THE UNIVERSITY OF ARIZONA® | Arizona's First University.

Using Integrative Modeling for Advanced Heterogeneous System Simulation

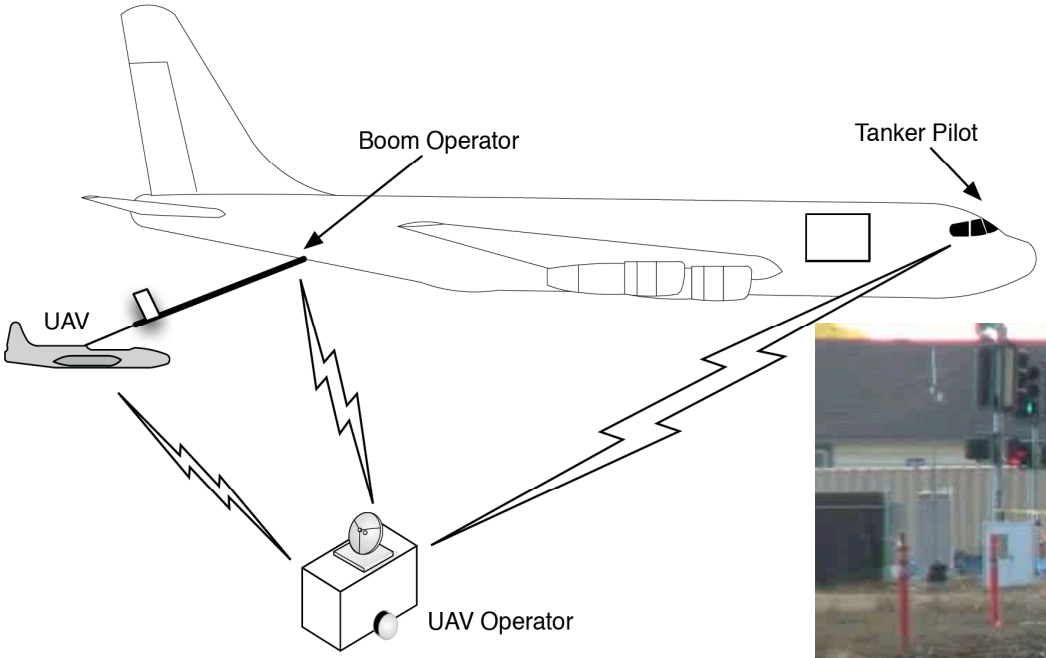
Tapasya Patki, Hussain Al-Helal, Jacob Gulotta, Jason Hansen, Jonathan Sprinkle

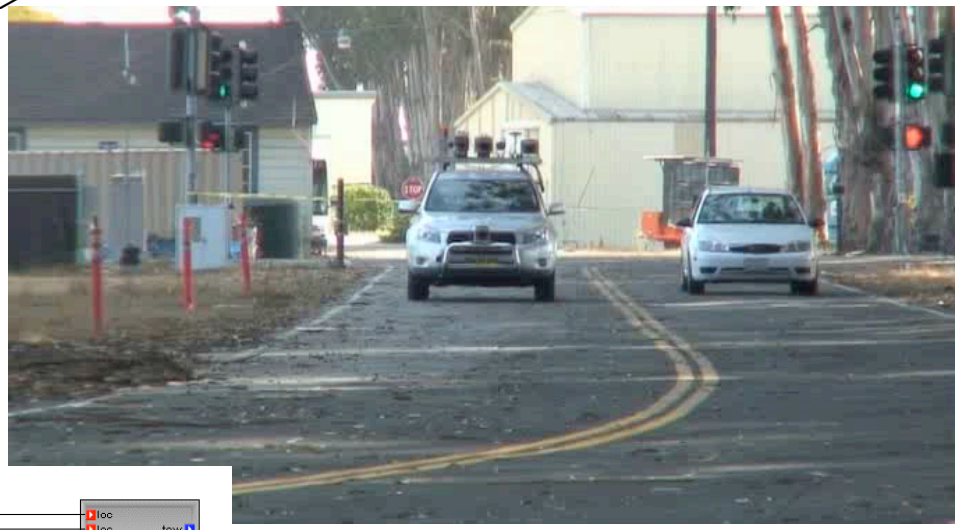
Thm (Kelly): If all you have is a hammer, look out for your thumb, and all that.

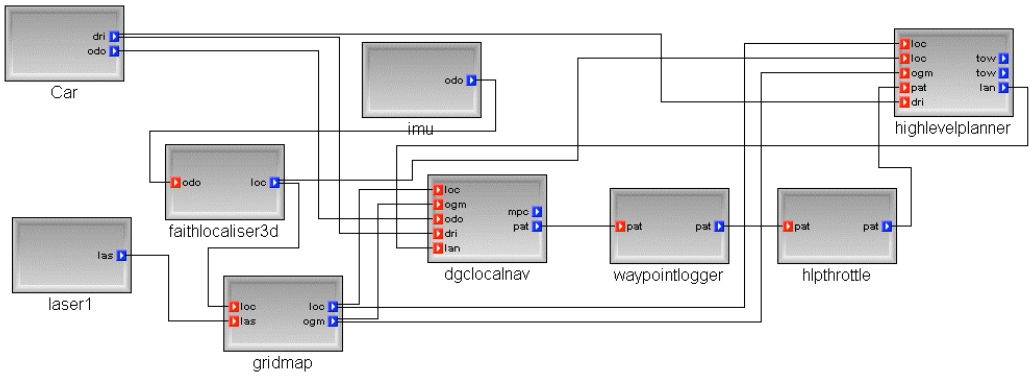
Corollary: If everyone has a hammer, then you will not use anyone else's nails.

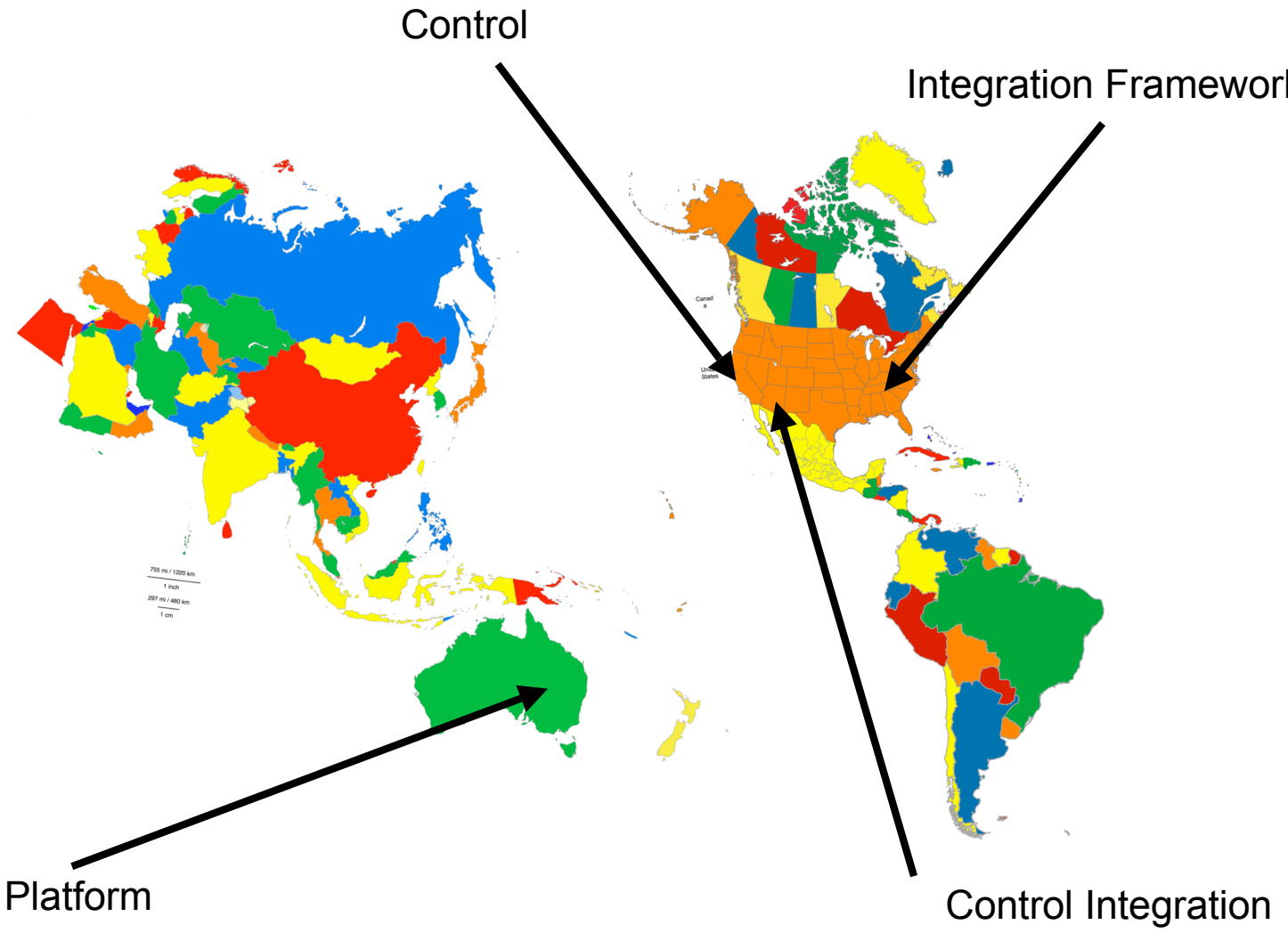
In collaboration with Vanderbilt University, and the University of California, Berkeley.
Supported by US AFOSR.

What kinds of heterogenous systems are we interested in?

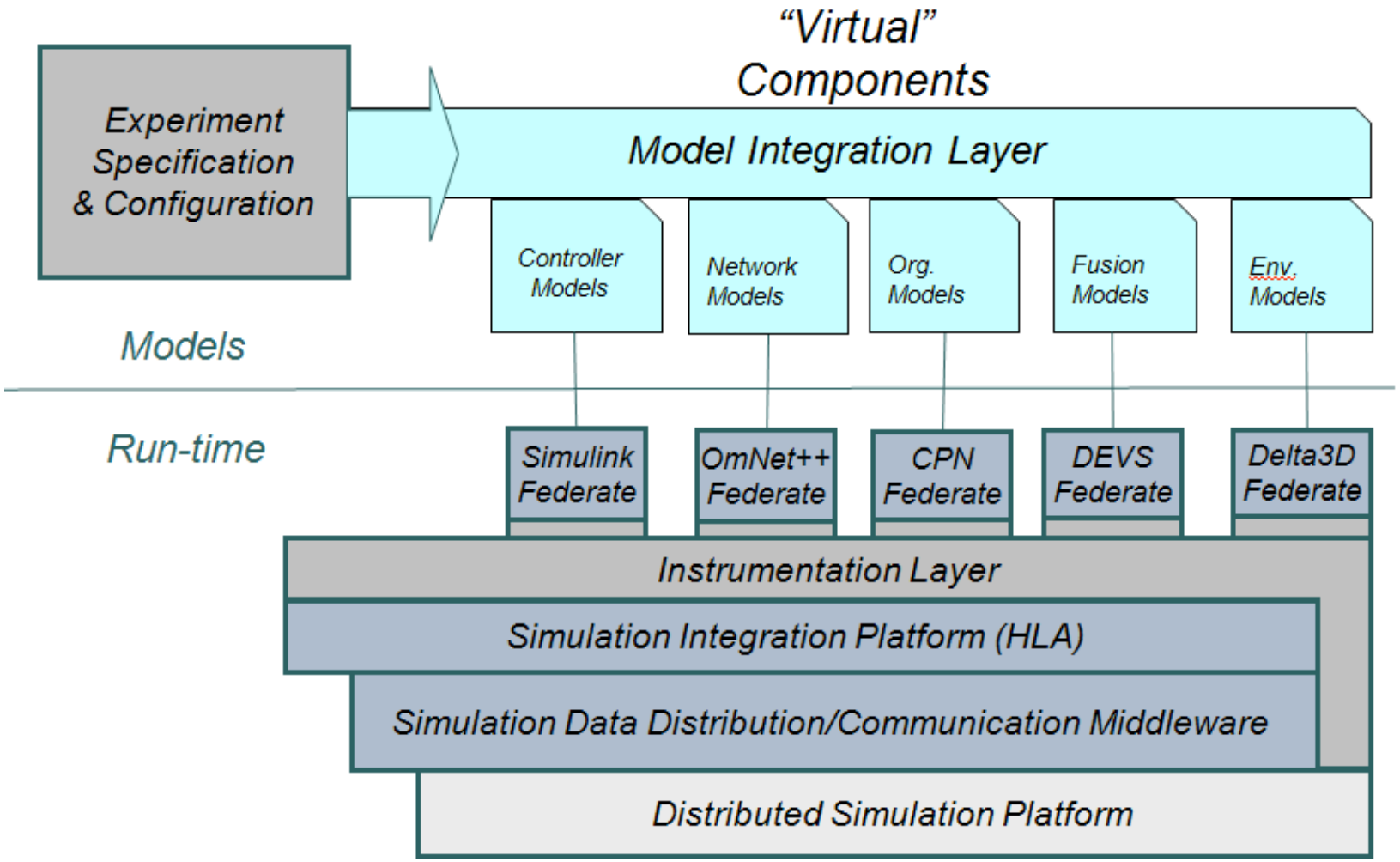






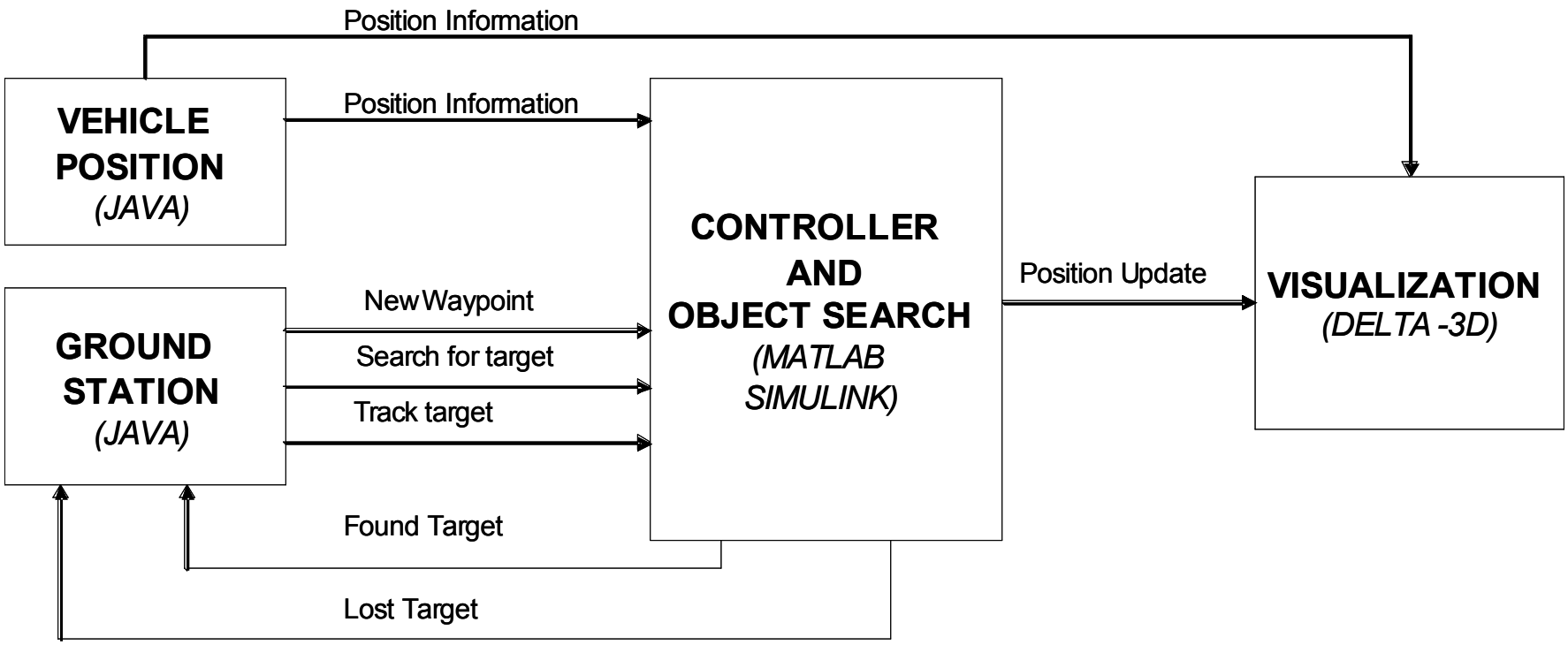

Control
Integration Framework
Platform
Control Integration

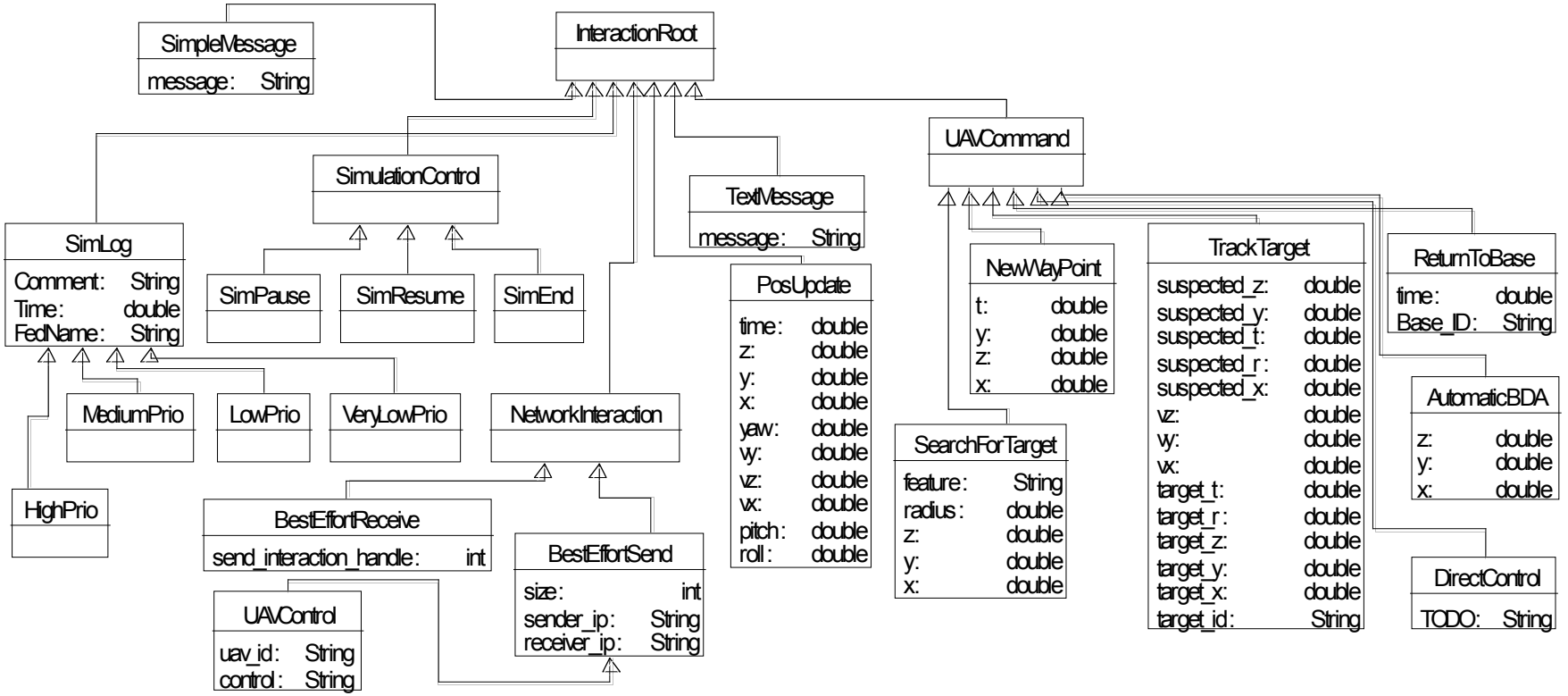
- Command and Control
  - Lots of decision makers, looking at tactical data, making command decisions.
  - Lots of monitors
  - Lots of data
  - Lots of decisions to make
- Types?
  - Tactical actors (manned/unmanned components)
  - GUI elements (human interfaces)
  - Vignettes (tactical tests)
- Example vignette:
  - A UAV is sent to a location to look for blue trucks. After a blue truck is spotted, the UAV reports its location. The C2 staff tell that UAV to “track” the blue truck. The UAV then stays as close as it can.
- Example problems:
  - What kind of UAV is it? What kinematic/dynamic properties does it have?
  - What connection/network settings do I use? Am I communicating via TCP/IP, or something more primitive?

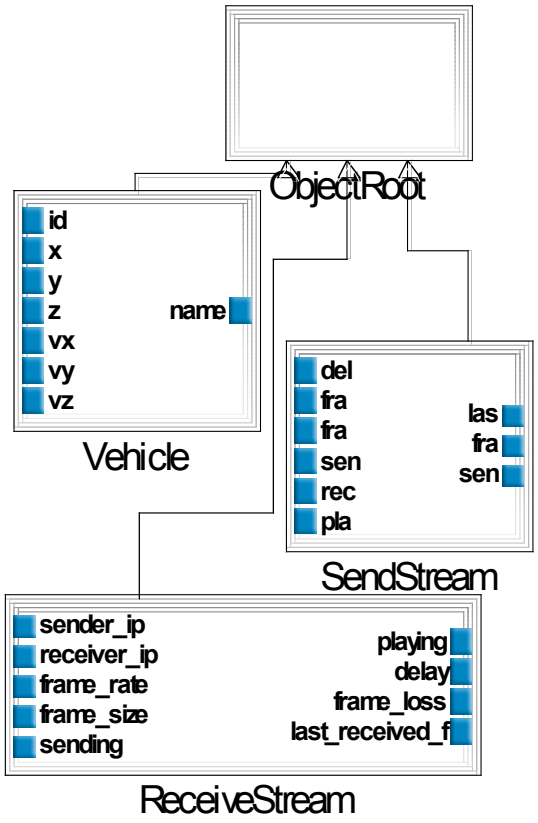


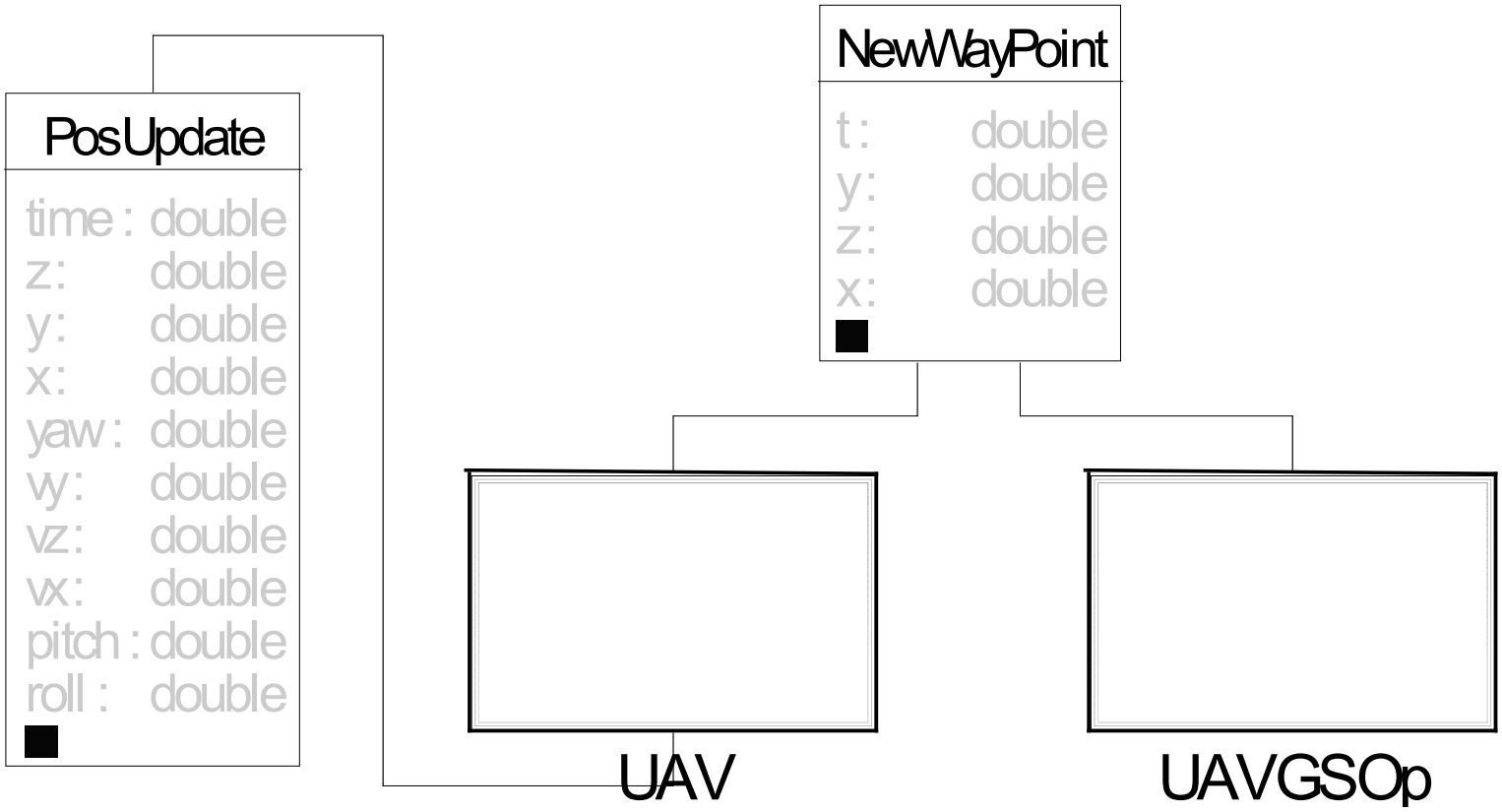
* From an unpublished manuscript by Balogh, et al.

The integrative modeling part…



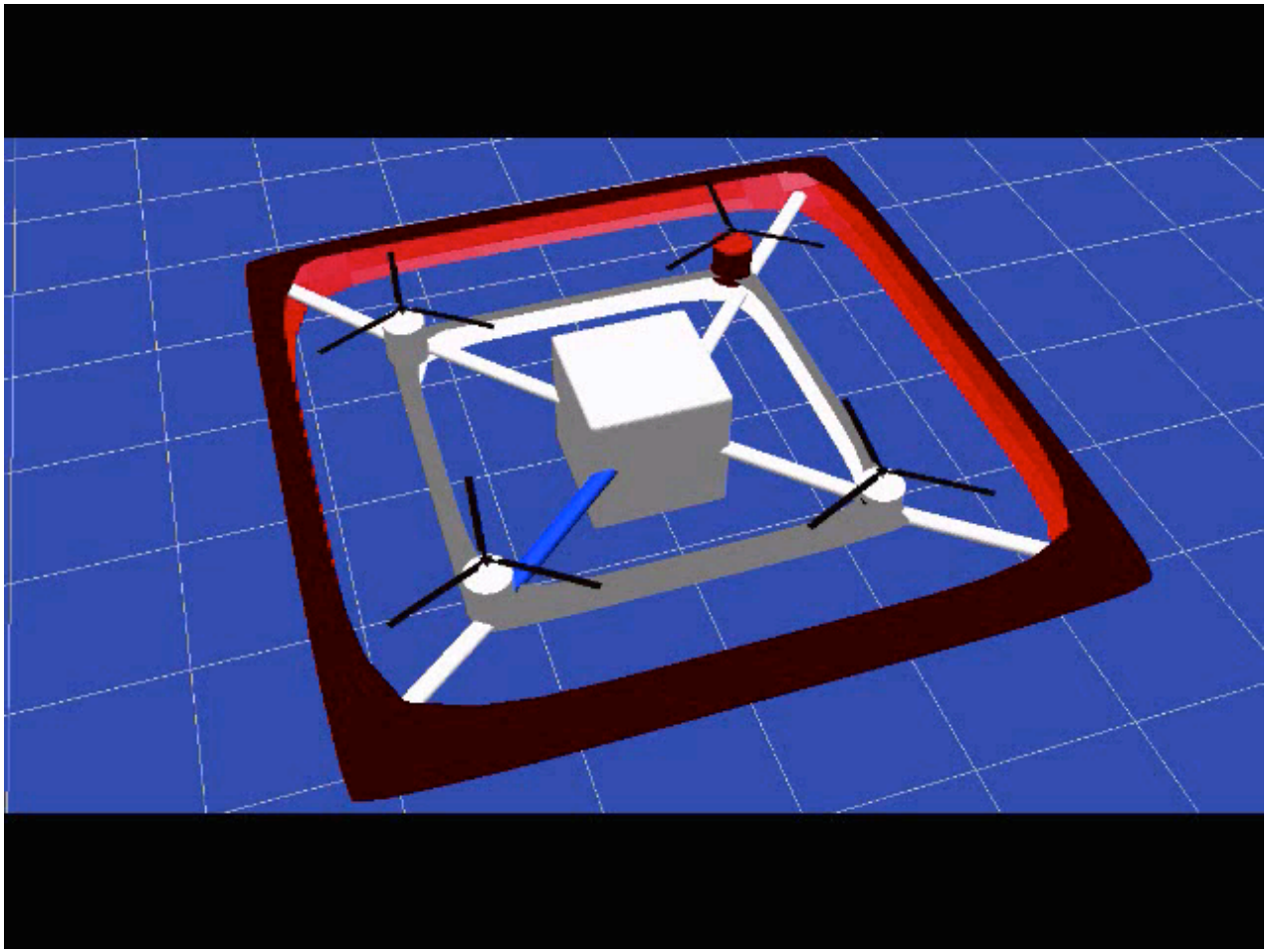

SimpleMessage
message: String
InteractionRoot
UAVCommand
SimulationControl
TextMessage
message: String
SimLog
Comment: String
Time: double
FedName: String
SimPause
SimResume
SimEnd
PosUpdate
time: double
z: double
y: double
x: double
yaw: double
vy: double
vz: double
vx: double
pitch: double
roll: double
NewWayPoint
t: double
y: double
z: double
x: double
TrackTarget
suspected_z: double
suspected_y: double
suspected_t: double
suspected_r: double
suspected_x: double
vz: double
vy: double
vx: double
target_t: double
target_r: double
target_z: double
target_y: double
target_x: double
target_id: String
ReturnToBase
time: double
Base_ID: String
AutomaticBDA
z: double
y: double
x: double
MediumPrio
LowPrio
VeryLowPrio
NetworkInteraction
SearchForTarget
feature: String
radius: double
z: double
y: double
x: double
HighPrio
BestEffortReceive
send_interaction_handle: int
BestEffortSend
size: int
sender_ip: String
receiver_ip: String
DirectControl
TODO: String
UAVControl
uav_id: String
control: String

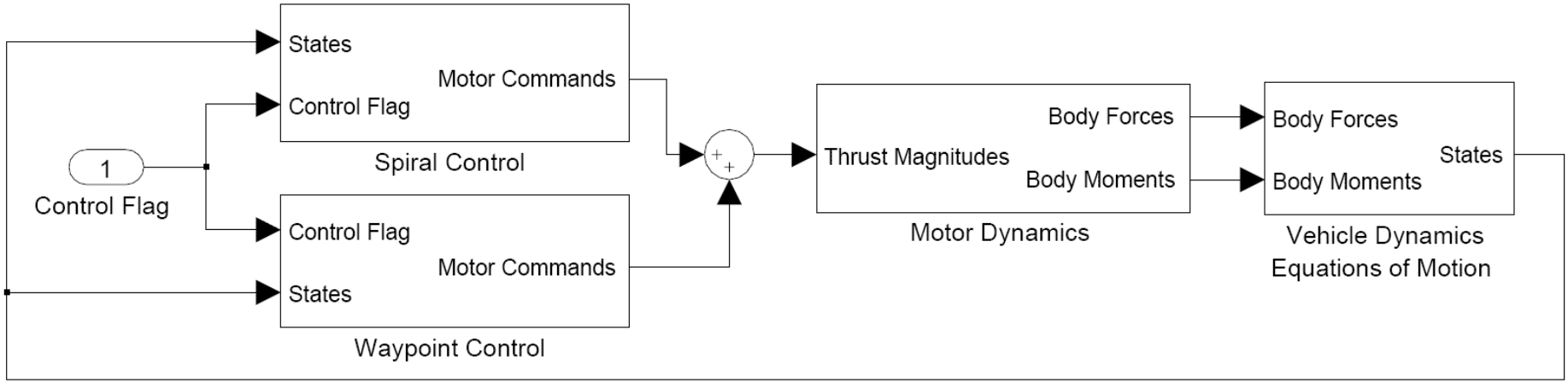

ObjectRoot
id
x
y
z
vx
vy
vz
name
Vehicle
del
fra
fra
sen
rec
pla
las
fra
sen
SendStream
sender_ip
receiver_ip
frame_rate
frame_size
sending
playing
delay
frame_loss
last_received_f
ReceiveStream

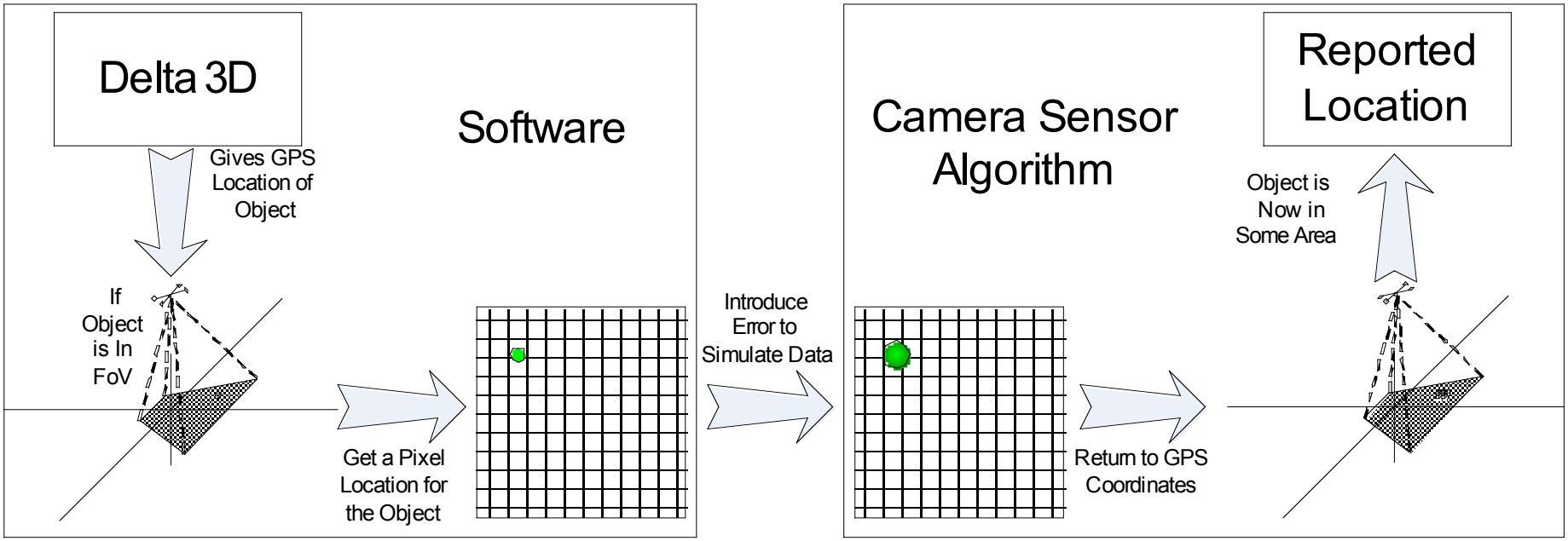

PosUpdate
time : double
z: double
y: double
x: double
yaw : double
wy: double
vz: double
vx: double
pitch : double
roll : double
NewWayPoint
t: double
y: double
z: double
x: double
UAV
UAVGSOp

A “default” rotorcraft

- A realistic model of the STARMAC was created using Blender
- Rendered model used to represent UAV during simulation
- Multiple instances of the rendered model can be used to simulate swarms of rotorcraft

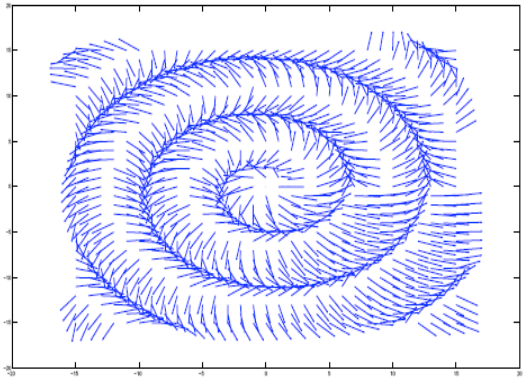



States
Motor Commands
Control Flag
Spiral Control
1
Control Flag
Control Flag
Motor Commands
States
Waypoint Control
Thrust Magnitudes
Body Forces
Body Moments
Motor Dynamics
Body Forces
States
Body Moments
Vehicle Dynamics
Equations of Motion

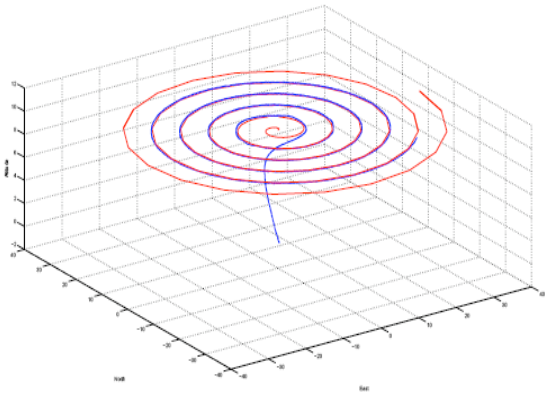

Delta 3D
Software
Gives GPS Location of Object
If Object is In FoV
Get a Pixel Location for the Object
Introduce Error to Simulate Data
Camera Sensor Algorithm
Return to GPS Coordinates
Reported Location
Object is Now in Some Area

- The GME Paradigm: what gets generated?
- Camera models: where are things executed?
- Vehicle dynamics: where does the logic live?

- In the "domain" of heterogeneous simulation
  - Every player has their own domain
  - That domain has the correct tools, etc., for doing development
- There are lots of hard pieces
  - Hard for domain-experts to understand middleware programming
  - Hard for middleware programmers to understand domain concepts
  - Hard for anyone to install everyone else's tools…
- But, major benefits, if pieces can be integrated easily
  - Allows *immediate* work on domain-problems, deferring integration work until later
  - Permits domain-specific work to use the tools of the domain
  - Showcases the power of code generation (when used by Jedi appropriately)



(a) Vector field showing spiral path following.



- We were able to stand up a significant demo within 3 months of (beginning) to install the software
- Our work concentrated on developing domain-specific pieces to improve visualization and design-time analysis
- The modeling infrastructure supported our development in the appropriate level of abstraction for future integration

- Future Work
  - More advanced control algorithms (mesh travel/stability)
  - More advanced code generation (autogenerate vignette scripts, etc.)

THE UNIVERSITY OF ARIZONA

We’re always looking for good graduate students!

http://www.ece.arizona.edu/~sprinkjm/research/c2wt/