

# **Systems Platform for Integrated Design in Real-time (SPIDR)**

**“Start Higher for a Better View”**

**Dr. Tatiana Kichkaylo and Dr. Gordon Roesler**  
**Information Sciences Institute**  
**University of Southern California**

## **We will describe a tool for design optimization and evaluation**

- Principles of operation
- **How it is different** from other optimization engines

## **Brief mention of successes**

- Being used at the conceptual design stage

## **Expanding and enhancing the tool's applicability**

- Enable programming by non-experts
- Evaluation of designs in later stages
- Interfaces with other applications for increased capability
- Dynamic simulation
- Hardware-in-the-loop

# Helping the engineer explore large spaces

## **Systems Engineering is about integrating concerns**

- **Functional subsystems**
- **Cost and reliability**
- **Manufacturability and deployment**
- **Schedule slips, funding dips, personnel...**

## **Everything affects everything**

- **Sophisticated models exist (and/or are under development) for various aspects**
- **Collaboration and project management tools help data flow (to some extent)**
- ***Optimization across concerns still done manually***

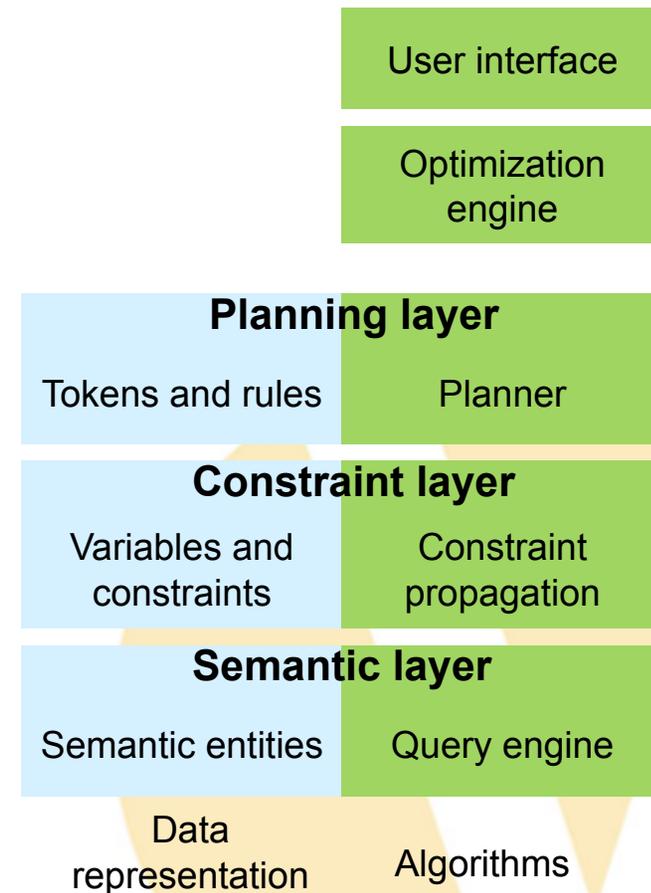
**In design, a very human tendency is to converge to a point solution ASAP—this causes potential optima to be missed**

## SPIDR relies on Artificial Intelligence techniques

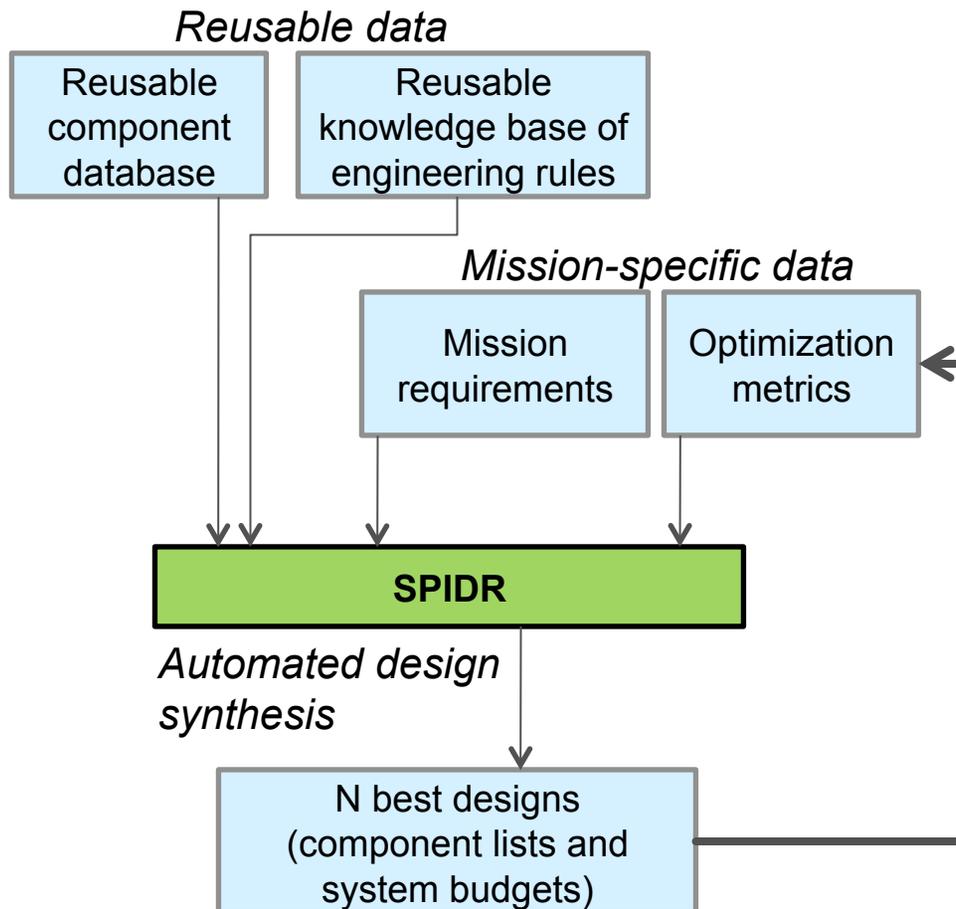
- Encode engineering knowledge as machine readable rules
- Allow discrete and continuous variables
- Variables are both inputs and outputs
- Use constraint-based solver to find best solutions with respect to some metric

## "Put all pieces into one bucket and shake"

- SPIDR is not a *model*. It is an engine.
- Since both components and CONOPS are rules, they are *co-optimized*



# SPIDR gives confidence in realism of results



Starting from high level allows the system to consider completely different configurations and approaches to find the best solution

The tool is agnostic to the meaning of rules and constraints, so it is easy to integrate multiple concerns

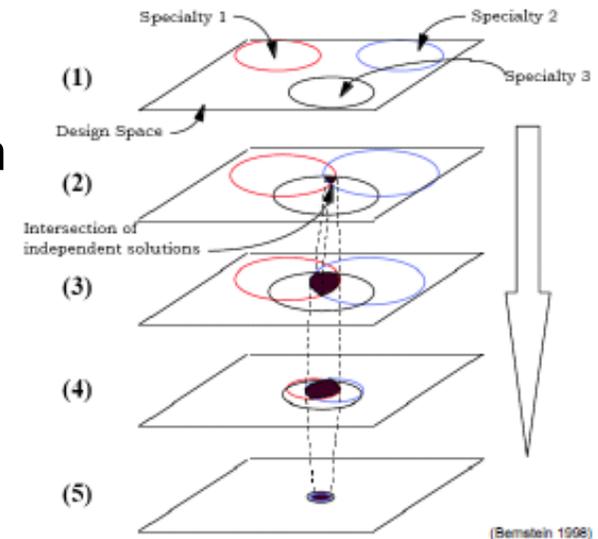
***User revises requirements and metrics as needed\****

*\* A Web interface enables review, inputs, and suggestions from remote experts*

# SPIDR embodies Set Based Design

## Set of designs is *implicit* in requirements and DB

- Original application: component-based design
  - First application: small satellites
  - One-shot optimization runs
  - Functional subsystems: select components given mission specification
- Next application: components and CONOPS optimized simultaneously (example follows)
  - System: autonomous surveillance vehicles, *with energy harvesting*
  - Added capability to optimize continuous as well as discrete variables
  - Remote access allowed SMEs to review/revise/explore in real time



← We are here

## Additional capabilities will greatly expand the utility

- Manufacturability and logistics concerns
- From Design to I&T to Acquisition
- Interfaces to other design/scientific/engineering applications

# SPIDR can address challenges further into the life cycle

## Incorporate dynamic simulations

## Third party models as constraints

- E.g., Matlab, specialized applications

## Non-functional constraints

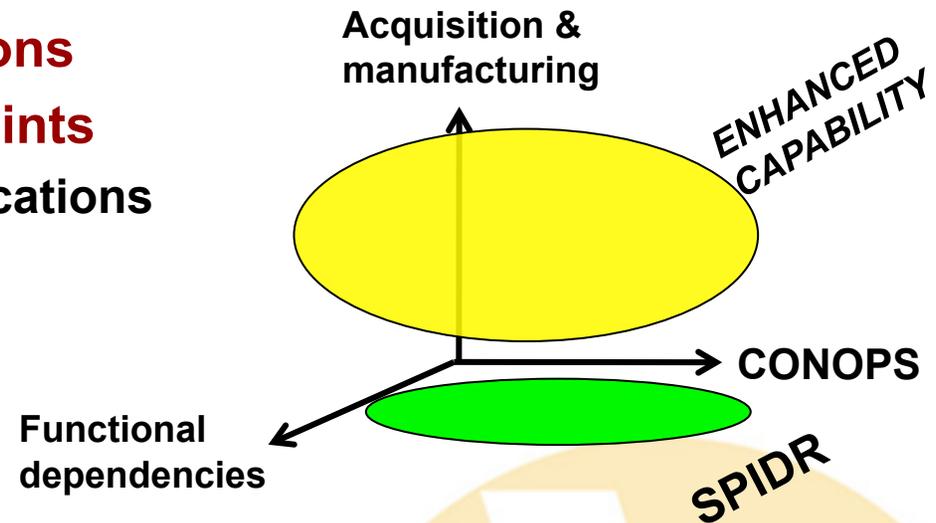
- Schedules
- Manufacturability
- Logistics
- Deployment

## Component arrangements within spatial envelope (3D)

## Flow into Integration and Testing, including ability to drive hardware-in-the loop simulations

## KEY STEP: End-user support for non-computer scientists

- User interfaces
- Training material and processes



# SPIDR example: How autonomous replenishment enhances autonomous surveillance

## Given

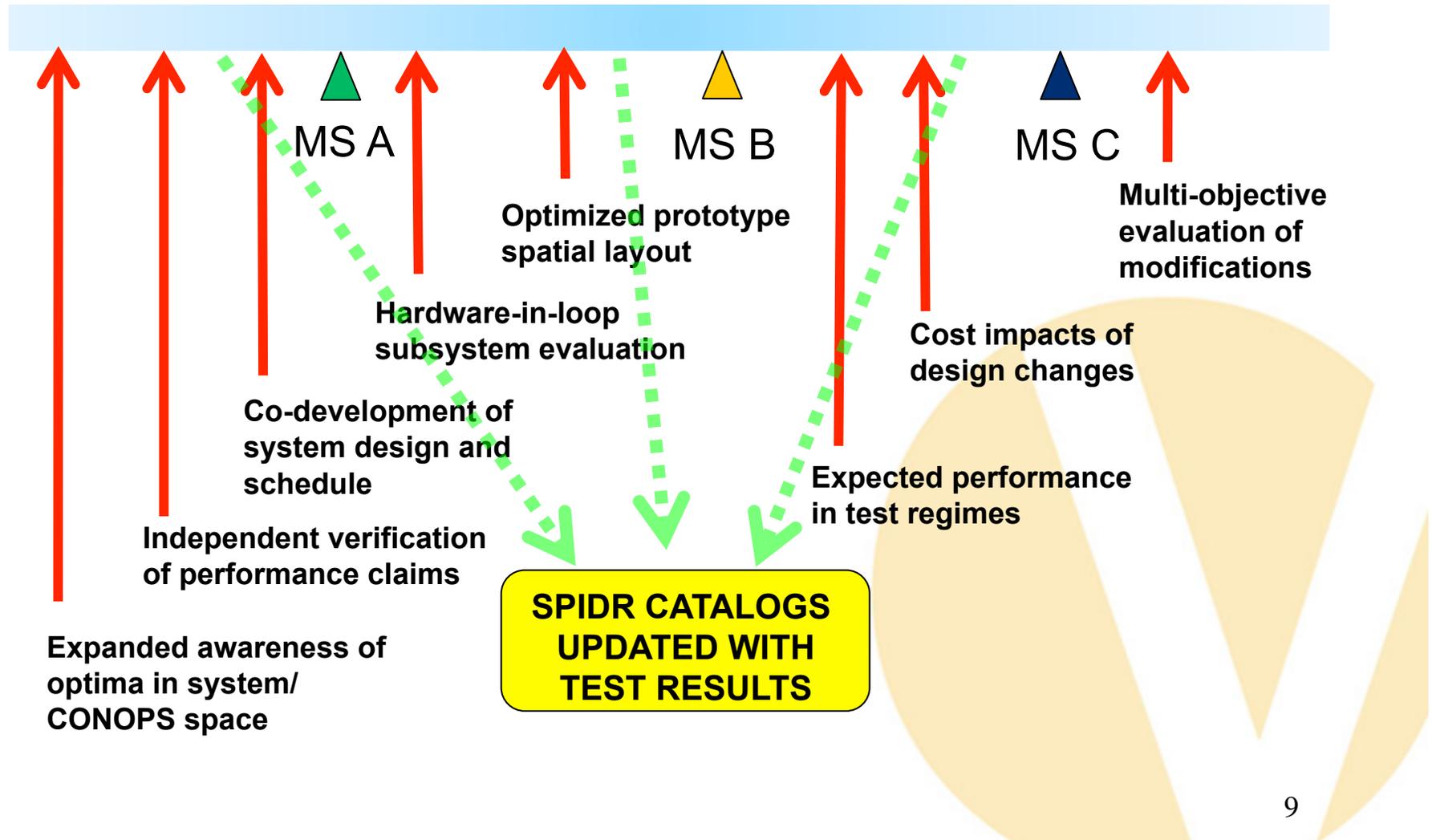
- Different areas of operation and different goals
- Different energy harvesting and energy storage technologies
- Different vehicle sizes, shapes, and component technologies
- Different operating speeds, operating environments
- Different fleet configurations
- Different deployment restrictions

## We found:

- The best approach to satisfy mission goals and how much it would cost (under varying constraints)
- The limiting factors (what technologies need to be developed or improved)
- Capabilities that would be enabled by new components and/or technologies
- The "sweet spots" on the cost-performance curve

# Potential contributions to the DOD Acquisition Process

SPIDR → SPIDER: expand **EVALUATION** capability



- ✓ • **Know system “sweet spots” and sensitivities**
- ✓ • **Early assessment of the impact of requirements**
- ✓ • **Easy verification of engineering basics**
  - **Enable reasoning under pressure and uncertainty**
  - **Knowing the envelope and designing closer to it**
  - **Maintain flexibility longer in the life cycle**
  - **Independent evaluation of designs and claims**
  - **Support the development of human SE capital**
  - **Support technical veracity**