Engineered Complex Systems

2017 NETL Crosscutting Research Project Review

3/21/2017

Project Investigator:

Ames Laboratory

Kenneth “Mark” Bryden

kmbryden@iastate.edu

Ames, IA 50010

DOE Award Number DE-AC02-07CH11358

Period of Performance 10/1/2015 – 9/30/2019

AMES LABORATORY
Creating Materials & Energy Solutions
U.S. DEPARTMENT OF ENERGY

Objective:

Develop and demonstrate the framework and tools needed to create holistic complex systems models that enable policy, engineering, and operational decisions for advanced fossil energy systems.

FY2016 Milestones:

- Implement the schema required to link models together using peer-to-peer ontologies on a basic energy system, demonstrating extensible model sets and model substitution
- Define the domain specific language needed to support the development of detailed fossil energy systems models based on model federation concepts

Goals and Objectives

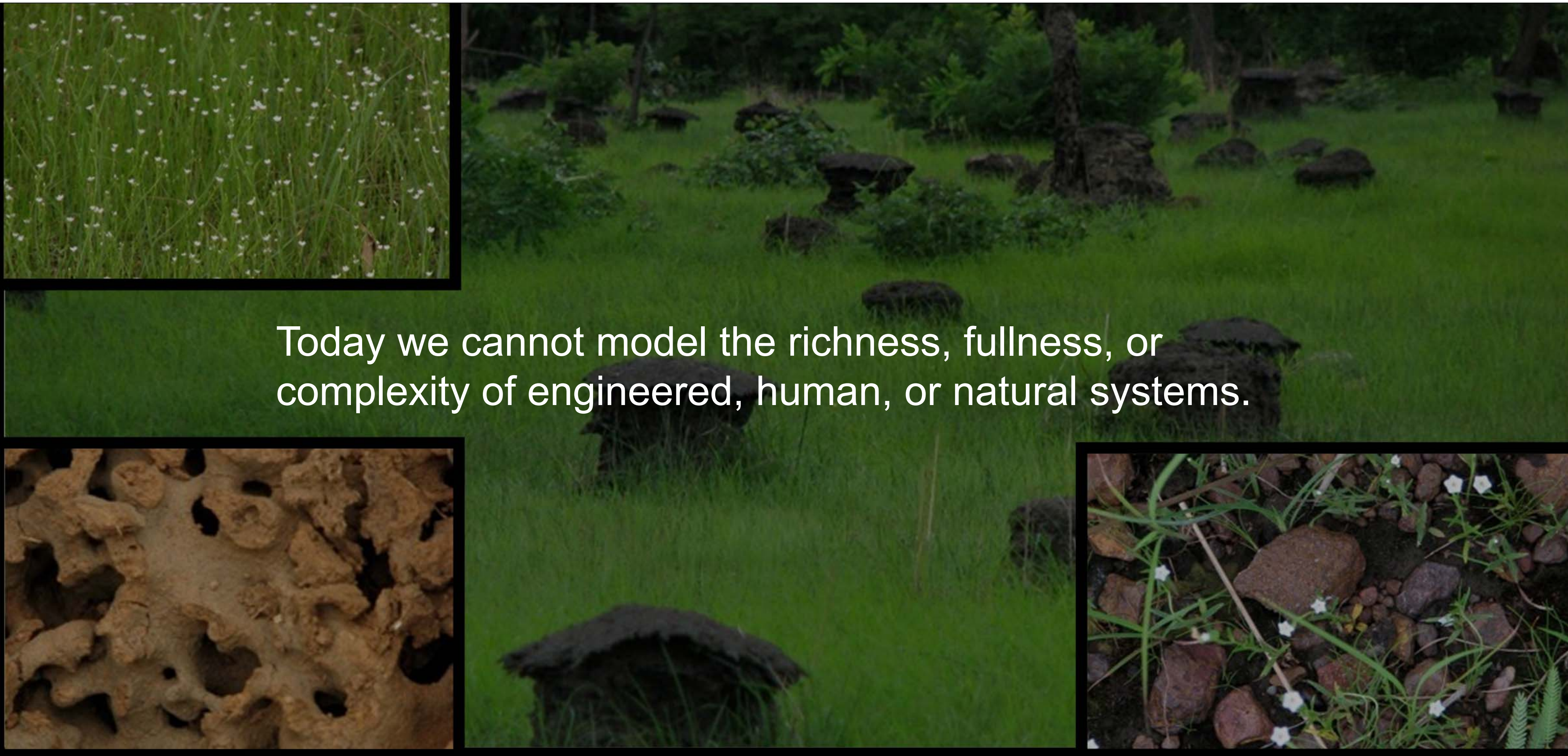
- Motivation
- Overview of federated modeling
- Federated model set proof of concept
- Federated modeling of Hyper system
- Domain specific language for federated modeling

Outline

Energy and environmental challenges

Interactions between
Engineered, human, and natural systems
are confounded by complexity

Holistic solutions are needed



New modeling approaches are needed

Many different models:

- No centralized, browsable storage
- Not readily accessible, citable, or maintained
- Hard to locate and use existing code

Codes do not work together:

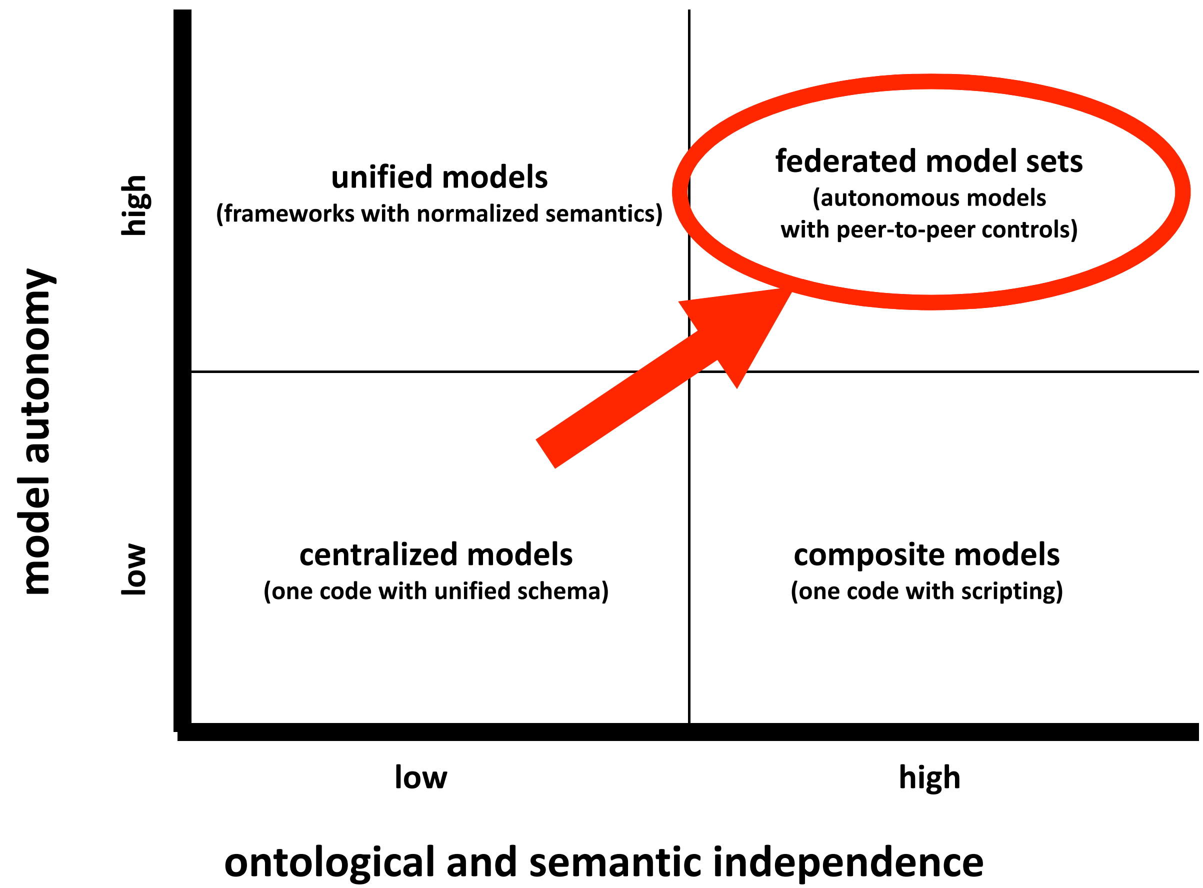
- Systems models use codes specifically built for them
- Hard to use existing codes in a new systems model
- Clunky

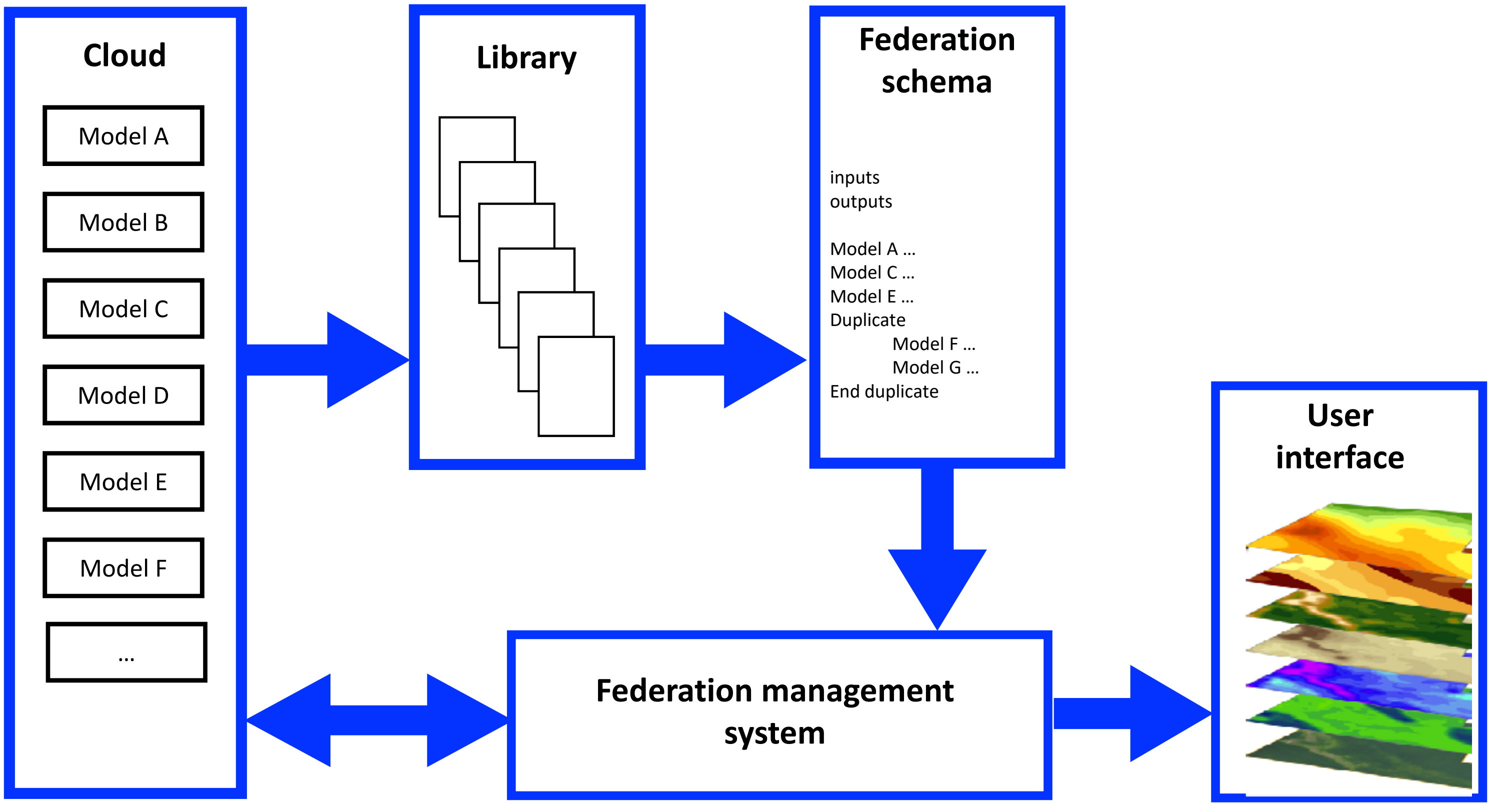
Systems modeling often lacks fidelity and granularity:

- Models are simplified, abstracted, averaged, order is reduced
- Interfaces may be lossy or imprecise

The advent of cloud computing has created a new computational platform:

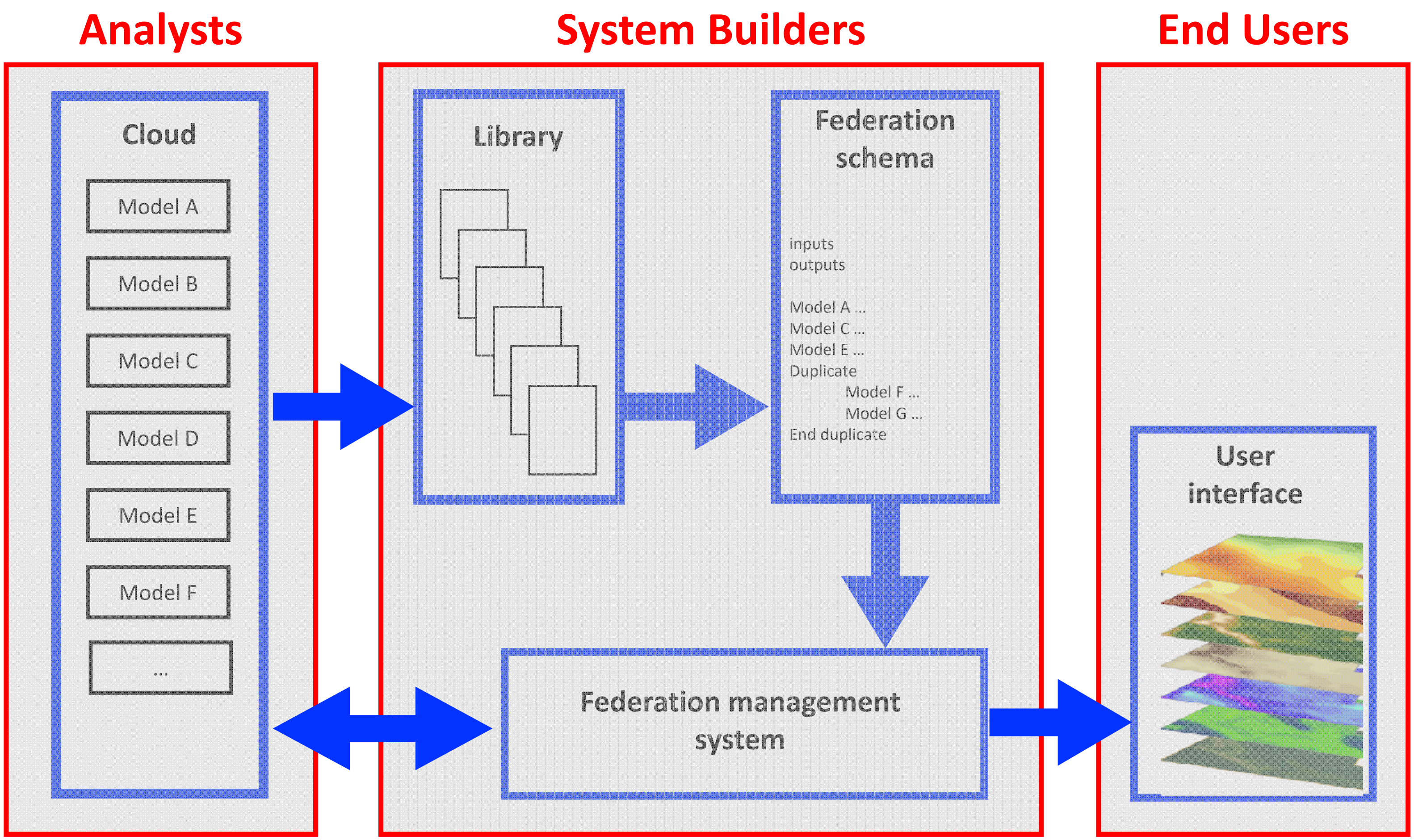
- On-demand computation and storage
- Immense scalability and flexibility
- Requires new architectural approach


model autonomy
high
low
unified models
(frameworks with normalized semantics)
federated model sets
(autonomous models
with peer-to-peer controls)
centralized models
(one code with unified schema)
composite models
(one code with scripting)
low
high
ontological and semantic independence



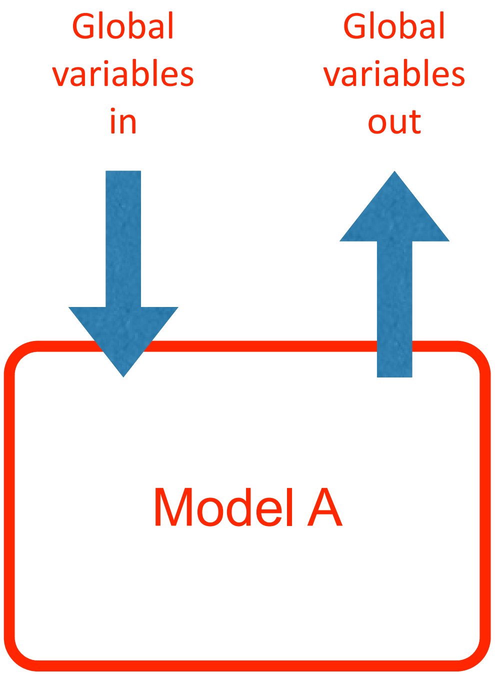
K. M. Bryden, Proceedings of the 7th International Congress on Environmental Modelling and Software (2014).

Approach



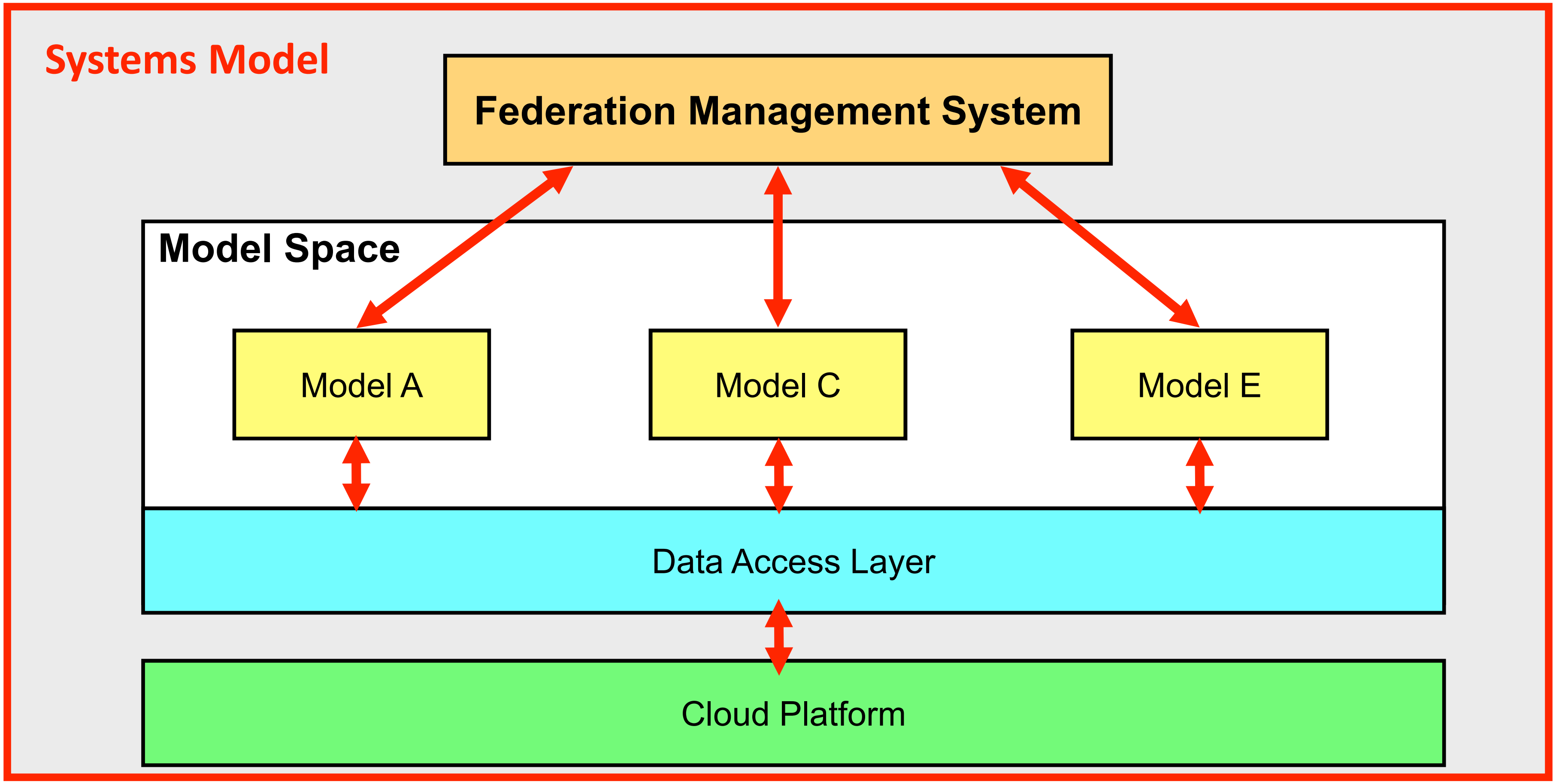
K. M. Bryden, Proceedings of the 7th International Congress on Environmental Modelling and Software (2014).

Workflow

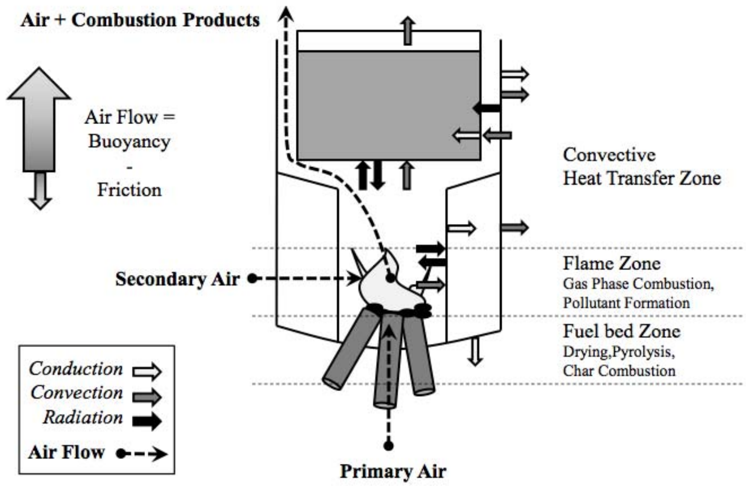


Each model is an independent microservice

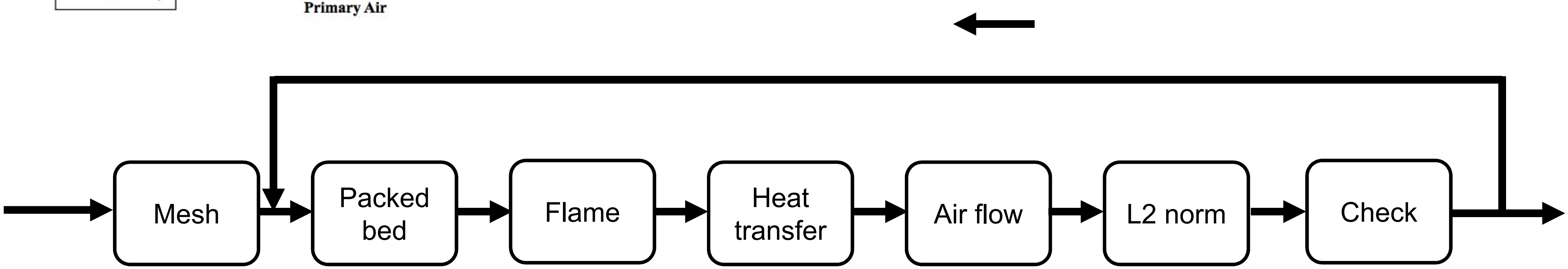
- Each microservice performs one specific task
- Microservices are stateless; no session information is retained
- Each work request must be complete and self-contained
- Models are reusable for other analysis
- Models can be strung together like beads on a complex weaving



Model space architecture



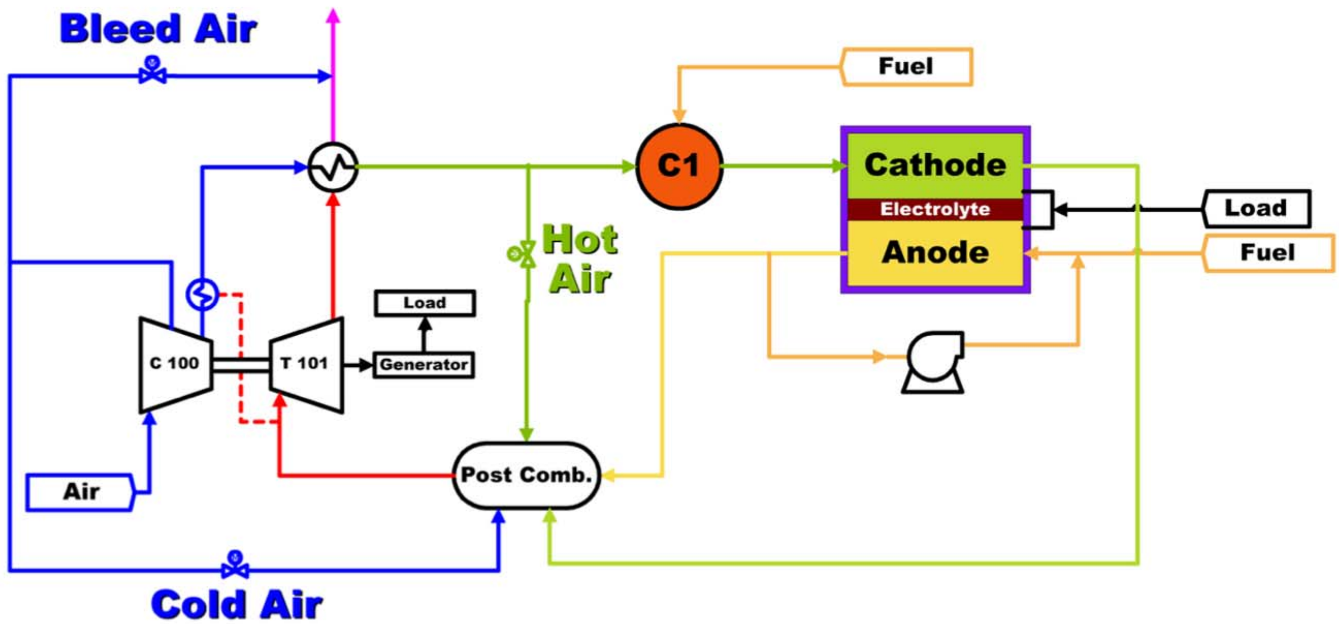
- Based on existing monolithic cookstove model
- Re-implemented as seven stateless microservices
- Federation management system to manage model execution
- Internal communication via message queues
- System model accessible as web information service



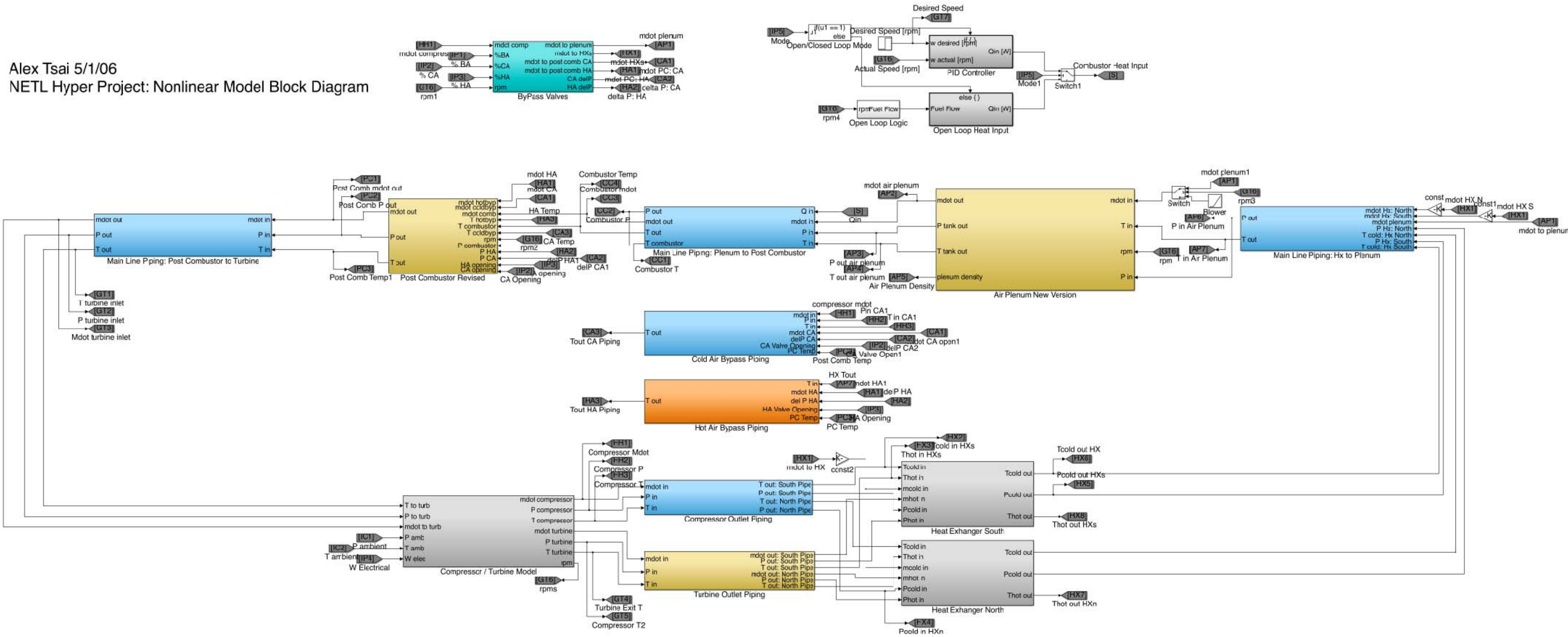
Small cookstove model

Case	Efficiency (%)		Execution Time (Seconds)	
	Monolithic Model	Federated Model Set	Monolithic Model	Federated Model Set
1	34.7	34.7	2.2	3.06
2	13.69	13.69	1.06	2.75
3	28.4	28.4	0.72	1.33

- Results of three design cases run on original and federated models
- Model output matches perfectly
- Federated model takes slightly longer to run.



- Current - a hybrid fuel cell and gas turbine system
- Goal - test the dynamic performance of any advanced power system that includes a gas turbine cycle



Hyper Simulink Model

- Required the development of an object oriented support framework
- Each major component implemented as an independent class
- Components reused where feasible
- Common interface for each model
- Validated against original Simulink model
- 18,000 lines, 65 files

- Running, proof-of-concept federated modeling system
  - Model microservices
  - Federation management system
  - Message queue
  - Web front-end
- Simple, experimental web user interface
- Growing collection of component models: cookstoves, Hyper, …

But…

In order to build a federated model set, you must write code in Java.

Where are we at?

Input blocks:

```
constants <system name>
 <constant 1>
 <constant 2>
 .
 .
 <constant n>
end constants
```

```
inputs <system name>
 <input 1>
 <input 2>
 .
 .
 <input n>
end inputs
```

Output block:

```
outputs <system name>
 <output 1>
 <output 2>
 .
 .
 <output n>
end outputs
```

System block:

```
system <system name> from library <library address>
 <model 1 name>
 <model 2 name>
 .
 .
 <model n name>
end system
```

A message contract consists of:

- A GUID
- A human-readable name
- A human-readable description
- A list of variables and types that make up the payload

Message Contract

Annual Hours of Fuel

GUID
e4c946

Data:

Name	Type
hours	double
type	string

Description:
Defines the annual hours used of a given fuel type.

Fuel Collection

Inputs

- **Fraction of Collected Fuel**
- **Annual Energy Usage**
- **Fraction of Fuel Preparation**
- **Opportunity Cost of Fuel Collection**

Outputs

Message Contract
e4c946

Description
Annual fuel collection time

Constraints
- Unit of time is hours.
- Numerical value must be positive.
- Fuel type must be one of known types.

Message Contract

Message Contract Number
e4c946

Data:

Name	Type
hours	double
type	string

Short title:

Annual Hours of Fuel

Description:

Defines the annual hours used of a given fuel type.

Fuel Cost

Inputs

Message Contract
e4c946

Description
Annual fuel collection time

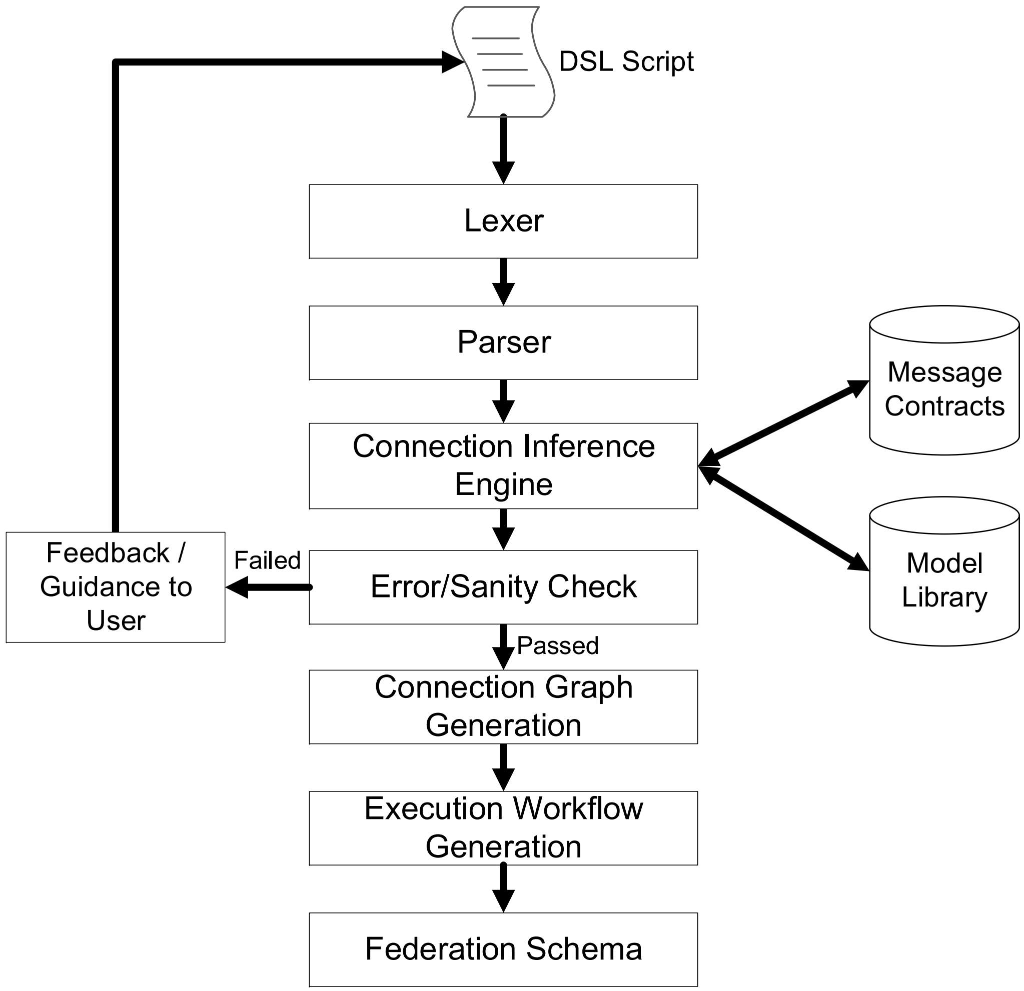
Constraints
- Unit of time is hours.
- Numerical value must be positive.
- Fuel type must be one of known types.

- **Fraction of Shadow Value of Time**
- **Cost of Human Labor**

Outputs

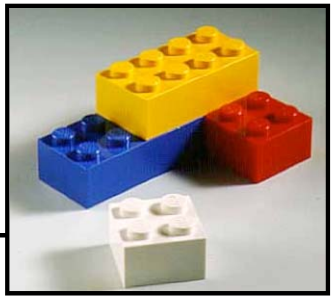
- **Annual Cost of Fuel Collection Time**

Model Connection with Message Contract

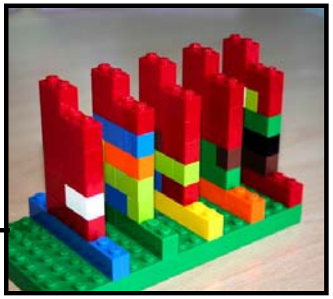


Domain Specific Language Toolset

Snap



Build



Do

New modeling paradigm

Acknowledgements

Current and former PhD students

Dan Bell
Steve Corns
Peter Finzell
Steven Gent
Gengxun Huang
Nate Johnson
Peter Johnson
Balu Karthikeyan
Nordica MacCarty
Doug McCorkle
David Muth
Zach Reinhart
Sunil Suram
Aditya Velivelli

Collaborators

Prof. Dan Ashlock
Dr. Aaron Bryden
Prof. Kris Bryden
Prof. Arne Hallam
Prof. Richard LeSar
Prof. Tom Shih

Major Funders

U.S. Department of Energy, Office of Fossil Energy

U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy

U.S. Army, Armament Research, Development and Engineering Center

Deere & Co

And many undergraduate and masters student researchers and our colleagues and friends in the ISU Virtual Reality Applications Center