Collaborative Modeling, Simulation, and Analytics with Splash

Nicole Barberis, **Peter J. Haas**, Cheryl Kieliszewski, Yinan Li, Paul Maglio, Piyaphol Phoungphol, Pat Selinger, Yannis Sismanis, Wang-Chiew Tan, Ignacio Terrizzano, Haidong Xue, SJSU CAMCOS

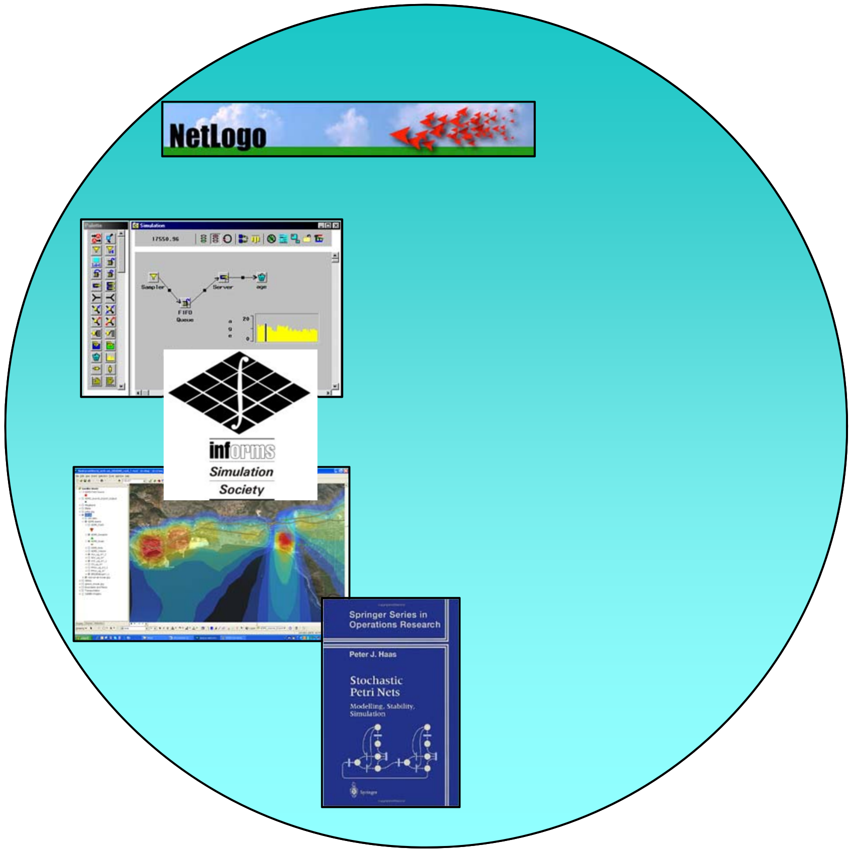
IBM Research – Almaden

Splash
Smarter Planet Platform
for Analysis and
Simulation of Health

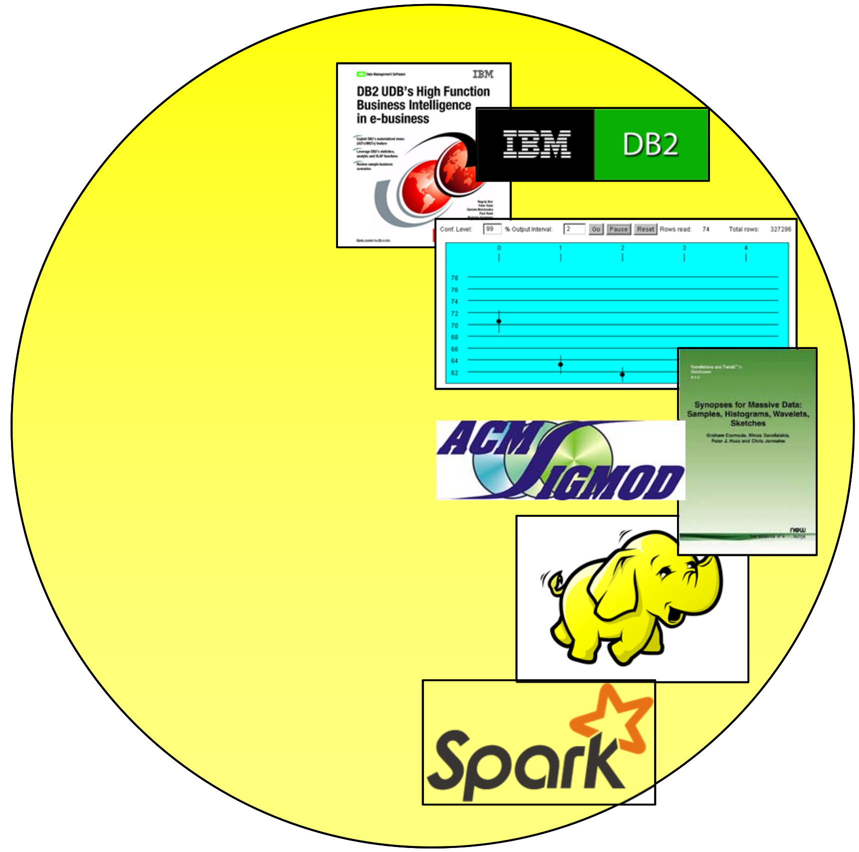
http://researcher.watson.ibm.com/researcher/view_project.php?id=3931

Some Context: Model-Data Ecosystems

My Two Communities

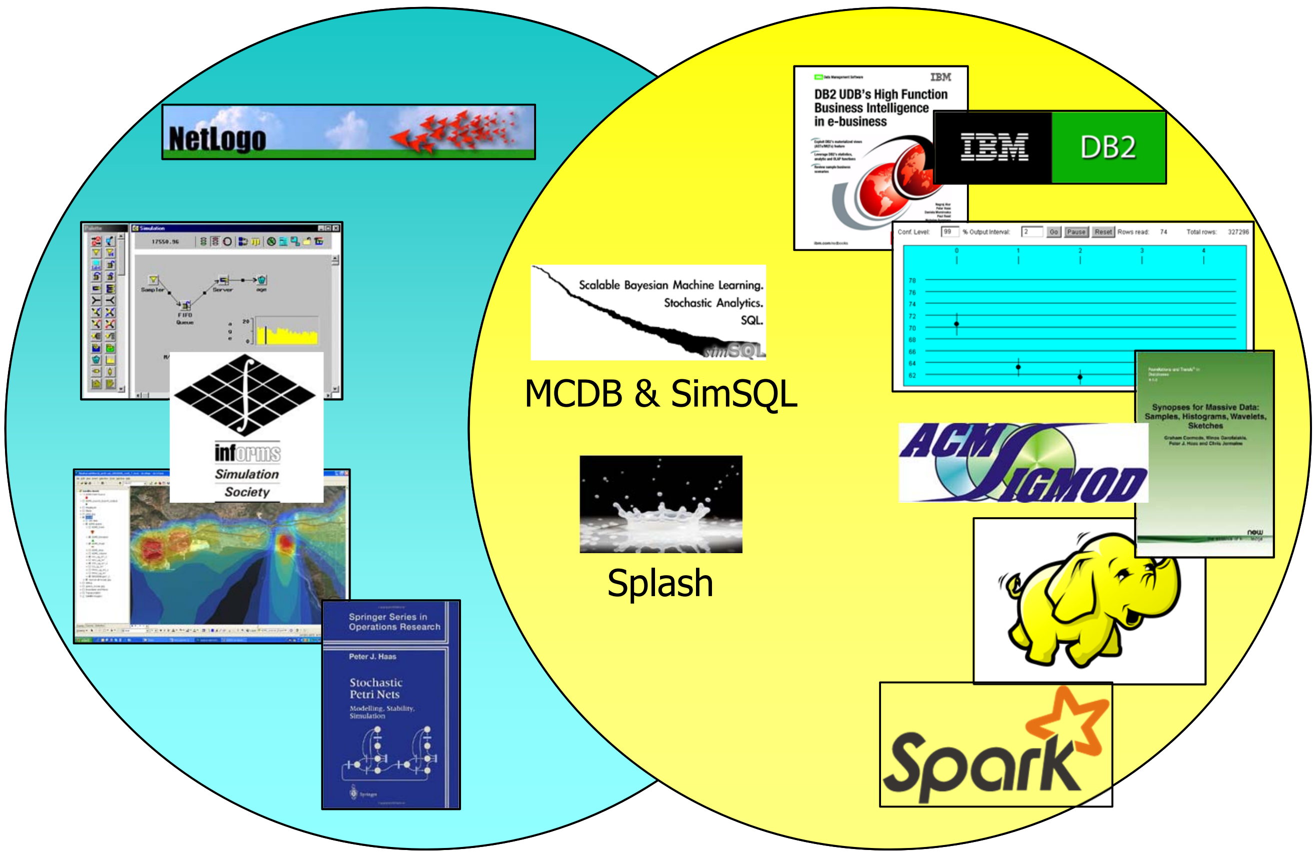


Modeling and Simulation



Information Management & Analytics

Opportunities for Innovation at the Intersection



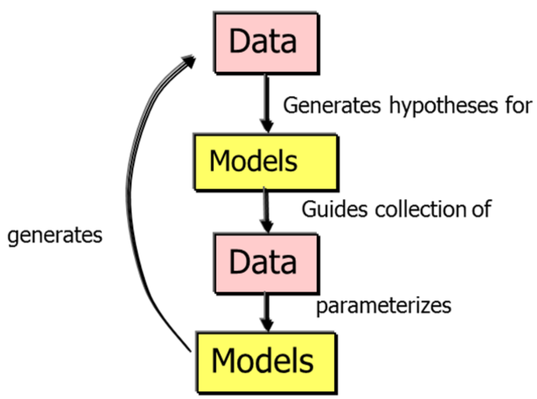
Modeling and Simulation

Information Management & Analytics

Some Further Thoughts and Examples [PODS 2014 Tutorial]

(In addition to large-scale scientific environments)

- Data-intensive simulation
  - Simulations within databases
  - Databases within simulations
  - Data harmonization at scale
- Information integration
  - Simulation as an information-integration tool
  - Combining real and simulated data
- And more!



Ecosystem of Data and Models

Model-Data Ecosystems: Challenges, Tools, and Trends

Peter J. Haas
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099 U.S.A.
phaas@us.ibm.com

ABSTRACT

In the past few years, research around (big) data management has begun to intertwine with research around predictive modeling and simulation in novel and interesting ways. Driving this trend is an increasing recognition that information contained in real-world data must be combined with information from domain experts, as embodied in simulation models, in order to enable robust decision making under uncertainty. Simulation models of large, complex systems (traffic, biology, population well-being) consume and produce massive amounts of data and compound the challenges of traditional information management. We survey some challenges, mathematical tools, and future directions in the emerging research area of model-data ecosystems. Topics include (i) methods for enabling data-intensive simulation, (ii) simulation and information integration, and (iii) simulation metamodeling for guiding the generation of simulated data and the collection of real-world data.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*decision support*; I.6 [**Simulation and Modeling**]: Simulation Support Systems

General Terms

Algorithms, Design

Keywords

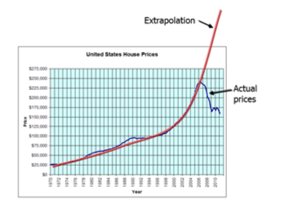
Simulation, data assimilation, information integration, decision support

1. INTRODUCTION: DATA IS STILL DEAD

In their *VLDB 2011* paper, "Data is dead…without what-if analytics", Haas et al. [27] point out that, outside of scientific or historical investigations and monitoring-type applications, the essential motivation underlying data processing and analytics is the need to support enterprise decision making under uncertainty. Thus the ultimate goal is to support deep predictive analytics that incorporate domain expertise in order to robustly predict the future consequences of decisions made today. From this perspective, data by itself is indeed "dead", reflecting the past state of the world. Descriptive analytics—such as simple querying, OLAP, data mining, machine learning, and time-series analysis—find important patterns and relationships in existing data, leading to insights about the real world as it currently stands. A "shallow" predictive approach that simply extrapolates current patterns into the future, however, can lead to very brittle predictions and subsequent bad decisions because it does not account for the fact that the mechanisms that generated the existing data can change. Figure 1 illustrates this point. A simple time series model was fit to median U.S. housing prices from 1970 to 2006 and then extrapolated to 2011. As can be seen, the resulting prediction failed spectacularly because it ignored expert information from economists, financial analysts, behavioral scientists, and others that might have helped in modeling the housing-price collapse that began in 2006. Thus data must be supplemented by models that embody expert knowledge about the constituent parts of systems and the way they behave and interact. For systems characterized by uncertainty, these models often take the form of stochastic simulations.

Eric Bonabeau, the author of *Swarm Intelligence*, makes a similar point in one of his blogs [9]:

> There is no doubt that the more *information* is used in building a model, the more accurate the model is likely to be. However, the notion that quantitative, nu-



Extrapolation of 1970-2006 median U.S. housing prices

Figure 1: The dangers of extrapolation

PODS'14, June 22–27, 2014, Snowbird, UT, USA.

ACM 978-1-4503-2375-8/14/06 ...$15.00.
http://dx.doi.org/10.1145/2594538.2594562.

76

Motivation for Splash

The Setting: Analytics for Decision Support

informs online

Analytics Section

"Analytics is…a complete [enterprise] problem solving and decision making process"

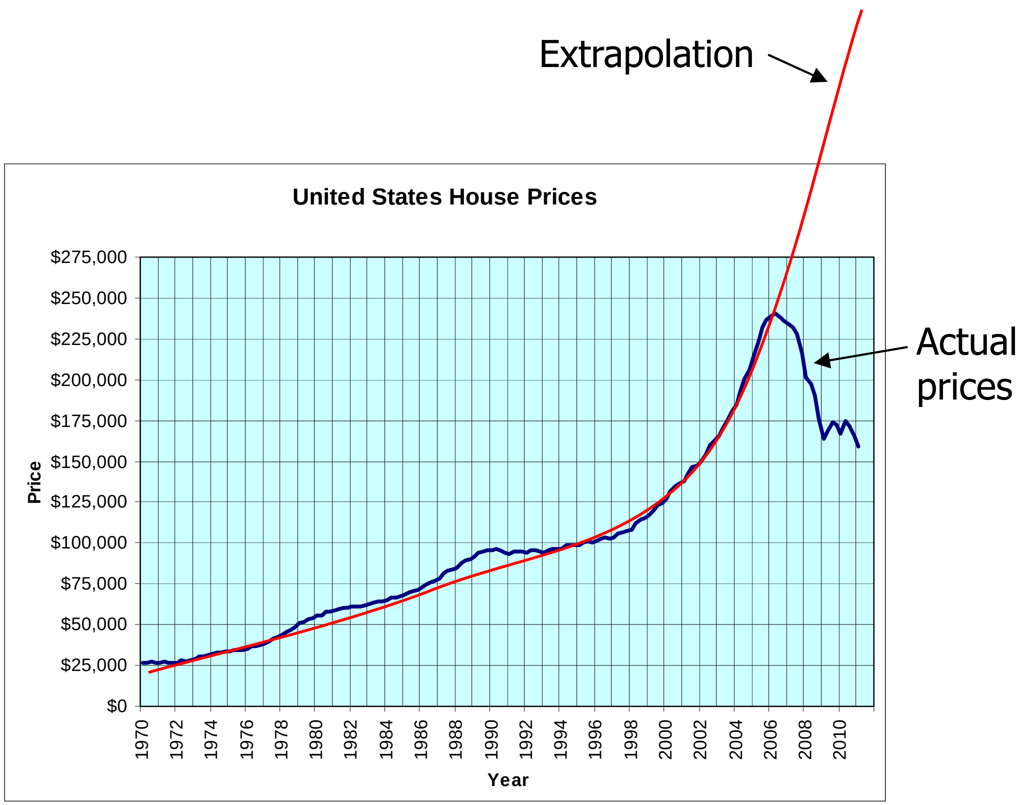
Descriptive Analytics: Finding patterns and relationships in historical and existing data

Splash

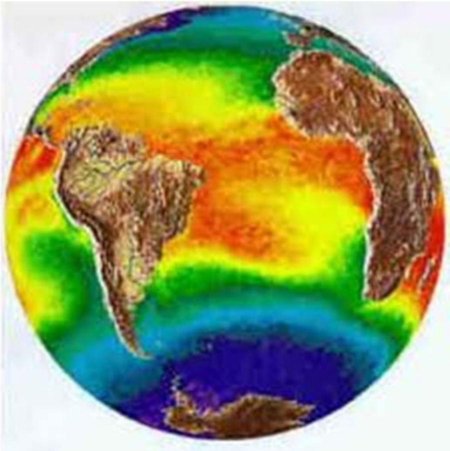
Predictive analytics: predict future probabilities and trends to allow what-if analysis

Prescriptive analytics: deterministic and stochastic optimization to support better decision making

Shallow Versus Deep Predictive Analytics



Extrapolation of 1970-2006 median U.S. housing prices



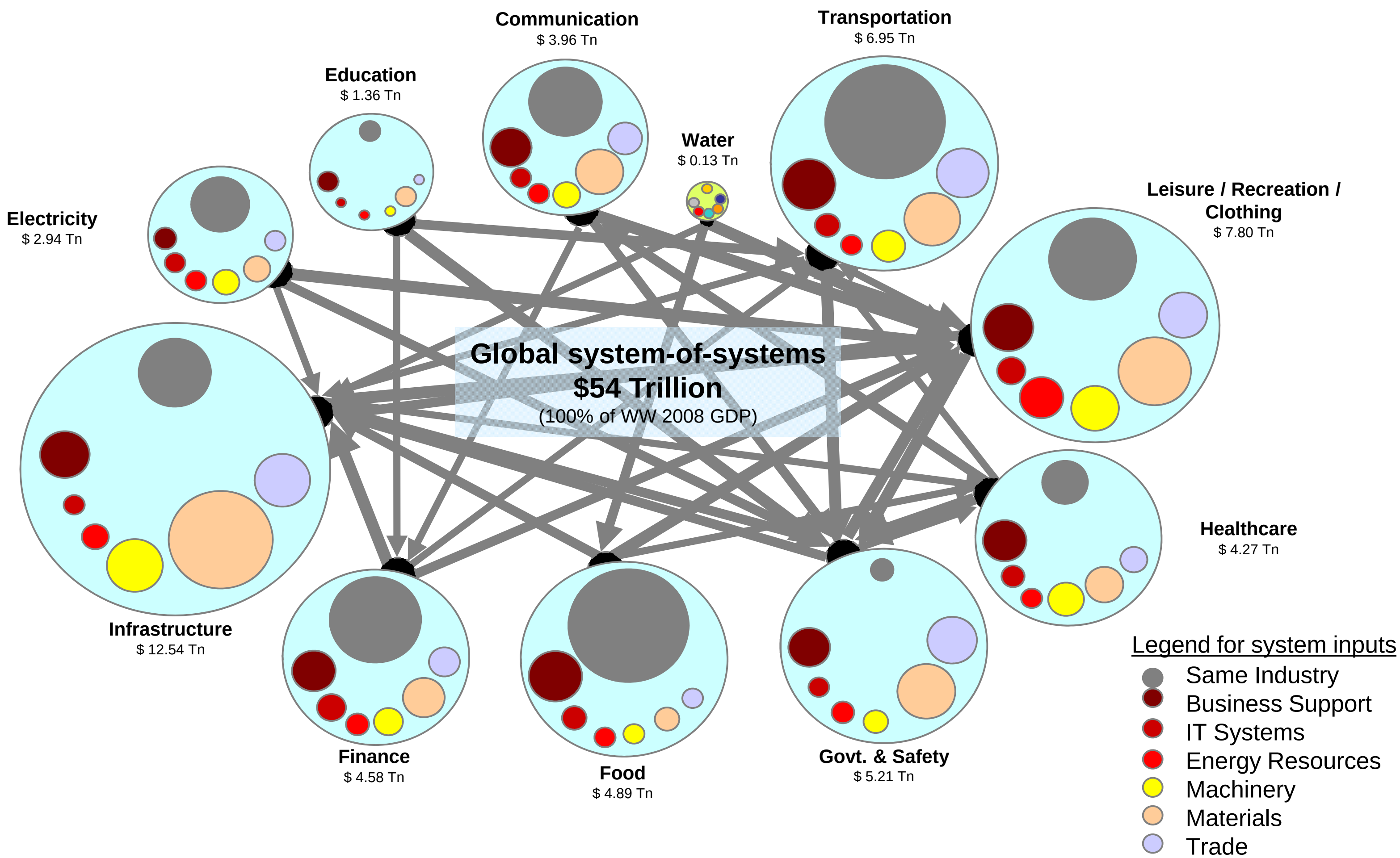
NCAR Community Atmosphere Model (CAM)

3.3 Eulerian Dynamical Core

$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} &= \boldsymbol{k} \cdot \nabla \times (\boldsymbol{n}/\cos\phi) + F_{\zeta_H}, \\
\frac{\partial \delta}{\partial t} &= \nabla \cdot (\boldsymbol{n}/\cos\phi) - \nabla^2 (E + \Phi) + F_{\delta_H}, \\
\frac{\partial T}{\partial t} &= \frac{-1}{a\cos^2\phi}\left[\frac{\partial}{\partial \lambda}(UT) + \cos\phi \frac{\partial}{\partial \phi}(VT)\right] + T\delta - \dot{\eta}\frac{\partial T}{\partial \eta} + \frac{R}{c_p^*}T_v\frac{\omega}{p} \\
&\quad + Q + F_{T_H} + F_{F_H}, \\
\frac{\partial q}{\partial t} &= \frac{-1}{a\cos^2\phi}\left[\frac{\partial}{\partial \lambda}(Uq) + \cos\phi \frac{\partial}{\partial \phi}(Vq)\right] + q\delta - \dot{\eta}\frac{\partial q}{\partial \eta} + S, \\
\frac{\partial \pi}{\partial t} &= \int_1^{\eta_t} \boldsymbol{\nabla} \cdot \left(\frac{\partial p}{\partial \eta}\boldsymbol{V}\right) d\eta.
\end{aligned}
$$

Big, Difficult, Important Problems Span Many Disciplines

Need collaborative cross-disciplinary modeling and simulation



IBM analysis based on OECD data.

POLICYFORUM

GLOBAL FOOD SUPPLY

Linking Policy on Climate and Food

Agriculture and the food system need to move center stage in preparing for UN climate negotiations in December 2011.

H. C. J. Godfray,[1] J. Pretty,[2] S. M. Thomas,[3*] E. J. Warham,[3] J. R. Beddington[3]

At the United Nations (UN) climate negotiations in Cancún, Mexico, in December 2010, the parties agreed to a global target of no more than 2°C warming above preindustrial levels. In an important new step, both developed and developing countries agreed to take urgent action to reduce greenhouse gas (GHG) emissions to meet this long-term goal. They also set important milestones on reducing deforestation and providing funds to help developing countries adapt to climate change.

sions as delegates prepare for the next UN negotiations in December 2011 in South Africa. We need to rethink the way we use land to produce food, and to bring the challenges of sustainability and reducing emissions to the fore. This has been a central theme of the UK Government's Foresight Programme on the Future of Food and Farming to which we, along with experts from 35 countries, have been contributors. The study took a broad approach to the food system, including its impact on the environment and especially climate change, as well as the special needs of the world's poorest. It demonstrates both the importance of incorporating agriculture into climate change discussions, and the urgency for action (*3*).

Agriculture and Climate Change

Agriculture is a major source of CO_2 emissions and contributes a disproportionate amount of other GHGs with high impact on warming [about 47% and 58% of total CH_4 and N_2O

emissions by 20% by 2020 (*8*), whereas the UK has set the legally binding target of reducing emissions by 34% by 2020 and at least 80% by 2050 (*9*). Ambitious goals such as these cannot be achieved without involving the food system. Policies for mitigating climate change will have a substantial effect on production. If applied inappropriately, these could have a detrimental effect on food availability, especially for the 925 million (*3*) who already experience chronic hunger and for the additional billion or so who suffer nutrient and vitamin deficiencies.

Land Use

The Cancún meeting made notable progress in an area with important ramifications for the food system. Pressure from expanding agriculture has led to much recent tropical deforestation, especially in South America and Southeast Asia. Land conversion releases large amounts of GHGs and is one of the most serious, although indirect, ways that pressure from the food system contributes to global warming. The UN initiative on Reducing Emissions from Deforestation and Forest Degradation (REDD) offers financial

POLICYFORUM

GLOBAL FOOD SUPPLY

Linking Policy on

H. C. J. Godfray,[1] J. Pretty,[2] S. M. Thomas,[3]* E. J.

At the United Nations (UN) climate negotiations in Cancún, Mexico, in December 2010, the parties agreed to a global target of no more than 2°C warming above preindustrial levels. In an important new step, both developed and developing countries agreed to take urgent action to reduce greenhouse gas (GHG) emissions to meet this long-term goal. They also set important milestones on reducing deforestation and providing funds to help developing countries adapt to climate change.

The food system is complex, and interventions often have unintended and deleterious effects on food security, or have major consequences that affect GHS emissions. Agricultural, economic, and climate modelers must compare their models more systematically, share results, and integrate their work to meet the needs of policy-makers.

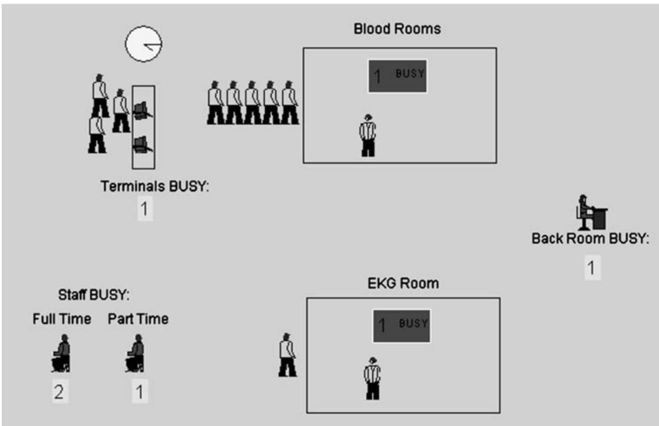
on warming [about 47% and 58% of total CH_4 and N_2O

Reducing Emissions from Deforestation and Forest Degradation (REDD) offers financial

World Health Organization

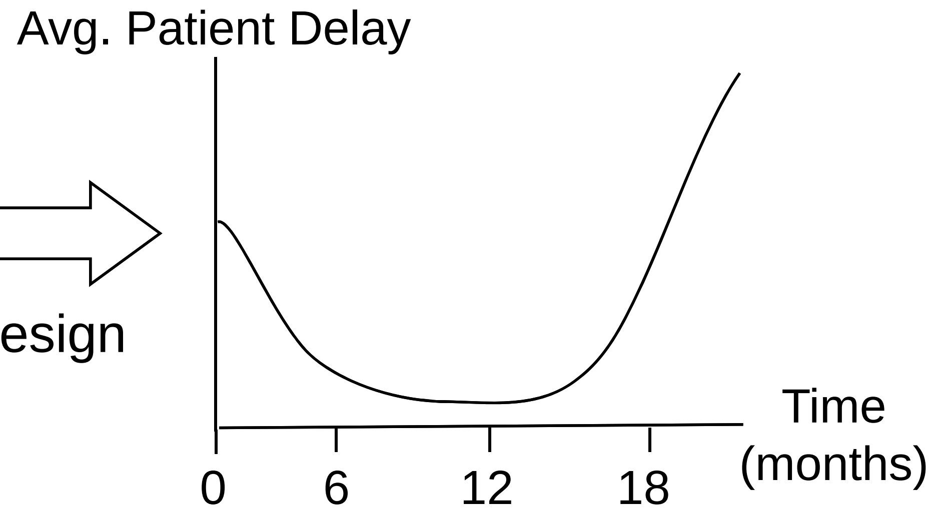
Health is a state of complete physical, mental, and social well-being and not merely the absence of disease or infirmity.

Example: Unintended Outcomes in Healthcare Optimization



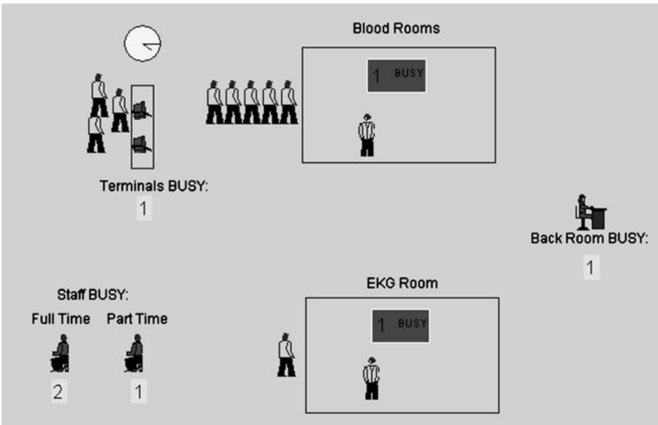
Simulation model of
Calgary Lab Services

Re-design



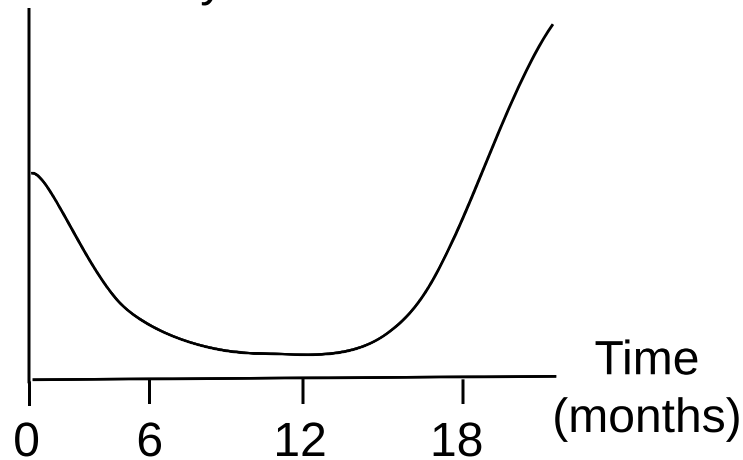
T. R. Rohleder & D. P. Bischak & L. B. Baskin (2007). Modeling patient service centers with simulation and system dynamics. Health Care Manage. Sci., 10:1–12.

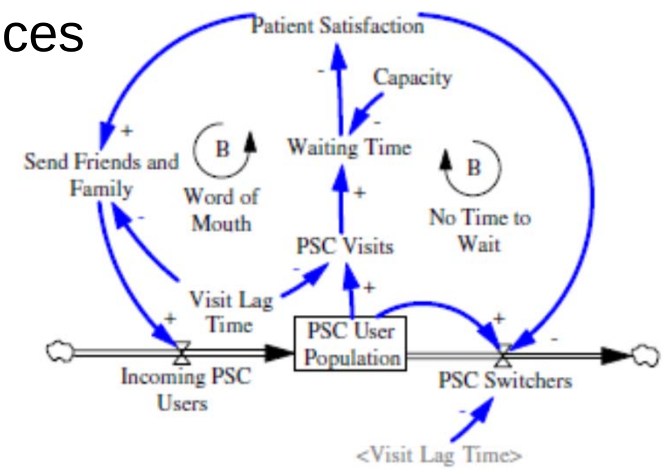
Example: Unintended Outcomes in Healthcare Optimization



Simulation model of
Calgary Lab Services

Re-design

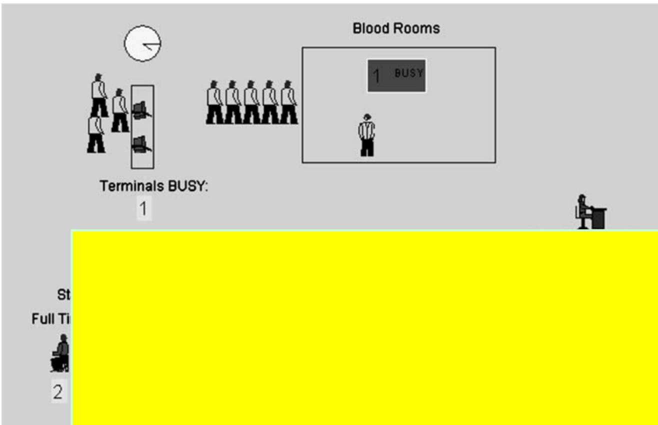




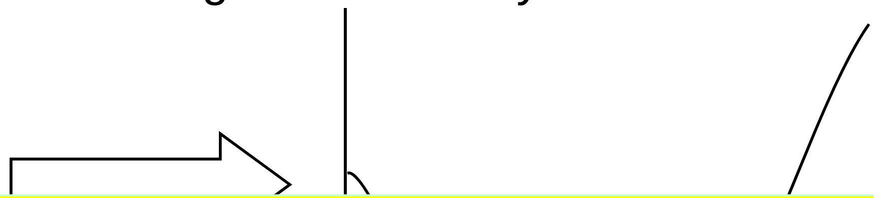
System-dynamics social model of lab use

T. R. Rohleder & D. P. Bischak & L. B. Baskin (2007). Modeling patient service centers with simulation and system dynamics. Health Care Manage. Sci., 10:1–12.

Example: Unintended Outcomes in Healthcare Optimization



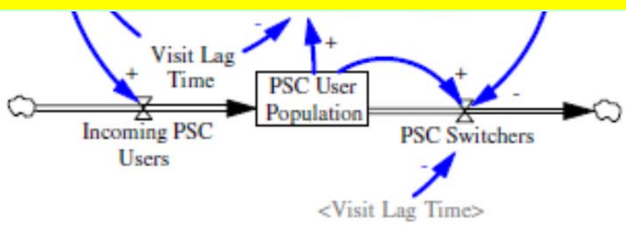
Avg. Patient Delay



ne
nths)

Moral:

Combine models across disciplines for more robust decision making

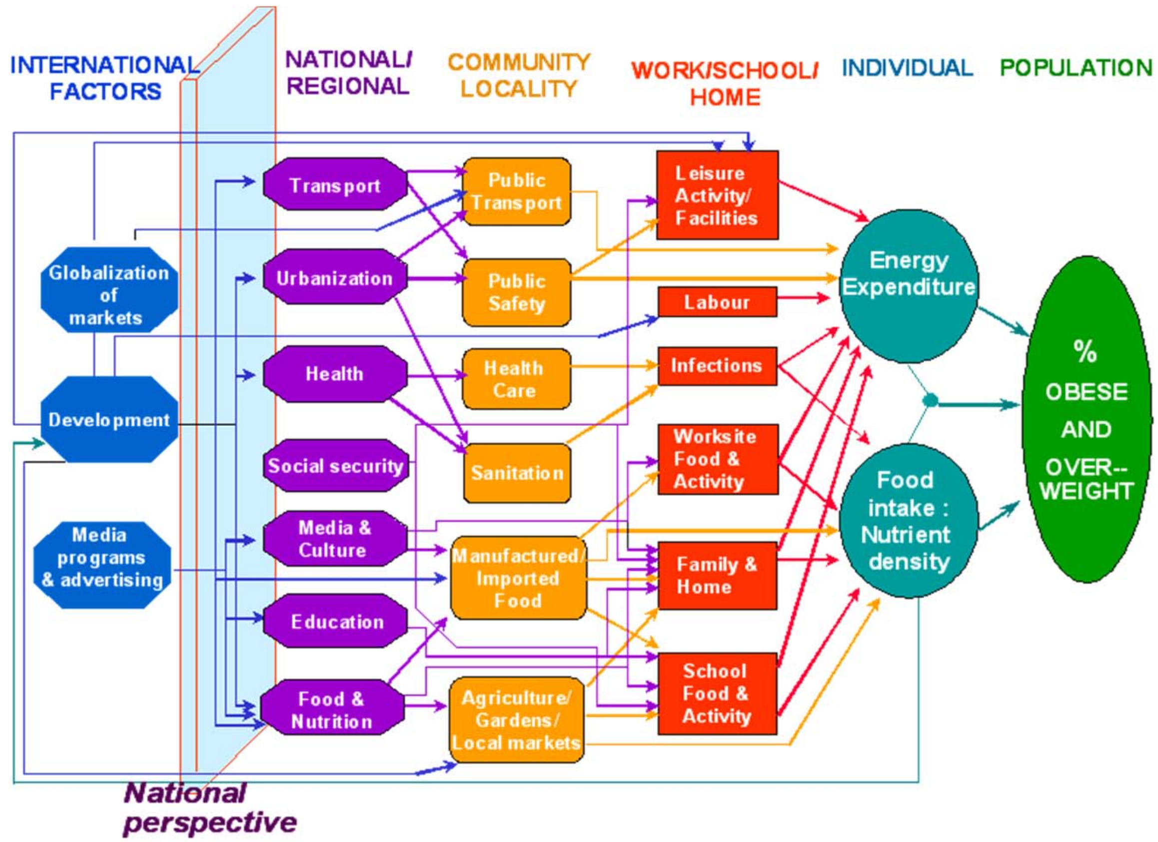


System-dynamics social model of lab use

T. R. Rohleder & D. P. Bischak & L. B. Baskin (2007). Modeling patient service centers with simulation and system dynamics. Health Care Manage. Sci., 10:1–12.

Combining Models Across Disciplines is HARD

- Domain experts have different worldviews
- Use different vocabularies
- Sit in different organizations
- Develop models on different platforms
- Don't want to rewrite existing models!



Huang, T. T, Drewnowski, A., Kumanyika, S. K., & Glass, T. A., 2009, "A Systems-Oriented Multilevel Framework for Addressing Obesity in the 21st Century," Preventing Chronic Disease, 6(3)

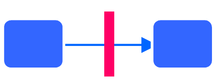
Prior approaches to Combining Models

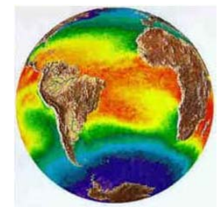
Monolithic models

- Create a monolithic model that encompasses all relevant domains

Integrated models

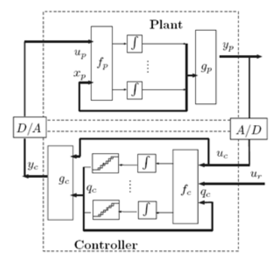
- Create modules that can be compiled into one
  - SpatioTemporal Epidemiological Modeler (STEM)
  - Community Atmospheric Model (CAM)

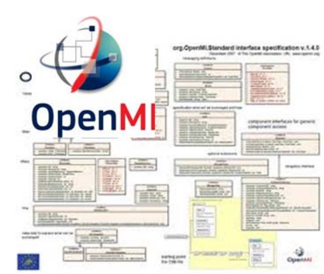




Tightly-coupled models

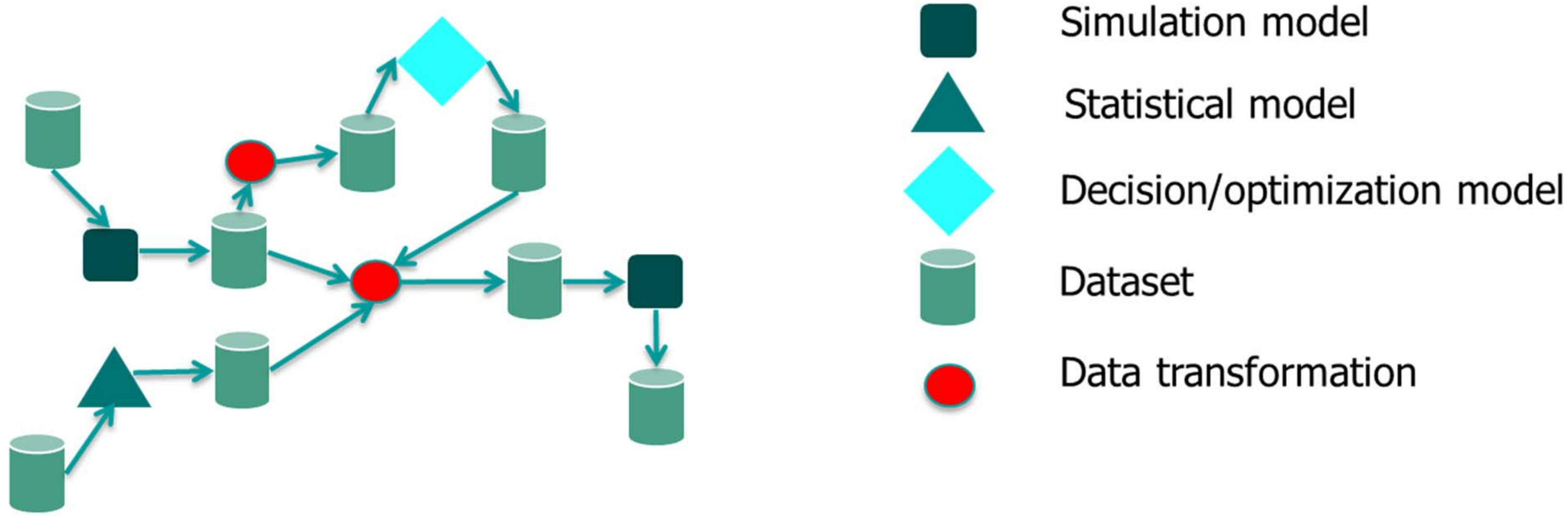
- Create modules that understand standard interfaces
  - DOD High Level Architecture (HLA)
  - Discrete-Event System Specification (DEVS)
  - Open Modeling Interface (OpenMI).





Splash: An Alternative Approach

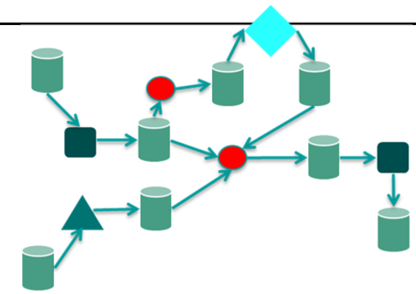
Loosely couple models and data via **data exchange**



Splash = data integration + workflow management + simulation

Re-use heterogeneous models and heterogeneous data that are curated by different domain experts

Some Benefits of Loose Coupling

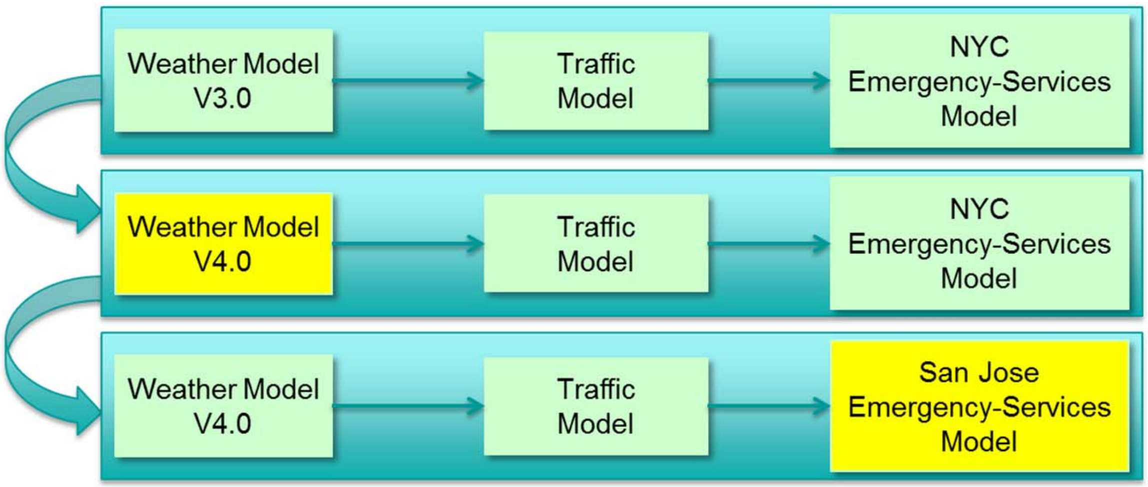


Facilitates cross-disciplinary modeling, analytics, and simulation for robust decision making under uncertainty

Enables re-use of models and datasets

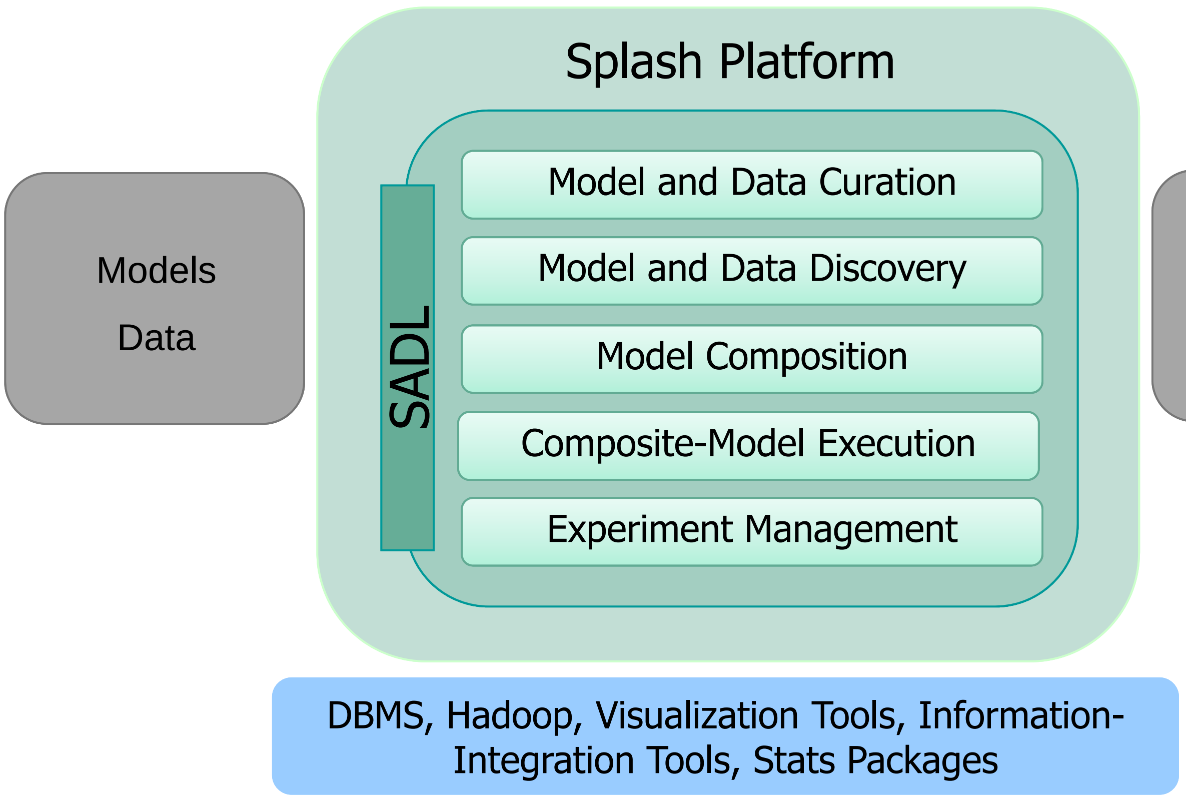
Encourages comprehensive documentation and curation of models via metadata

Allows model flexibility:
- Upgrading to state-of-the-art
- Customizing for different users

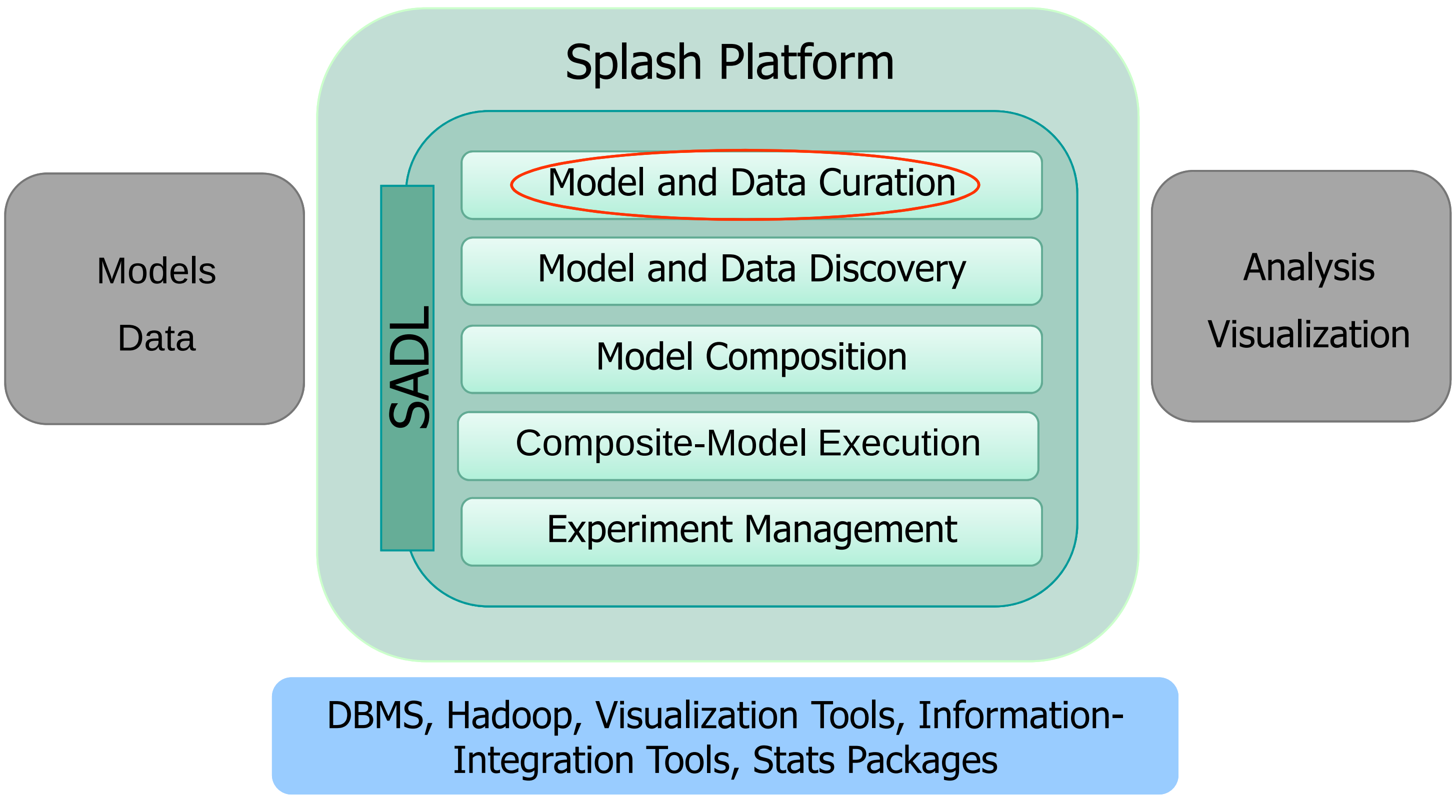


Splash

A prototype platform and service for integrating existing data, models, and simulations to gain insight needed for complex decision making related to policy, planning, and investment.



Model and Data Curation



Splash Actor Description Language (SADL)

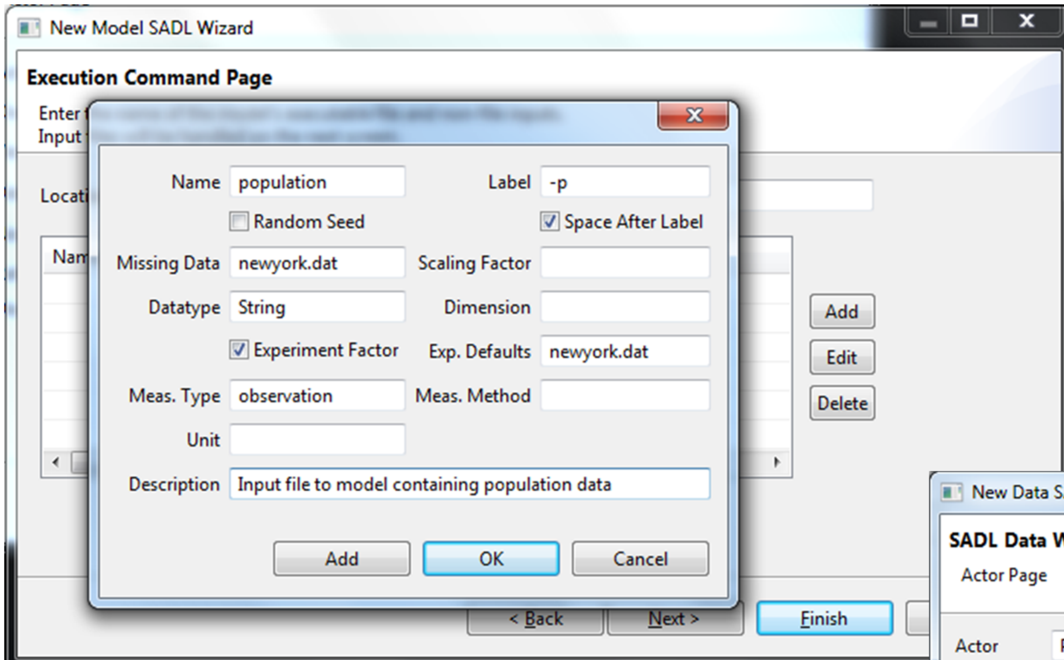
- SADL provides "schemas and constraints" for models, transformations, and data, enabling interoperability

- SADL file for data (can exploit XSD)
  - **Attribute names, semantics, units**
  - **Constraints**
  - **How to access**
  - **Security**
  - **Experiment-management info**

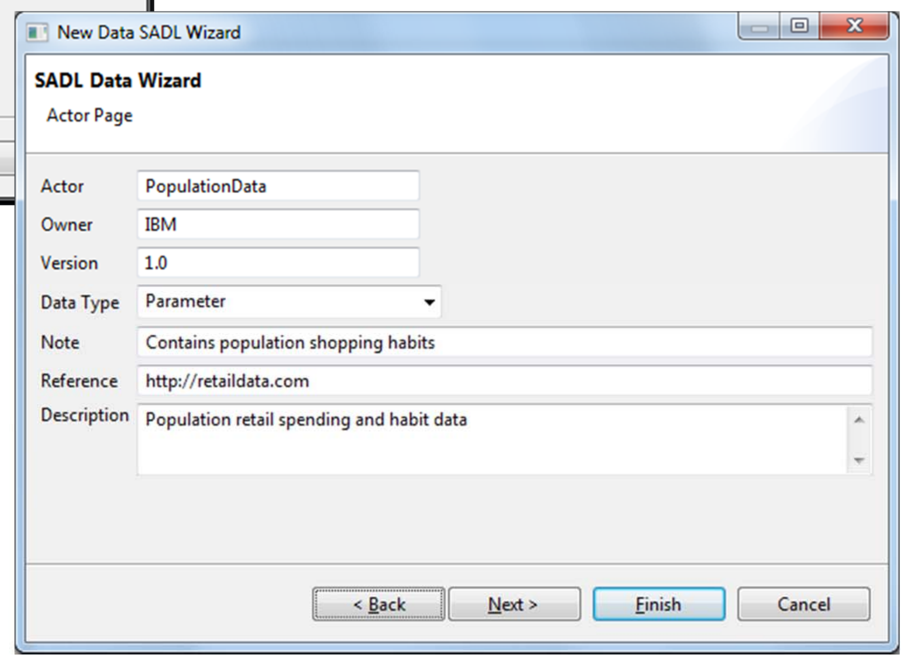
- SADL file for a model:
  - **Inputs and outputs** (pointers to SADL files for data sources and sinks)
  - **How to execute** (info needed to synthesize command line)
  - **Semantics and assumptions**
  - **Provenance** (papers, ratings, ownership, security, change history, …)
  - **RNG info**

```
<Actor name="BMI Model" type = "model" model_type = "simulation"
       sim_type = "continuous-deterministic" owner="Jane Modeler">
  <Description>
  Predict weight change over time based on an individual's energy   and food
  intake. Implemented in C. Reference: http://csel/asu.edu/?q=Weight
  </Description>
  <Environment>
     <Variable name="EXEC_DIR" default="/Splash" description="executable
  directory path"/>
    <Variable name="SADL_DIR" default="/Splash/SADL" description="schema
  directory path"/>
  </Environment>
  <Execution>
    <Command>$EXEC_DIR/Models/BMIcalc.out</Command>
    <Title>Run BMI model</Title>
  </Execution>
'  <Arguments>
    <Input name="demographics" sadl="$SADL_DIR/BMIInput.sadl"
           description="demographics data"/>
    <Output name="people" sadl="$SADL_DIR/BMIOutput.sadl"
           description="people's daily calculated BMI"/>
  </Arguments>
</Actor>
```

Registration: Use Wizards to Create Model and Data SADL Files

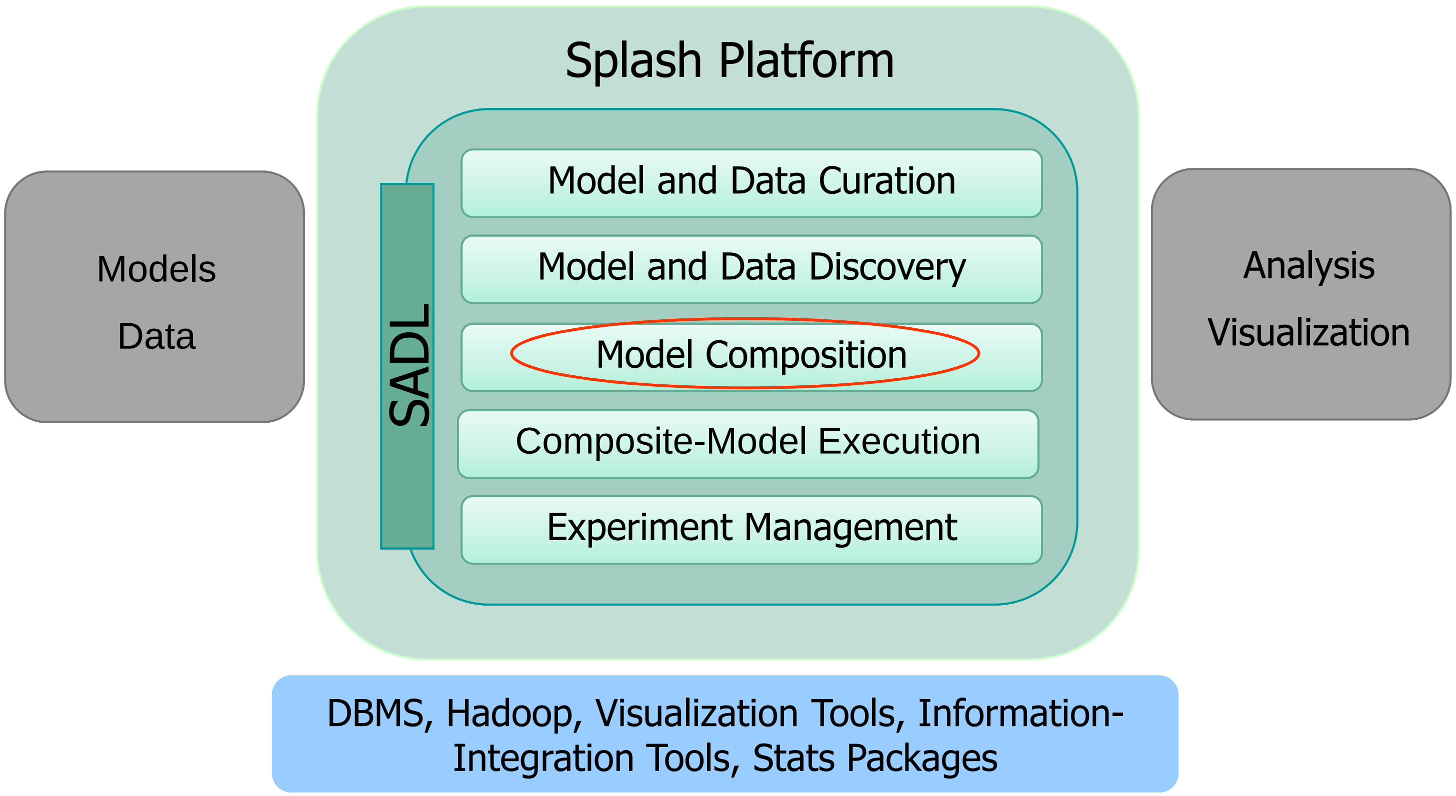


Model Wizard offers step by step guidance to generate the Model's SADL, and the command line for invocation

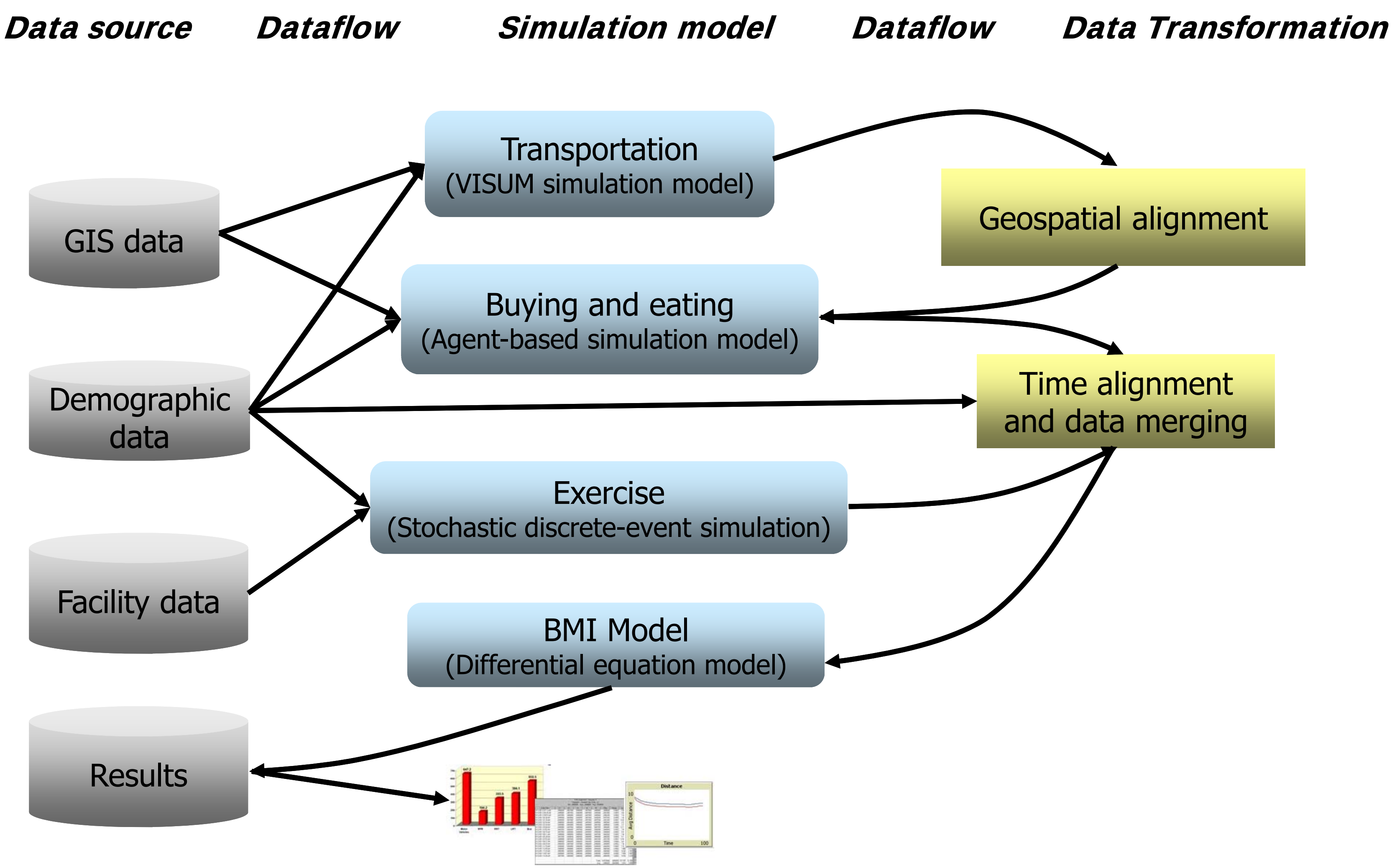


Data Wizard generates SADL for model input and output files

Model Composition

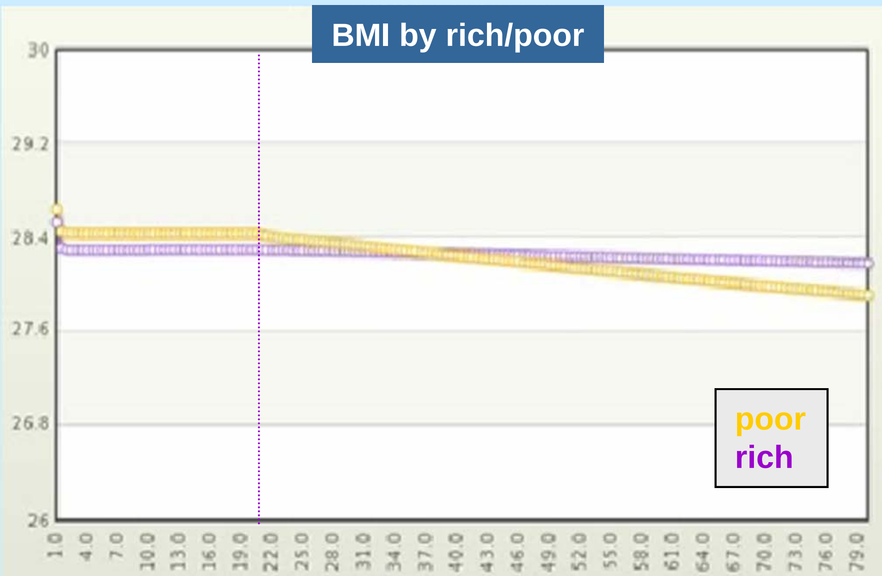


Obesity Example

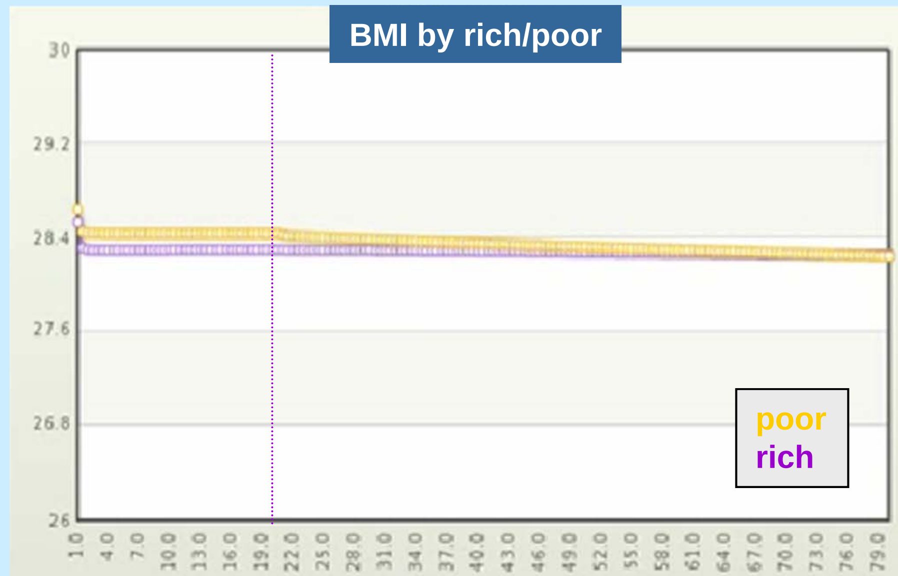


Sample Results

If we open a new “healthy” food store in a “bad” neighborhood…



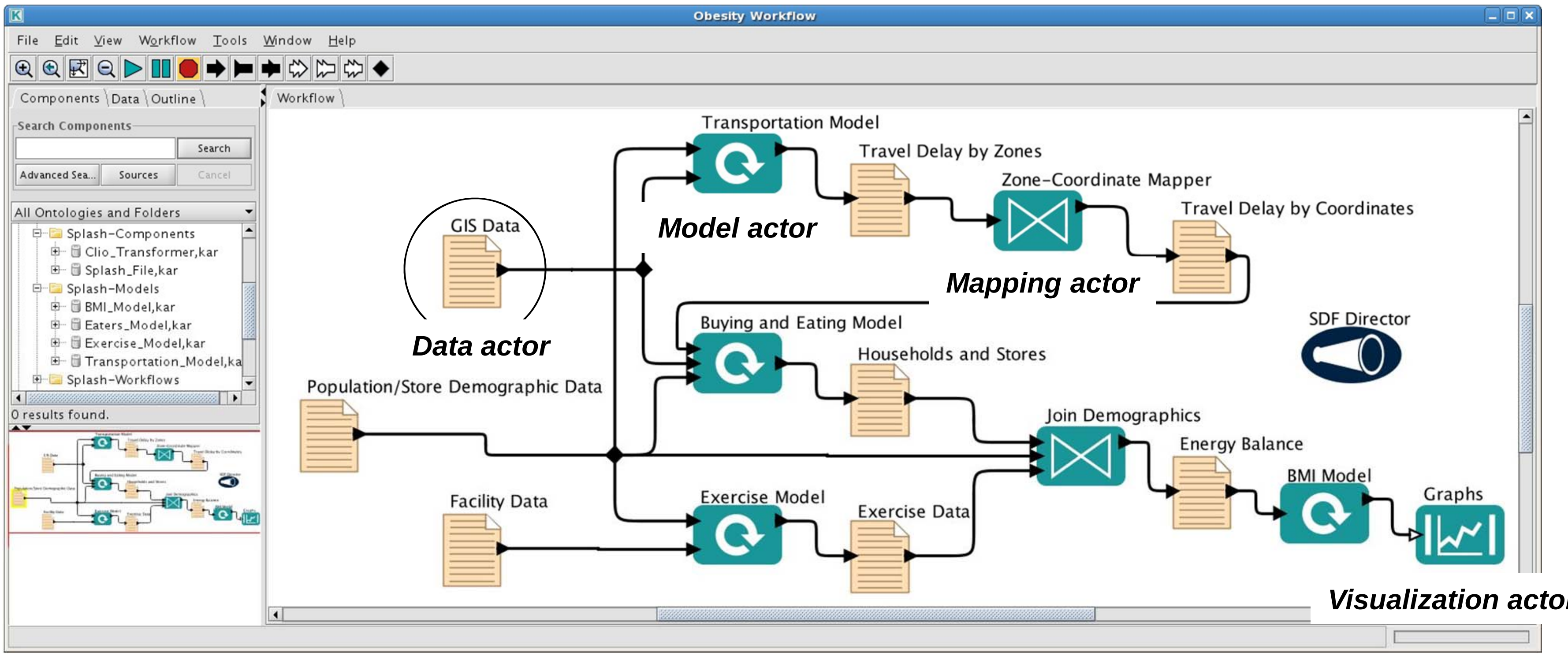
Without traffic model



Including traffic model

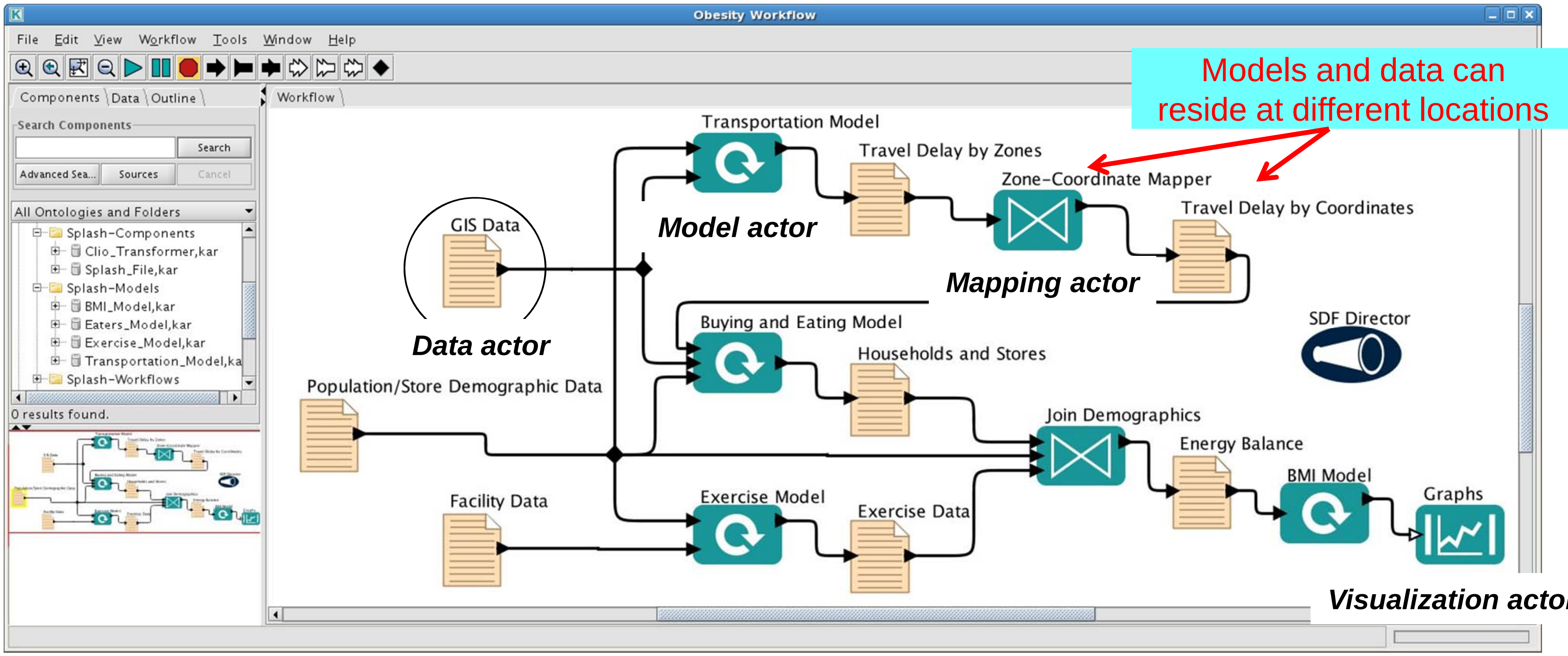
* Many assumptions, sample only, your mileage may vary …

Implemented Obesity Example



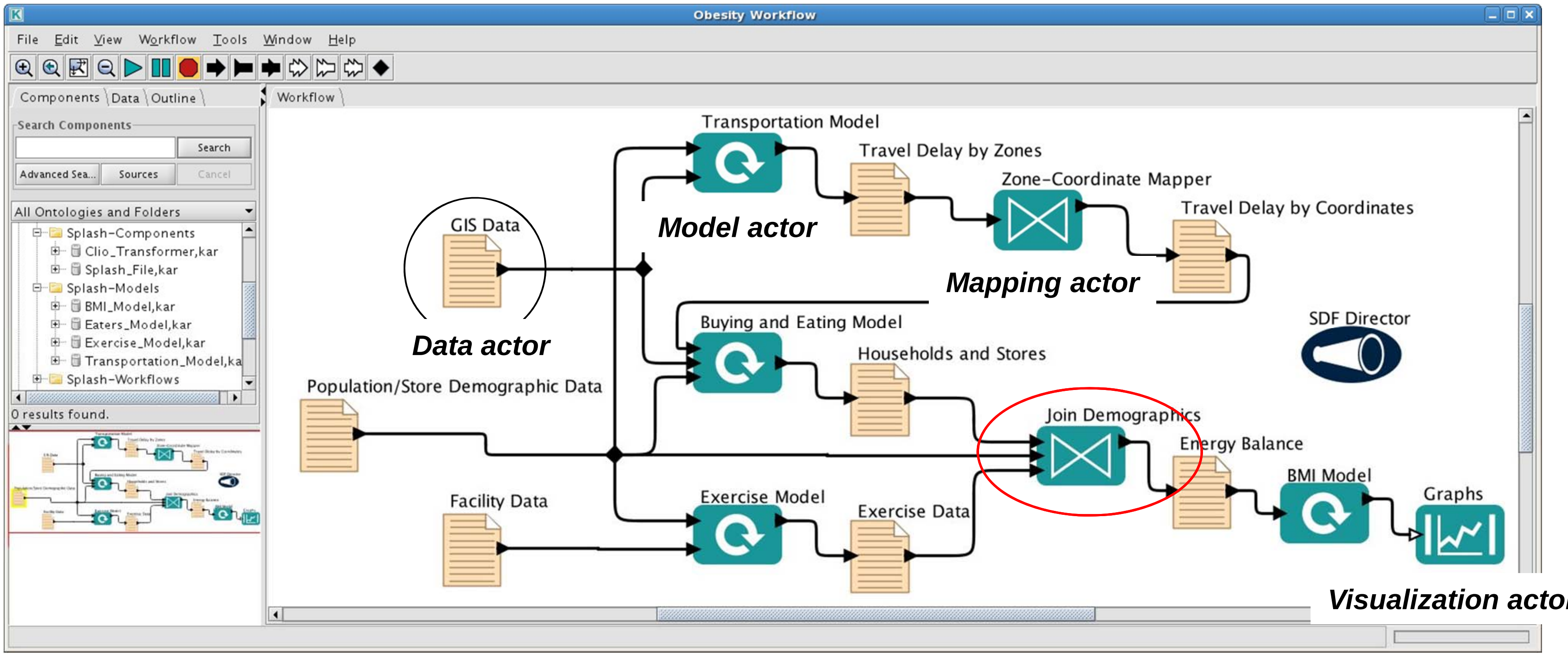
- ***Data actors:*** input and output files, databases, web services, etc.
- ***Model actors:*** simulation, optimization, statistical models
- ***Mapping actors:*** data transformations, time and space alignment
- ***Visualization actors:*** graphs, reports, etc.

Implemented Obesity Example



- ***Data actors:*** input and output files, databases, web services, etc.
- ***Model actors:*** simulation, optimization, statistical models
- ***Mapping actors:*** data transformations, time and space alignment
- ***Visualization actors:*** graphs, reports, etc.

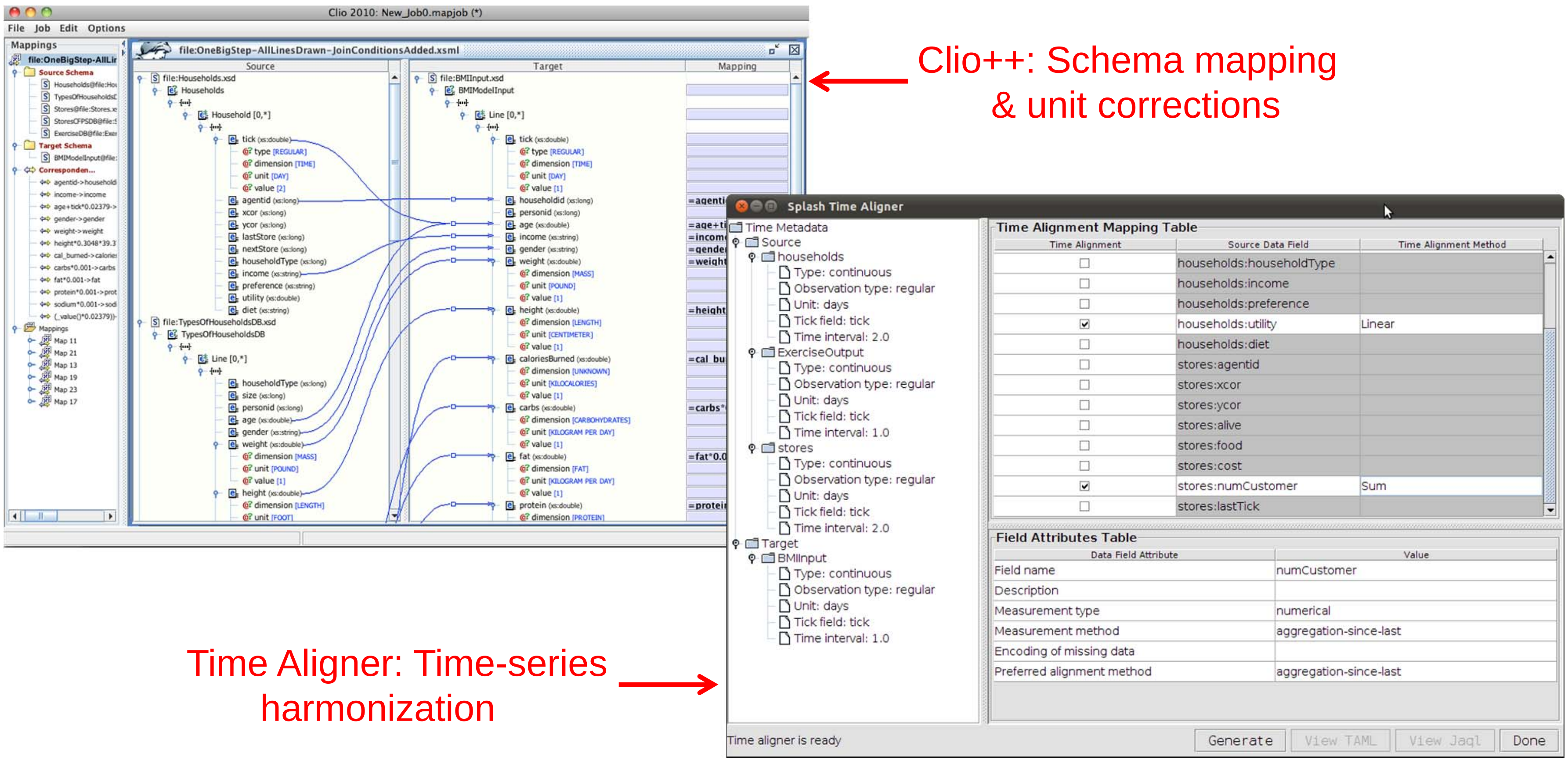
Implemented Obesity Example



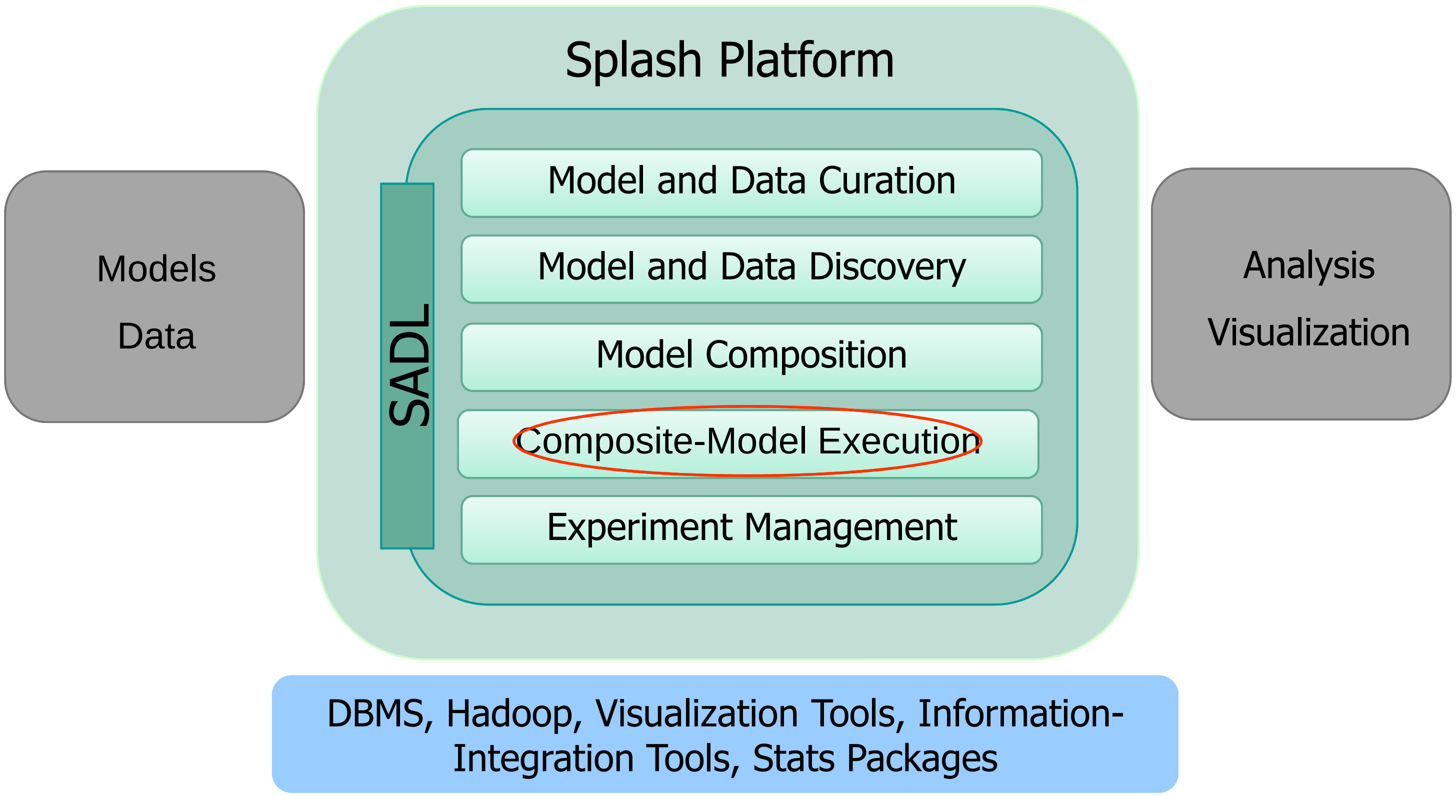
- ***Data actors:*** input and output files, databases, web services, etc.
- ***Model actors:*** simulation, optimization, statistical models
- ***Mapping actors:*** data transformations, time and space alignment
- ***Visualization actors:*** graphs, reports, etc.

Data Transformations Between Models

- Transformation design tools for structural (schema) and time alignments
- SADL metadata used to automatically detect mismatches
- Splash generates code for massive-scale transformation on Hadoop at simulation time



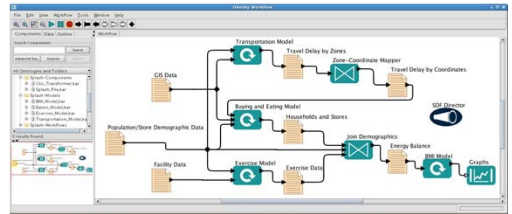
Composite-Model Execution



Executing a Composite Model: The Need for Runtime Efficiency

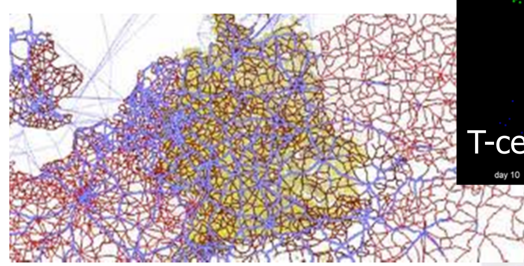
A huge parameter space to explore (many model runs)

- Ex: 3 models + 10 params/model + 2 vals/param = over 1 billion model runs
- Even worse for stochastic models (multiple Monte Carlo replications)
- Experimental design can help

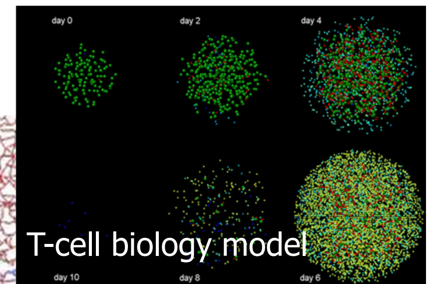


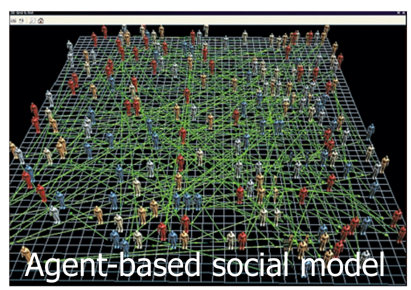
Each model run can be extremely time consuming

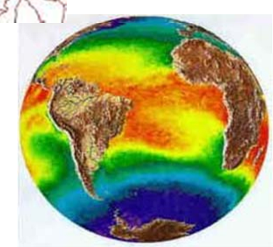
- Large-scale, high resolution models produce and consume massive amounts of time-series and other data
- CPU-intensive computations
- Composing models (with data transformations) intensifies the problem



Regional traffic model

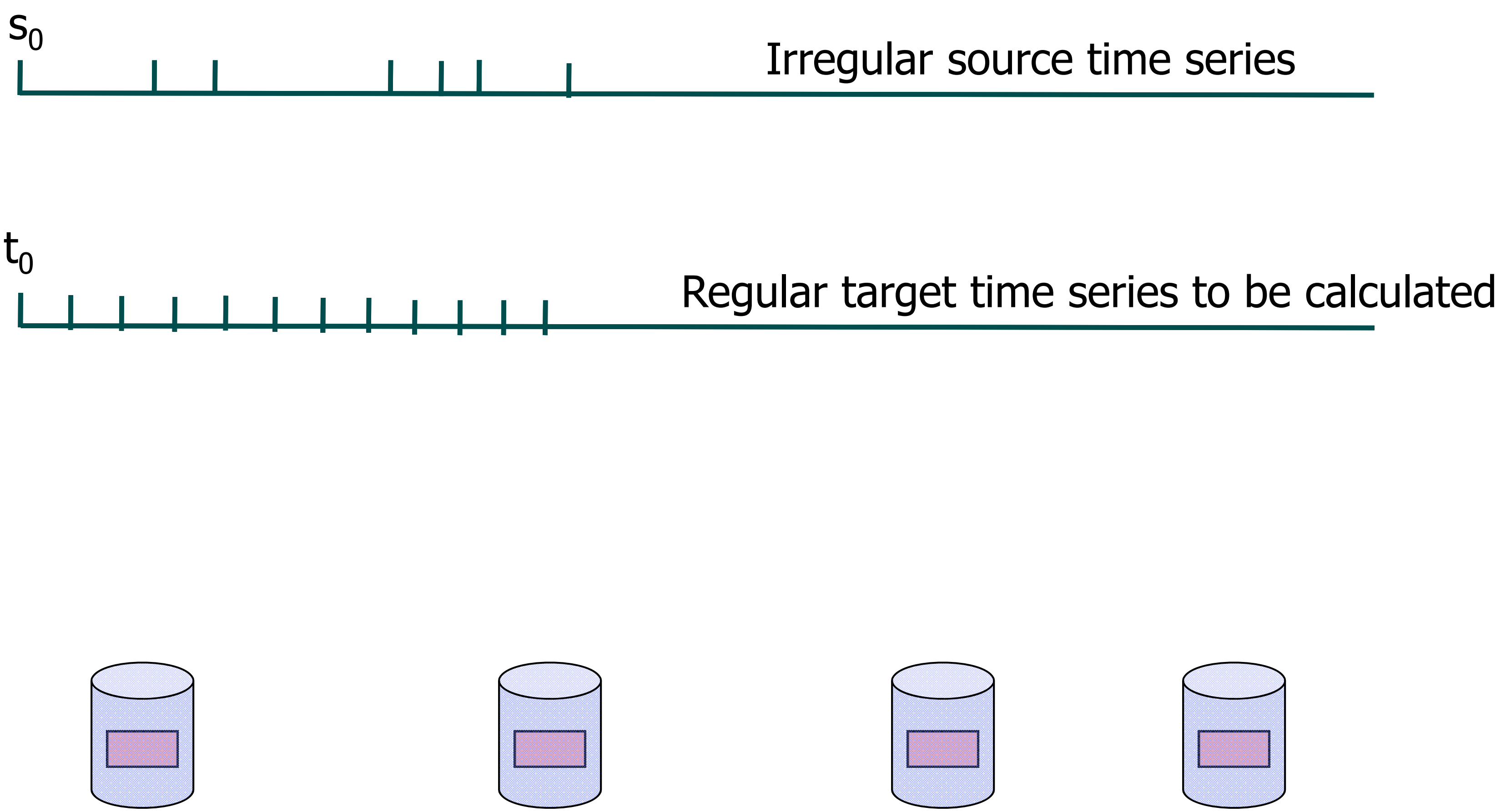






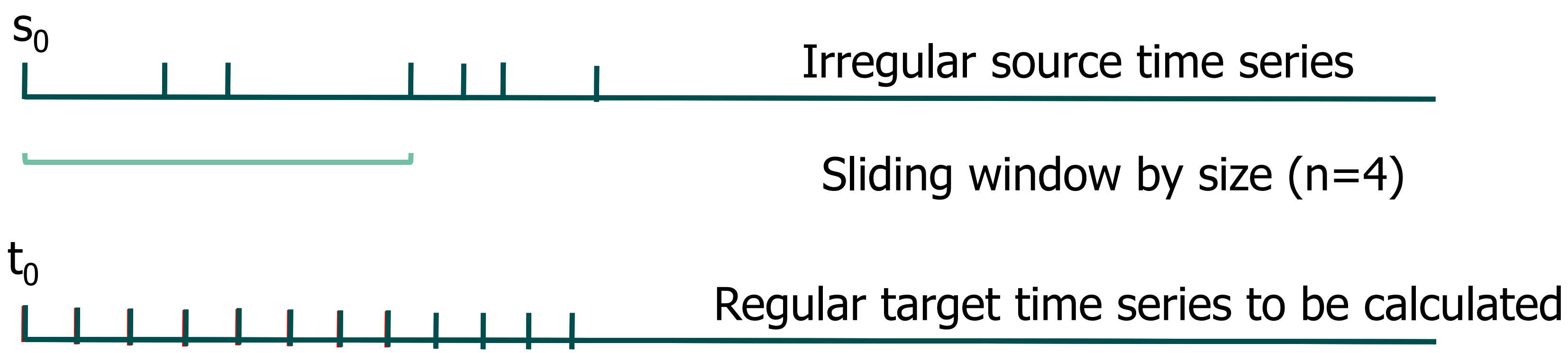
NCAR Community Atmosphere Model (CAM)

Time alignment with MapReduce



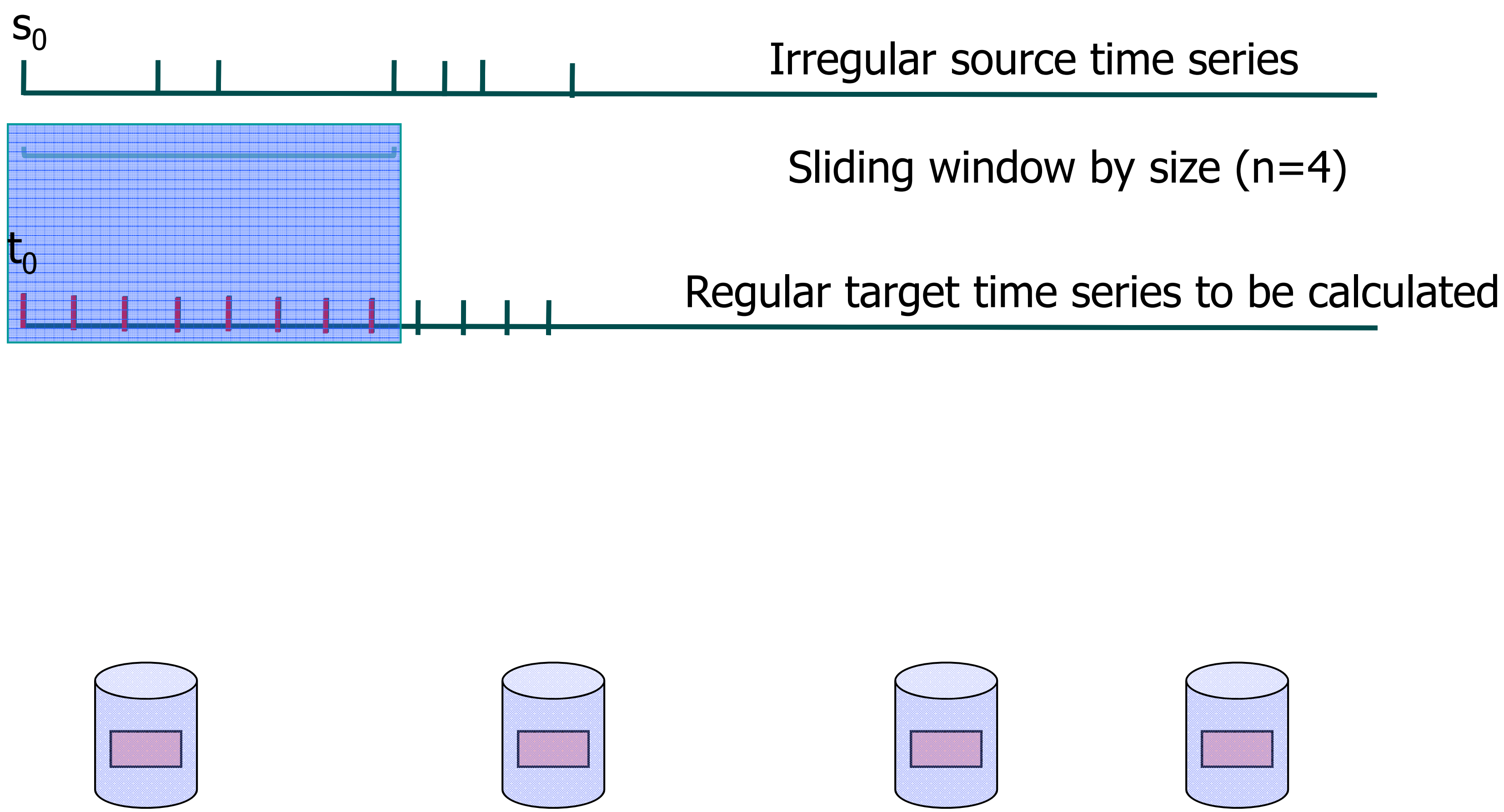
Interpolation, nearest neighbor, aggregation (since-last, since-start)

Time alignment with MapReduce



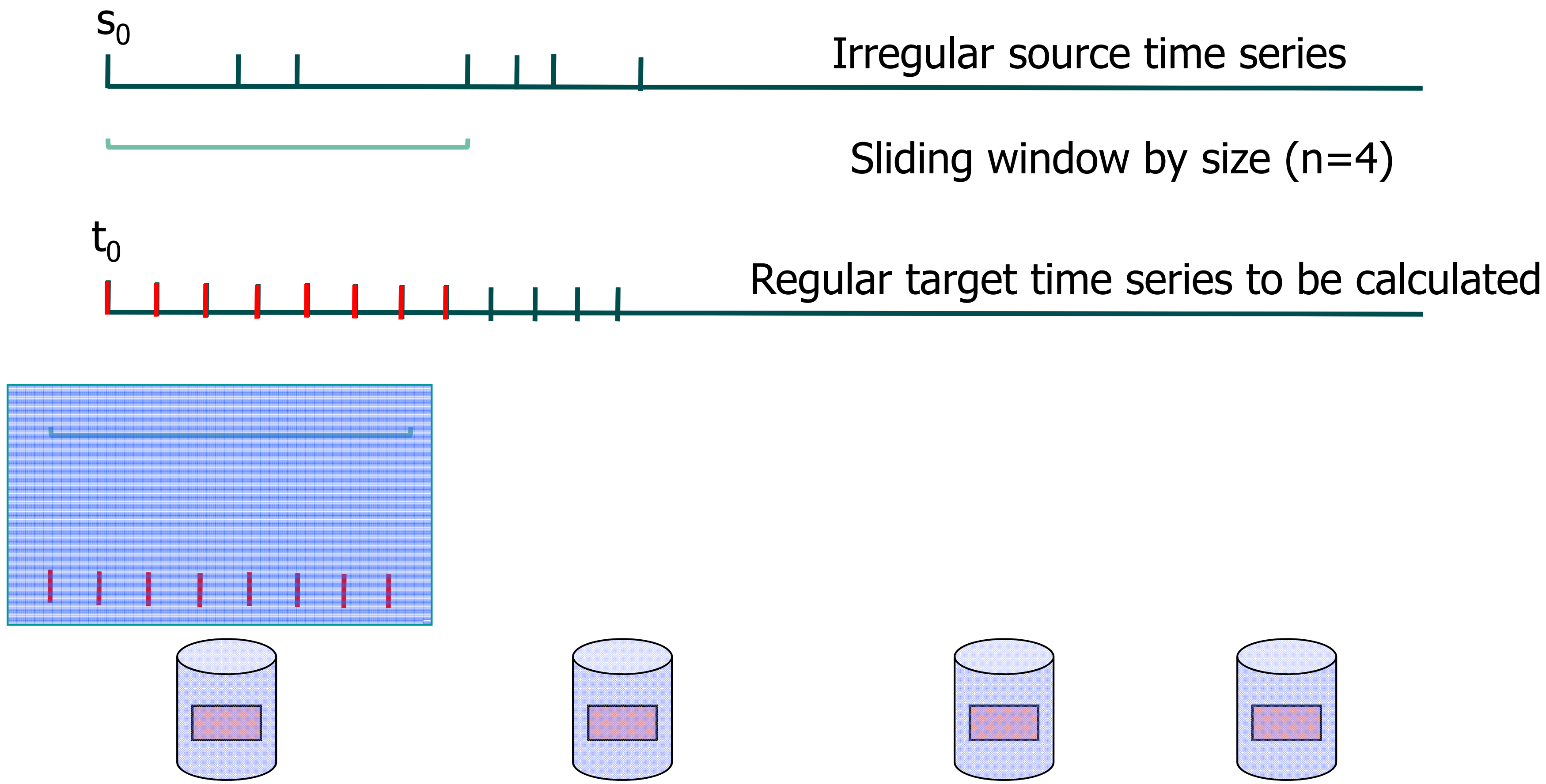
Interpolation, nearest neighbor, aggregation (since-last, since-start)

Time alignment with MapReduce



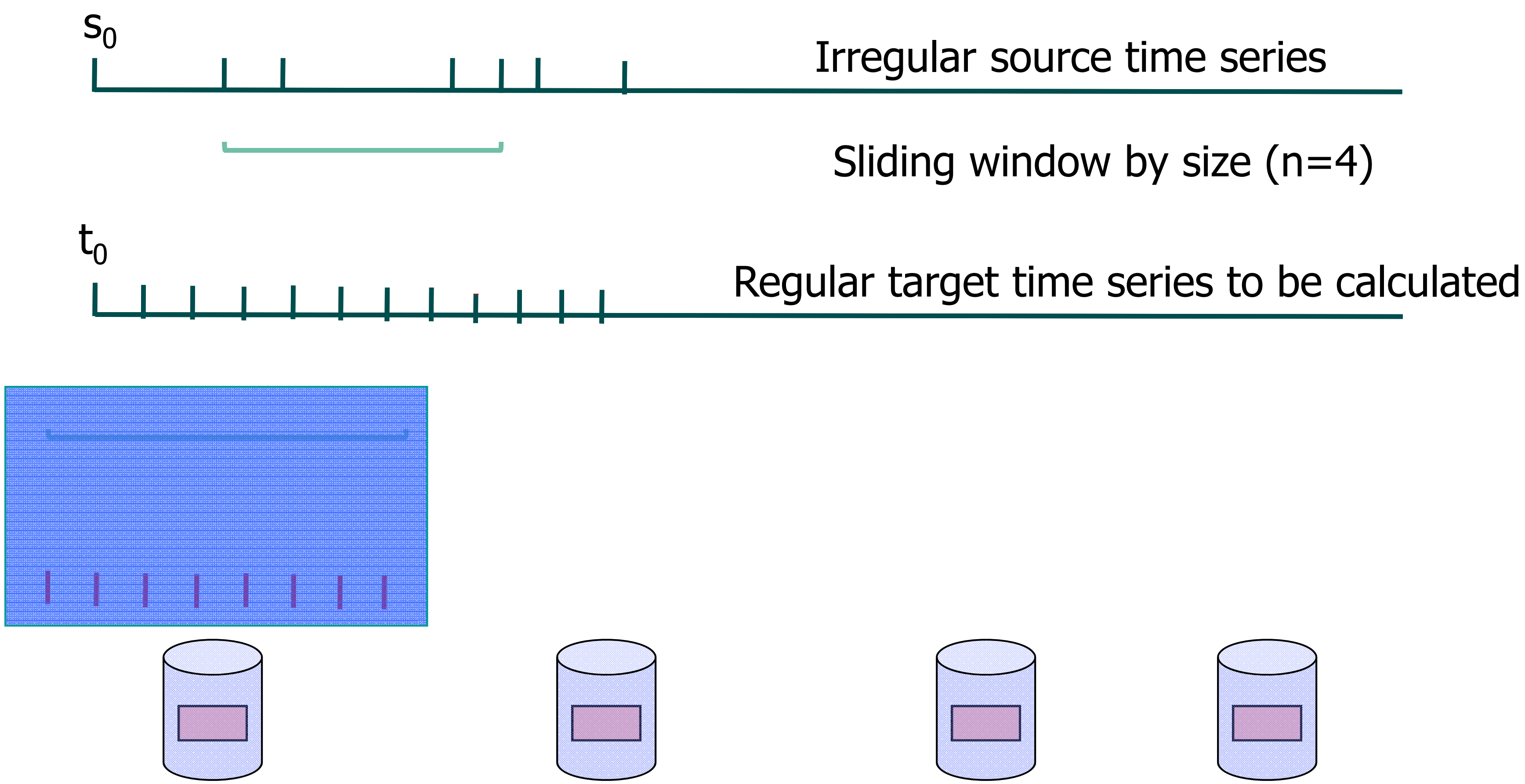
Interpolation, nearest neighbor, aggregation (since-last, since-start)

Time alignment with MapReduce



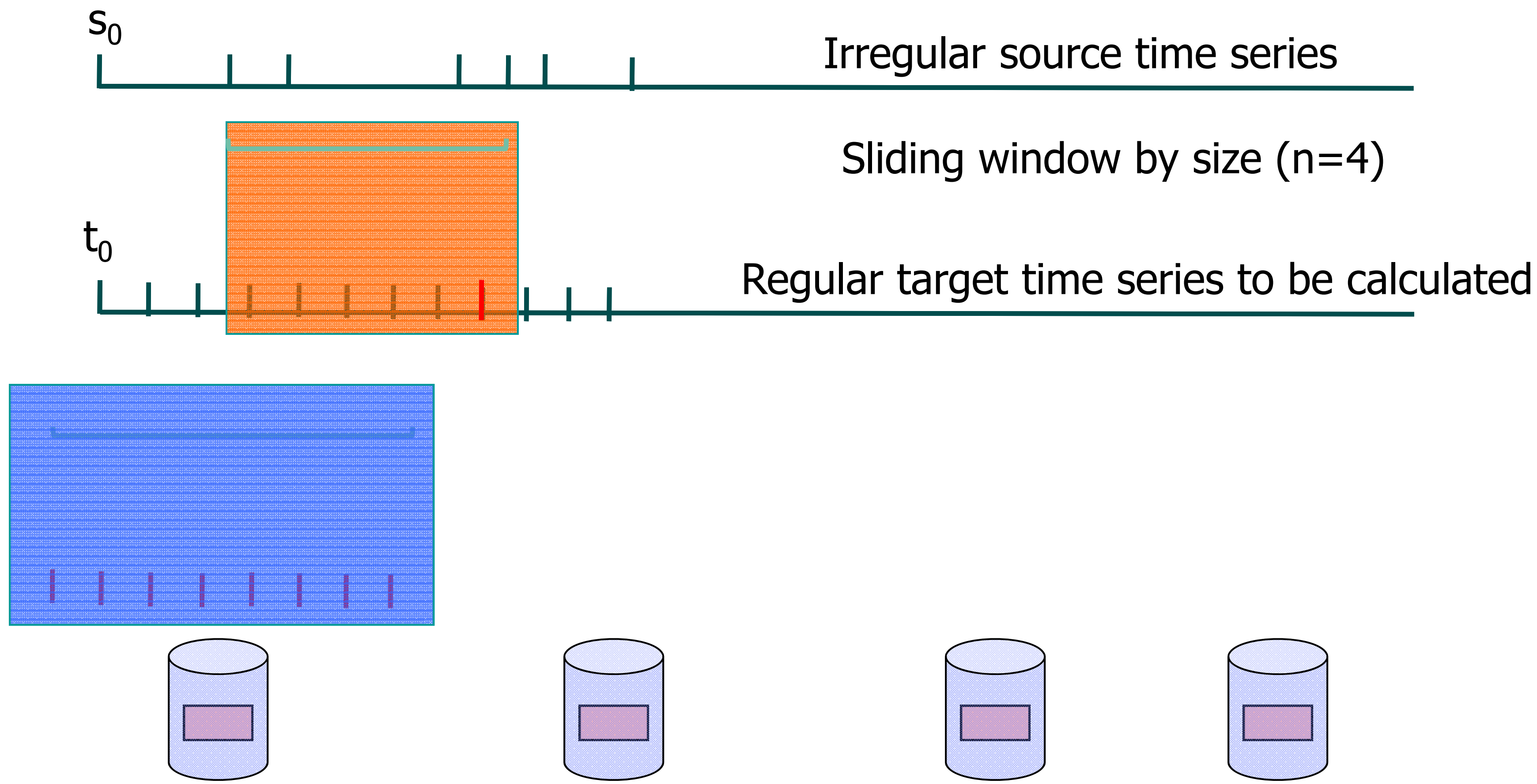
Interpolation, nearest neighbor, aggregation (since-last, since-start)

Time alignment with MapReduce



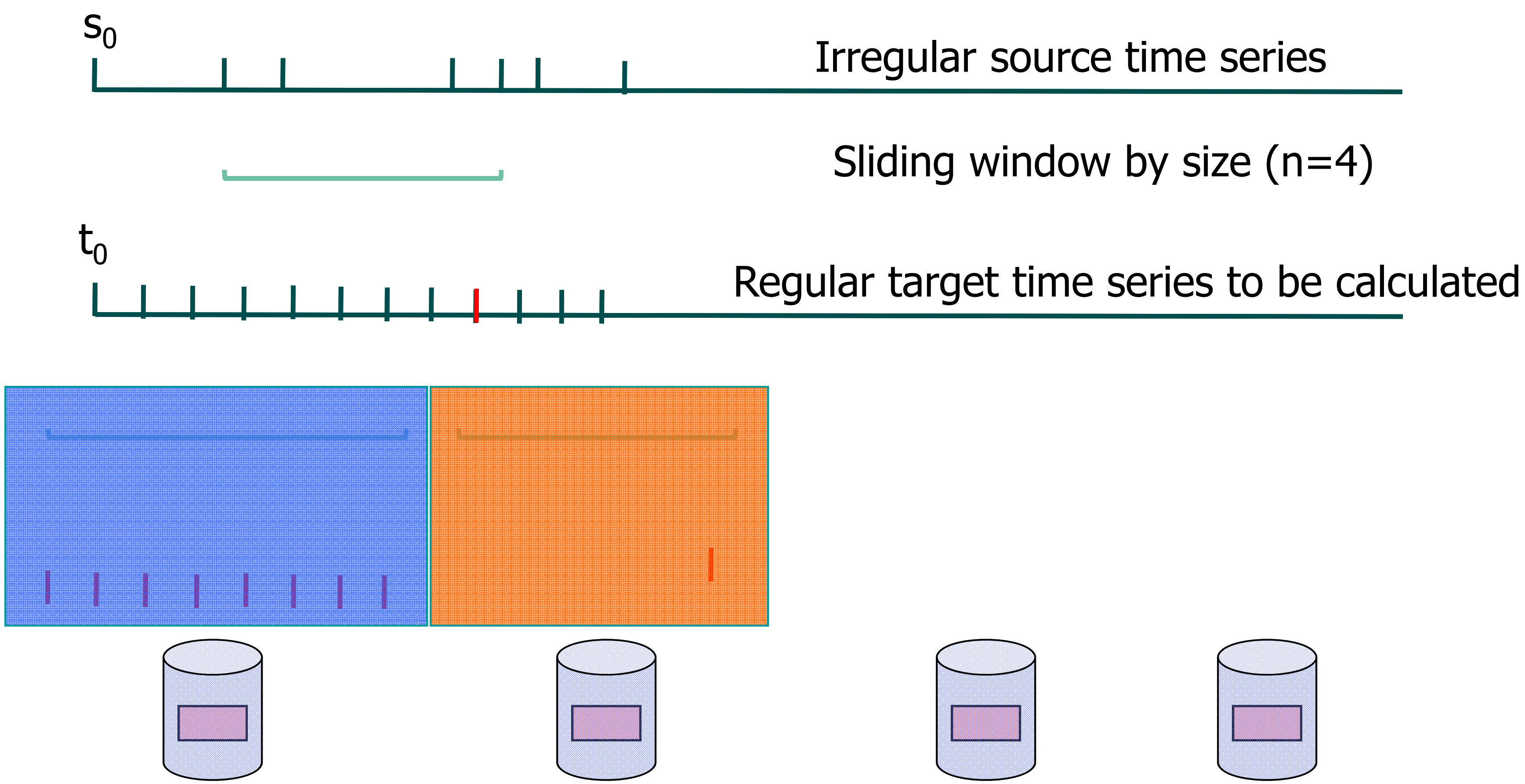
Interpolation, nearest neighbor, aggregation (since-last, since-start)

Time alignment with MapReduce



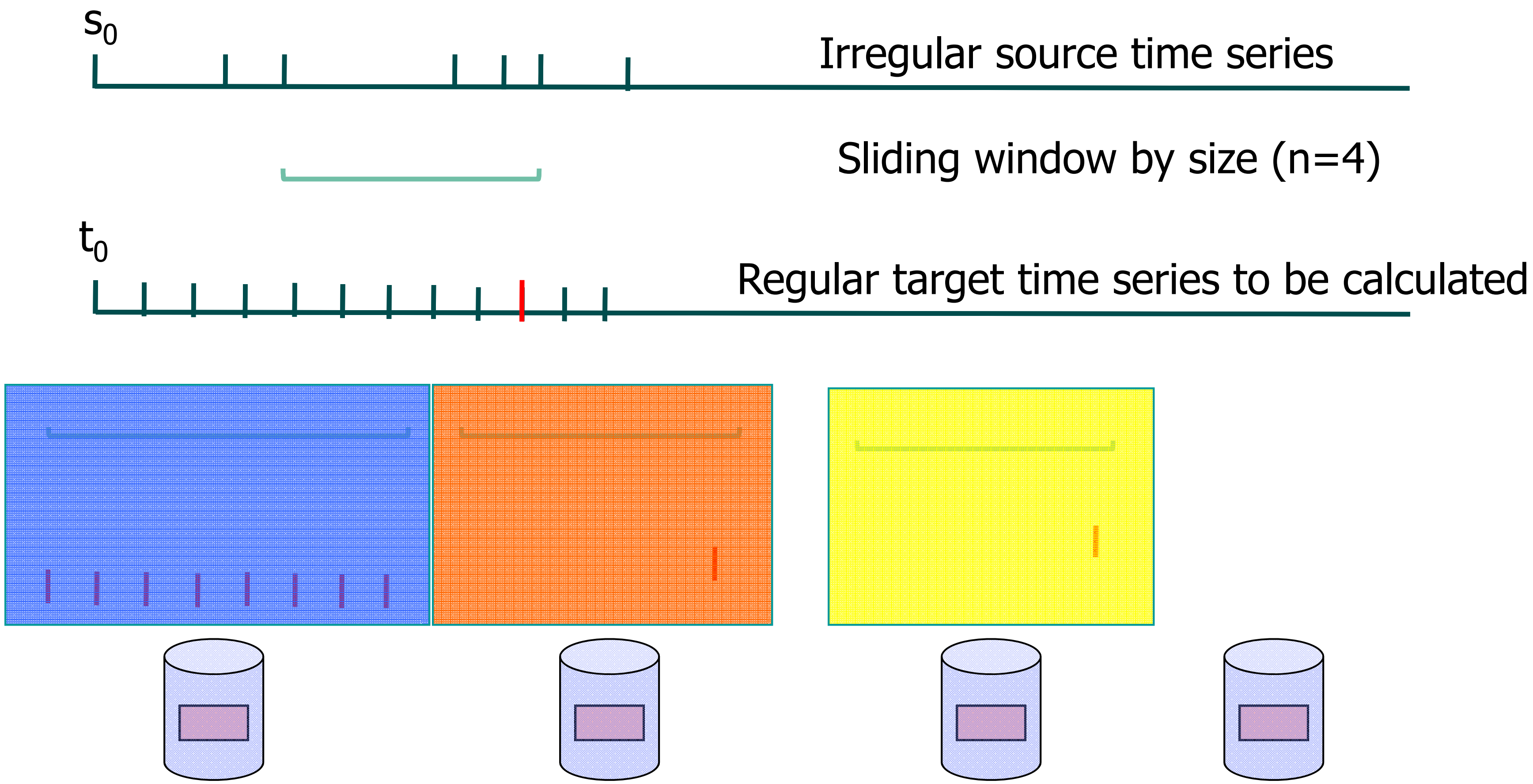
Interpolation, nearest neighbor, aggregation (since-last, since-start)

Time alignment with MapReduce



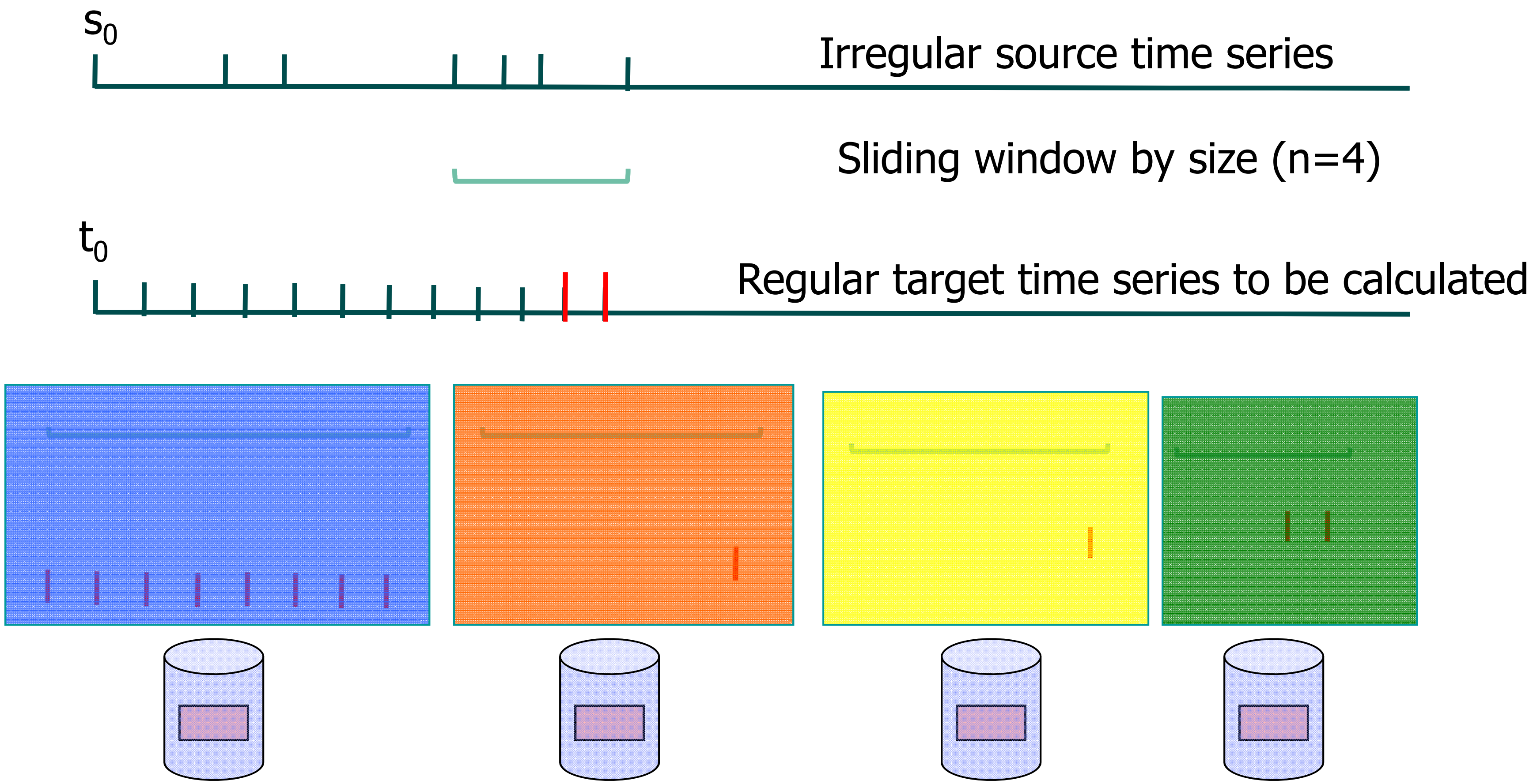
Interpolation, nearest neighbor, aggregation (since-last, since-start)

Time alignment with MapReduce



Interpolation, nearest neighbor, aggregation (since-last, since-start)

Time alignment with MapReduce

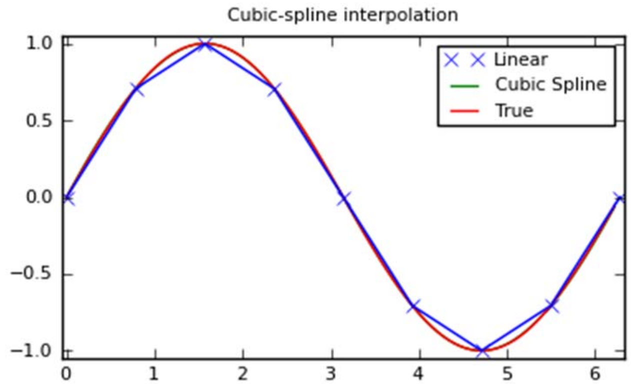


Interpolation, nearest neighbor, aggregation (since-last, since-start)

Cubic-Spline Interpolation in MapReduce

- **Recall:** Source outputs 1 tick per two days; target needs one tick per day

- **(Natural) cubic spline** widely used
  - Uniformly approximates f and f'
  - Error of $O(h^4)$ as knot spacing $h \to 0$
  - Default method in SAS



- Given **source** and **target** time series:

$$S = \langle (s_0, d_0), (s_1, d_1), \ldots, (s_m, d_m) \rangle \text{ and } T = \langle (t_0, \tilde{d}_0), (t_1, \tilde{d}_1), \ldots, (t_n, \tilde{d}_n) \rangle$$

- Given **window** W_i for t_i: $W_i = \langle (s_j, d_j, \sigma_j), (s_{j+1}, d_{j+1}, \sigma_{j+1}) \rangle$ where $[s_j, s_{j+1})$ contains t_i

$$\tilde{d}_i = f(W_i) = \frac{\sigma_j}{6h_j}(s_{j+1} - t_i)^3 + \frac{\sigma_{j+1}}{6h_j}(t_i - s_j)^3 + \left(\frac{d_{j+1}}{h_j} - \frac{\sigma_{j+1} h_j}{6} \right)(t_i - s_j) + \left(\frac{d_j}{h_j} - \frac{\sigma_j h_j}{6} \right)(s_{j+1} - t_i)$$

$$h_j = s_{j+1} - s_j$$

Question: How to Compute Spline Constants?

- Must **solve** $Ax = b$ (m-1 rows and columns):

$$A = \begin{pmatrix} \frac{h_0 + h_1}{3} & \frac{h_1}{6} & 0 & \cdots & 0 & 0 & 0 \\ \frac{h_1}{6} & \frac{h_1 + h_2}{3} & \frac{h_2}{6} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{h_{m-3}}{6} & \frac{h_{m-3} + h_{m-2}}{3} & \frac{h_{m-2}}{6} \\ 0 & 0 & 0 & \cdots & 0 & \frac{h_{m-2}}{6} & \frac{h_{m-2} + h_{m-1}}{3} \end{pmatrix} \qquad b = \begin{pmatrix} \frac{d_2 - d_1}{h_1} - \frac{d_1 - d_0}{h_0} \\ \frac{d_3 - d_2}{h_2} - \frac{d_2 - d_1}{h_1} \\ \vdots \\ \frac{d_m - d_{m-1}}{h_{m-1}} - \frac{d_{m-1} - d_{m-2}}{h_{m-2}} \end{pmatrix}$$

- **Prior work**
  - Some solutions require **evenly spaced** source points
  - Some solutions require **precomputation** (somehow) of A^{-1}
  - Other solutions for **vector machines**, **MPI architectures**, **GPUs**
    - Require a lot of **data shuffling** (reduce steps) in Hadoop adaptation
    - Example: **Parallel Cyclic Reduction (PCR)** uses $\log_2 m$ map-reduce jobs

- **Our approach: minimize** $L(x) = \|Ax - b\|_2^2 = \sum_i (A_{i.}x - b_i)^2 = \sum_i L_i(x)$

Our Solution: Distributed Stochastic Gradient Descent (DSGD)

- Originally for **matrix completion,** e.g., Netflix ratings problem [GHS KDD11]

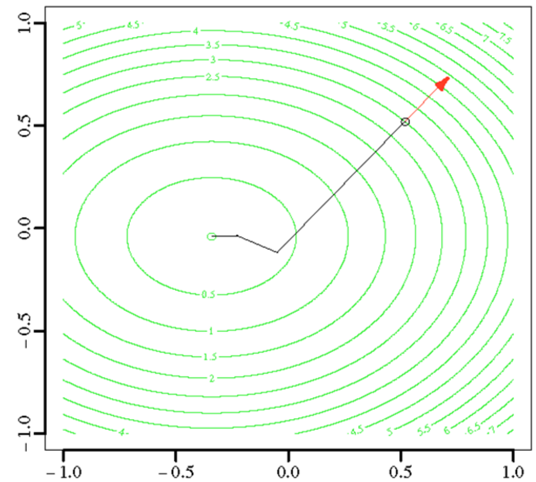
- Uses **stochastic gradient descent (SGD)** to minimize L
  - **Deterministic** gradient descent (**DGD**): $x^{(n+1)} = x^{(n)} - \varepsilon_n L'(x^{(n)})$

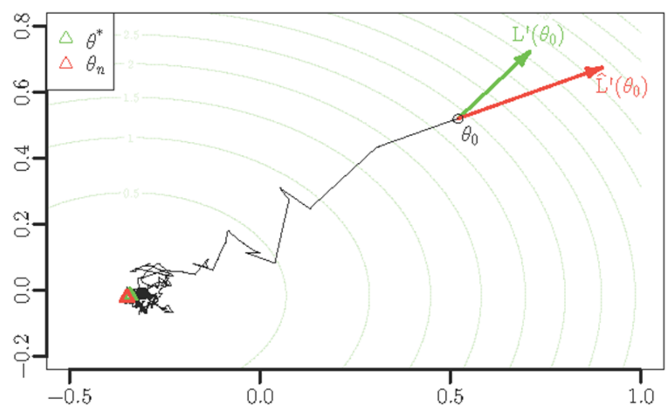
    where $L'(x^{(n)}) = \sum_{i=1}^{m-1} L_i'(x^{(n)})$
  - **Stochastic** gradient descent: $x^{(n+1)} = x^{(n)} - \varepsilon_n \hat{L}'(x^{(n)})$

    where $\hat{L}'(x^{(n)}) = (m-1)L_I'(x^{(n)})$

    and I is randomly chosen from $[1..m-1]$
  - Avoids getting stuck at local minima
  - **Problem:** SGD is not a parallel algorithm

- **Idea:** run SGD on subsets (strata) of rows, randomly switch strata; choose "sparse" strata that allow parallel execution of SGD
  - Converges to overall solution with probability 1 under mild conditions





Choosing Strata

Goal: Permit parallel execution of SGD within each stratum

Key observation: $L'_i(x) = \begin{pmatrix} 0 & \ldots & 0 & u_{i,i-1} & u_{i,i} & u_{i,i+1} & 0 & \ldots & 0 \end{pmatrix}$

Updating x_i only affects (and is affected by) x_{i-1} and x_{i+1}

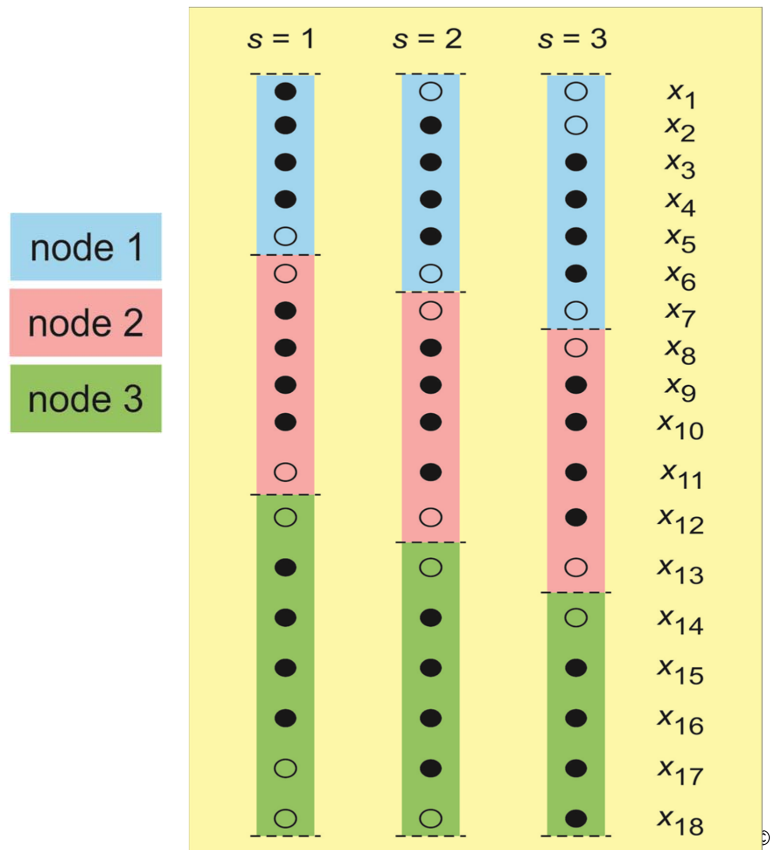
where $u_{i,j} = 2a_{i,j}(a_{i,i-1}x_{i-1} + a_{i,i}x_i + a_{i,i+1}x_{i+1})$

Stratum choice:

- Can implement as map-only Hadoop job (almost no data shuffling)
- Exploit discrepancy between logical splits and physical blocks

Empirical study:

- 2x-3x faster than best-of-breed PCR alg.
- 10 scans vs log m for PCR
- PCR requires extra sort
- PCR requires massive data shuffling (network bottleneck)



Speeding up Composite Simulations: Result Caching

Motivating example: Two models in series, 100 reps

Model 1 → Model 2

Deterministic   Stochastic

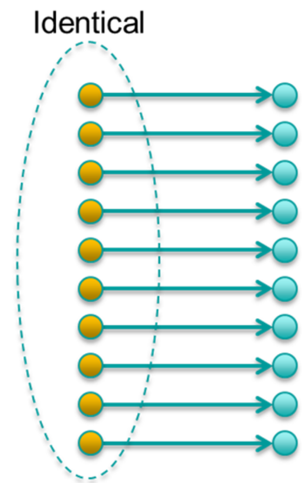
- Naïve approach: execute composite model (i.e., Models 1 & 2) 100 times

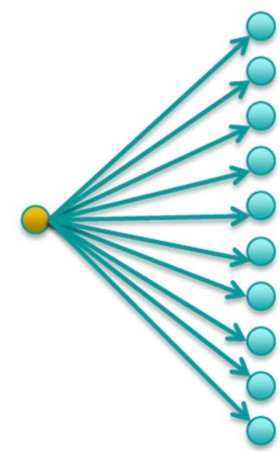
- A better approach:

Model 1 → Cache → Model 2

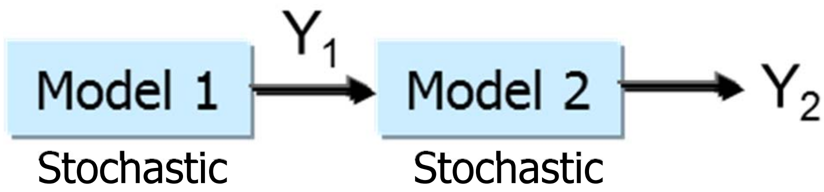
  – Execute Model 1 once and cache result
  – Read from cache when executing Model 2

Question: Can result-caching idea be generalized?

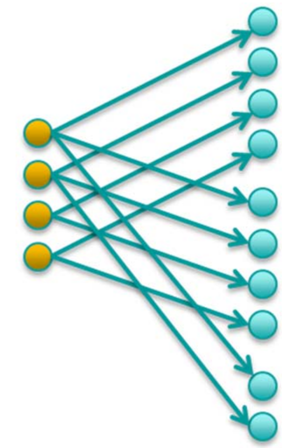




General Method for Two Stochastic Models in Series



Goal: Estimate $\theta = E[Y_2]$ based on n replications



Result-caching approach:

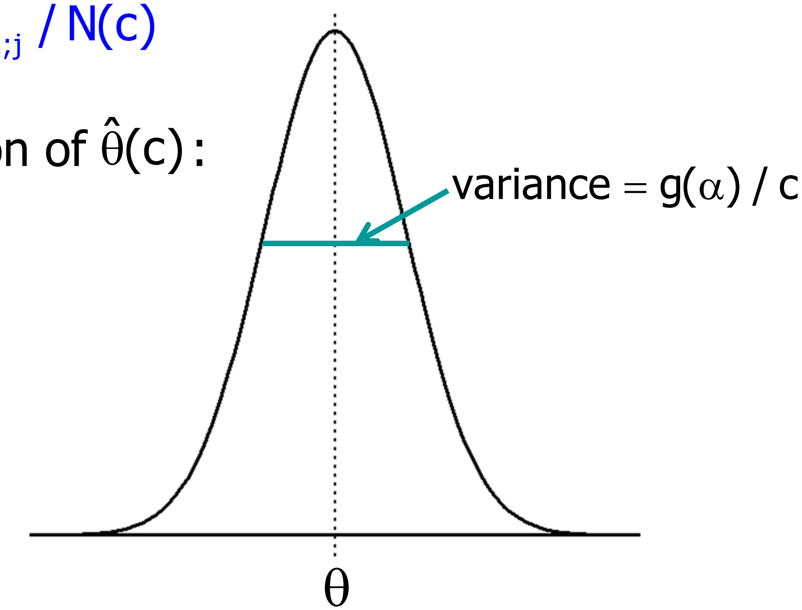
1. Set $m_n = \lceil \alpha n \rceil$ for some $\alpha \in (0,1]$ (the re-use factor)
2. Generate m_n outputs from Model 1 and cache them
3. To execute Model 2, cycle through Model 1 outputs
4. Estimate θ by $\theta_n = \sum_{i=1}^{n} Y_{2;i} / n$

Ex: $n=10$, $m_n = 4$

Optimizing the Re-Use Factor for Maximum Efficiency

Q: How to trade off cost and precision?

- Assume a (large) fixed computational budget c
- Random cost model: correlated pair (τ_i, Y_i)
  - $\tau_i =$ (random) cost of producing an observation Y_i
  - $N(c) =$ # of observations of Y_2 generated under c
  - $\hat{\theta}(c) = \sum_{j=1}^{N(c)} Y_{2;j} / N(c)$
- Approx. distribution of $\hat{\theta}(c)$:



$$g(\alpha) = (\alpha E[\tau_1] + E[\tau_2])\left\{Var[Y_2] + \left(2r_\alpha - \alpha r_\alpha(r_\alpha + 1)\right)Cov[Y_2, \tilde{Y}_2]\right\} \qquad r_\alpha = \lfloor 1/\alpha \rfloor$$

(cost per obs.) x (contributed variance per obs.)

The Optimal Re-Use Factor

Optimal solution

- Assume that $\mathrm{Cov}[Y_2, \tilde{Y}_2] \geq 0$
- Optimal value of α:

$$\alpha^* \approx \left(\frac{E[\tau_2] / E[\tau_1]}{\left(\mathrm{Var}[Y_2] / \mathrm{Cov}[Y_2, \tilde{Y}_2]\right) - 1} \right)^{1/2}$$

(truncate at 1/n or 1)

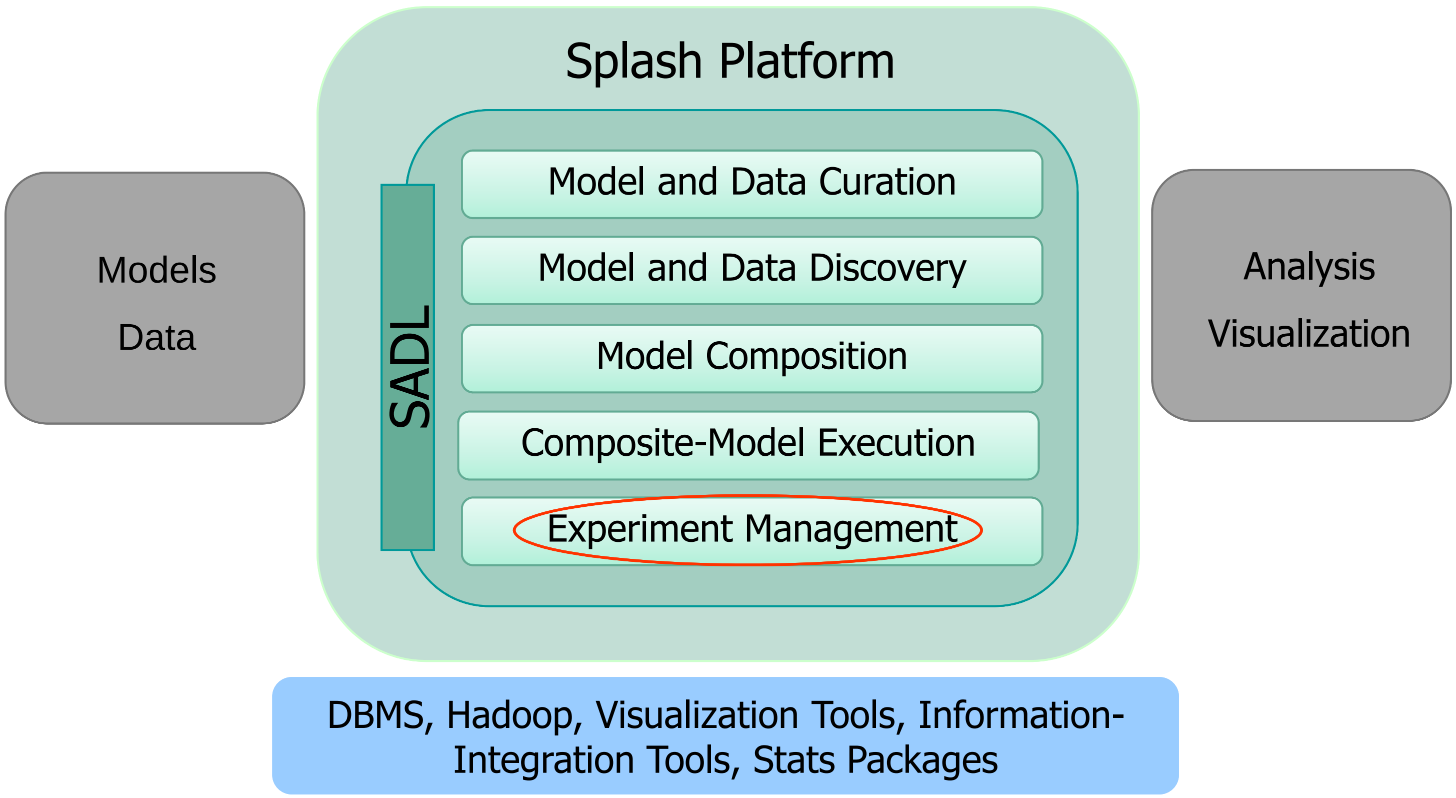
Observations

– If E[Model 1 cost] >> E[Model 2 cost], then high re-use of output

– If Model 2 insensitive to Model 1 (Cov << Var), then high re-use

– If Model 1 is deterministic (Cov = 0), then total re-use

Experiment Management (and Optimization)



Experiment Design and Efficiency

Trades off execution cost versus level of detail that can be estimated

Coarse resolution is OK for sensitivity analysis etc.

Resolution III design

Run	Factors						
	A	*B*	*C*	*D*=*AB*	*E*=*AC*	*F*=*BC*	*G*=*ABC*
def	-1	-1	-1	1	1	1	-1
afg	1	-1	-1	-1	-1	1	1
beg	-1	1	-1	-1	1	-1	1
abd	1	1	-1	1	-1	-1	-1
cdg	-1	-1	1	1	-1	-1	1
ace	1	-1	1	-1	1	-1	-1
bcf	-1	1	1	-1	-1	1	-1
abcdefg	1	1	1	1	1	1	1

Example: 1st-order polynomial metamodel for scaled data (7 factors)

$$Y = \beta_0 + \beta_1 x_1 + \cdots \beta_7 x_7$$
$$+ \beta_{1;2} x_1 x_2 + \cdots + \beta_{6;7} x_6 x_7 + \beta_{1;2;3} x_1 x_2 x_3 + \cdots + \text{noise}$$
$$x_1, \ldots, x_7 \in \{-1, 1\} \quad \text{(full factorial = 128 runs)}$$

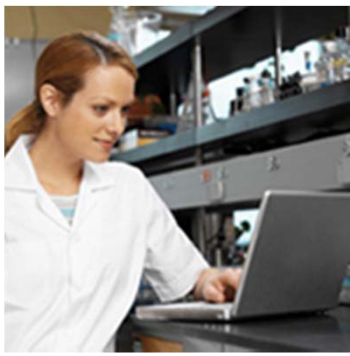
Fractional-factorial experimental designs

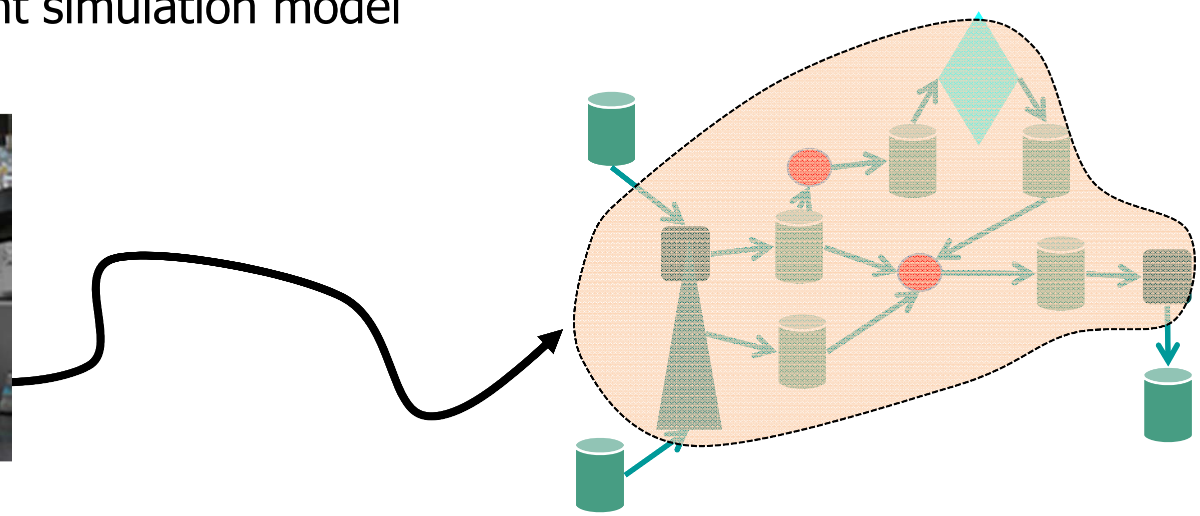
To estimate	If you can ignore	Resolution	# runs
Main effects	All high-order effects	III	8
Main effects	3rd-order and higher	IV	16
Main effects + 2-way interactions	3rd-order and higher	V	64

Running experiments in Splash

Goal

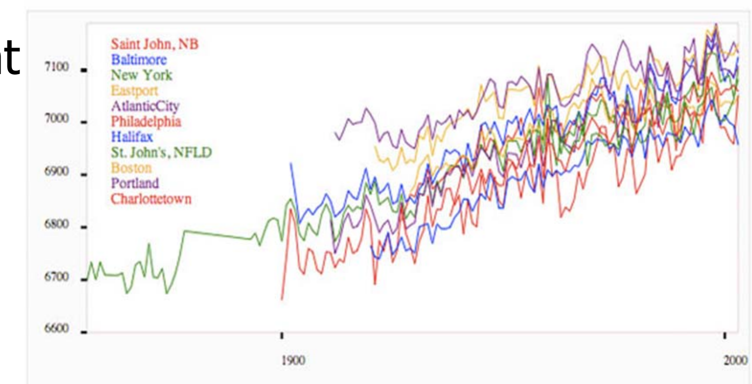
- Provide a facility that gives the illusion of executing **one** coherent simulation model





Main Challenges

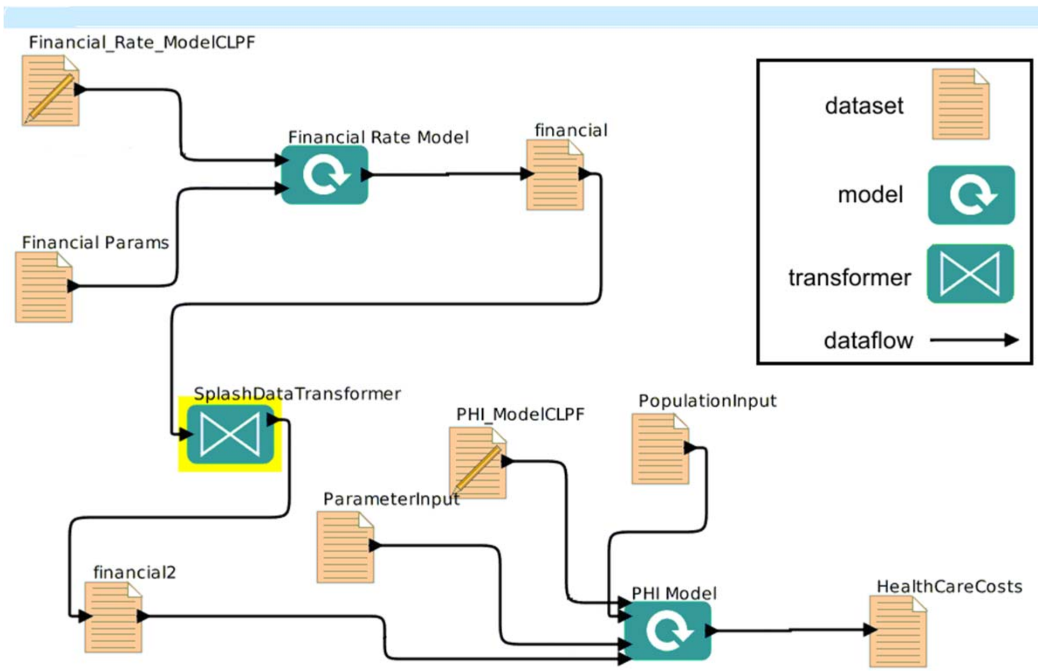
- Automate the coordination between experiment conditions and inputs to different submodels.
- Automate the combination of different replications of different submodels.



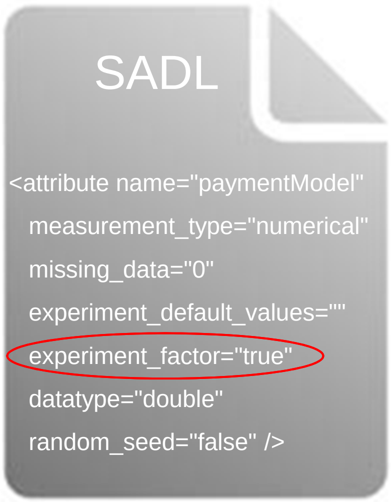
Example: Healthcare Payer Model

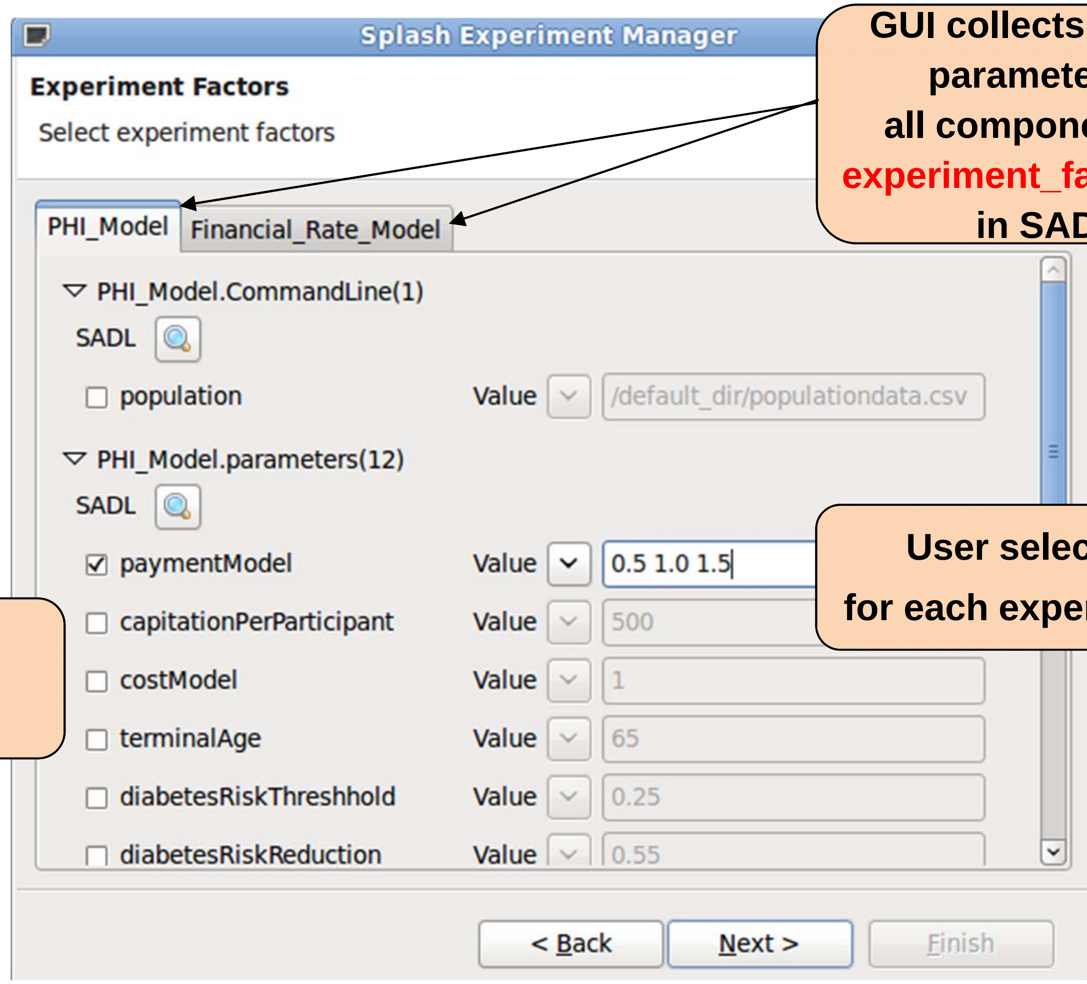
Composition of two models

- Emory/Georgia Tech Predictive Health Institute model [Park et al. 2012]
  - Simple agent-based model of prevention and wellness program
  - For investigation of payment systems (capitated vs outcome-based)
- Simple logarithmic random walk model of interest & inflation rates



Experiment Manager (Specifying Experimental Factors)



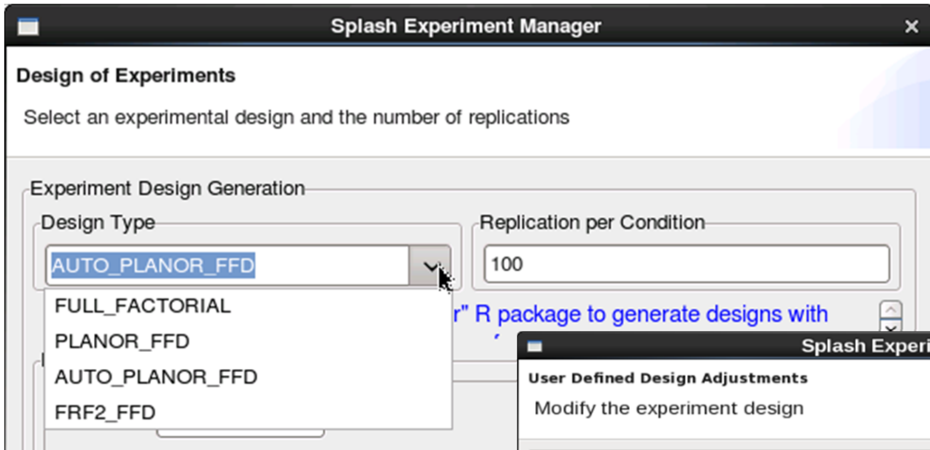


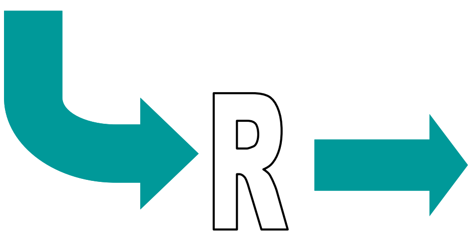
GUI collects simulation parameters from all component models experiment_factor = TRUE in SADL file

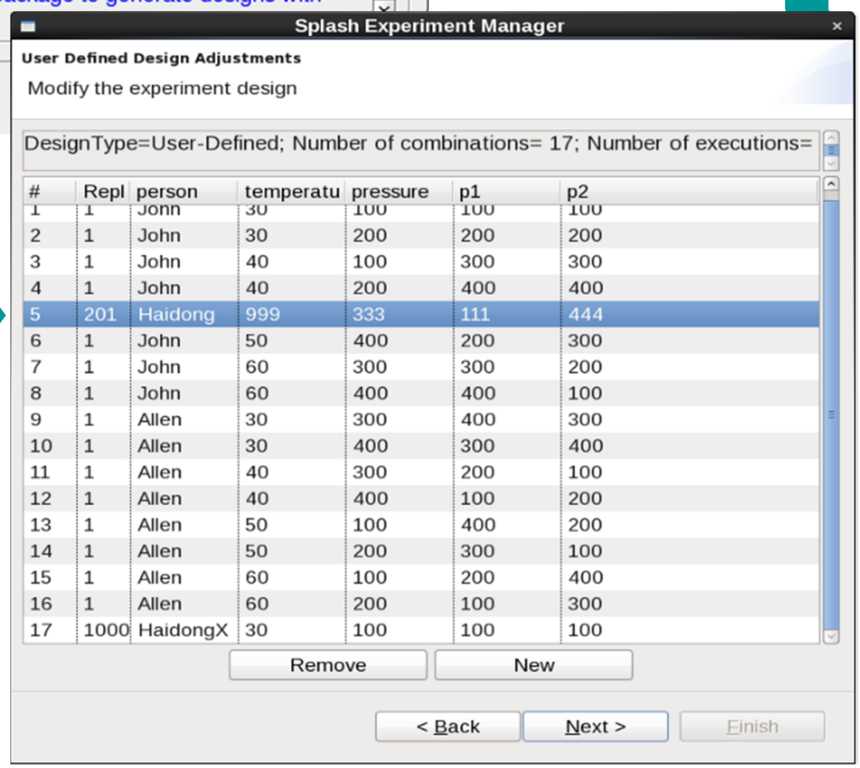
User selects values for each experiment factor

User selects subset of parameters as experiment factors

Experiment Design in Splash





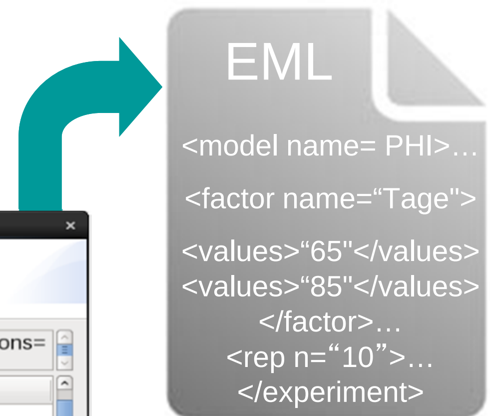


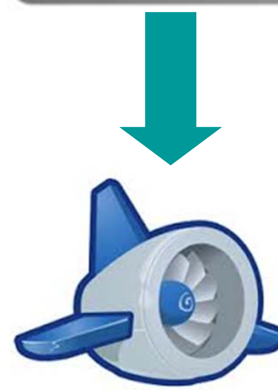
#	Repl	person	temperatu	pressure	p1	p2
1	1	John	30	100	100	100
2	1	John	30	200	200	200
3	1	John	40	100	300	300
4	1	John	40	200	400	400
5	201	Haidong	999	333	111	444
6	1	John	50	400	200	300
7	1	John	60	300	300	200
8	1	John	60	400	400	100
9	1	Allen	30	300	400	300
10	1	Allen	30	400	300	400
11	1	Allen	40	300	200	100
12	1	Allen	40	400	100	200
13	1	Allen	50	100	400	200
14	1	Allen	50	200	300	100
15	1	Allen	60	100	200	400
16	1	Allen	60	200	100	300
17	1000	HaidongX	30	100	100	100

Editable design

(Factor values and
# of Monte Carlo reps
for each condition)

Design Persistence





Execution Engine

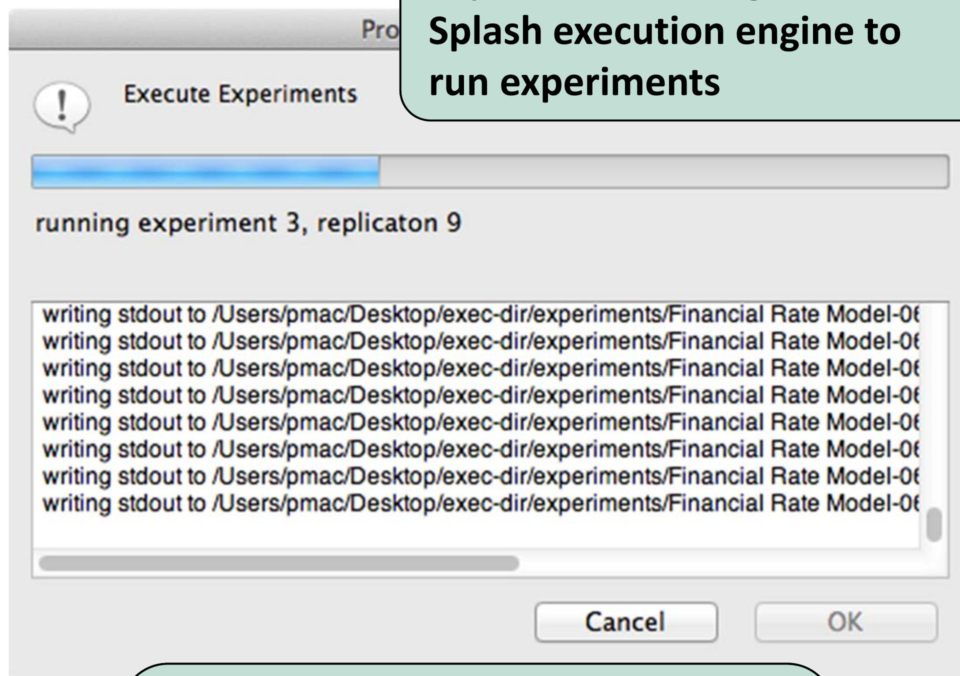
Experiment Manager (Running an Experiment)

Technical challenges include:

- **Routing parameter values to models**
  - Different sources: command line args, parameter files, stdin, …
  - Synthesizing the parameter files that a model expects (templating)

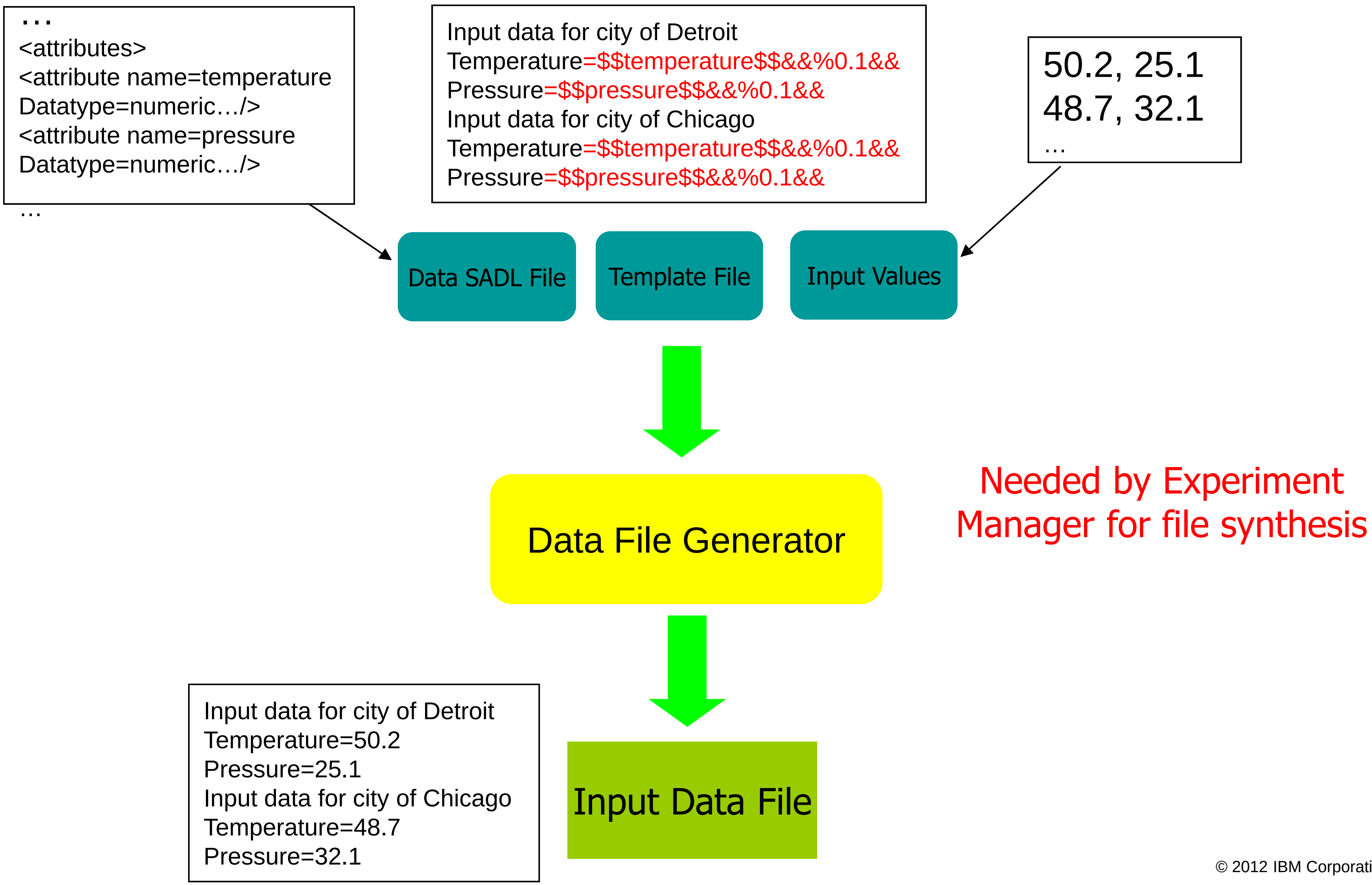
- **Managing PRNG seeds**
  - Avoiding cycle overlaps
  - PRNG info in SADL file
  - Diagnostics (future work)

Experiment Manager invokes Splash execution engine to run experiments

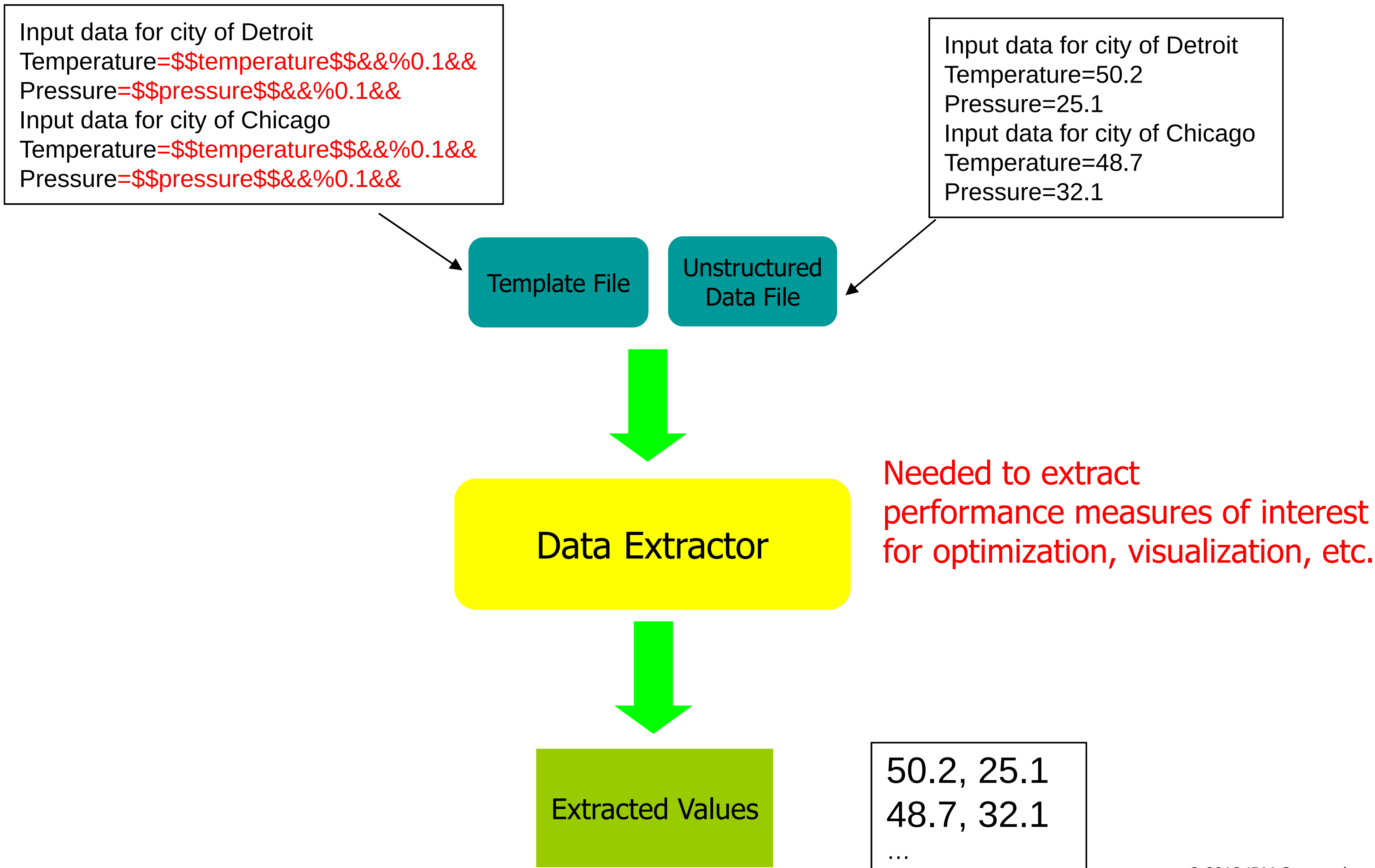


Intermediate and final outputs can be saved in a file tree for
- **Provenance tracking**
- **Traceability**
- **Drill down**

Template-Based Data File Generation Process

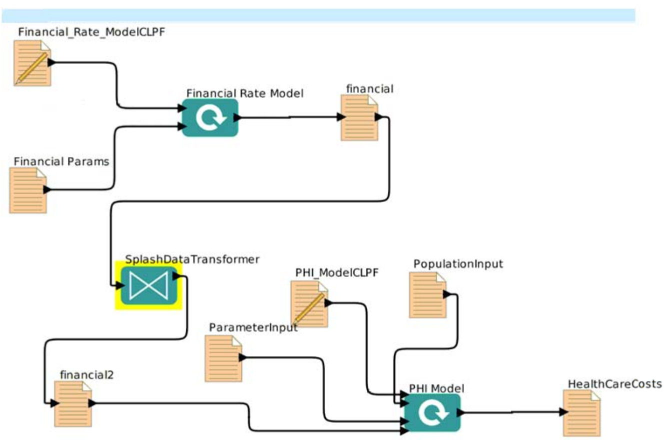


Template-Based Data Extraction Process

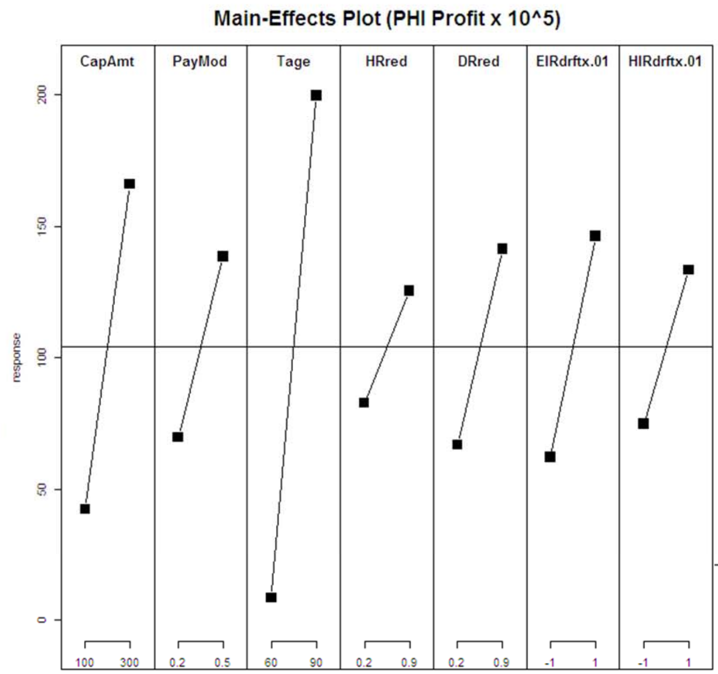


Efficient Sensitivity Analysis

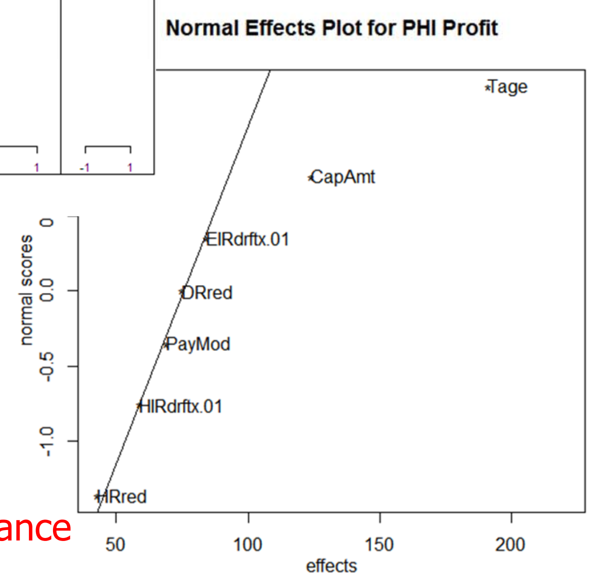
- Main-effects plots:
  - High/low values
  - Orthogonal fractional factorial experiment design (160 vs 2560 runs)



PHI healthcare payer model + interest-rate model
(Park et al., *Service Science*, 2012)



Identify the most important profit drivers (CapAmt & Tage)



Check statistical significance of graphical results

Optimization Functionality: Ranking and Selection

- Rinott procedure for finding best among small number of designs
- Executes min. # of runs needed to distinguish systems

Equal number of stage-1 replications per design

For $i = 1$ to k

Execute n_0 stage-1 replications of model i to obtain $Y_{i,1}, \ldots, Y_{i,n_0}$

Set $\bar{X}_i = (1/n_0)\sum_{j=1}^{n_0} Y_{i,j}$ and $V_i = (1/(n_0 - 1))\sum_{j=1}^{n_0}(Y_{i,j} - \bar{X}_i)^2$

Set $N_i = \max\left(n_0, \left\lceil \frac{h^2 V_i}{\delta^2} \right\rceil\right)$, where $h = h(C, \delta)$ is a tabulated constant

Execute $N_i - n_0$ additional replications of model i to obtain $Y_{i,n_0+1}, \ldots, Y_{i,N_i}$

Compute $\bar{Y}_i = (1/N_i)\sum_{j=1}^{N_i} Y_{i,j}$

Select system with largest value of $\bar{Y}_i$ as the best system

Compute MCB intervals for $i = 1, 2, \ldots, k$:

$a_i = \min(0, \max_{j \neq i} \bar{Y}_j - \bar{Y}_i - \delta)$ and $b_i = \max(0, \max_{j \neq i} \bar{Y}_j - \bar{Y}_i + \delta)$

System determines number of stage-2 replications

Results are combined and ranked

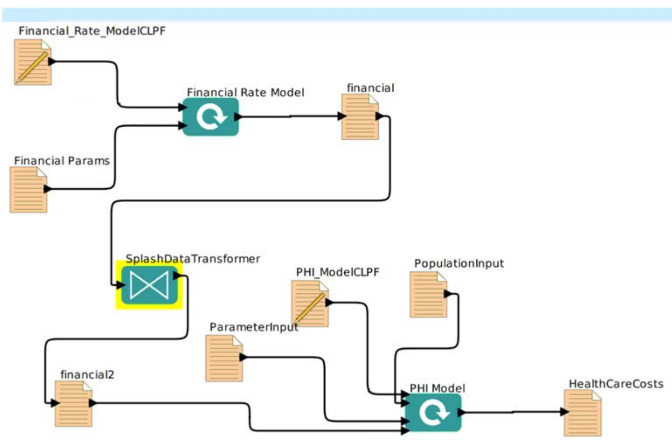
Simultaneous 100*C*% MCB conf. intervals.

Selects design within δ of optimum with probability $\geq C$

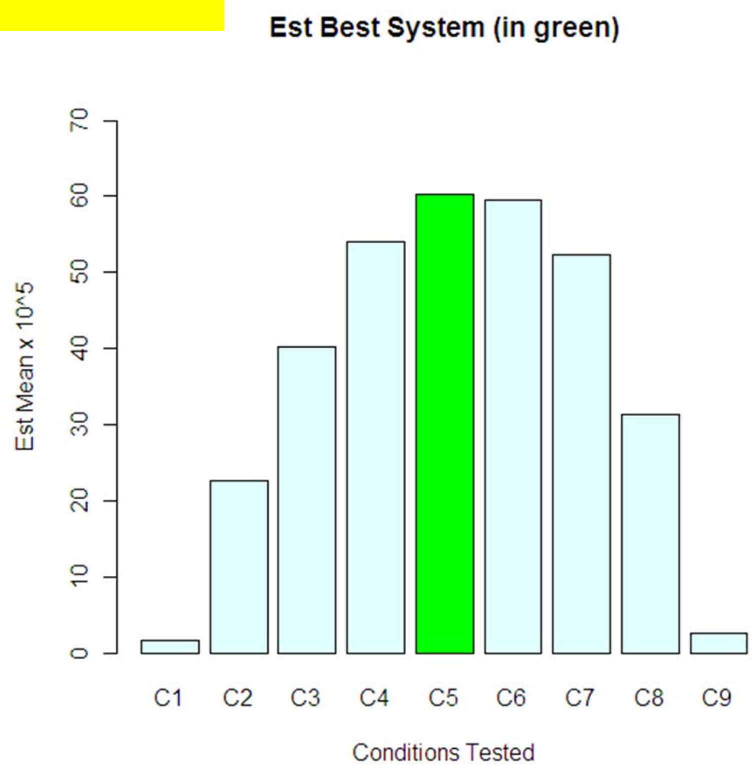
Results for PHI Profitability: Estimated Best System

"Conditions" = payment schemes for wellness program (0 = full capitation, 1 = pay-for-outcome)

Look at weighted schemes: 0.1, 0.2, … , 0.9

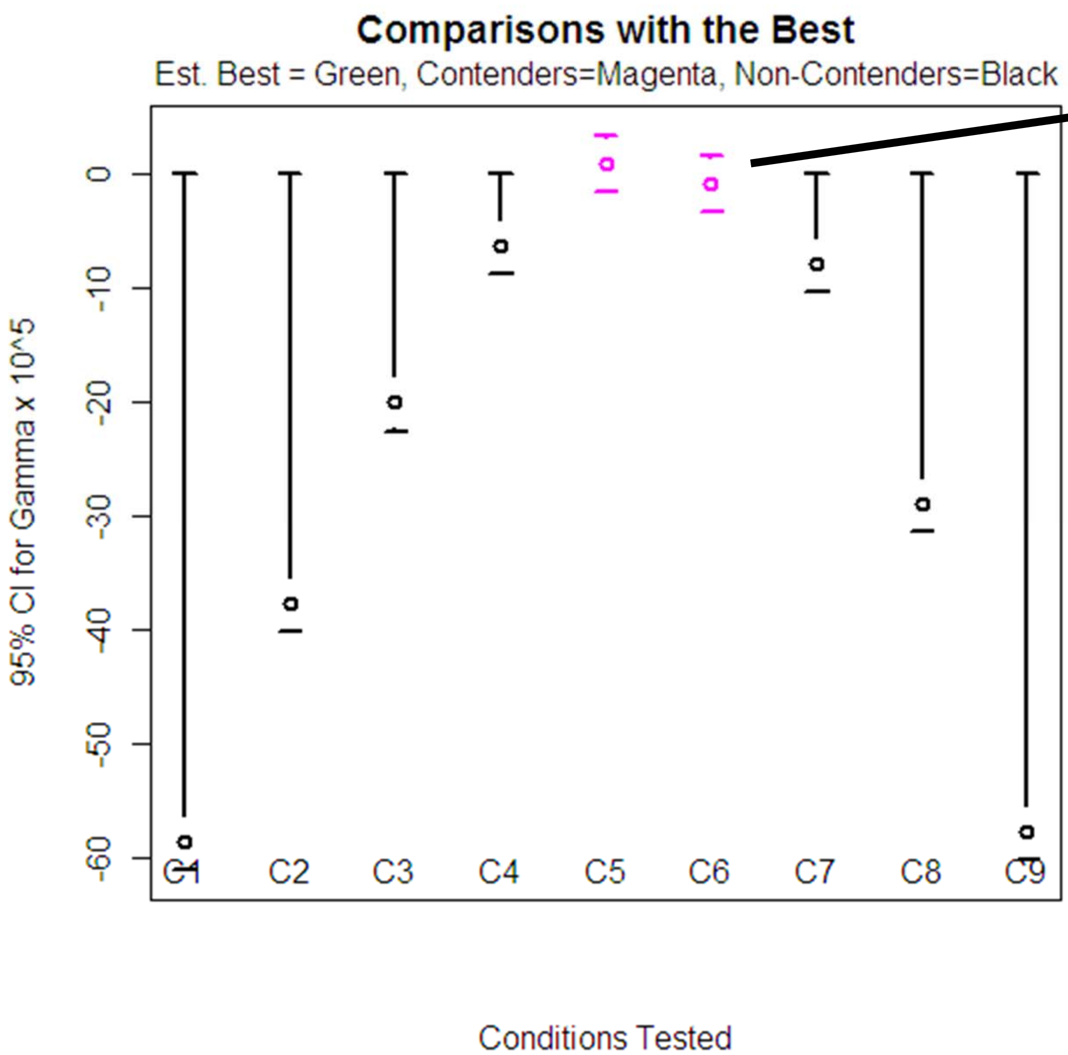


PHI healthcare payer model + interest-rate model
(Park et al., *Service Science*, 2012)



With prob = 95%, C5 = 0.5 is the "best system" (within indifference zone = $250K)

Results Continued: Multiple Comparisons with the Best



Identifies set of best solutions

Simultaneous 95% confidence intervals on difference between each system and best of others

Simulation Metamodeling (Joint Work with SJSU CAMCOS)

"Simulation on demand"

1. Run simulations in advance to get values at multiple "design points"
2. Fit a (stochastic) response surface
3. Decision maker can explore surface in real time
4. Can apply stochastic optimization techniques to find peaks and valleys
5. Can use for factor screening

Technique: Stochastic Kriging
(Ankenman et al., *Oper. Res.*, 2010)

- Robust, global fit
- Gives approximate model response + uncertainty estimates (MSE)
- Efficient allocation to of runs to minimize integrated mean-square error (IMSE)
- Metamodel added to Splash repository

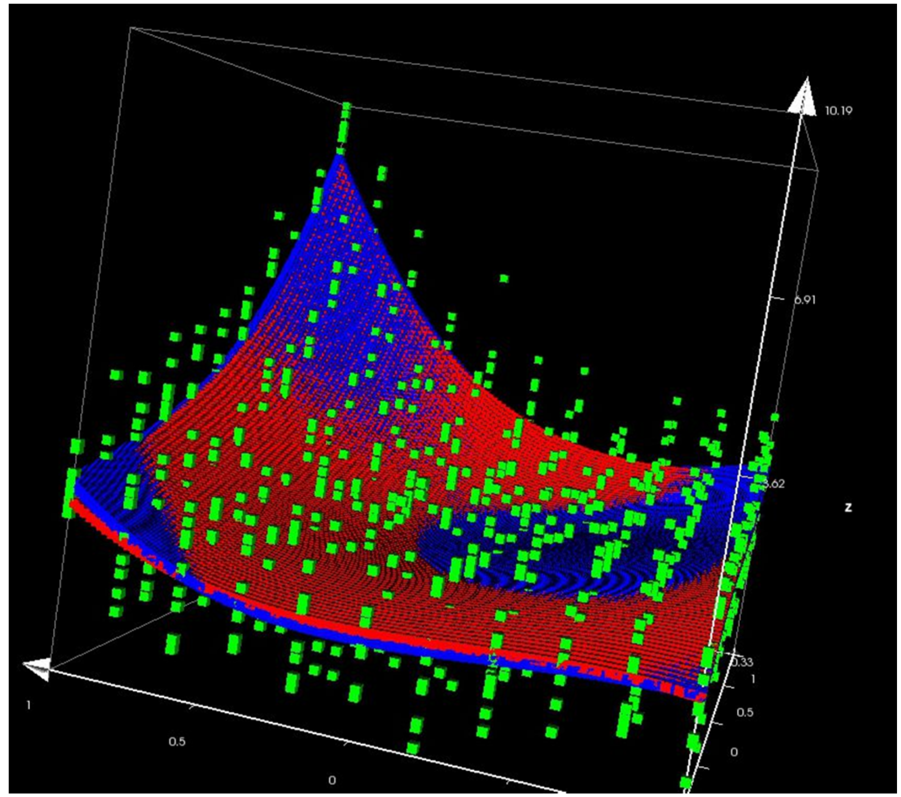
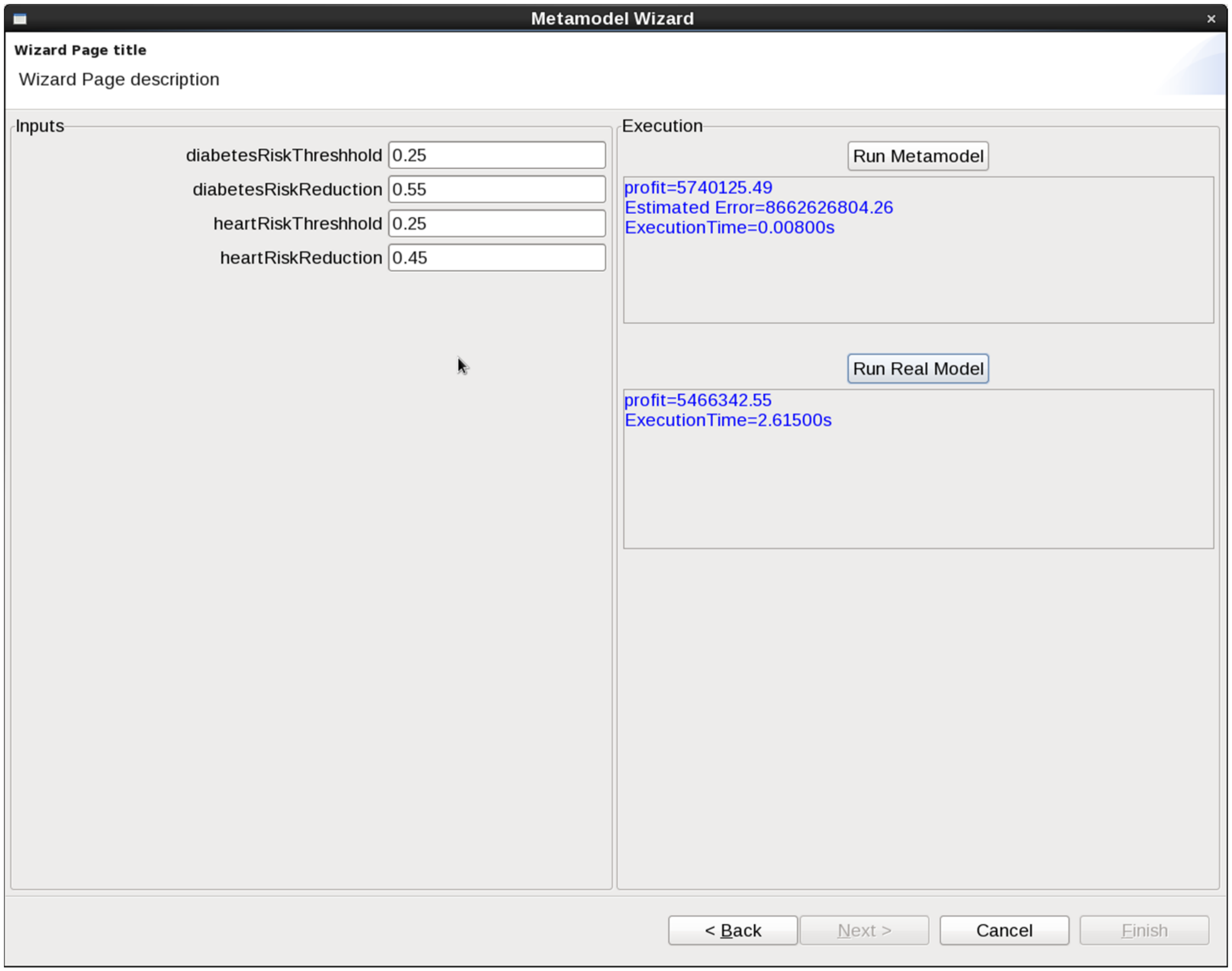


Image: SJSU CAMCOS

Models uncertainty due to both interpolation and simulation variability

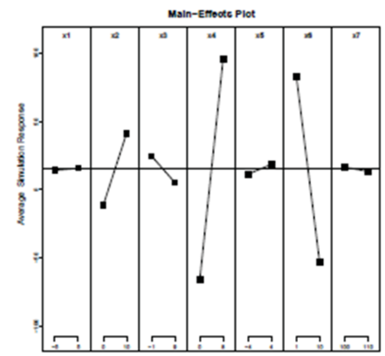
Assessment of PHI metamodel



Metamodel gives good approximation to real results (1.6% error in this example)

Faster by over two orders of magnitude

Factor screening (Joint with SJSU CAMCOS)



Goal: identify most important subset of drivers

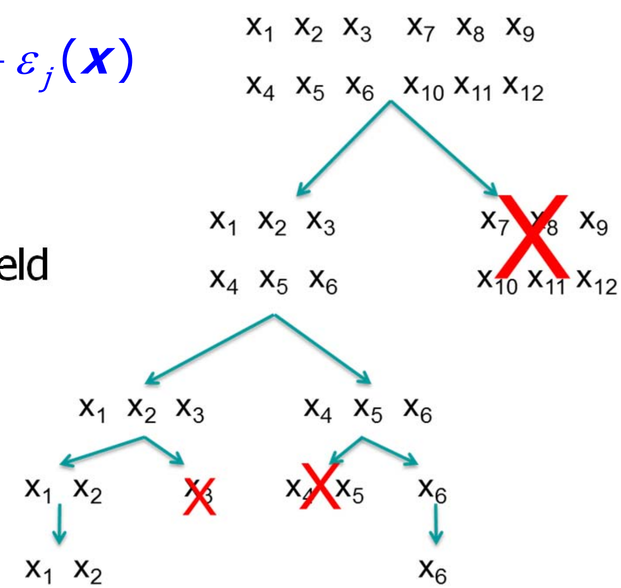
- Drivers captured in metamodel parameters

Ex: Linear models $Y(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_7 x_7 + \varepsilon$

- Main effects used for screening
- For Gaussian noise, positive effects: sequential bifurcation

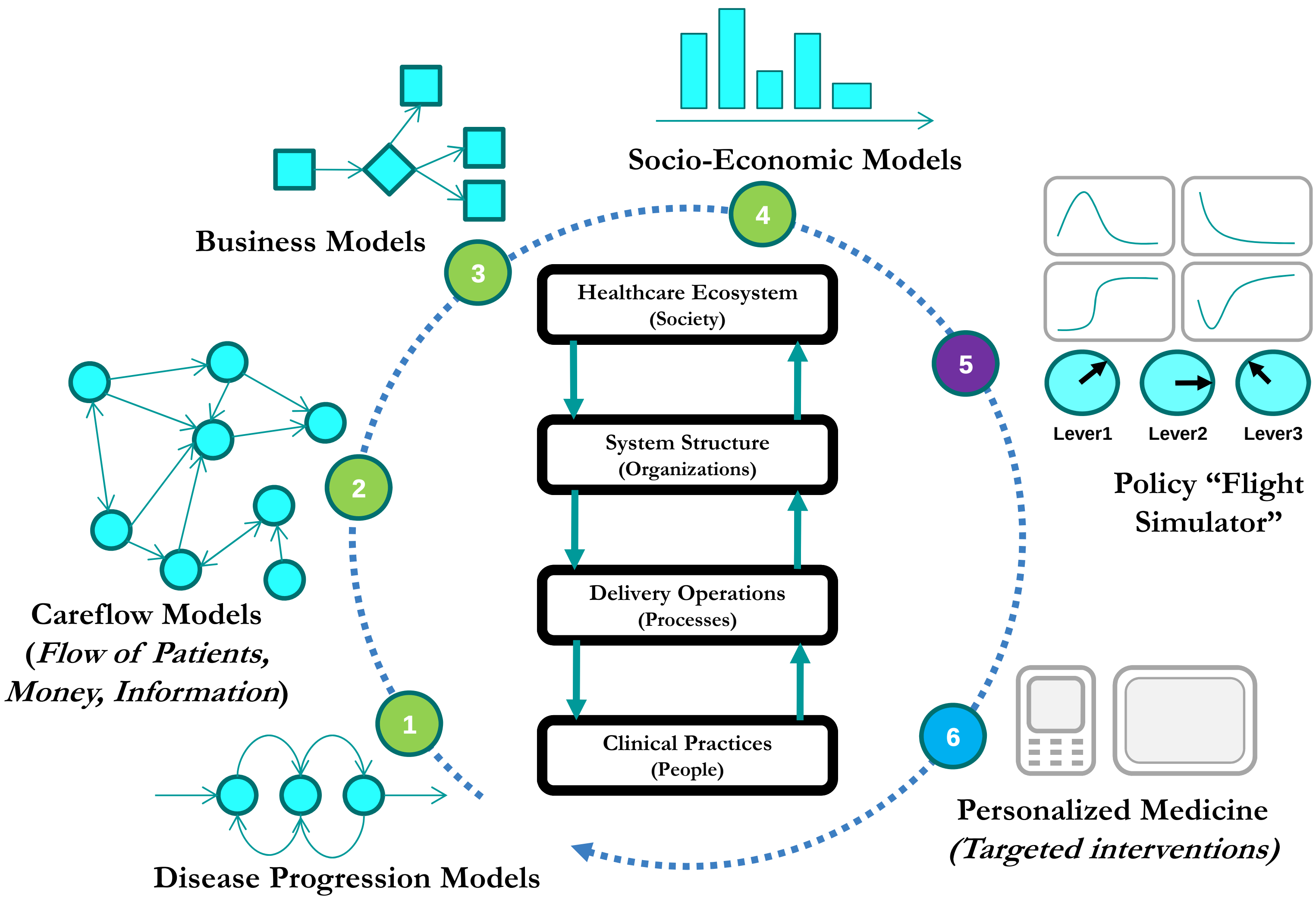
Ex: Gaussian process models $Y_j(\boldsymbol{x}) = \beta_0 + M(\boldsymbol{x}) + \varepsilon_j(\boldsymbol{x})$

- Special case of stochastic kriging
- $\varepsilon_j(\boldsymbol{x})$ = simulation noise
- $M(\boldsymbol{x})$ = interpolation uncertainty, modeled as Gaussian field
  - For any $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r$ vector $V = (M(\boldsymbol{x}_1), \ldots, M(\boldsymbol{x}_r))$ is multivariate normal
  - $\mathrm{Cov}[M(\boldsymbol{x}_i), M(\boldsymbol{x}_j)] = \tau^2 \prod_{k=1}^{n} \exp(-\theta_k (x_{i,k} - x_{j,k})^2)$
- Small $\theta_k \Rightarrow$ small effect of k^{th} factor
- Bayesian "posterior quantiles" method for screening



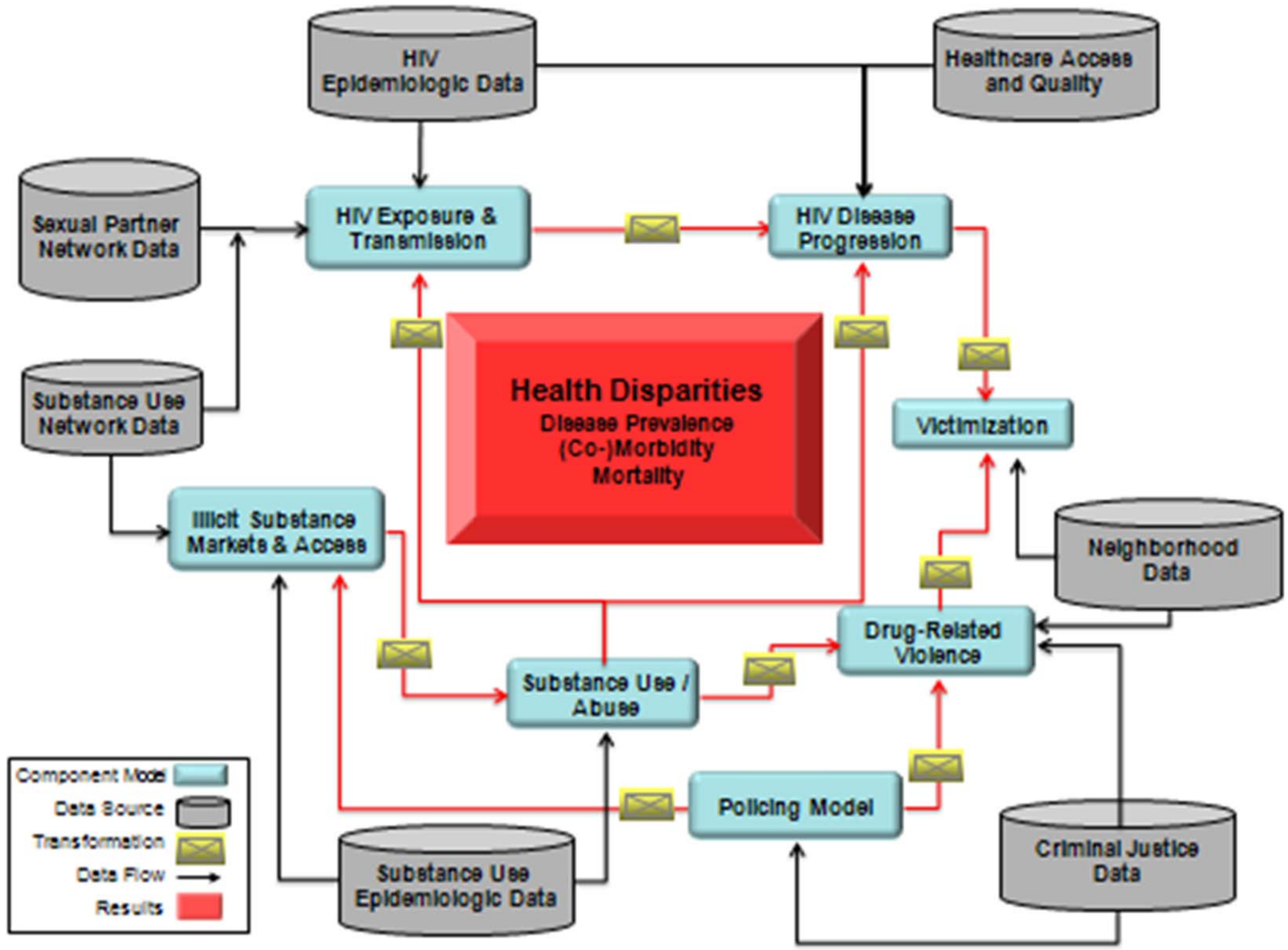
Some Potential Splash Applications

Multi-level, End-to-End Modeling



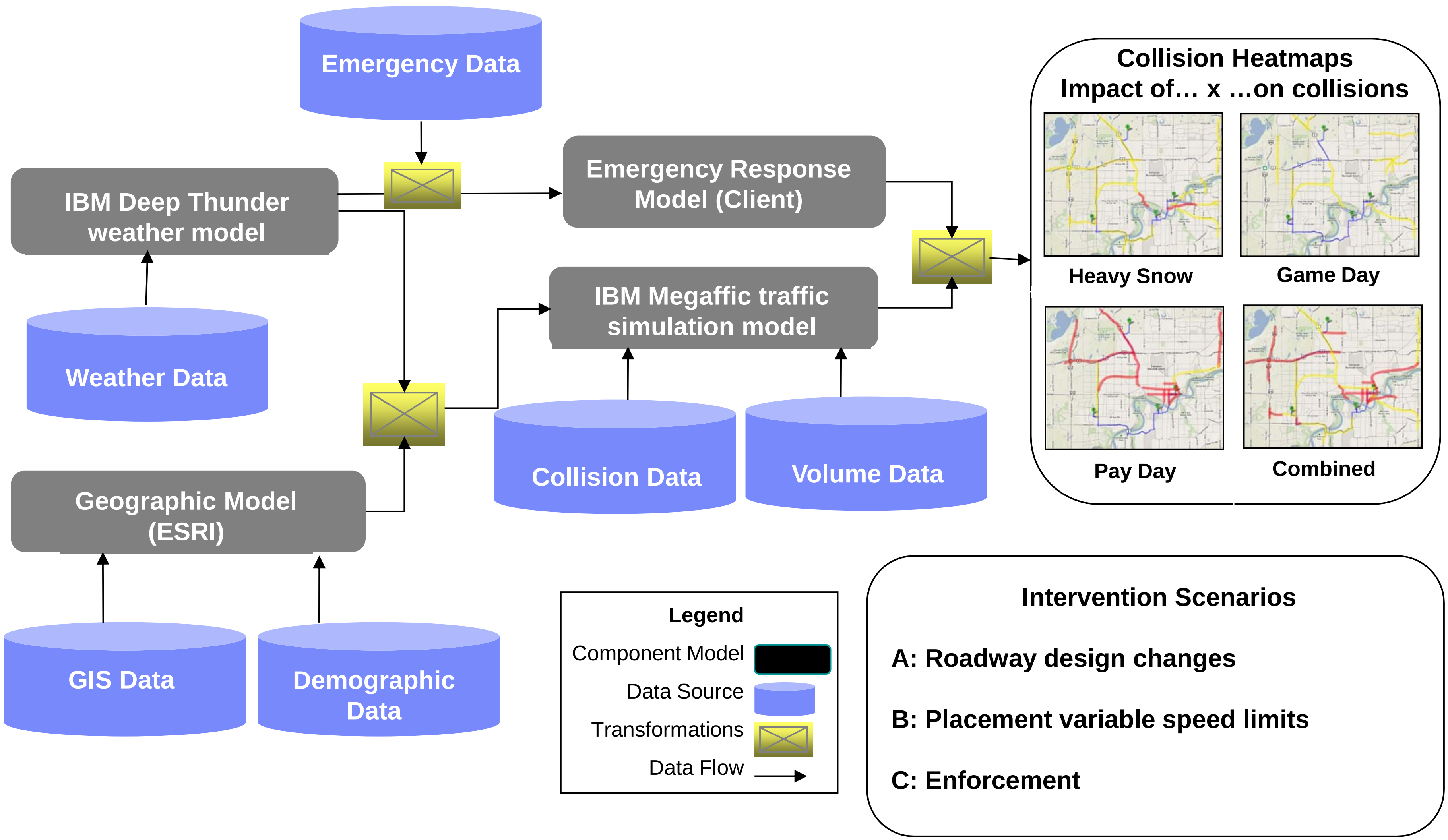
Rouse, W. B. & Cortese, D. A. (2010). Introduction, in W. B. Rouse & D. A. Cortese (Eds.), *Engineering the System of Healthcare Delivery*. IOS Press.

Cross-domain, Syndemic Modeling



Richard Rothenberg et al., Georgia State University, 2011

Composite model for traffic safety

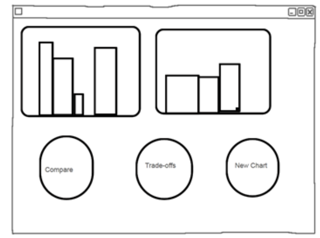


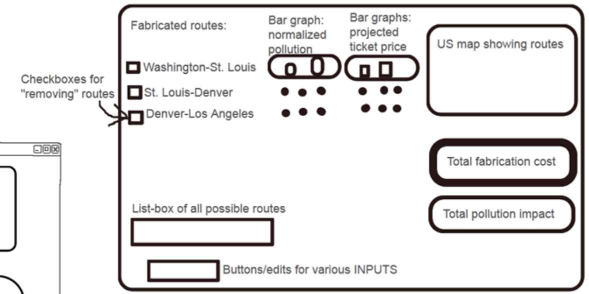
Open Research Questions

How to Determine User Requirements?

Common to Analysts and Scientists

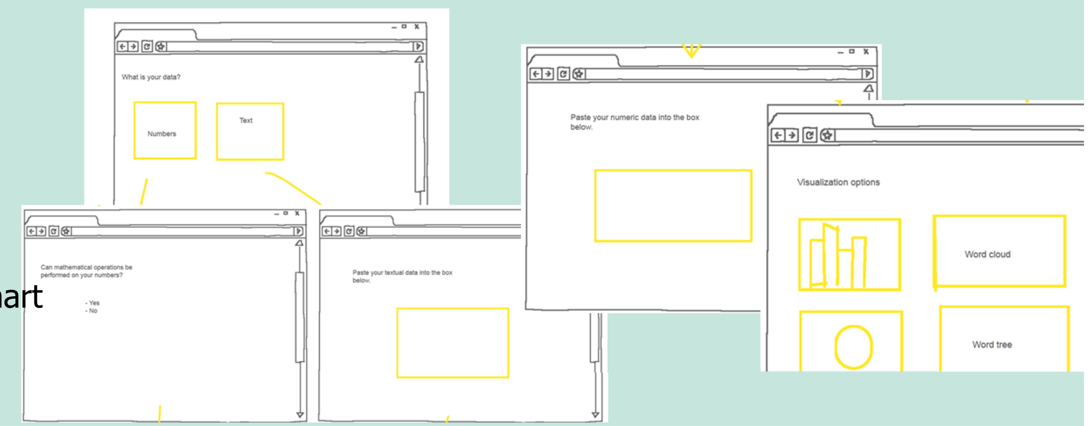
- Examine schemas (data) and variables (models) prior to selection
- Compare output of simulation results to examine trade-offs and simulation selection
- Dashboard with summary of models and data sources used to run a simulation





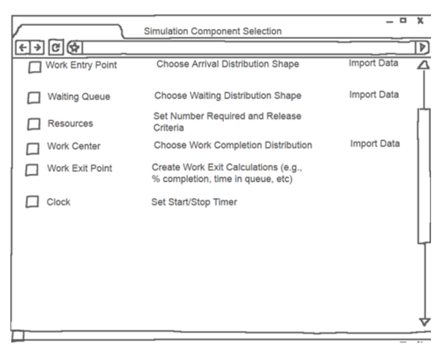
Specific to Analysts

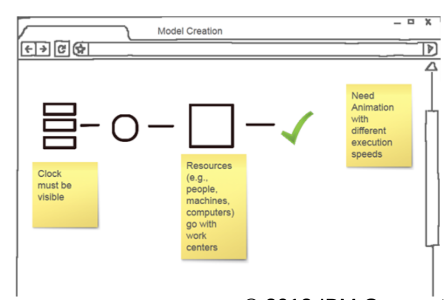
- Guidance and recommendations
- Pre-defined templates for simulation set-up and analyzing simulation output
- Recommendations for what template to use and the steps to run a simulation
- Recommended output visualization – suggest one chart style would be better than another style to explain relationships in data



Specific to Scientists

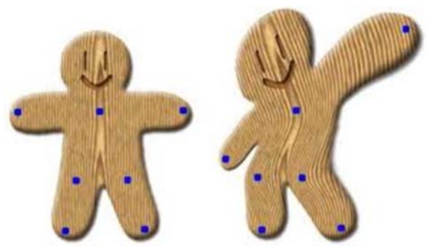
- Feature to assess the veracity and provenance of model and data sources
- Ability to upload their own sources to supplement the existing sources
- High levels of interaction with the models & data when previewing search results prior to running the simulation





Database Research++

- **Data search → model-and-data search**
  – Find compatible models, data, and mappings (using metadata)
  – Involves semantic search technologies, repository management, privacy and security



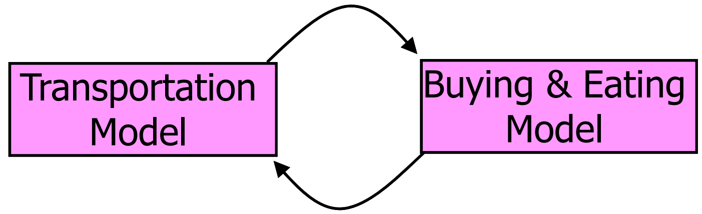
- **Data integration → model integration**
  – Simulation-oriented data mapping
  – Geospatial alignment [e.g., Howe & Maier 2005]
  – Hierarchical models with different resolutions
  – Complex data transformations (e.g., raw simulation output to histogram)

- **Query optimization → simulation-experiment optimization**
  – Optimally configure workflow among distributed data and models
  – Factoring common operations across different mappings in the workflow
  – Avoiding redundant computations across experiments (e.g., result caching)
  – Statistical issues: managing pseudorandom numbers and Monte Carlo replications

Some Deep Problems

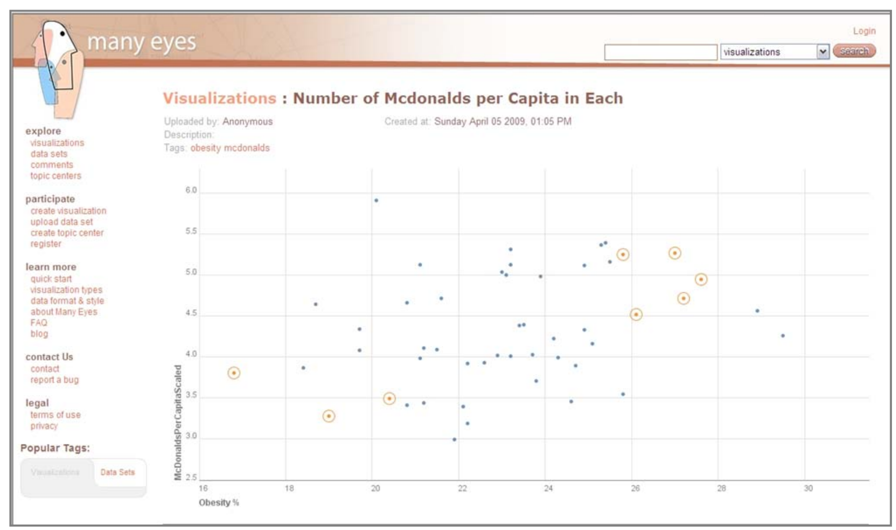
- **Causality approximation**
  - Fixed-point + perturbation approaches
  - System support
  - Theoretical support

- **Deep collaborative analytics**
  - Visualizing and mining the results
  - Understanding and explaining results:
    - Provenance [e.g., J. Friere et al.]
    - Root-cause analysis
  - Trusting results
    - Model validation
    - ManyEyes++, Swivel++



$$\dot{f}_n(t) = \Lambda_1\left(f_n(t), g_{n-1}(t)\right)$$
$$\dot{g}_n(t) = \Lambda_2\left(f_{n-1}(t), g_n(t)\right)$$

$$\left.\begin{aligned}\dot{f}(t) &= \Lambda_1\left(f(t), g(n\Delta t)\right)\\ \dot{g}(t) &= \Lambda_2\left(f(n\Delta t), g(t)\right)\end{aligned}\right\} \text{ for } t \in \left[n\Delta t, (n+1)\Delta t\right)$$



Conclusion

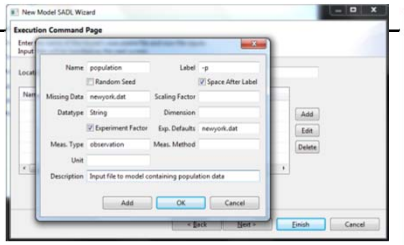
- **Splash:**
  - composition of heterogeneous models and data
    to support cross-disciplinary decision making in complex systems
  - Loose coupling of models through data exchange
  - Combines data-integration, simulation, and workflow technologies
- **Key features**
  - SADL metadata language for curation and functionality
  - Automated detection of data mismatches
  - Semi-automated design of scalable data transformations (schema and time alignment)
  - Runtime accelerators
    - MapReduce framework for scalable data transformations
    - Map-only Hadoop method for cubic-spline interpolation
    - Result-caching to minimize # of model executions
  - Experiment-manager allows sensitivity analysis, factor screening and optimization
  - Simulation metamodeling for real-time model exploration
- **Many open research questions!**

Questions?

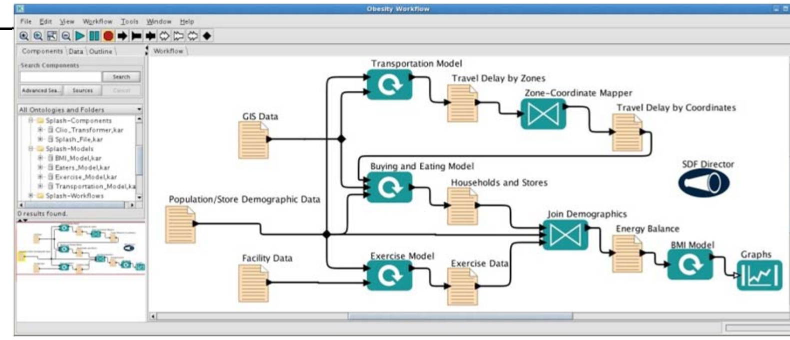
Splash project page:
http://researcher.watson.ibm.com/researcher/view_project.php?id=3931

Backup Slides

Splash Technology for Loose Coupling via Data Exchange

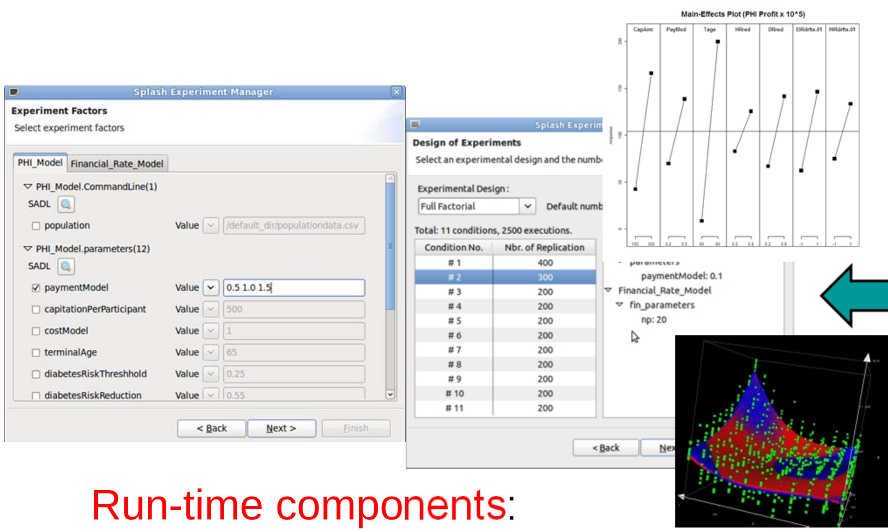


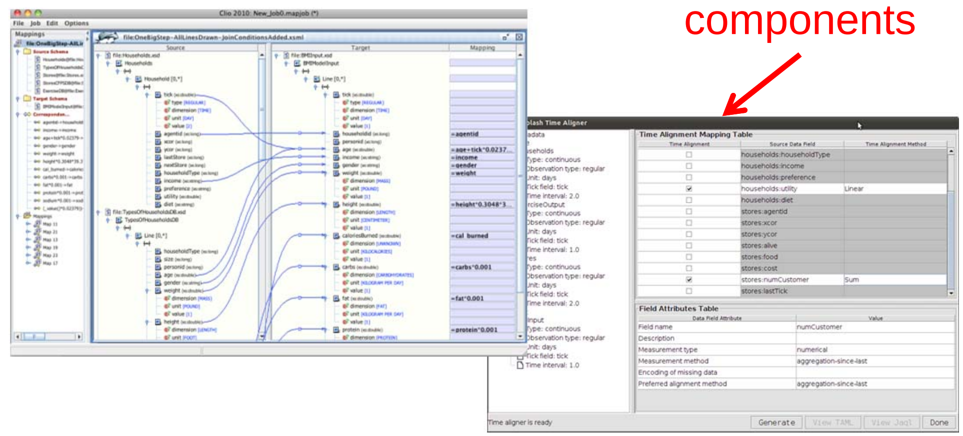
SADL metadata language



Kepler adapted for model composition

Design-time components





Run-time components:

- Kepler adapted for model execution
- Experiment Manager
 (sensitivity analysis, metamodeling, optimization)

Data transformation tools:

- Clio++
- Time Aligner (MapReduce algorithms)
- Templating mechanism

Distributed SGD, Continued

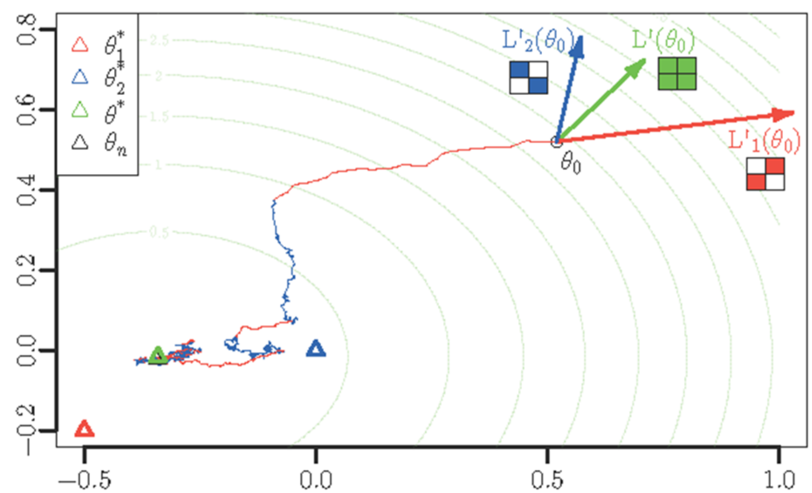
- Divide the m-1 rows into three **strata**: U^1, U^2, U^3

- **Decompose** loss function:

$$L(x) = \tfrac{1}{3}L^1(x) + \tfrac{1}{3}L^2(x) + \tfrac{1}{3}L^3(x)$$
$$\text{where } L^s(x) = 3\sum_{i \in U^s} L_i(x)$$

- Define (random) **stratum sequence** $\gamma_1, \gamma_2, \ldots$

- Execute SGD **w.r.t.** L^{γ_k} at k^{th} step in parallel



- **Theorem:** Suppose that $x^* = A^{-1}b$ exists and
  - $\varepsilon_n = O(n^{-\alpha})$ for some $\alpha \in (0.5, 1)$
  - $(\varepsilon_n - \varepsilon_{n+1}) / \varepsilon_n = O(\varepsilon_n)$
  - $\{\gamma_n : n \geq 0\}$ is regenerative with $E[\tau_1^{1/\alpha}] < \infty$ and $E[X_1(s)] = 0$

  Then $x^{(n)} \to x^*$ with probability 1

- Stratum sequence occasionally restarts probabilistically
- Time τ between restarts has finite $1/\alpha$ moment
- Sequence spends ≈1/3 of its time on each stratum

- **Proof:** [GHS11] + Liapunov-function argument

Hadoop Implementation

- Physical **blocks** and logical **splits**
  - InputFormat operator creates splits (one split per mapper)
  - A split is mostly on one block
  - Splits are usually disjoint
  - Map job: each mapper first obtains all split data (small amount of data movement)
  - Reduce job: massive shuffling of data over network
- We allow splits to **overlap** by two rows
- DSGD is implemented as a **map-only** job (no data shuffling!)

1	$a_{1,1}$	$a_{1,2}$	b_1	$\boldsymbol{x_1}$
2	$a_{2,1}$	$a_{2,3}$	b_2	$\boldsymbol{x_2}$
3	$a_{3,2}$	$a_{3,4}$	b_3	$\boldsymbol{x_3}$
4	$a_{4,3}$	$a_{4,5}$	b_4	$\boldsymbol{x_4}$
5	$a_{5,4}$	$a_{5,6}$	b_5	x_5
6	$a_{6,5}$	$a_{6,7}$	b_6	x_6
7	$a_{7,6}$	$a_{7,8}$	b_7	$\boldsymbol{x_7}$
8	$a_{8,7}$	$a_{8,9}$	b_8	$\boldsymbol{x_8}$
9	$a_{9,8}$	$a_{9,10}$	b_9	$\boldsymbol{x_9}$
10	$a_{10,9}$	$a_{10,11}$	b_{10}	$\boldsymbol{x_{10}}$
11	$a_{11,10}$	$a_{11,12}$	b_{11}	x_{11}
12	$a_{12,11}$	$a_{12,13}$	b_{12}	x_{12}
13	$a_{13,12}$	$a_{13,14}$	b_{13}	x_{13}

split 1

split 2

stratum s = 1

(mapper 2 modifies x_7)

Hadoop Implementation

- Physical **blocks** and logical **splits**
  - InputFormat operator creates splits (one split per mapper)
  - A split is mostly on one block
  - Splits are usually disjoint
  - Map job: each mapper first obtains all split data (small amount of data movement)
  - Reduce job: massive shuffling of data over network
- We allow splits to **overlap** by two rows
- DSGD is implemented as a **map-only** job (no data shuffling!)

1	$a_{1,1}$	$a_{1,2}$	b_1	x_1
2	$a_{2,1}$	$a_{2,3}$	b_2	x_2
3	$a_{3,2}$	$a_{3,4}$	b_3	x_3
4	$a_{4,3}$	$a_{4,5}$	b_4	x_4
5	$a_{5,4}$	$a_{5,6}$	b_5	x_5
6	$a_{6,5}$	$a_{6,7}$	b_6	x_6
7	$a_{7,6}$	$a_{7,8}$	b_7	x_7
8	$a_{8,7}$	$a_{8,9}$	b_8	x_8
9	$a_{9,8}$	$a_{9,10}$	b_9	x_9
10	$a_{10,9}$	$a_{10,11}$	b_{10}	x_{10}
11	$a_{11,10}$	$a_{11,12}$	b_{11}	x_{11}
12	$a_{12,11}$	$a_{12,13}$	b_{12}	x_{12}
13	$a_{13,12}$	$a_{13,14}$	b_{13}	x_{13}

split 1

split 2

stratum s = 2

(mapper 2 modifies x_7)

Hadoop Implementation

- Physical **blocks** and logical **splits**
  - InputFormat operator creates splits (one split per mapper)
  - A split is mostly on one block
  - Splits are usually disjoint
  - Map job: each mapper first obtains all split data (small amount of data movement)
  - Reduce job: massive shuffling of data over network
- We allow splits to **overlap** by two rows
- DSGD is implemented as a **map-only** job (no data shuffling!)

1	$a_{1,1}$	$a_{1,2}$	b_1	x_1
2	$a_{2,1}$	$a_{2,3}$	b_2	x_2
3	$a_{3,2}$	$a_{3,4}$	b_3	$\boldsymbol{x_3}$
4	$a_{4,3}$	$a_{4,5}$	b_4	$\boldsymbol{x_4}$
5	$a_{5,4}$	$a_{5,6}$	b_5	$\boldsymbol{x_5}$
6	$a_{6,5}$	$a_{6,7}$	b_6	$\boldsymbol{x_6}$
7	$a_{7,6}$	$a_{7,8}$	b_7	x_7
8	$a_{8,7}$	$a_{8,9}$	b_8	x_8
9	$a_{9,8}$	$a_{9,10}$	b_9	$\boldsymbol{x_9}$
10	$a_{10,9}$	$a_{10,11}$	b_{10}	$\boldsymbol{x_{10}}$
11	$a_{11,10}$	$a_{11,12}$	b_{11}	$\boldsymbol{x_{11}}$
12	$a_{12,11}$	$a_{12,13}$	b_{12}	$\boldsymbol{x_{12}}$
13	$a_{13,12}$	$a_{13,14}$	b_{13}	x_{13}

split 1

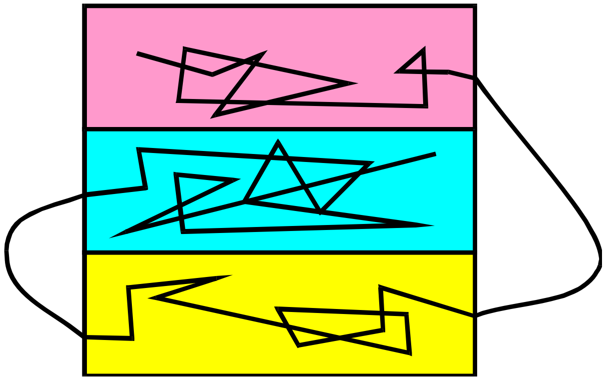
split 2

stratum $s = 3$

(x_7 affects mapper 1)

Other Implementation Details

- **Initial guess**
  - Ignore off-diagonal elements
  - Works well due to "diagonal dominance"
- **Stratum sequence** as in [GHS11]
  - Meander in a stratum for a while, then jump to next stratum
  - Tension between thorough exploration of stratum and randomness
  - Visit all k rows in stratum: at each "sub-epoch" select one of $k!$ orders at random
  - Similar strategy for jumping between strata
  - Convergence Theorem still applies
- **Step-size sequence**
  - Constant during sub-epoch
  - "Bold driver" heuristic
  - Experiment with initial step size (in parallel on small subsequences)



Optimizing the Re-Use Factor for Maximum Efficiency

To define (asymptotic) efficiency, consider budget-constrained setting [Fox & Glynn 1990; Glynn & Whitt 1992]

- Cost of producing n outputs from Model 2:

$$C_n = \sum_{j=1}^{m_n} \tau_{1;j} + \sum_{j=1}^{n} \tau_{2;j}$$

$\tau_{i;j} =$ (random) cost of producing j^{th} observation of Y_i

- Under (large) fixed computational budget c
  - Number of Model 2 outputs produced:

$$N(c) = \max\{n \geq 0 : C_n \leq c\}$$

  - Estimator:

$$U(c) = \theta_{N(c)} = N(c)^{-1} \sum_{j=1}^{N(c)} Y_{2;j}$$

Optimizing the Re-Use Factor II

The key limit theorem as budget increases to infinity

Suppose that $E[\tau_1 + \tau_2 + Y_2^2] < \infty$. Then $U(c)$ is asymptotically $N(\theta, g(\alpha)/c)$.

where $r_\alpha = \lfloor 1/\alpha \rfloor$ and

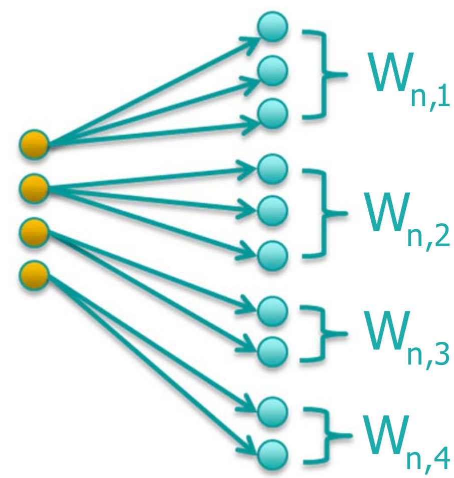
$$g(\alpha) = \underbrace{(\alpha E[\tau_1] + E[\tau_2])}_{\text{(cost per obs.)}} \times \underbrace{\left\{ Var[Y_2] + \left(2r_\alpha - \alpha r_\alpha (r_\alpha + 1)\right) Cov[Y_2, \tilde{Y}_2] \right\}}_{\text{(contributed variance per obs.)}}$$

$Cov[Y_2, \tilde{Y}_2] =$ covariance of two Model 2 outputs that share a Model 1 input

- Thus, minimize $g(\alpha)$ [or maximize asymptotic efficiency $= 1/g(\alpha)$]

Proof Outline

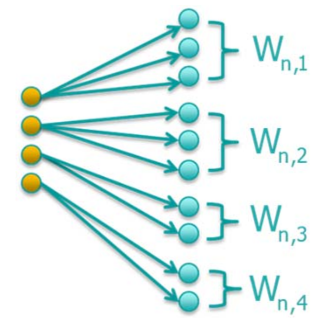
- Set $W_{n,j} = \sum_{i=1}^{n} Y_{2;i}\, I[\text{input for ith run of Model 2 is } Y_{1;j}]$
- Thus $\theta_n = \left(\frac{m_n}{n}\right) m_n^{-1} \sum_{j=1}^{m_n} W_{n;j} \approx \alpha \cdot m_n^{-1} \sum_{j=1}^{m_n} W_{n;j}$
- By Theorem 1 in [Glynn & Whitt 1992], it suffices to show that
  - $C_n / n \xrightarrow{a.s.} \alpha c_1 + c_2$ (straightforward to show)
  - $W_{n,1}, W_{n,2}, \ldots, W_{n,m_n}$ obeys a "Lindeberg-Feller" FCLT



$W_{n,j}$ and $W_{n,j'}$ are independent for $j \neq j'$

- Can establish standard "Lindeberg condition" which suffices for FCLT (Billingsley 1999)
- Some additional fussy details due to the cycling through Model 1 outputs

Point and Interval Estimates



Typical scenarios:

- Compute $100(1-\delta)\%$ confidence interval for θ under fixed budget c
- Estimate θ to within $\pm 100\varepsilon\%$ with probability $100(1-\delta)\%$

Issue: n is unknown a priori (so can't compute m_n)

- Solution: estimate n from n_0 pilot (or prior) runs

$W^{(c)}_{n,j}$ is "centered" version of $W_{n,j}$

- Can show: $\dfrac{\sqrt{n}(\theta_n - \theta)}{\sqrt{h_n(\alpha)}} \Rightarrow N(0,1)$ where $h_n(\alpha) = n^{-1}\sum_{j=1}^{m_n}\left(W^{(c)}_{n,j}\right)^2$

  so that CI from n runs is $\left[\theta_n - z_\delta\left(h_n(\alpha)/n\right)^{1/2}, \theta_n + z_\delta\left(h_n(\alpha)/n\right)^{1/2}\right]$

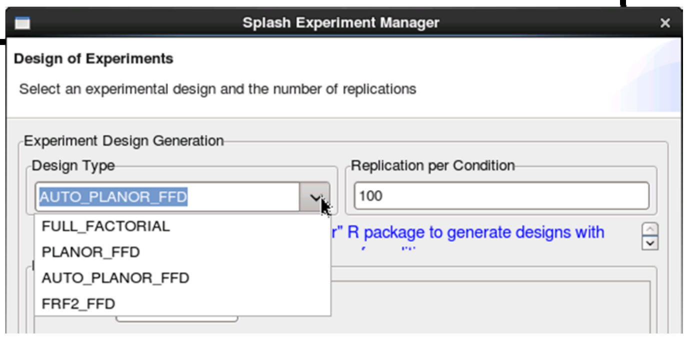
  where z_δ is $(1+\delta)/2$ normal quantile

- Can set
  - $n \approx c/(\alpha c_1 + c_2)$ for fixed budget
  - $n \approx h_{n_0}(\alpha)\left(z_\delta / \varepsilon\theta_{n_0}\right)^2$ for fixed precision

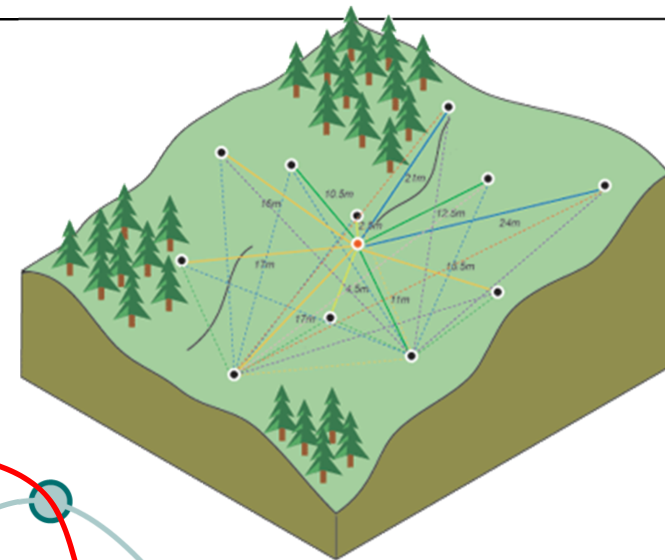
Interface to R system for experimental design

Method	Provider	Notes
Full Factorial Design	Experiment Manager	▪Simple, fast design generation ▪Exhaustive factor combinations -> slow execution
Planor Fractional Factorial Design	R – planor package http://cran.r-project.org/web/packages/planor/vignettes/PlanorInRmanual.pdf	▪Supports arbitrary factor levels ▪Leverages R design generation ▪Checks statistical feasibility of user's proposed design ▪Slow design generation, fast experiment execution
Auto Planor Fractional Factorial Design	R – planor package http://cran.r-project.org/web/packages/planor/vignettes/planorVignette.pdf	▪Supports arbitrary factor levels ▪Leverages R design generation ▪Automatically finds smallest feasible experiment ▪Slower design generation, fast experiment execution
FRF2 Fractional Factorial Design	R – FrF2 package http://cran.r-project.org/web/packages/FrF2/FrF2.pdf	▪Only supports 2-level factors ▪Fast generation ▪Fast execution
Custom	User Specified	▪Any design above may be used as basis

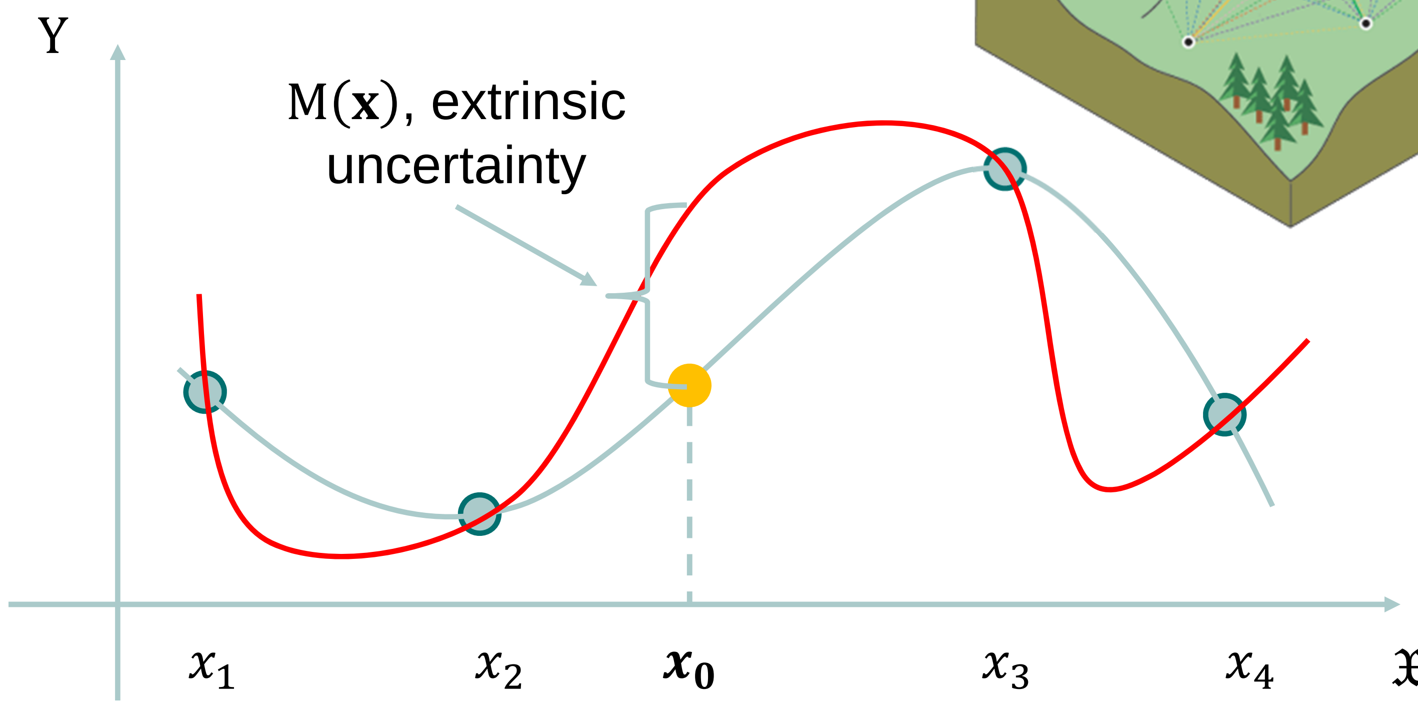
As new designs are introduced in R, the interface is in place to take advantage of these.



Standard Kriging



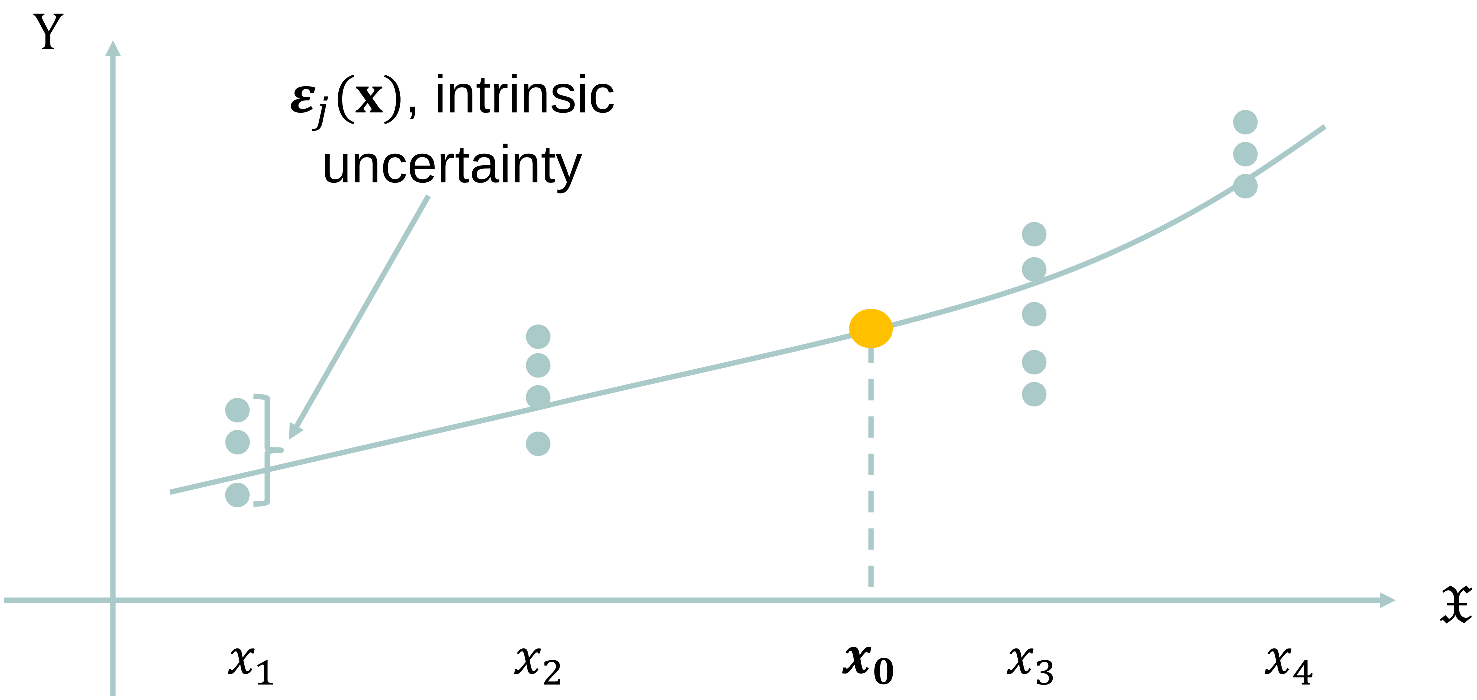
$$\mathcal{Y}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta} + \mathrm{M}(\mathbf{x})$$



Images: SJSU

Stochastic Kriging

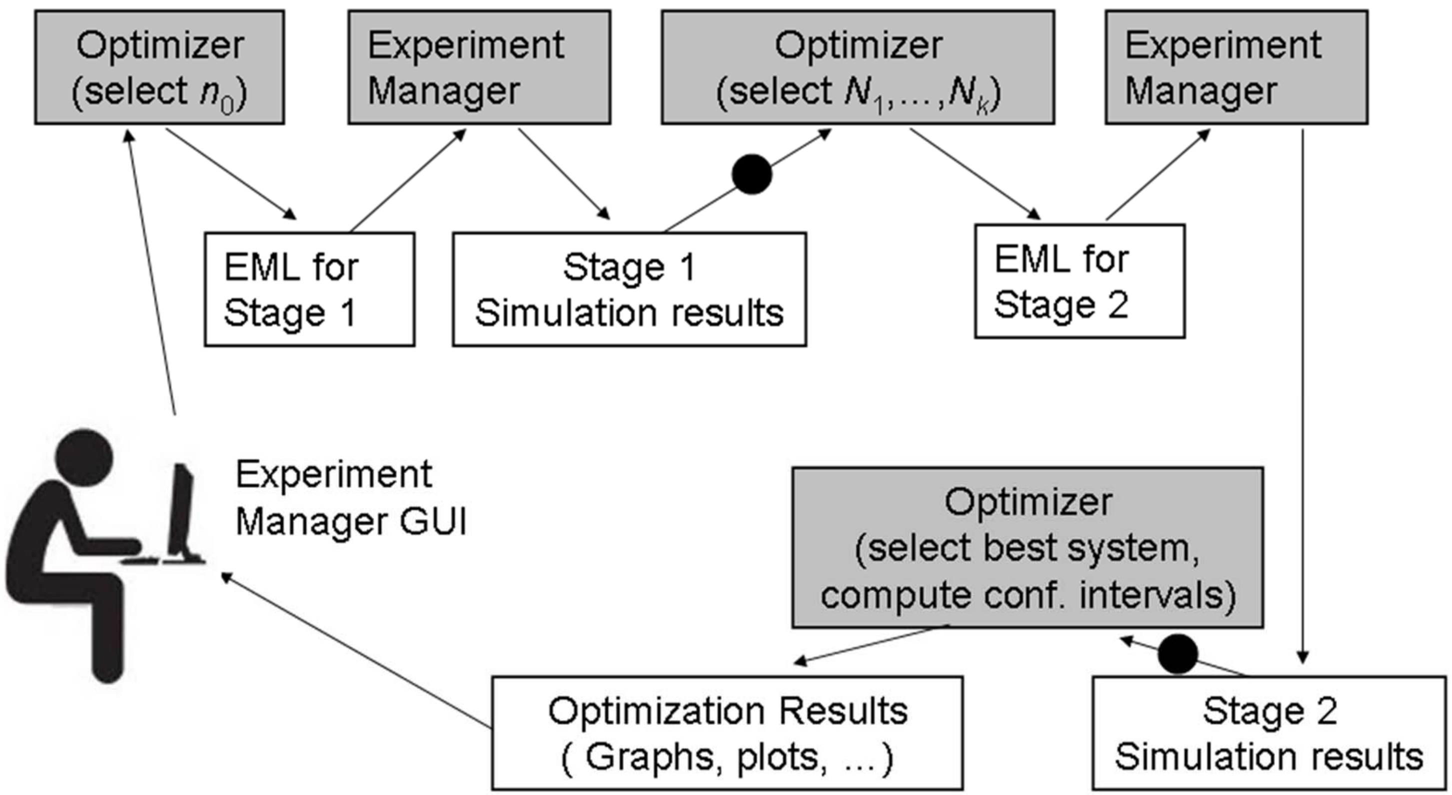
$$\mathcal{Y}_j(\mathbf{x}) = \mathbf{f}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta} + \mathrm{M}(\mathbf{x}) + \boldsymbol{\varepsilon}_j(\mathbf{x})$$



MLE estimate: $$\widehat{\mathrm{Y}}(\mathbf{x}_0) = \beta_0 + \Sigma_{\mathrm{M}}(\mathbf{x}_0,\cdot)^{\mathrm{T}}\left[\Sigma_{\mathrm{M}} + \widehat{\Sigma}_{\varepsilon}\right]^{-1}(\bar{\mathcal{Y}} - \beta_0\mathbf{1}_{\mathrm{k}})$$

Images: SJSU

Optimization Process Flow



- Optimizer is R code,
- Orchestration via Python scripts

● = template-based data extraction