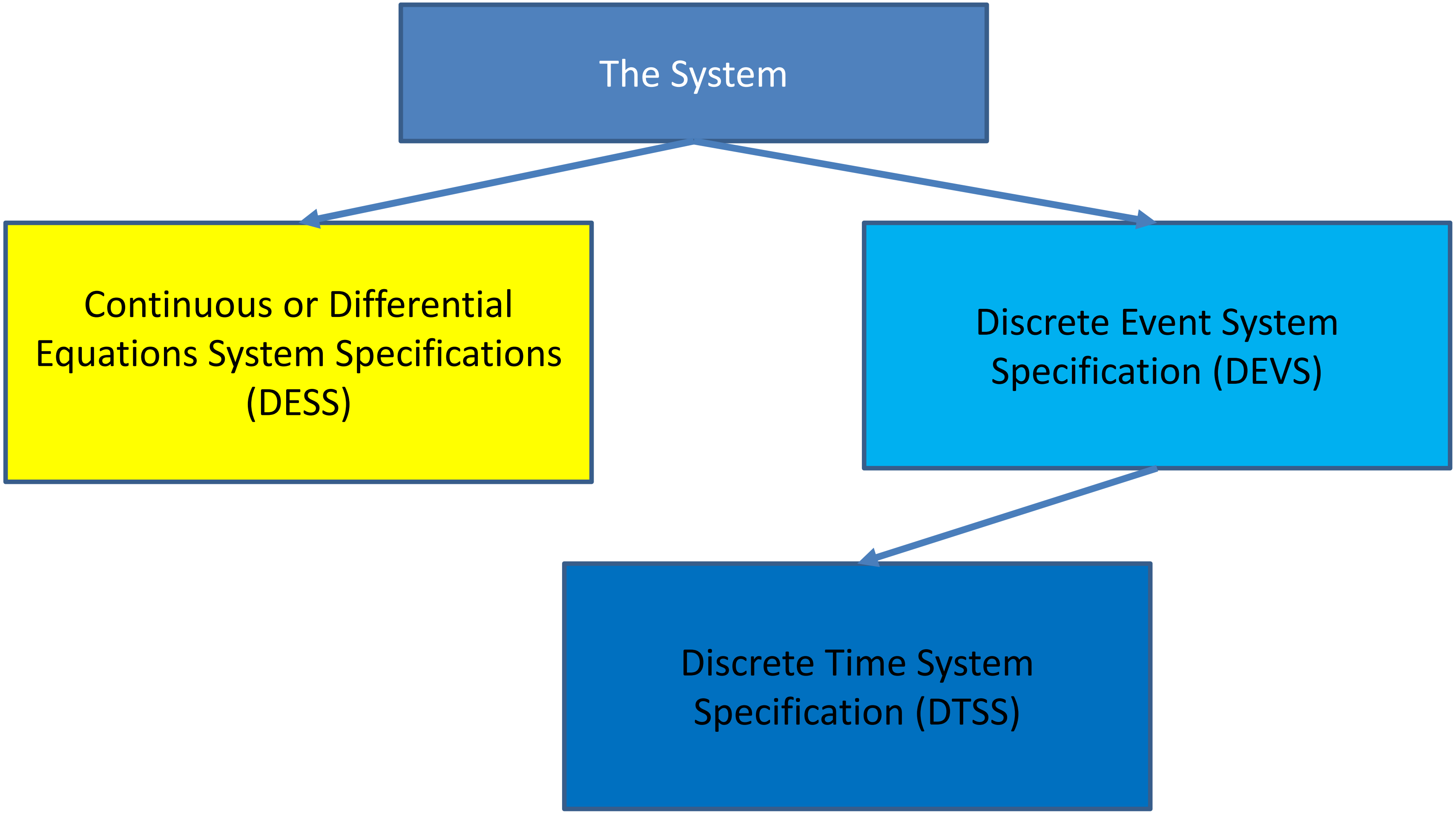
- Virtual Reality
- Improved Interfaces
- Better Animation
- Agent-based Modeling

How do you know the model is correct?

- Simulation Validity
  - Structural Simulation
  - Behavioral Simulation
  - Predictive Simulation

- Simulation Verification
  - How the simulation is built
  - What estimates are used
  - What are the unknowns
  - How accurate and of what fidelity are the results

Types of Simulation



Dynamic Simulations

- Can be
  - Continuous – changes constantly over time. For any exact time (i.e. t = 1 hour, 2 minutes, 13.9987665 seconds) there is a (potentially) exact value
  - Discrete – changes occur at specific and separated points in time. For example, a customer can arrive at a bank at 3:14:15, but not again until 3:14:16. It is "impossible" (i.e. a non-event) for a customer to arrive halfway between two time steps.

- Continuous and Dynamic simulations (mixed models) are possible

Discrete Event Simulations

- Discrete – changes occur at specific and separated points in time.

- For example, a customer can arrive at a bank at 3:14:15, but not again until 3:14:16. It is "impossible" (i.e. a non-event) for a customer to arrive halfway between two time steps.

Deterministic vs. Stochastic

- Deterministic – there is no element of randomness in the simulation.

- Stochastic – some part of the simulation is based on randomness. Randomness can be event-based (customer entrance time) or probability-based (the odds of an event, like a structural failure) occurring.

And the difference…

- DESS – specific time dq/dt = a*x + b for all t
  - Every time t has a specific value, not necessarily dependent upon any other time t
- DTSS – q (t + 1) = some function of q(t)
  - Every time has value (state) based on previous time
- DEVS – there is a time t_n of the next event.
  - The state at the next event is a function of all events that have preceded the event.

All good simulations based on a model

- A simulation must be designed to either
  - Model a real system. The system can them be used for comparisons and verification and validation
  - Model an imaginary system (that might or might not be build in the future). Verification and validation much harder.

How do you know the model is correct?

- Validity
  - Structural Simulation
  - Behavioral Simulation
  - Predictive Simulation

- Verification
  - How the simulation is built
  - What estimates are used
  - What are the unknowns
  - How accurate and of what fidelity are the results

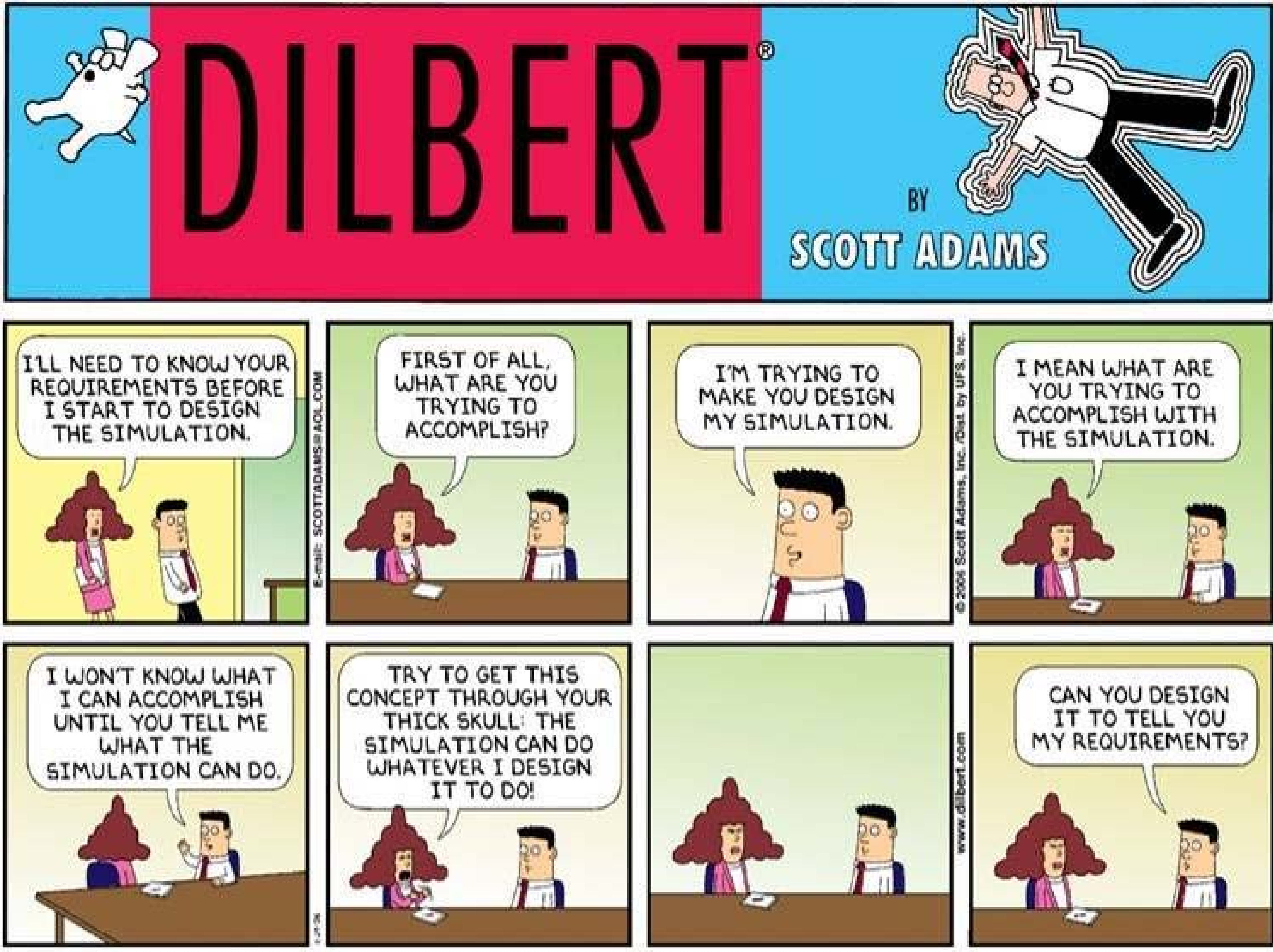
What’s new in Simulation today?

- Graphical simulation languages can be used to create the model, run the simulation, and explore the outputs.

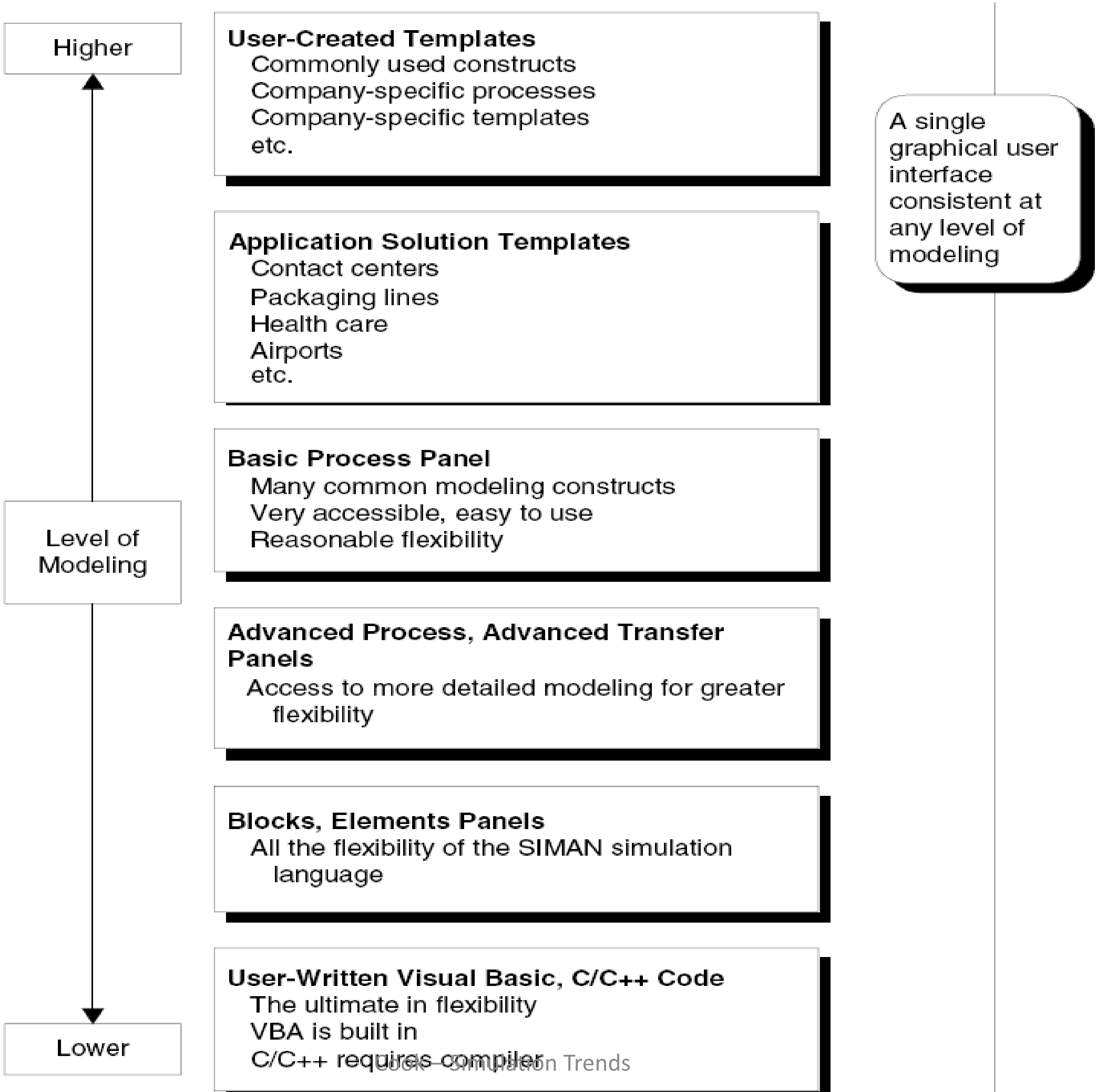
- GUIs of varying levels of detail and specificity can be used to build complex graphical models

How to build a M&S

- Hierarchical structure
  - Multiple levels of modeling
  - Mix different modeling levels together in same model
  - Often, start high then go lower as needed
- Get ease-of-use advantage of simulators without sacrificing modeling flexibility



DILBERT®
BY
SCOTT ADAMS
I'LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SIMULATION.
E-mail: SCOTTADAMS@AOL.COM
FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?
I'M TRYING TO MAKE YOU DESIGN MY SIMULATION.
© 2006 Scott Adams, Inc. /Dist. by UFS, Inc.
I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH WITH THE SIMULATION.
I WON'T KNOW WHAT I CAN ACCOMPLISH UNTIL YOU TELL ME WHAT THE SIMULATION CAN DO.
TRY TO GET THIS CONCEPT THROUGH YOUR THICK SKULL: THE SIMULATION CAN DO WHATEVER I DESIGN IT TO DO!
www.dilbert.com
CAN YOU DESIGN IT TO TELL YOU MY REQUIREMENTS?



Higher
Level of Modeling
Lower
User-Created Templates
Commonly used constructs
Company-specific processes
Company-specific templates
etc.
Application Solution Templates
Contact centers
Packaging lines
Health care
Airports
etc.
Basic Process Panel
Many common modeling constructs
Very accessible, easy to use
Reasonable flexibility
Advanced Process, Advanced Transfer Panels
Access to more detailed modeling for greater flexibility
Blocks, Elements Panels
All the flexibility of the SIMAN simulation language
User-Written Visual Basic, C/C++ Code
The ultimate in flexibility
VBA is built in
C/C++ requires compiler
A single graphical user interface consistent at any level of modeling

Visual and Graphical M&S Trends

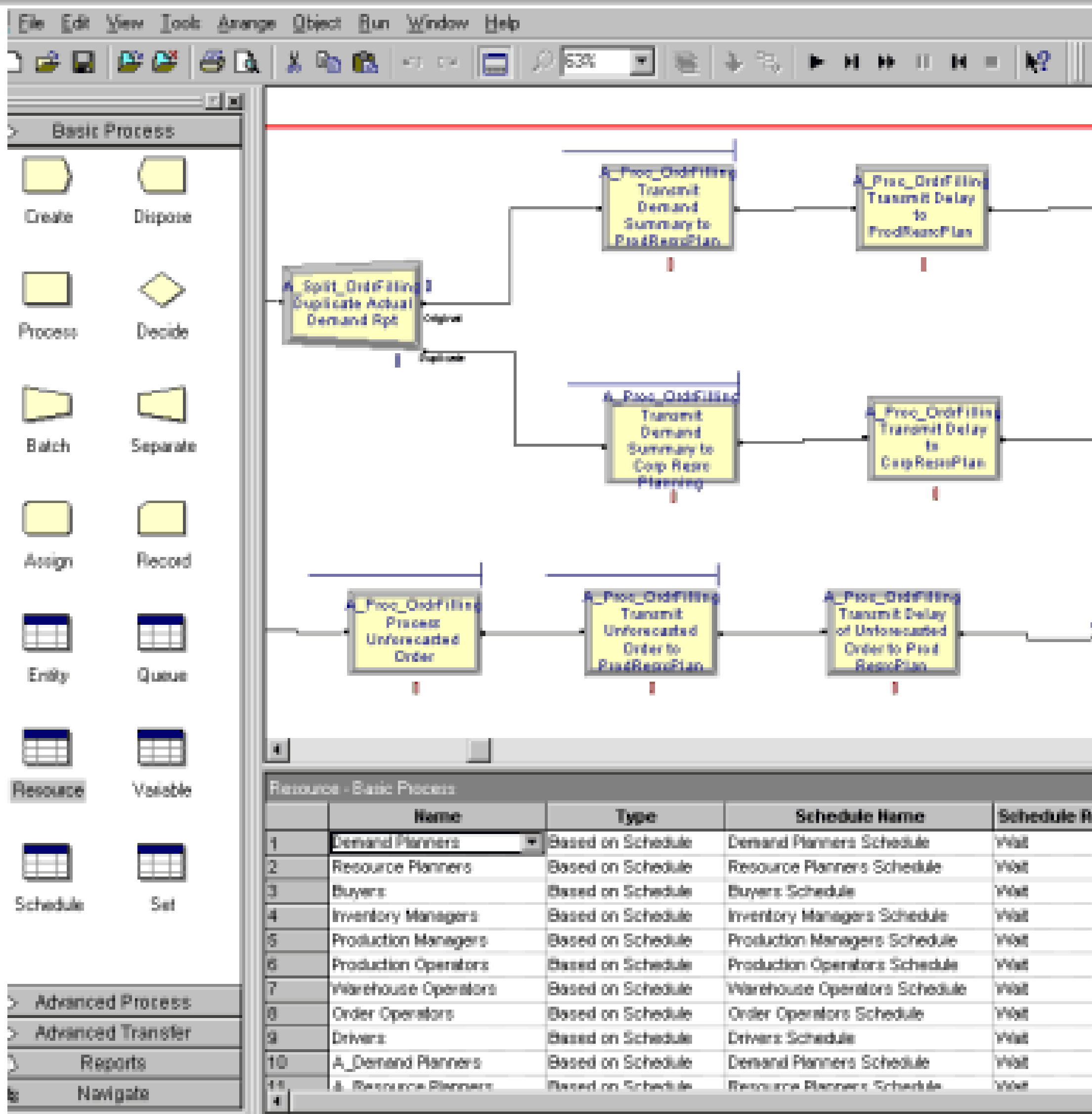
- **Highly realistic training scenarios**
- **Training realistic than games designed for entertainment**
- **Focus on end-user experience that matches real life (virtual reality)**

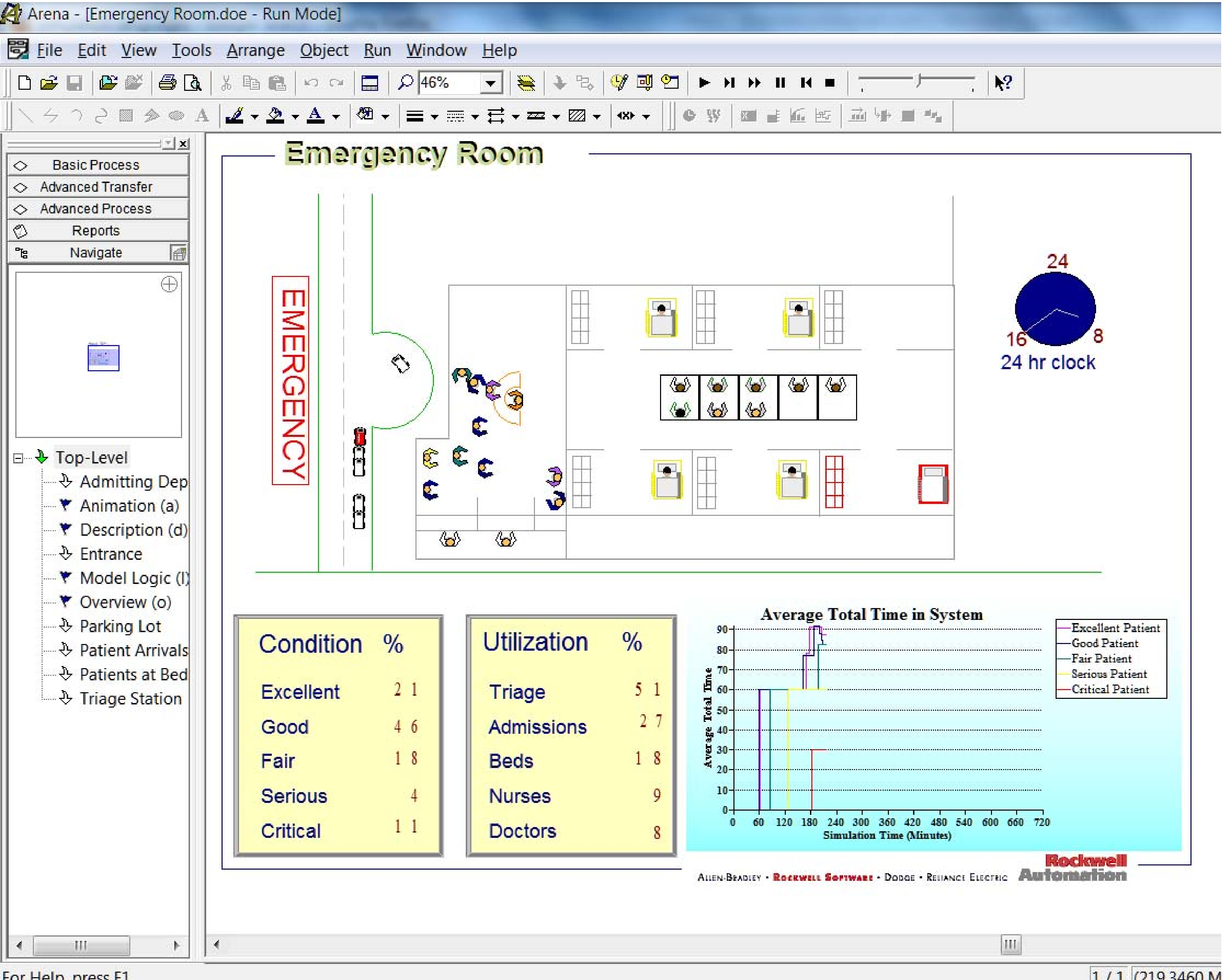
However = lots of $$s NOT needed!

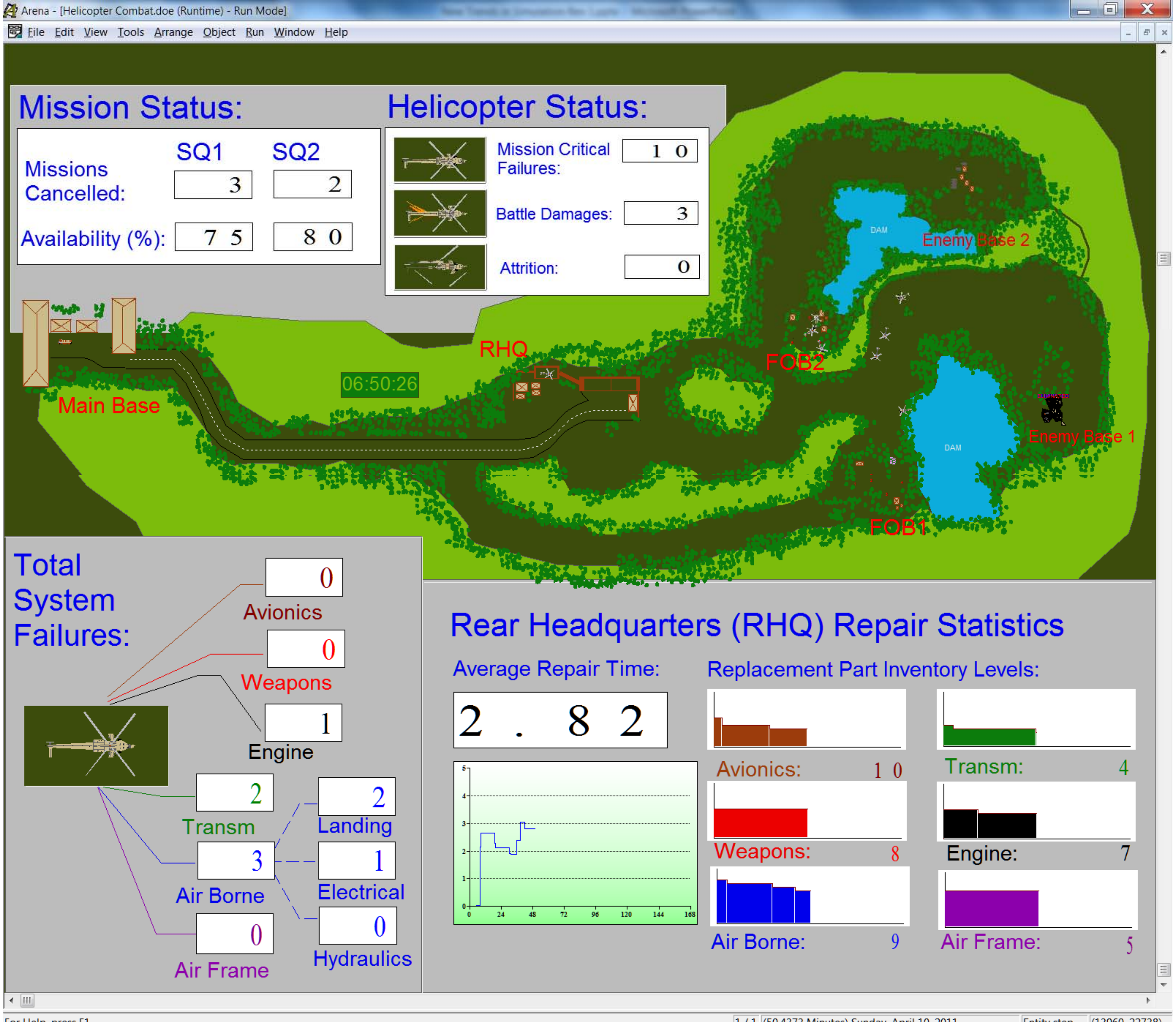
- Simple languages and available tools let you transform models and simulations into easy to visualize products that can be used for
  - Proof of concept
  - Ease of user understanding
  - Graphical display of formerly table-driven data

- This speeds up the “model to reality” timeline

One Example - Arena




Arena - [Emergency Room.doe - Run Mode]
File Edit View Tools Arrange Object Run Window Help
46%
Basic Process
Advanced Transfer
Advanced Process
Reports
Navigate
Top-Level
Admitting Dep
Animation (a)
Description (d)
Entrance
Model Logic (l)
Overview (o)
Parking Lot
Patient Arrivals
Patients at Bed
Triage Station
Emergency Room
EMERGENCY
24
16
8
24 hr clock
Condition %
Excellent 21
Good 46
Fair 18
Serious 4
Critical 11
Utilization %
Triage 51
Admissions 27
Beds 18
Nurses 9
Doctors 8
Average Total Time in System
Average Total Time
Simulation Time (Minutes)
Excellent Patient
Good Patient
Fair Patient
Serious Patient
Critical Patient
Allen-Bradley • Rockwell Software • Dodge • Reliance Electric
Rockwell Automation
For Help, press F1
1 / 1
(219.3460 M


Arena - [Helicopter Combat.doe (Runtime) - Run Mode]
File Edit View Tools Arrange Object Run Window Help
Mission Status:
SQ1 SQ2
Missions Cancelled: 3 2
Availability (%): 75 80
Helicopter Status:
Mission Critical Failures: 10
Battle Damages: 3
Attrition: 0
Main Base
RHQ
FOB2
FOB1
Enemy Base 2
Enemy Base 1
DAM
DAM
06:50:26
Total System Failures:
Avionics 0
Weapons 0
Engine 1
Transm 2
Landing 2
Air Borne 3
Electrical 1
Air Frame 0
Hydraulics 0
Rear Headquarters (RHQ) Repair Statistics
Average Repair Time:
2.82
0 24 48 72 96 120 144 168
Replacement Part Inventory Levels:
Avionics: 10
Transm: 4
Weapons: 8
Engine: 7
Air Borne: 9
Air Frame: 5
For Help, press F1
1 / 1 (50.4373 Minutes) Sunday, April 10, 2011
Entity step
(13960, 22738)

Questions or comments??

Dr. David A. Cook

Department of Computer Science

Stephen F. Austin State University

Nacogdoches, TX 75962

cookda@sfasu.edu