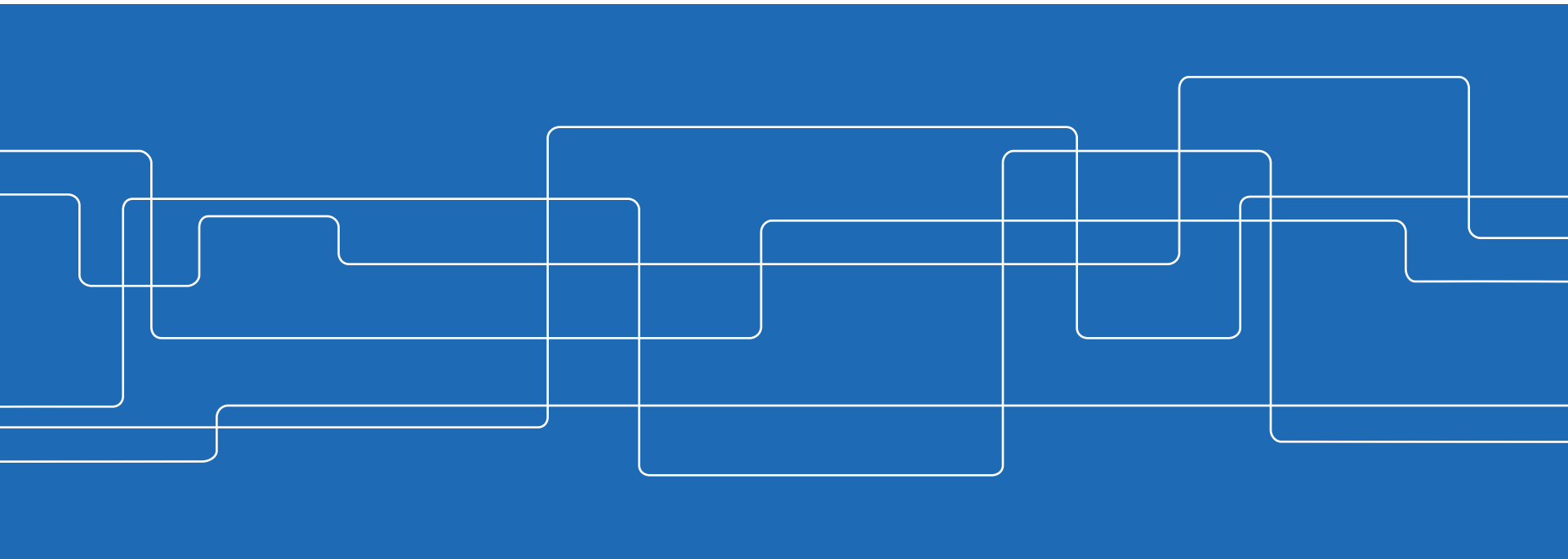




# Search-Based Design of Large Software Systems-of-Systems

Robert Lagerström, Pontus Johnson, and Mathias Ekstedt  
KTH Royal Institute of Technology





# Outline

Today:

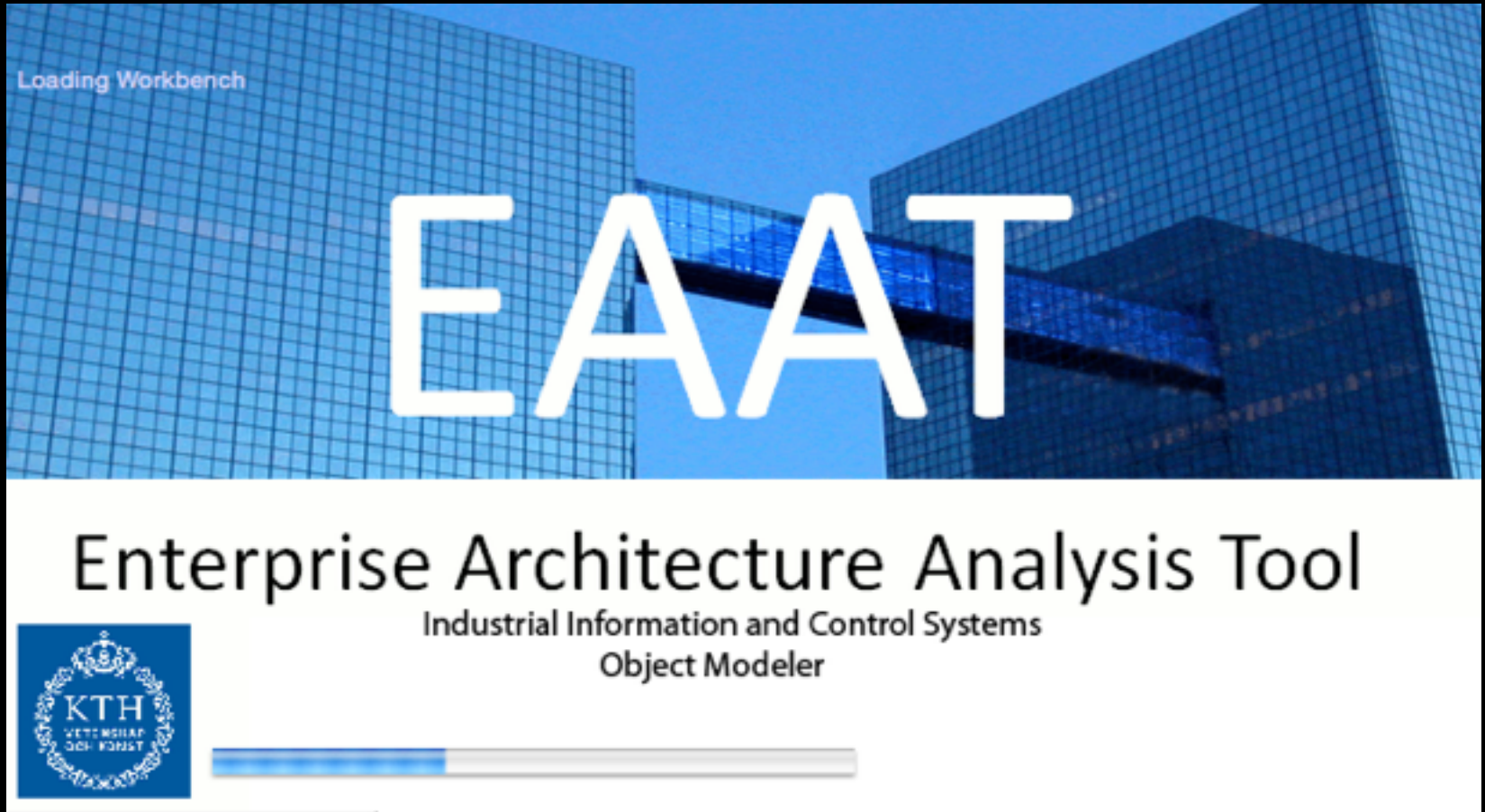
Introduction to a systems-of-systems modeling and analysis tool

Search-based software engineering

Tomorrow:

Search-based design of systems-of-system?

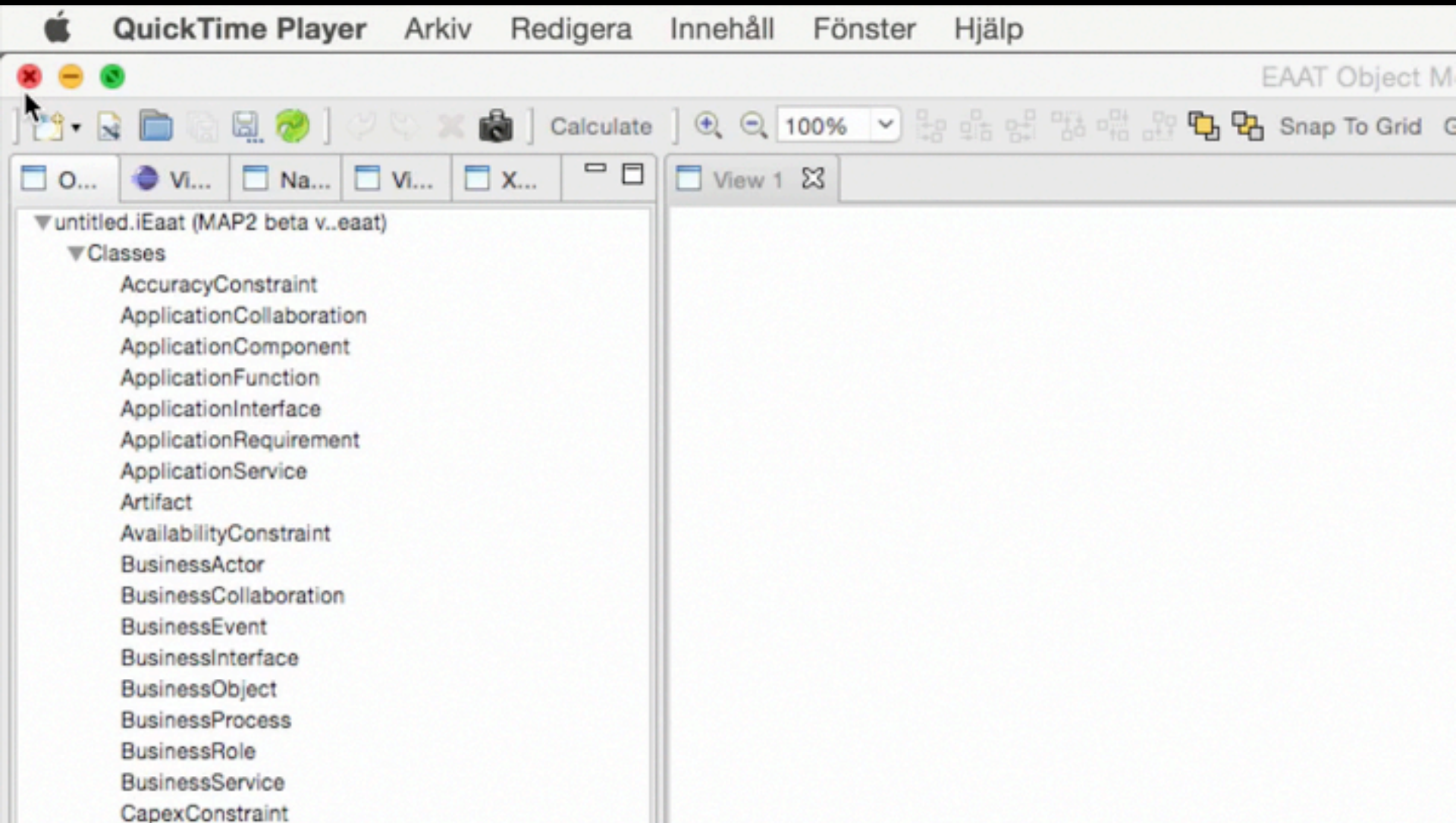
# Our EA tool



A demo...

(Download at: [www.ics.kth.se/eaat](http://www.ics.kth.se/eaat))

# Adding Applications





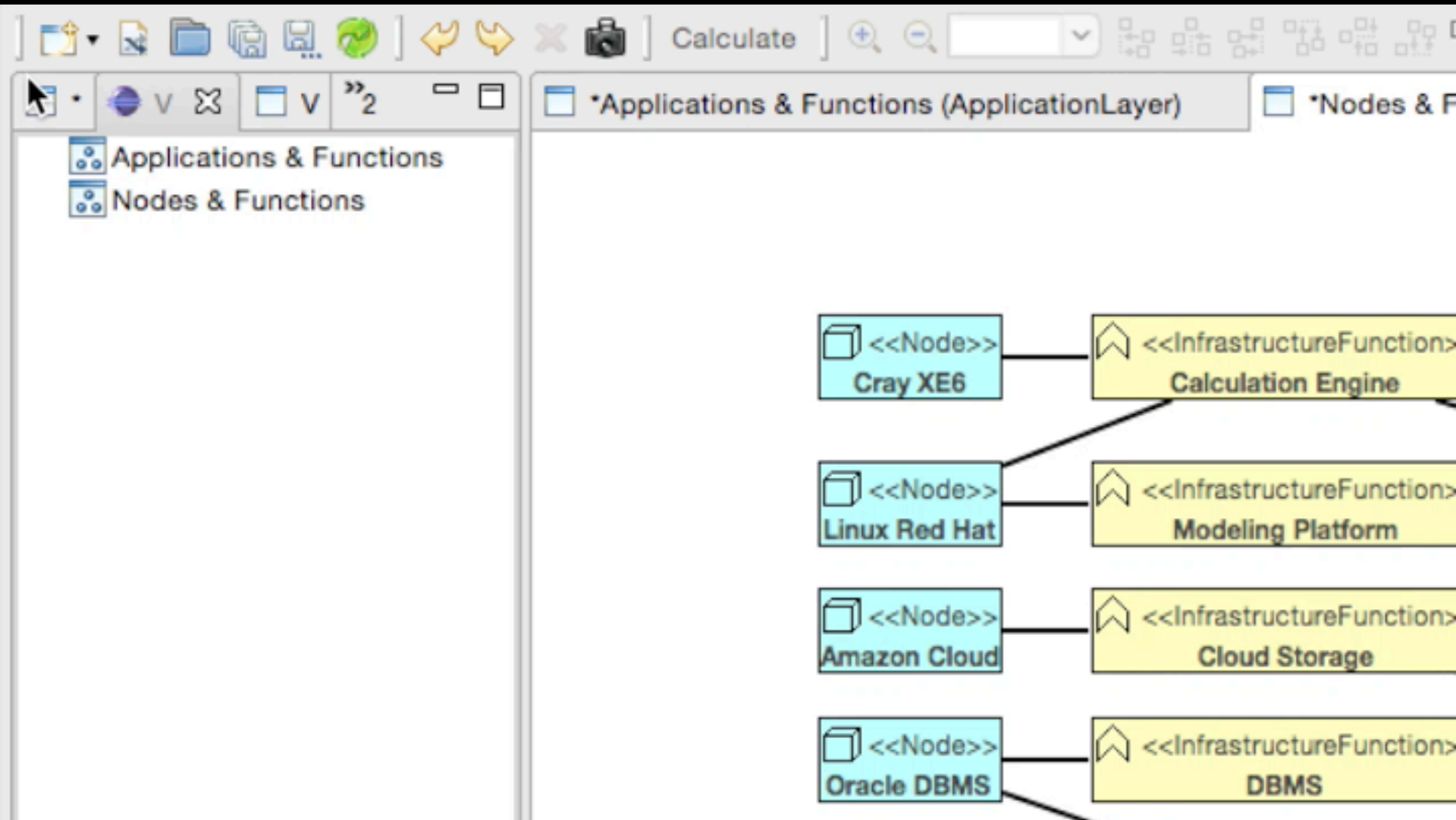
# Adding Infrastructure

The screenshot displays a software development environment with a class diagram in the center. The diagram shows a hierarchy of classes. On the left, a 'Classes' pane lists the following classes: ApplicationCollaboration, ApplicationComponent, ApplicationFunction, ApplicationInterface, ApplicationService, and DataObject. The main diagram area shows a series of blue boxes representing 'ApplicationComponent' subclasses and yellow boxes representing 'ApplicationFunction' subclasses. The 'ApplicationComponent' subclasses are: AMR System, CRM System, DSM System, SCADA, Forecasting System, DSM, and Planning System. The 'ApplicationFunction' subclasses are: EMS, AMR, Customer Support, DSM, Grid Control, Forecasting, Generation Control, and Production Planning. Lines connect each 'ApplicationComponent' box to its corresponding 'ApplicationFunction' box. On the right, a 'Property' pane shows a table with the following data:

Property	Value
Attributes	false
Link Label	false
Link Role N...	false
Movable L...	false
ObjectConn...	true
Mode	Default
Name	View 2

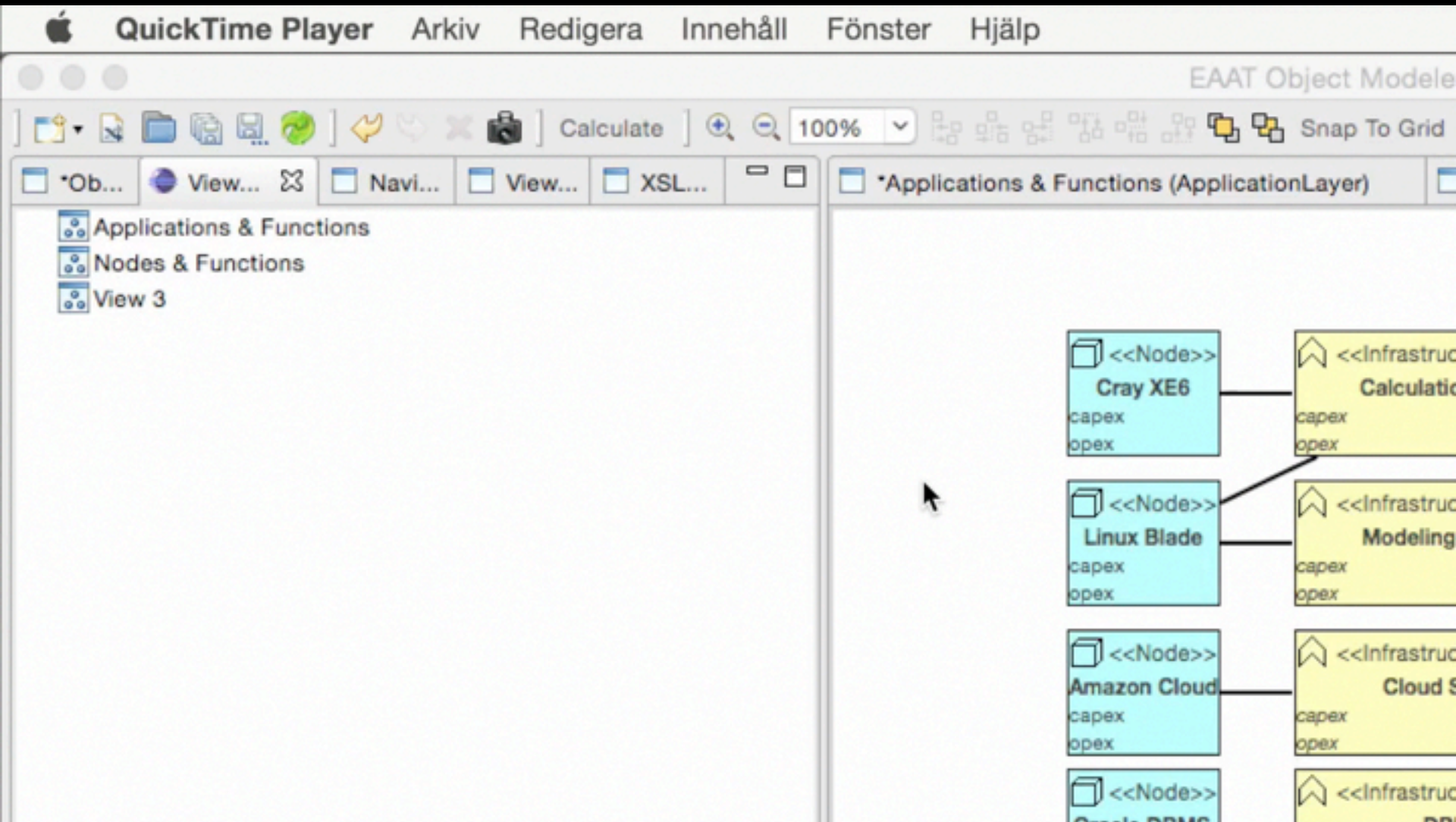
At the bottom of the window, there is a status bar with several tabs: Error Log, Calculation Problems, Invalid Invariants, Views Containing Object, DSM Element, and Console. The console shows the text: 'Xtext OCL for 'null' : null'.

# Cost View





# Calculating Infrastructure Costs





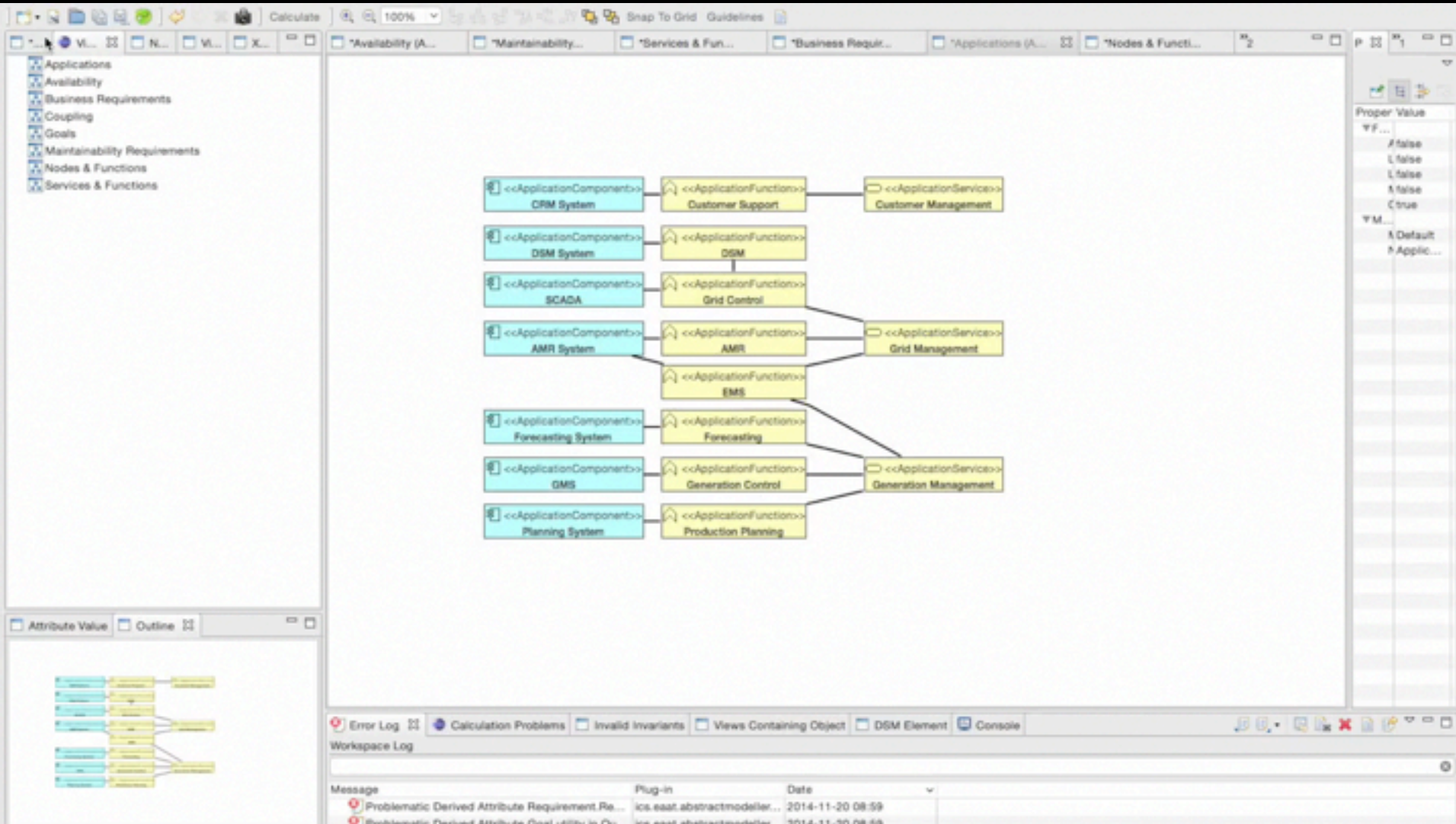
# Calculating Application Costs

The screenshot displays a software application interface for calculating application costs. The main workspace shows a hierarchical diagram of infrastructure components. On the left, a tree view lists 'Applications & Functions', 'Infrastructure', and 'Nodes & Functions'. The main diagram consists of nodes for hardware (Cray X86, Linux Blade, Amazon Cloud, Oracle DBMS, PostgreSQL), infrastructure functions (Calculation Engine, Modeling Platform, Cloud Storage, DBMS, Temporary Storage, Fault Reporting), and infrastructure services (Inference Service, Data Storage, Reading Service, Fault Detection). A Properties panel on the right shows details for the selected 'Infrastructure' object. The bottom status bar includes an Error Log and a Console window.

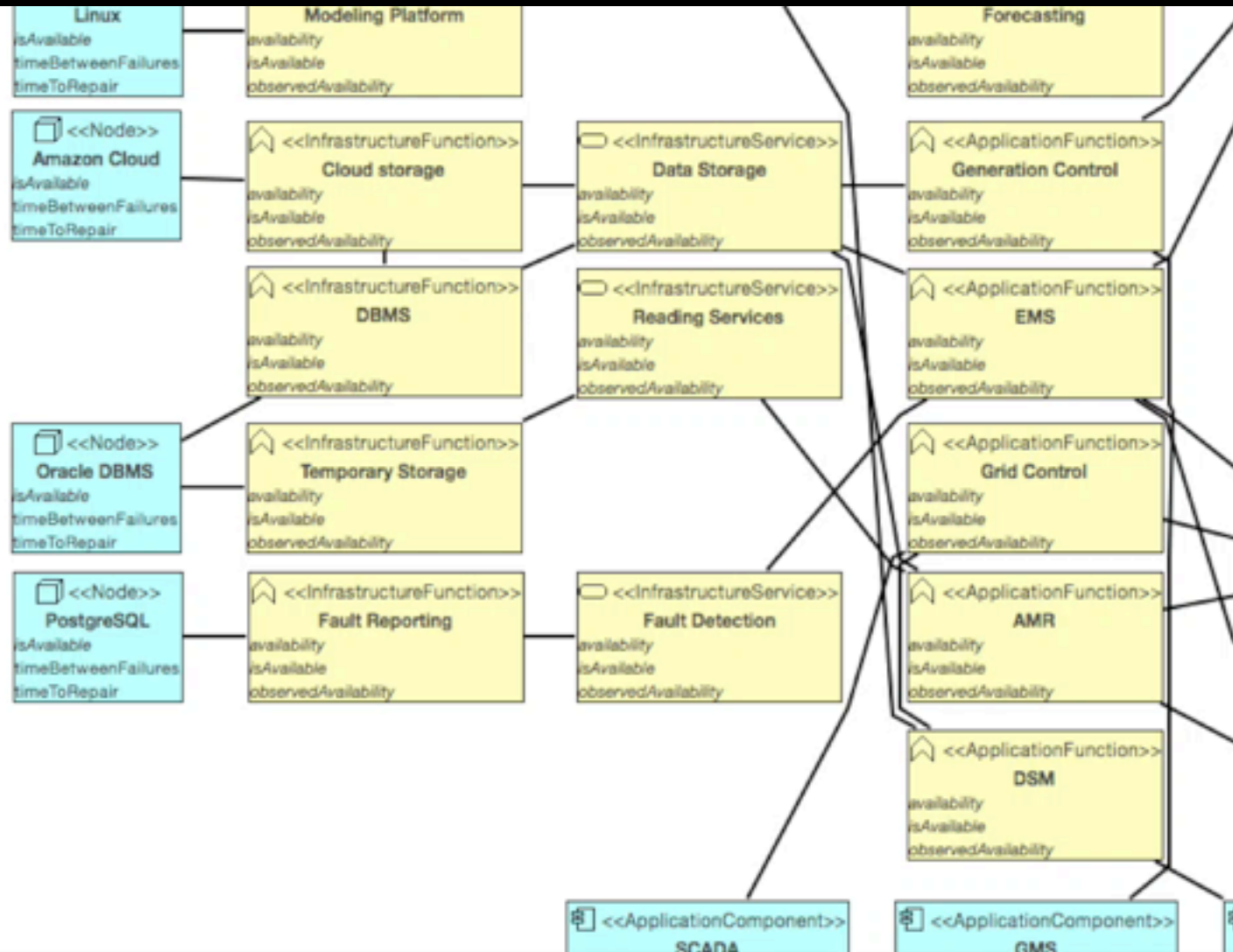
Property	Value
Attributes	true
Link Label	false
Link Role Names	false
Moveable table...	false
ObjectConnection	true
Mode	Default
Name	Infrastructure

Message	Plug-in	Date
Calculation result is empty. Sampling did not s...	ics.esaat.concretemodelle...	2014-11-18 14:41
Unable to create menu item "non-erl" of ut...	non-erl	2014-11-18 15:03

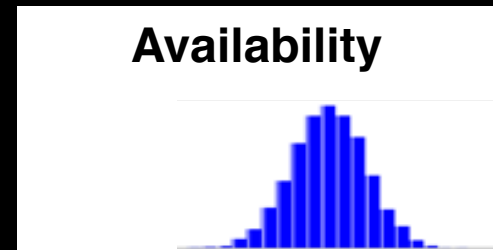
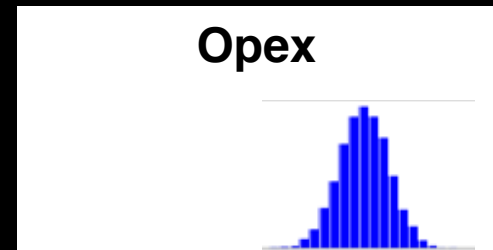
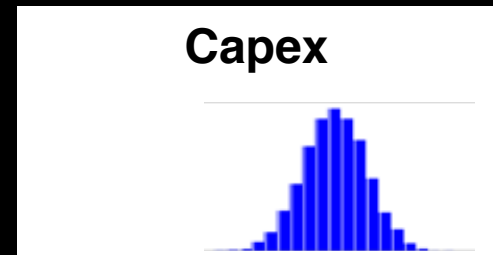
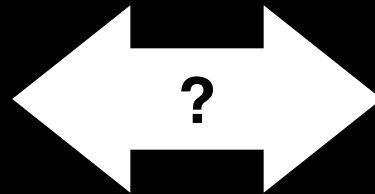
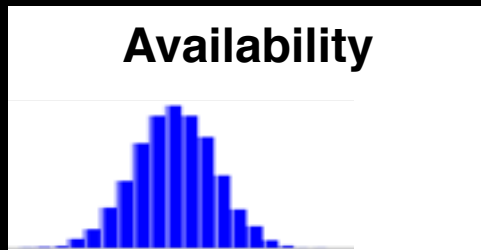
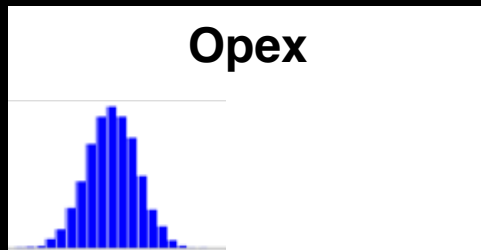
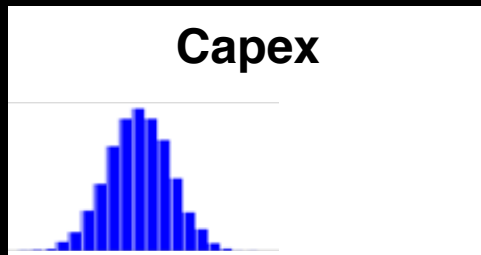
# Calculating Availability



# Introducing Redundancy

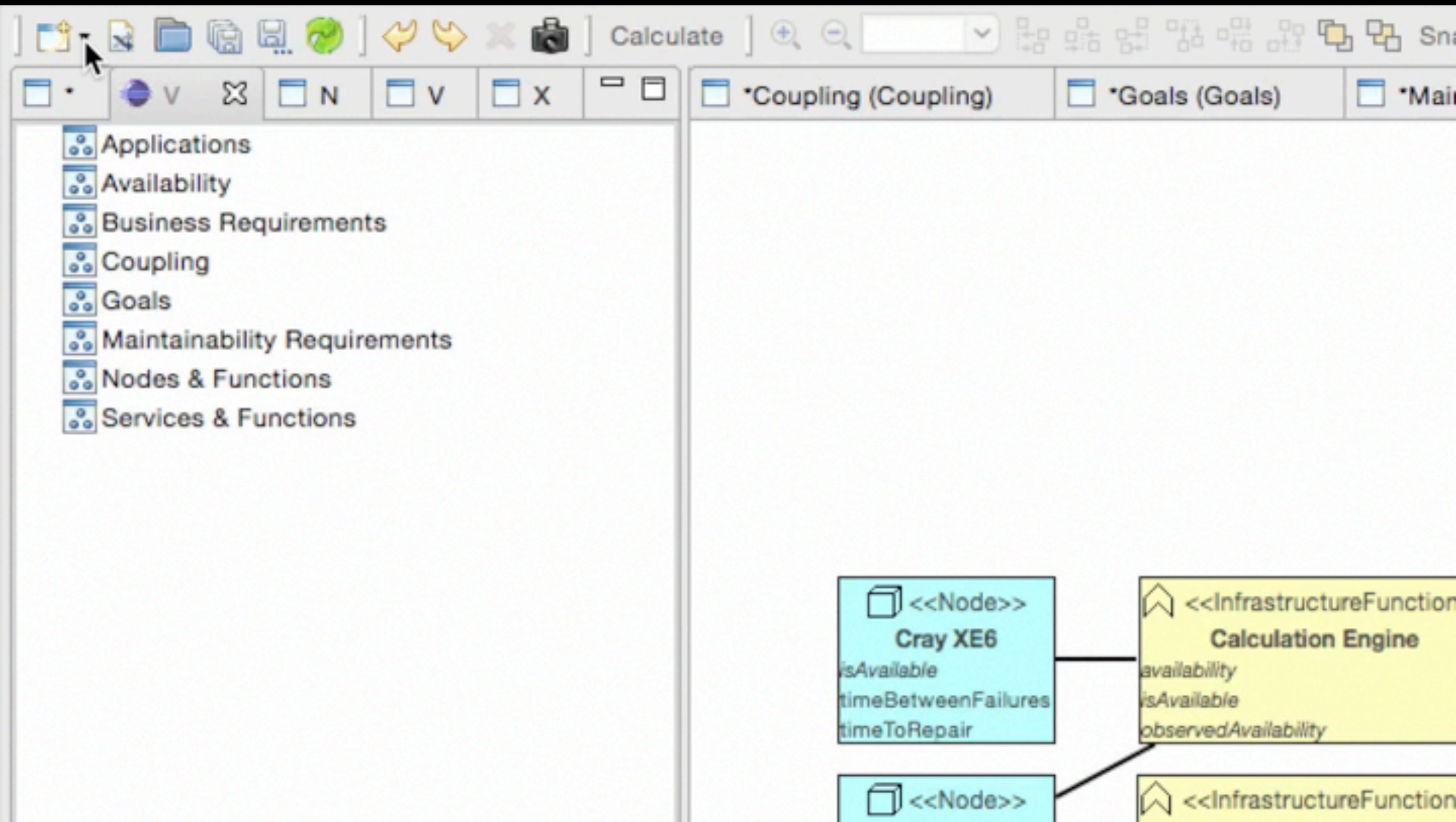


# Design Trade-offs

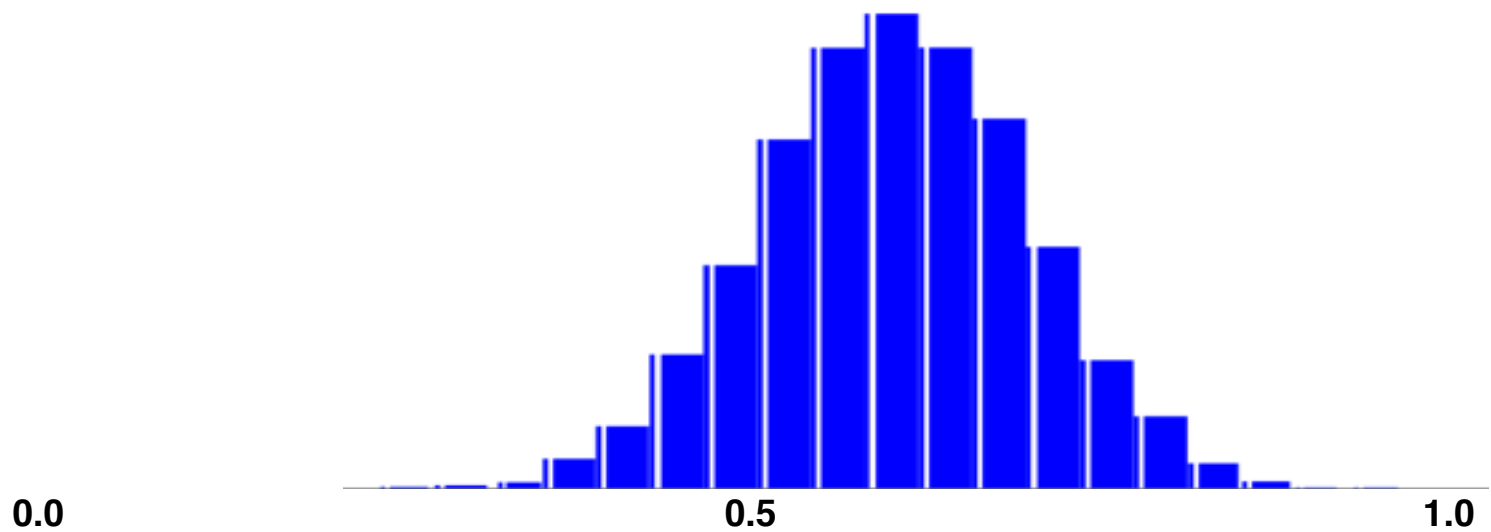




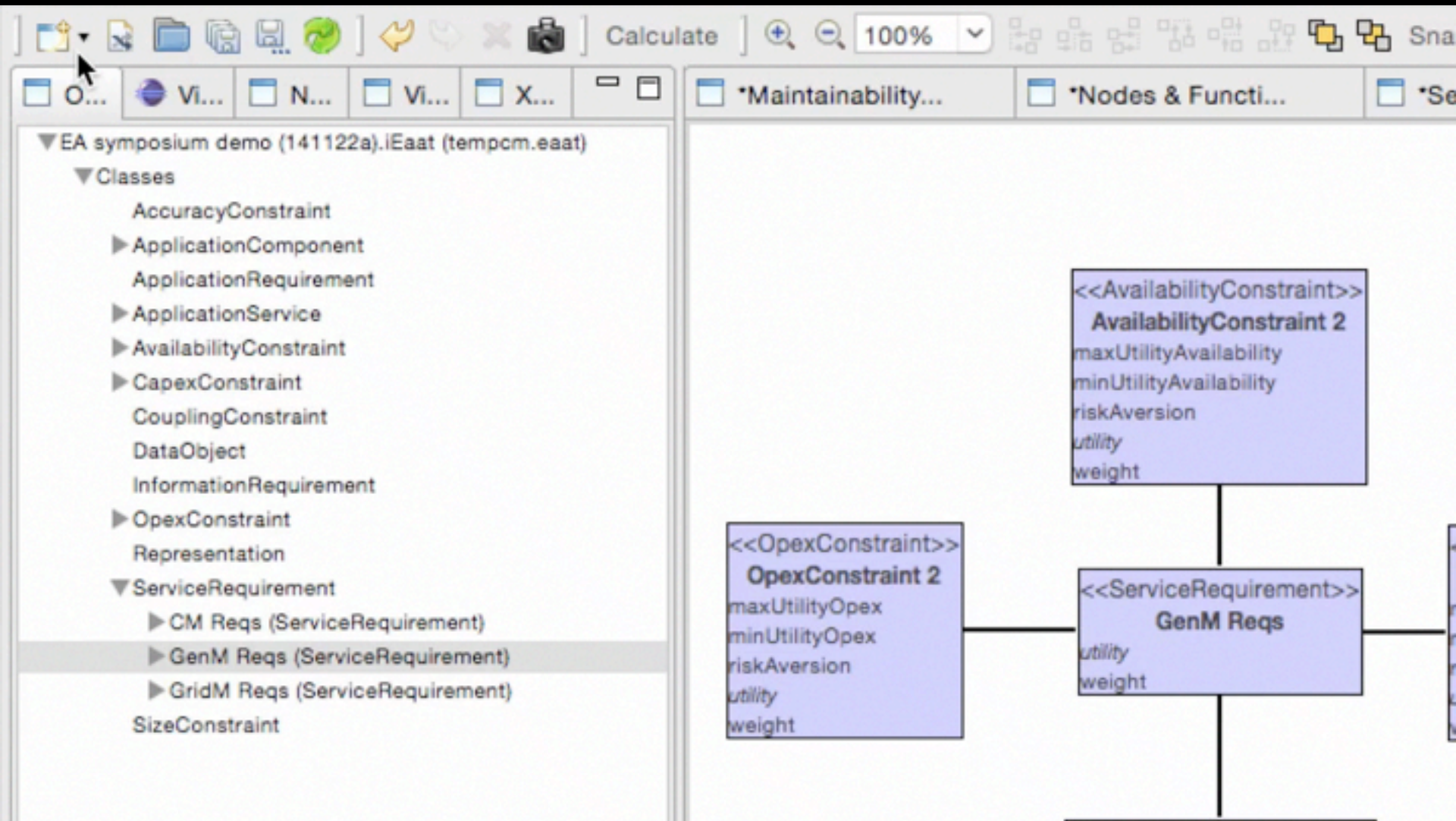
# Specifying Requirements



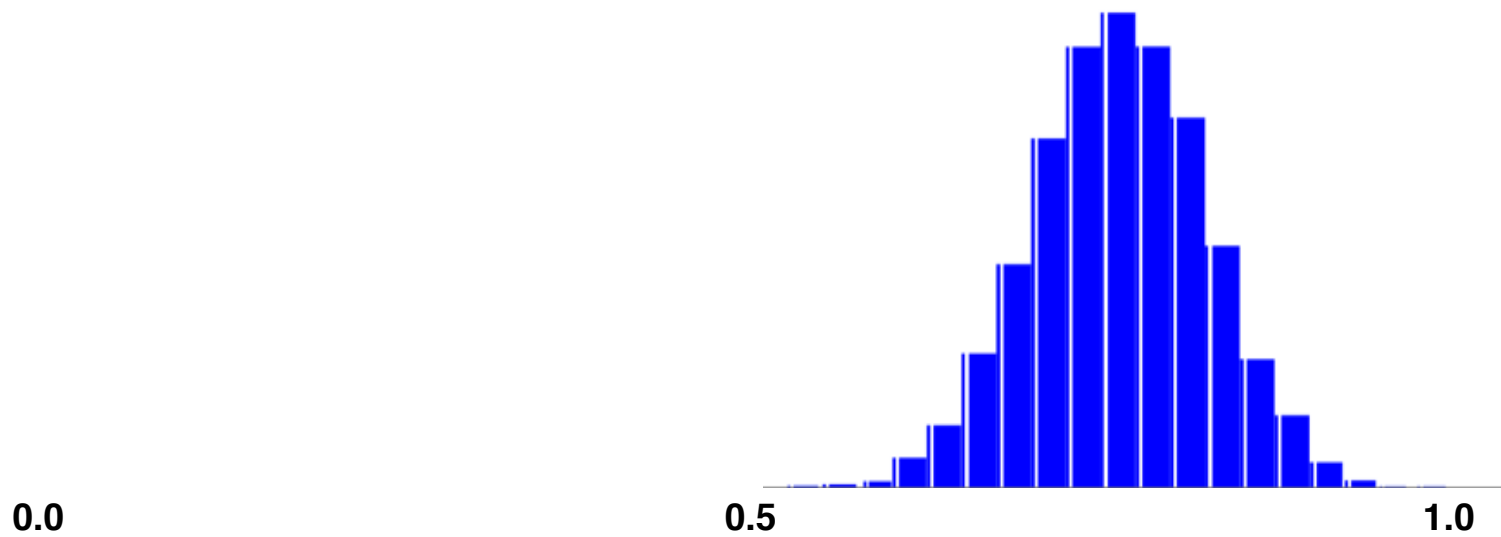
# CustomerManagementReqs.utility



# Prioritizing Requirements



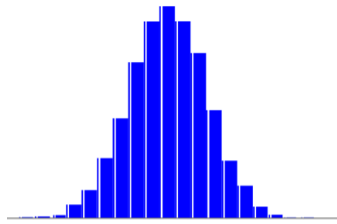
# OperationalQuality.utility





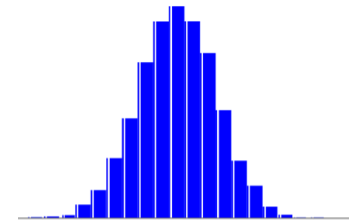
# Design Trade-off Aggregation

OperationalQuality.utility

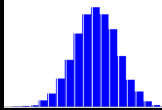


?

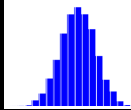
OperationalQuality.utility



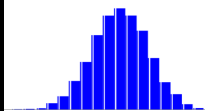
Capex



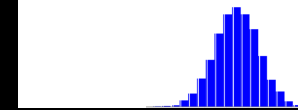
Opex



Availability



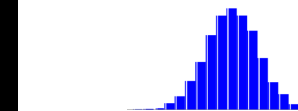
Capex



Opex



Availability





# Current quality attribute coverage

Requirements (utility)

Application service Availability

Application service cost

Data accuracy

Application component coupling

Application size

System interoperability

Information security



# Search-Based Software Engineering

Optimization techniques:

Hill climbing

Genetic algorithms

Simulated Annealing

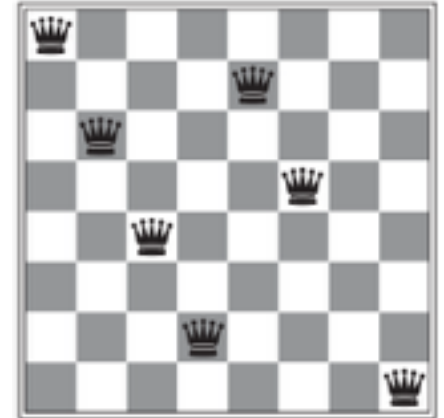
Particle swarm optimization

Ant colony optimization

Mark Harman, Bryan F. Jones,  
Search-based software engineering,  
Information and Software Technology, 2001



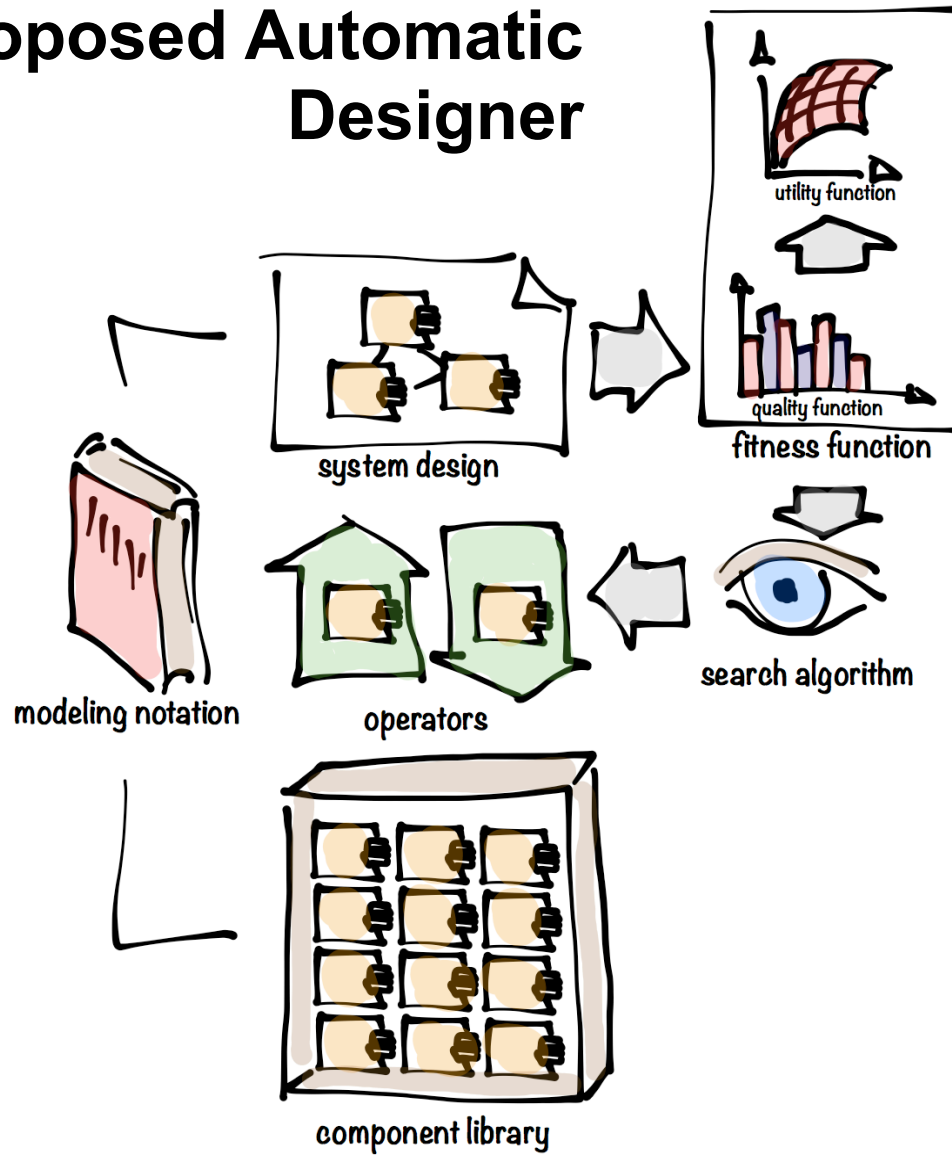
## Definition of a (search) problem



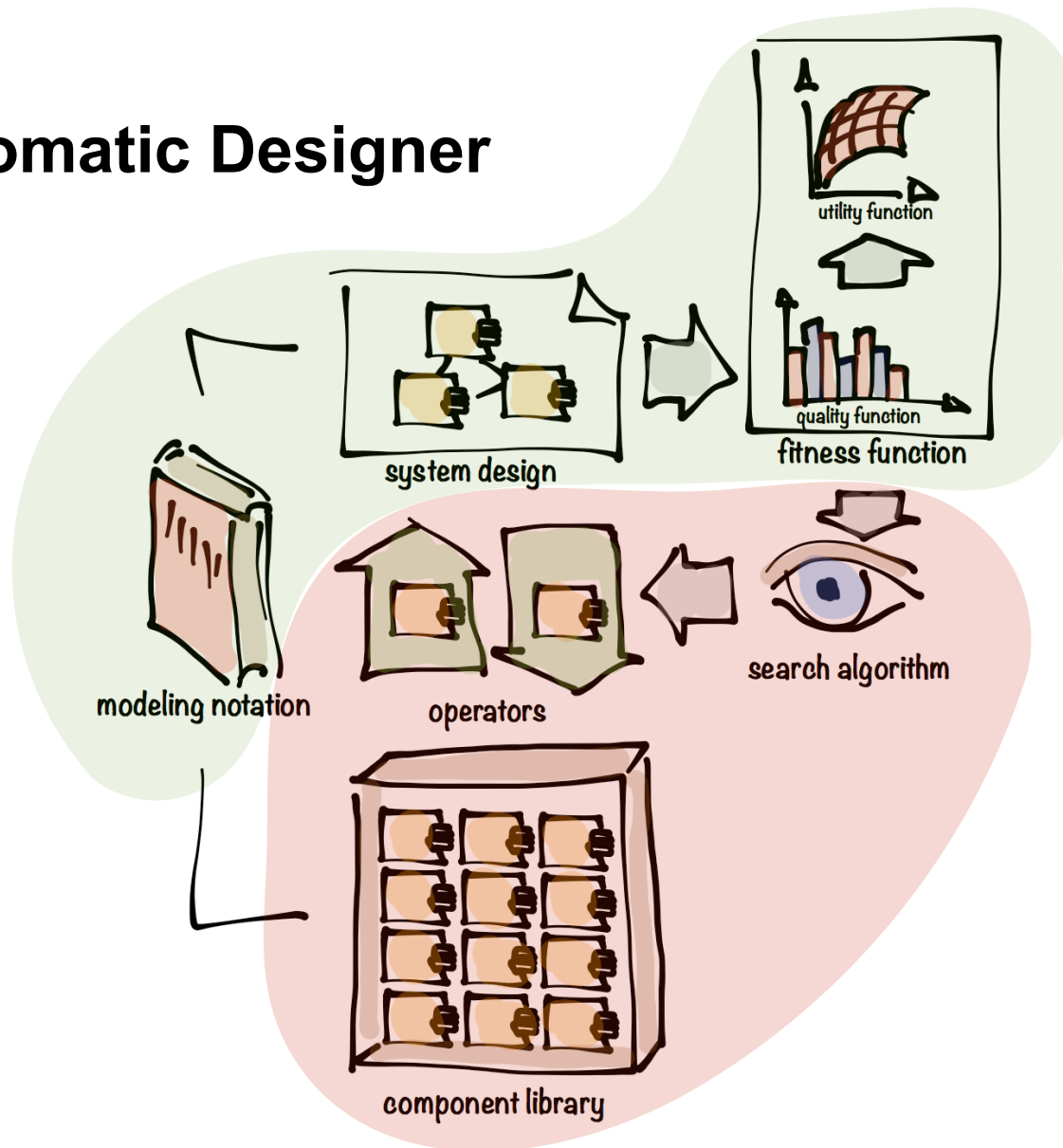
- The **initial state** that the agent starts in.
- A description of the possible **actions** available to the agent.
- A description of what each action does; **the transition model**.
- The **fitness function/goal test** is the characterization of what is considered to be a good solution.
- A **path cost** function that assigns a numeric cost to each path.



# Proposed Automatic Designer



# Automatic Designer



# Joint Utility Function

