A cloud-based modeling framework for complex advanced power systems

Dr. Mark Bryden
Energy and environmental challenges

Increasing energy use
Increasing impact on the environment
Increasing resource scarcity
Interactions between Engineered, human, and natural systems are confounded by complexity.

Holistic solutions are needed
Today we cannot model the richness, fullness, or complexity of engineered, human, or natural systems.

New modeling approaches are needed
Many different models
- No uniform, active storage space
- 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
- Algebraic expression, ODEs, reduced-order models
- Averaging and message passing
Proposed Design Process

Problem Definition → Conceptual Design → Preliminary Design → Detailed Design → Validation & Verification → Final Design Selection & Production

Systems Modeling

- Engineering Design
  - Conceptual Design
  - Preliminary Design
  - Detailed Design
  - Validation & Verification

Analysis is preserved as a part of the product.
Demonstrate the framework and tools needed to create detailed systems models based on reusable, accessible models that enable policy, engineering, and operational decisions for advanced fossil energy systems.

This requires the development of

- Web enabled information objects
- Federated model sets
- User interaction tools
• 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: CSP - Fossil Hybrid
Three questions

- Why can’t we integrate analysis into engineering decision making on-the-fly?
- Why isn’t engineering analysis like a game?
- Why do we continually make new models?
Federated Modeling

ontological and semantic independence

model autonomy

unified models (frameworks with normalized semantics)

centralized models (one code with unified schema)

federated model sets (autonomous models with peer-to-peer controls)

composite models (one code with scripting)
Approach

**Federation schema**

- Inputs
- Outputs
- Model A ...
- Model C ...
- Model E ...
- Duplicate
- Model F ...
- Model G ...
- End duplicate

**Cloud**

- Model A
- Model B
- Model C
- Model D
- Model E
- Model F
- ...

**Library**

**Federation management system**

**User interface**

What’s needed

Models → Cloud-based information objects

Cloud-based information objects → Federated models

Federated models → Engineering decisions
**Work flow**

**Analysts**
- Cloud
  - Model A
  - Model B
  - Model C
  - Model D
  - Model E
  - Model F
  - ...

**Systems Modelers**
- Library
- Federation schema
  - inputs
  - outputs
  - Model A ...
  - Model C ...
  - Model E ...
  - Duplicate
    - Model F ...
    - Model G ...
  - End duplicate

**Users**
- User interface

**Modeling strategy**
- High fidelity
- Empirical and mechanistic simulation

**Analysts**

**Systems Modelers**

**Users**

Components and information flow

Cloud
- Model A
- Model B
- Model C
- Model D
- Model E
- Model F

Library

Federation schema
- Model A ...
- Model B ...
- Model C ...
- Model E ...
- Duplicate
  - Model F ...
  - Model G ...
- End duplicate

User interface

Federation management system
Components and information flow
Each request is treated as an independent request

- communication is an independent set of a request and a response
- no session information is retained
- the code has no knowledge of the actions of other codes within the federation
- each member of the federation performs a specific task
- models are reusable for other analysis
- models can be strung together like beads on a complex weaving
App implementation (microservices architecture)

User

Web application

Back end (server based)
- extract relevant data from microservice responses
- use in system model
- provide needed info to front end

Front end

User

Microservices

Weather data from weather model
Traffic data from predictive model
Public event data service
• independently deployable
• easy assembly of various models and information sources
• models can be implemented using different programming languages, databases, hardware, and software environments
• direct replacement of a federation member can be performed without disruption to other members or the federation
• microservices perform (provide) one task only
• microservices are reusable
Model space architecture
Small cookstove model

Air Flow = Buoyancy - Friction

Air + Combustion Products

Convective Heat Transfer Zone

Primary Air

Secondary Air

Flame Zone
Gas Phase Combustion,
Pollutant Formation
Fuel bed Zone
Drying, Pyrolysis,
Char Combustion

Conduction
Convection
Radiation
Air Flow

Mesh → Packed bed → Flame → Heat transfer → Air flow → L2 norm → Check

Air flow
Update and increase the detail as needed
• What is the tradeoff between cost, acceptance, and impact for a particular geometry change to an energy technology?

• How do I maximize the impact of the energy system?

• If I change materials what will be impact of the on the environment and the people?
Components
- Cookstoves
- Solar hot water
- Lights

Village Energy Model
- Rebound
- Climate impacts
- User acceptance

Agronomic Model
- Erosion
- Fertility
- Crop yield

Systems level design
<table>
<thead>
<tr>
<th></th>
<th>Current Practice</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Software</strong></td>
<td>Libraries</td>
<td>Web-enabled microservices</td>
</tr>
<tr>
<td><strong>Language</strong></td>
<td><strong>Specific</strong></td>
<td>Agnostic</td>
</tr>
<tr>
<td></td>
<td>(overcome with cross-compilers)</td>
<td></td>
</tr>
<tr>
<td><strong>Integration Cost</strong></td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Data re-use</strong></td>
<td>Not implicit</td>
<td>Implicit</td>
</tr>
<tr>
<td><strong>Latency</strong></td>
<td>Designed to be low</td>
<td>Can be higher</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(needs to be addressed)</td>
</tr>
</tbody>
</table>
## Work Plan

### Models to cloud-based information objects
- 3. Extract models for Hyper
- 6. Build analysis-as-a-web-service
- 7. Test analysis-as-a-web-service

### Federated modeling tools
- 11. Build federated model execution system
- 12. Test federated model system

### User interaction tools
- 15. Exploration toolkit
- 16. Decision space toolkit

### Final Integration and demonstration
- 17.
Models to cloud-based information objects

- Complete establishing a separable model set for the NETL Hyper facility
- Developed and implemented the needed linking schema for a simple energy problem
- Outlined the schema needed to link models together using a peer-to-peer ontology

User interaction tools

- Chose Unity™ as the user interaction environment
- Initiated development of a simple example based on “hand crafted” connection points
New modeling paradigm
Decision making environments that integrate all the information, models, and other artifacts related to a product or process.
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. 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