Synergy - a language and framework for robot design

“a language to represent robots with a framework to generate simulation, optimize design and generate control software increases the effectiveness of design”

Lalitesh K Katragadda
Robotics Institute
Carnegie Mellon

Lunar Rover Initiative
Field Robotics Center

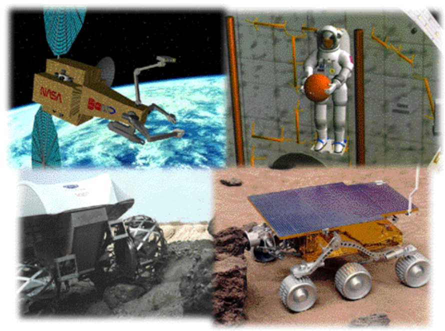
Carnegie Mellon

motive and purpose

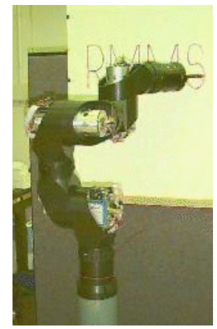
robots spread

robots diversify

safety

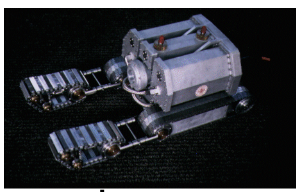


exploration



factory

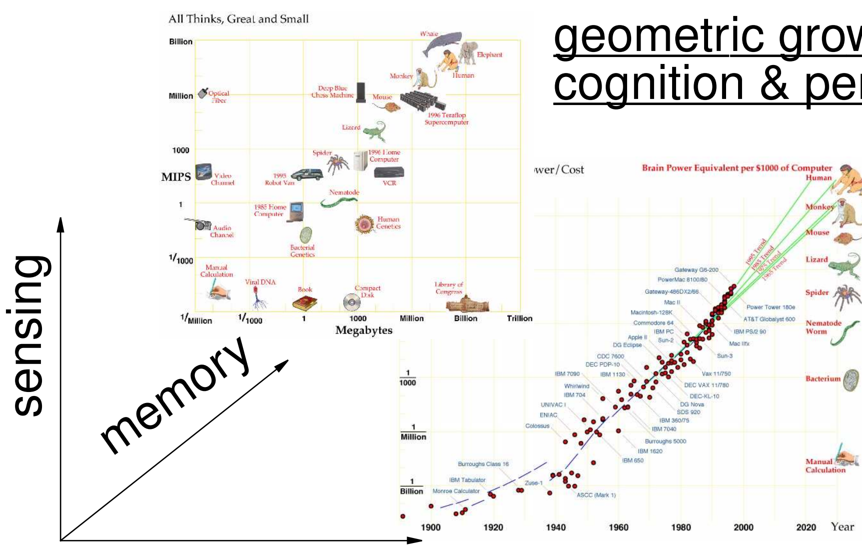
hazardous



underwater

service

geometric growth
cognition & perception

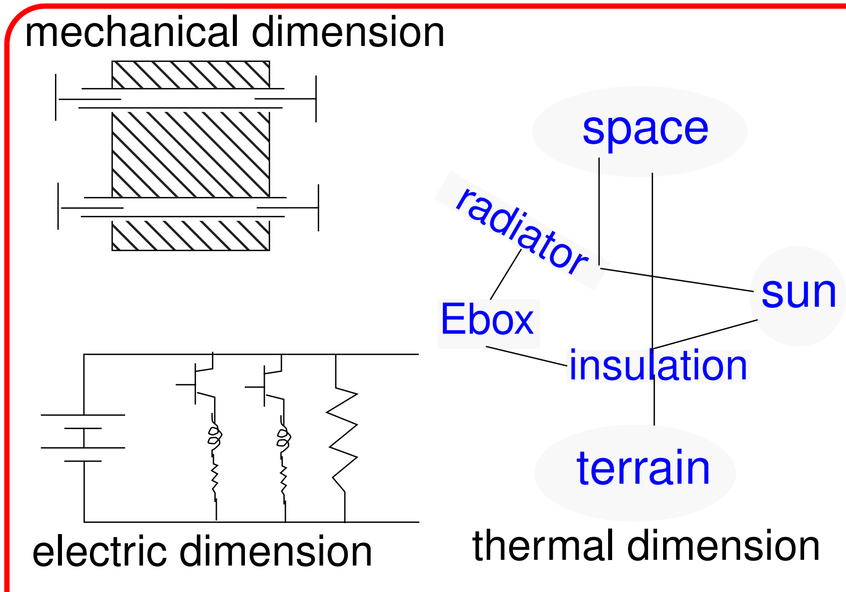


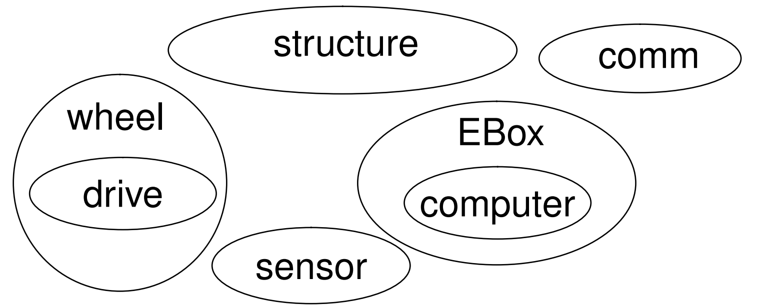
computing

long road
to dream

Carnegie
Mellon

seven blind designers and a robot



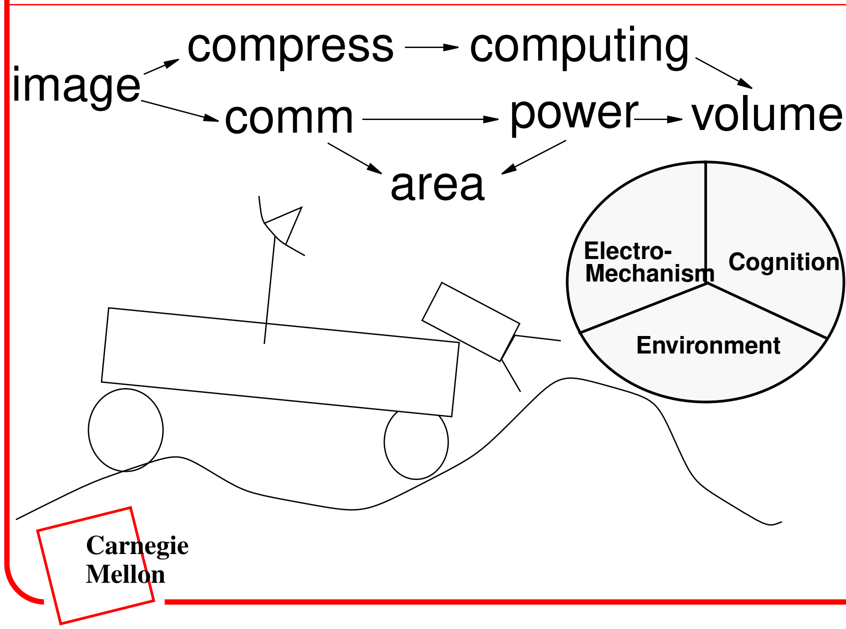


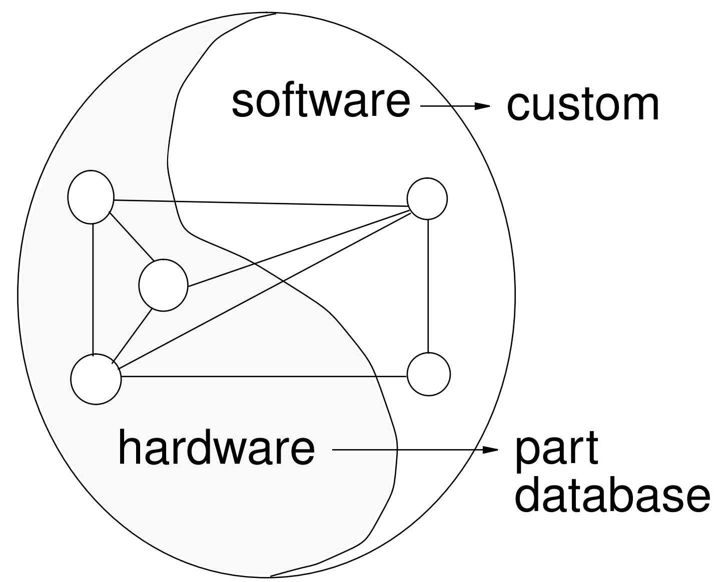
<u>Parts of an Elephant</u>

Mind equipped to see only 3-6 any time

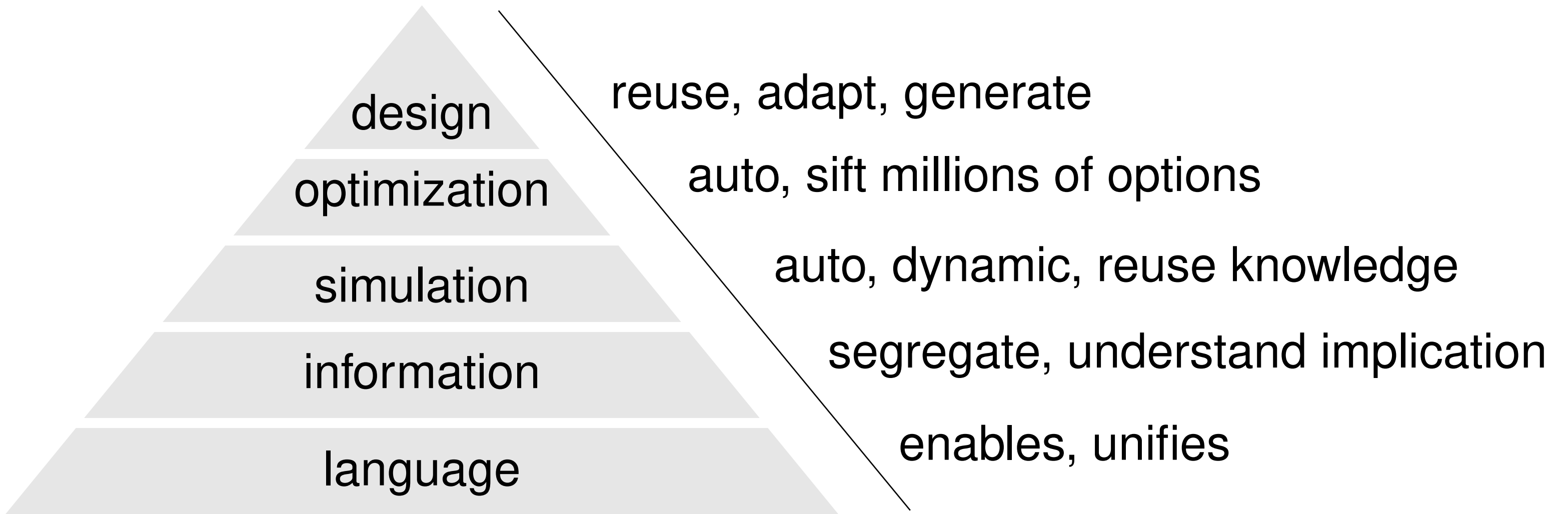
Designers get smarter w expertise

Designers stretched, overlook often





genesis of a design environment



Themes:

- Comprehensive: enable and unify robot dimensions
- Segregate: knowledge, tracking & complexity from design and representation
- Enable rapid reaction to design and information change
- Minimize effort and expertise needed by designer
- Maximize flexibility, modularity and reusability of design

synergy

Software	Needs
[L] language spreadsheet	[O]
[S] simulator	[L] [C]
[O] optimizer	[L] [S]
[C] control software generator	[L] [S]

Approach:

Focus on a useful and comprehensive language spreadsheet

Develop framework for [S] [O] [C]

Formulate critically needed control software generator

Implement first-cut versions of all three and integrate with [L]

Carnegie Mellon

background on design environments

Tool Name	Domain size	Representativ Comprehension	Multi dimensionality	Dynamic modeling	Optimize ability	Software generation
Viper	Lo	Med	Lo	Med	Med	No
ProE	Hi	Lo	Lo	Lo	Lo	No
Algor	Hi	Med	Med	Lo	Med	No
TeleGrip	Med	Lo	Lo	Med	Lo	Lo
VLSI	Lo	Med	Med	Lo	Hi	Hi
Manipul.	Lo	Hi	Hi	Hi	Med	Hi
Synergy	Hi	Hi	Hi	Med	Med	Med

general, integrated, multi-dimensionality absent

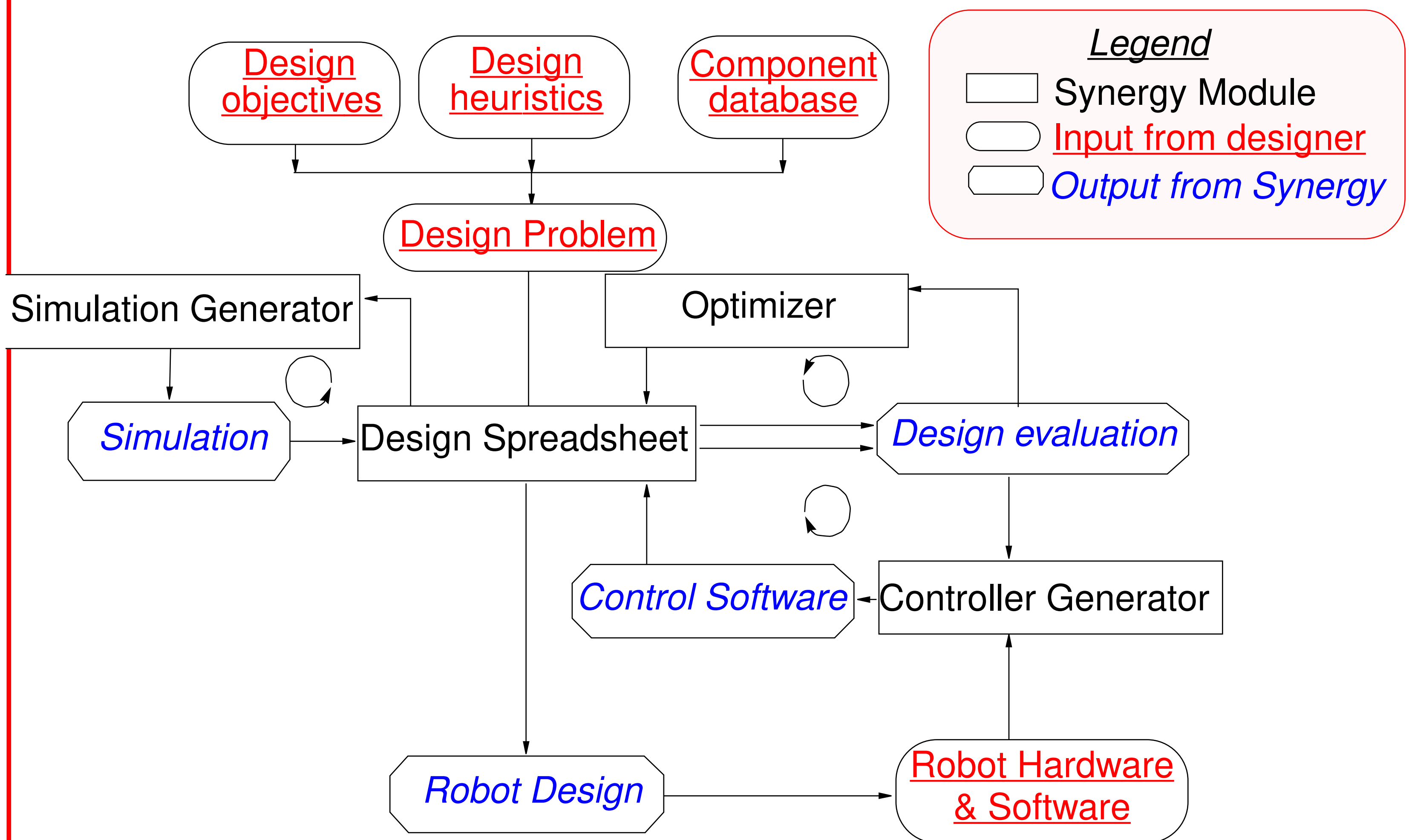
Existing Tools

Robots
- Graphic displays
- Dynamics modeling
- Sensor simulation
- Software interface
- Geometric modeling
- Robot subsets

Commercial Artifacts (chips, cars)
- Specialized System models
- Detailed component analysis
- Static dimensional analysis

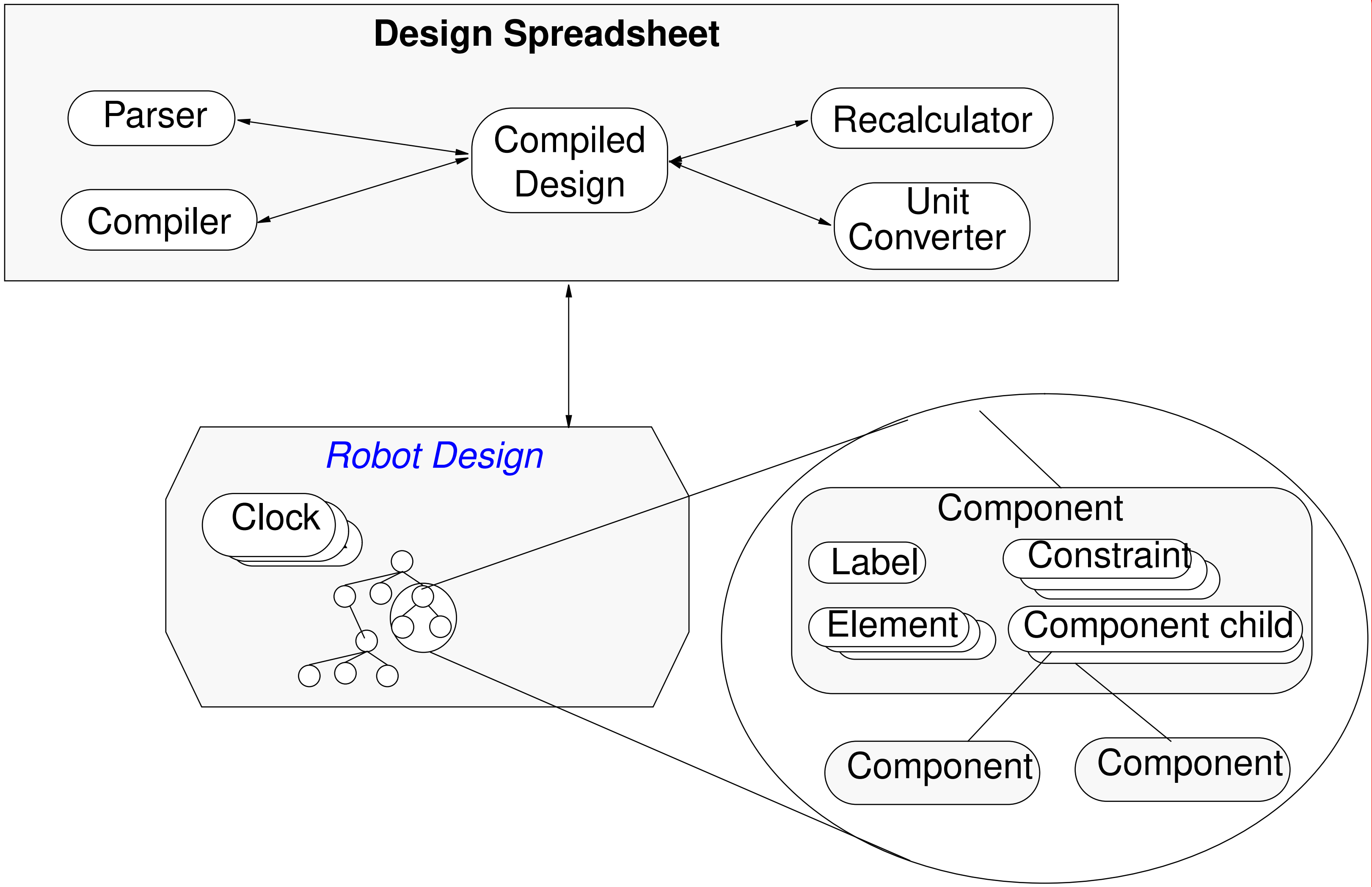
Carnegie Mellon

synergy overview



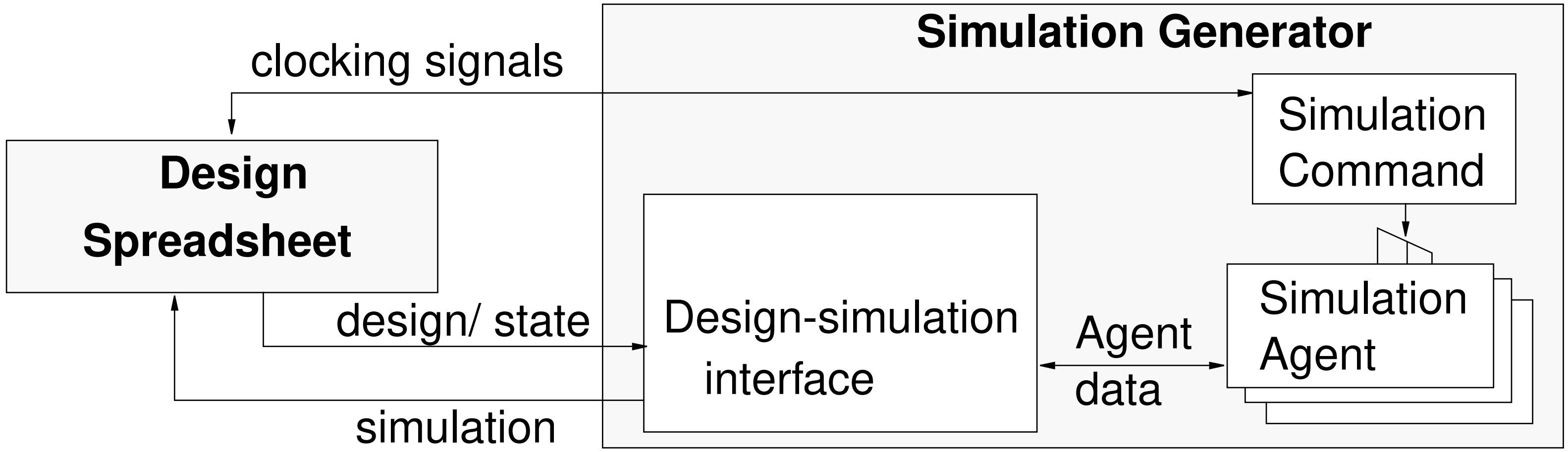
Carnegie Mellon

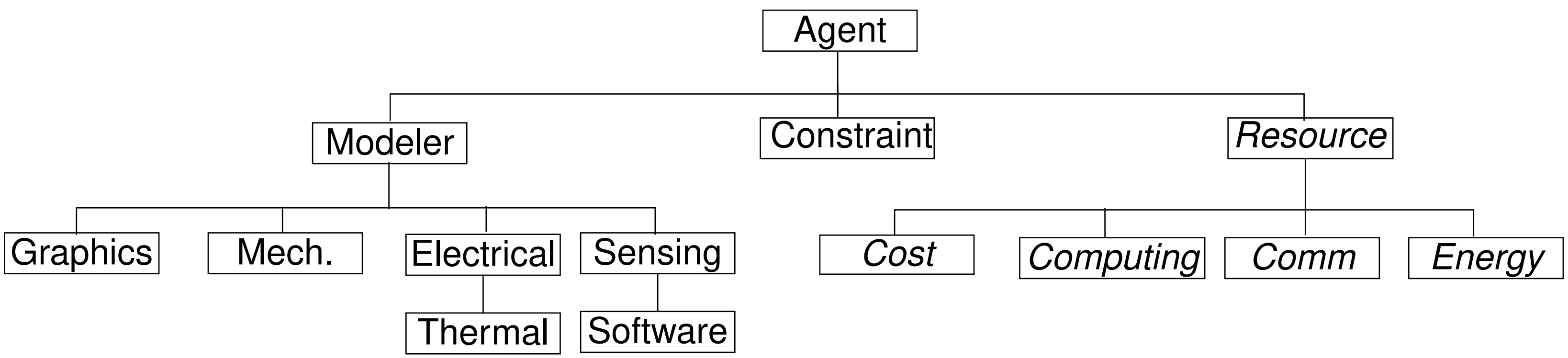
design spreadsheet



Carnegie Mellon

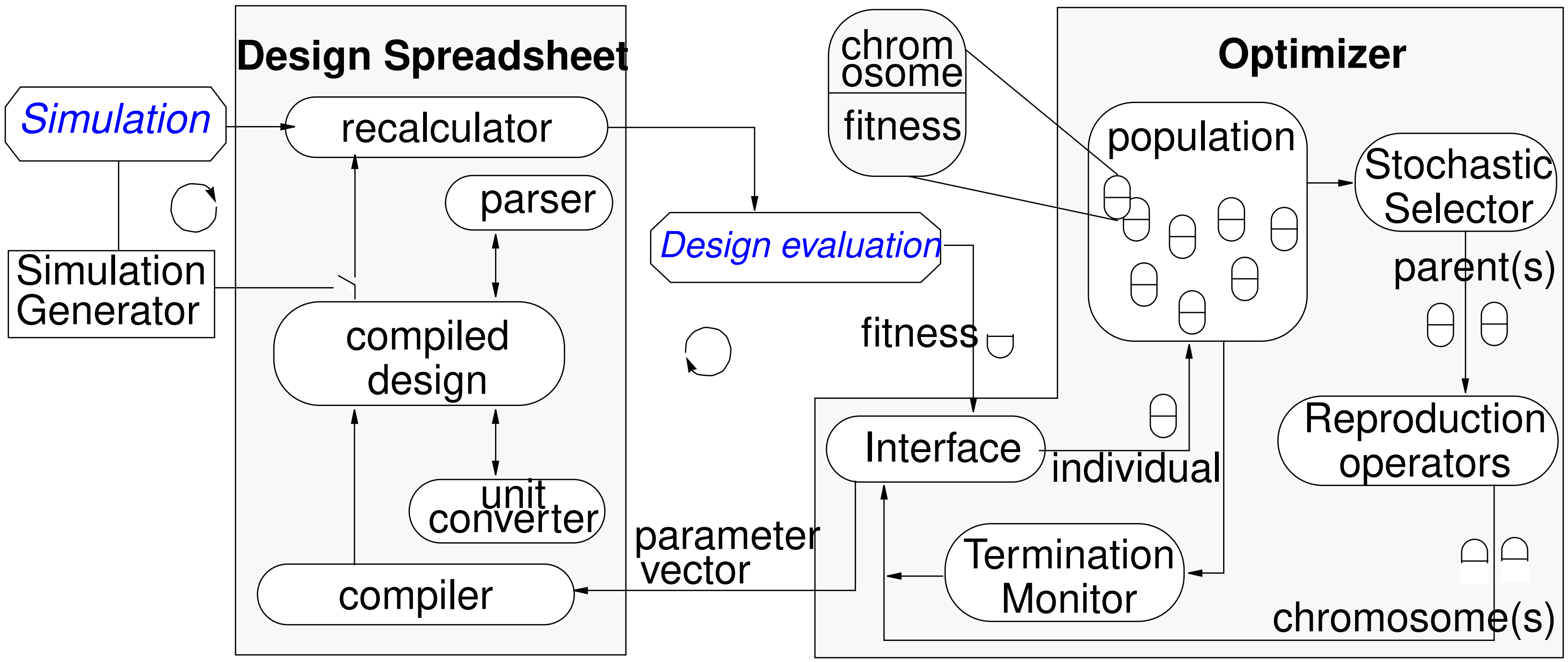
simulation generation





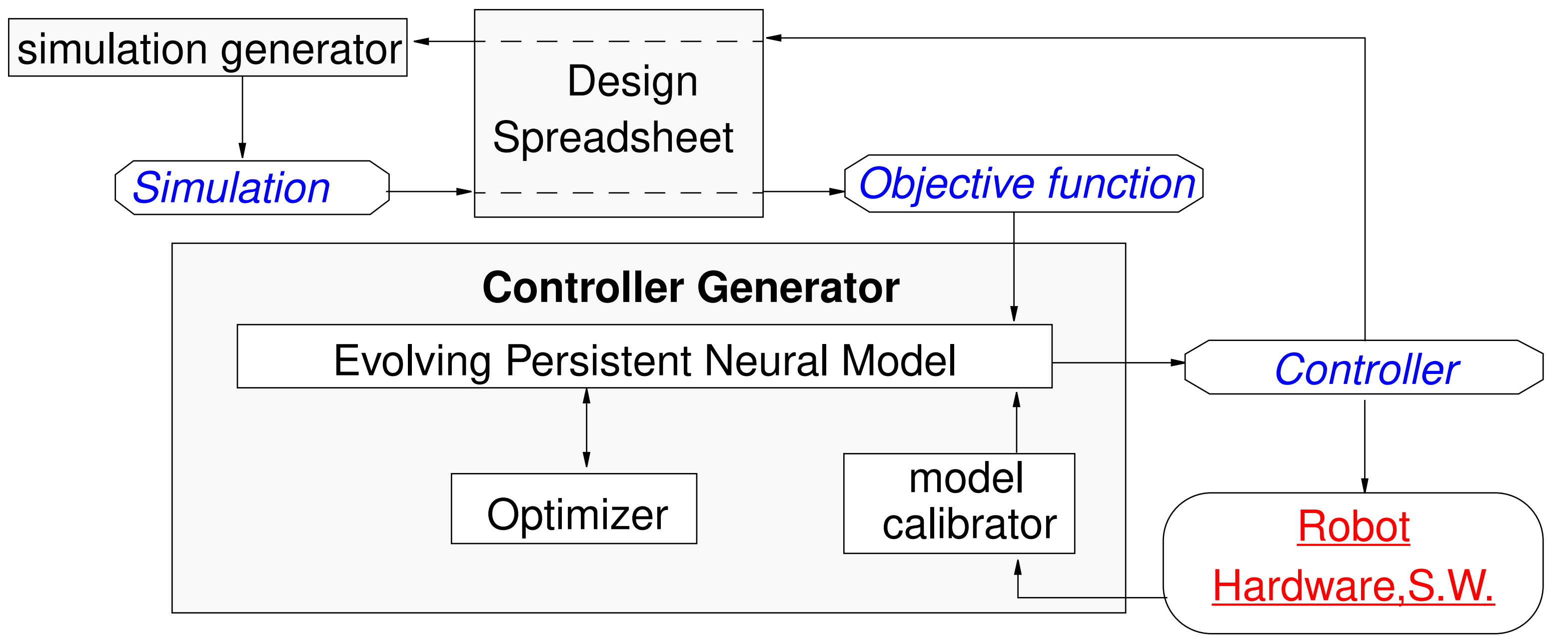
Carnegie Mellon

design optimization

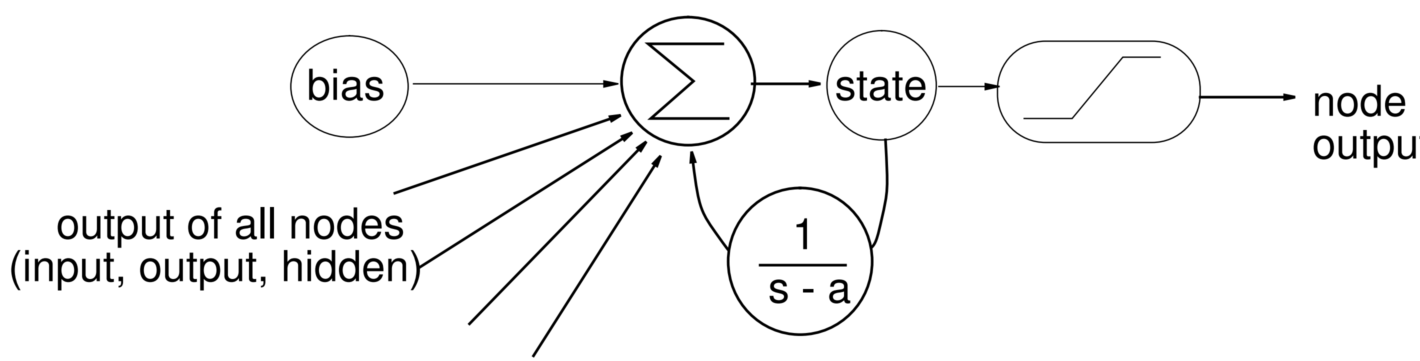


Carnegie Mellon

control software generation



Neuron Model



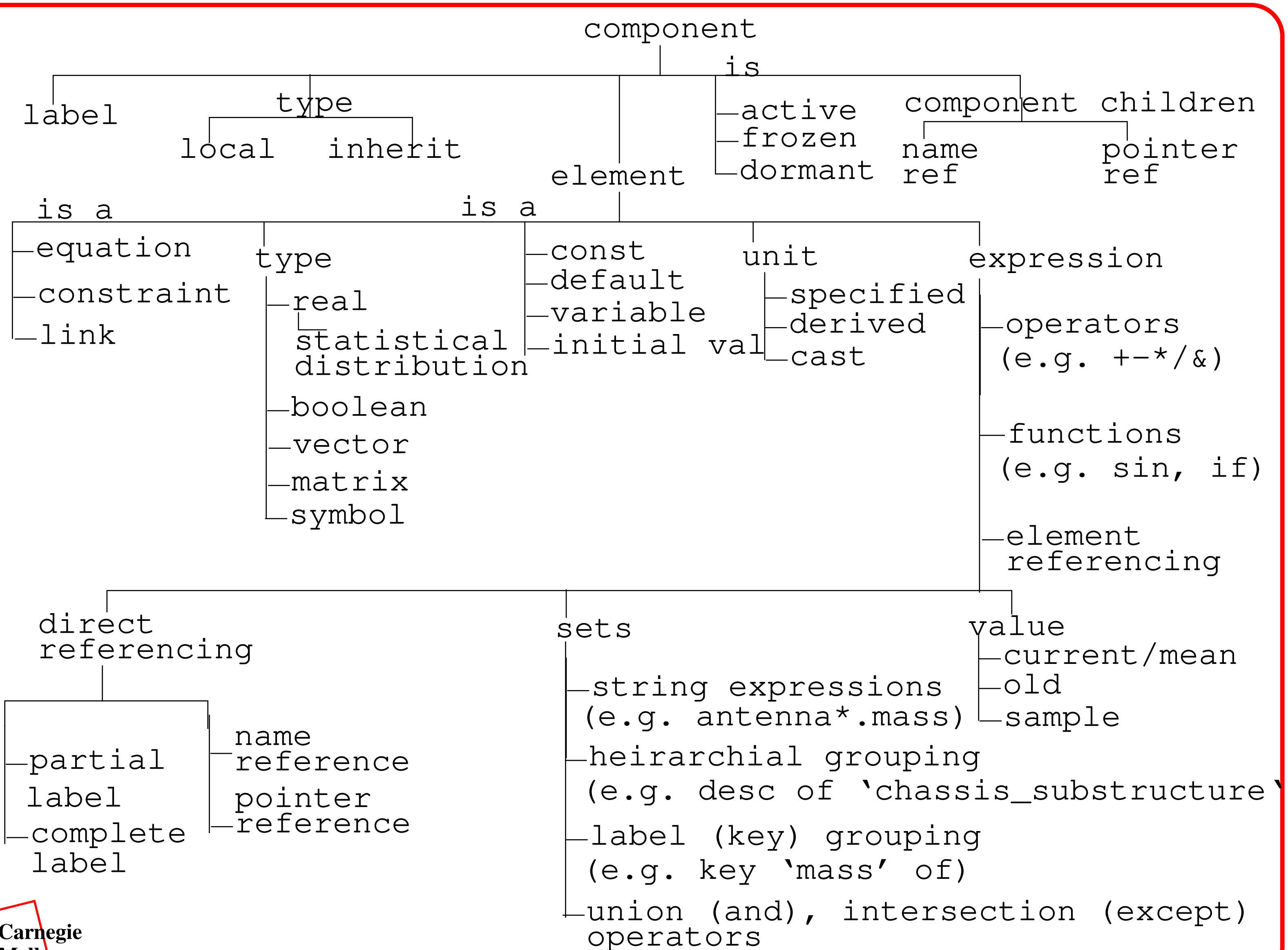
Carnegie
Mellon

example component

```
Component CameraPanoSphere {
    type insulated;
    resolution = 1024 * 1024 * 3 byte;
    FovH = 360 degree;
    FovV = 270 degree;
    N = (int) FovHneed*FovVneed/(FovH*FovV);
    freq = 1 Hz;
    area = 20 cm * 20 cm * 5 * N;
    mass = N * 3 kg;
    power = (8 watt + 10 watt*(freq/14 Hz)) * N;
    cost = 1.0 megadollar + 1.0 megadollar*N;
    Component compression  : Fractal{}
    mipsDint = N * resolution * freq *
        compression.computing;
    downlink = N * resolution * freq / compression.ratio;
    resolution <= 1024*1024*3 byte;
    freq <= 14 Hz; }
component comm {
transparent downlink = sum (key downlink of rover); ...
```

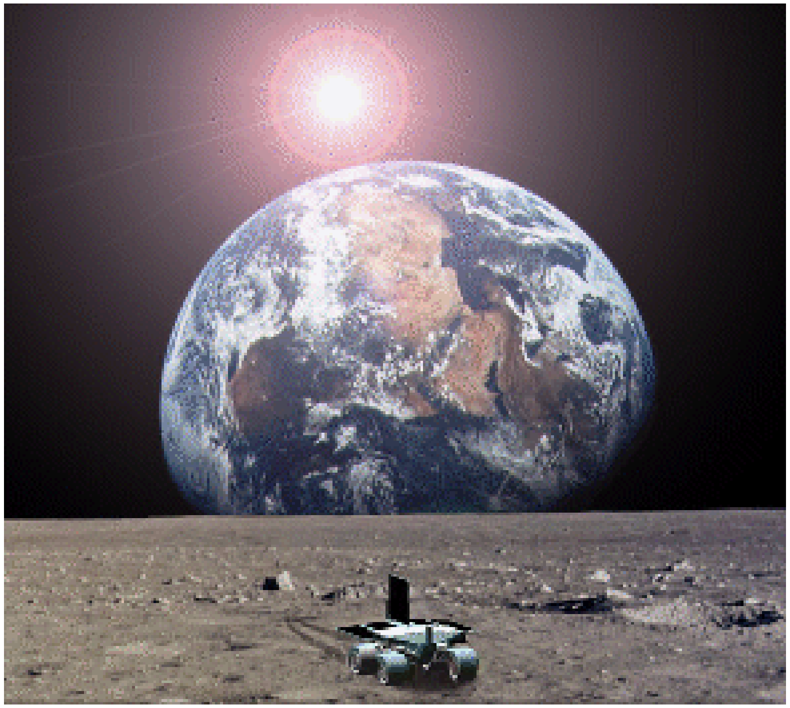
Carnegie Mellon

language structure



case study and demonstration

lunar rover design



real chartered problem
unique and challenging
broad technology
multiple designs

Carnegie
Mellon

lunar rover design

Four design objectives:

- Ice discovery
- Edutainment
- Earth rise 2001
- Lunar mining (ISRU)

Technology encoded in language:

- launcher profiles
- landing propulsion analysis
- Camera and sensor analysis
- Comm link analysis, options (parabolic, phased, annular)
- Computing analysis, chip option, connector analysis
- Safeguarding analysis and constraints
- Locomotion torque, power, flotation, wheel size analysis
- Others: Thermal; Power; Programmatics, margins

Carnegie Mellon

example technology

Propulsion:

$$\mu = e^{\left(\frac{deltaV}{Vjet}\right)} ; M_{deltaV} = \left(\frac{\mu - 1}{\mu}\right) M_f ; V_{jet} = I_{sp} \cdot 9.81$$

$$M_{hover} = \frac{M_f}{\left(\frac{I_{sp} \cdot 9.81}{(g \cdot T)} - 1.0\right)}$$

Thermal:

$Q + H = P_r + P_i$;Q: enclosure heat, H: RHU heat

Pr: Radiator dissipation, Pi: Insulation loss

$$P_i = \langle 5.67 \times 10^{-8} \cdot A_i \cdot Ei_{IR}\left(T_i^4 - T_{ie}^4\right)\rangle - \langle 1358 \cdot a_S \cdot Ei_{solar}\rangle$$

Ai: insul. area; Ei: emissivity; ir/solar: emission spectra

As: enclosure area incident to sun; Ti/ie: int/ext Temp,K

Carnegie Mellon

lunar designs generated

Comparison with former designs:

	Ice Discovery	Edutainment	Earthrise
Synergy Design	258 kg	163 kg	96 kg
Former Design	329 kg	205 kg	110 kg
Improvement	71 kg, 21%	42 kg, 20%	14 kg, 13%

Perturbation analysis of various payload additions:

Perturbation	Ice Discovery	Edutainment	ISRU	Earthrise
Mass, 1 kg	2.89 kg, 0.94 W	3.02 kg, 1.74 W	2.80 kg, 1.61 W	2.61 kg, 0.42 W
Power, 1 W	0.61 kg, 1.66 W	0.56 kg, 1.78 W	0.31 kg, 1.65 W	0.30 kg, 1.51 W
Communication, 1 kbps	0.52 kg, 1.05 W	0.03 kg, 0.06 W	0.00 kg, 0.17 W	0.01 kg, 0.04 W
component max T, -1^{o} C	0.02 kg, 0.01 W	0.00 kg, 0.00 W	0.30 kg, 0.30 W	0.40 kg, 0.06 W
component min T, $+1^{o}$ C	0.06 kg, 0.00 W	0.07 kg, 0.04 W	0.01 kg, 0.12 W	0.00 kg, 0.00 W
Drive efficiency, 1.0/1.1	3.69 kg, 10.5 W	3.38 kg, 11.5 W	8.53 kg, 50.9 W	0.26 kg, 1.76 W

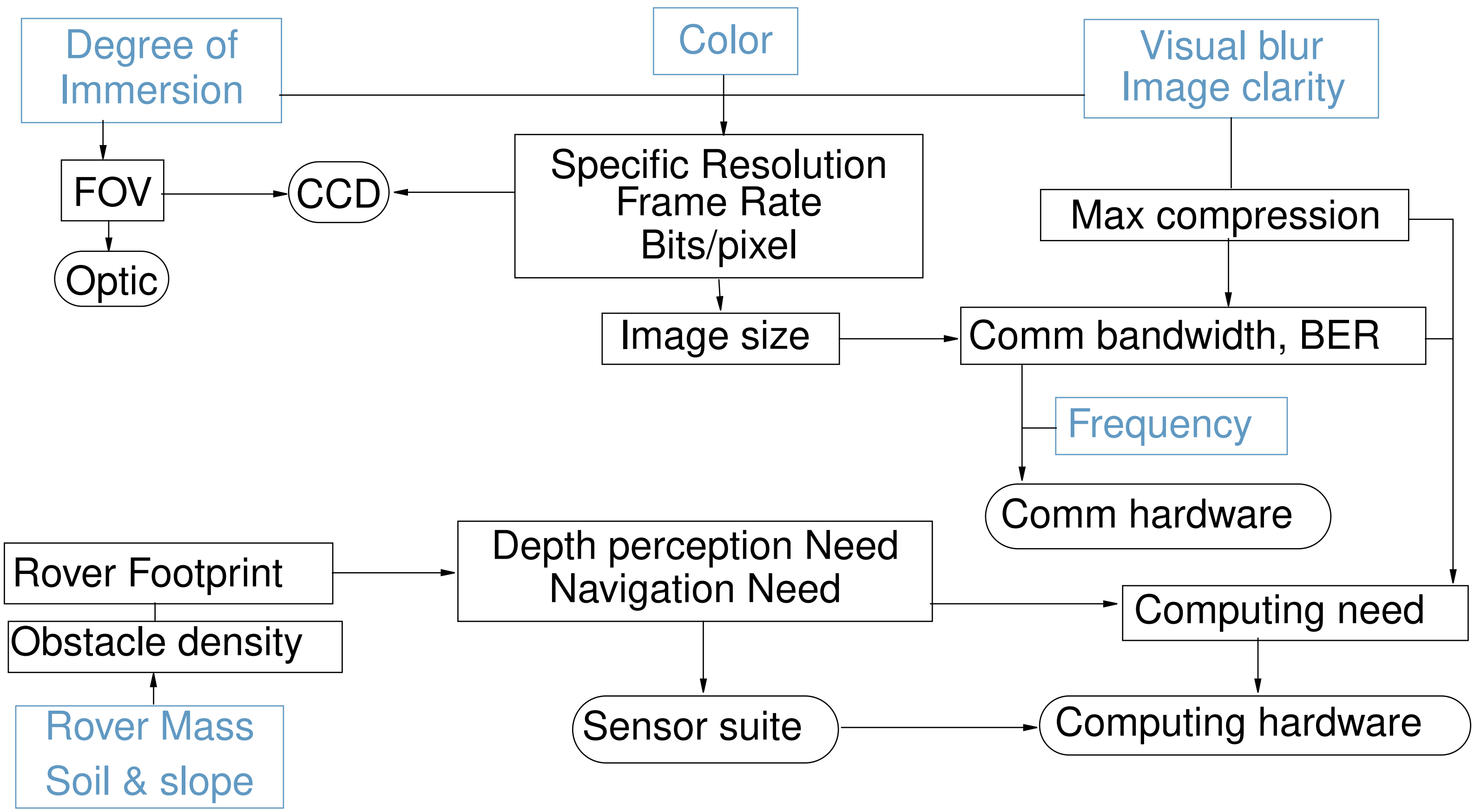
Carnegie
Mellon

ice discovery rover design trade-off

Traded Quantity	Old Value	New Value	Major Quantity Affected	Delta Change, New Value	Rover & spacecraft Mass
Baseline Design					314 kg, 1311 kg
Imaging Speed	1 Hz	0.1 Hz	Com gain	6.7 dB, 0.8 dB	312 kg, 1302 kg
Cold sortie hrs	30 hrs	20 hrs	Battery	-18 kg, 30 kg	265 kg, 1143 kg
Comm Xmit power	15 W, fixed	12 W, var	Com power	-10 W, 40 W	258 kg, 1118 kg

Carnegie
Mellon

payload causal dependency

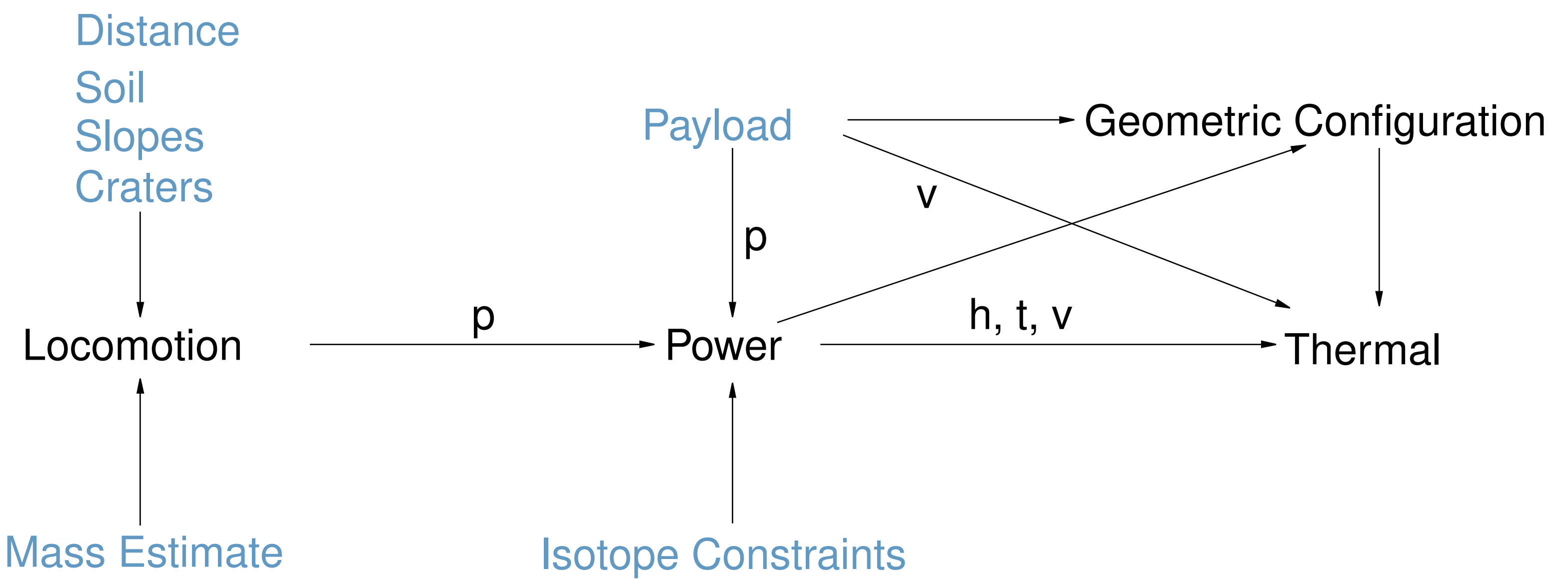


Carnegie Mellon

budget resource flow

Software regenerates design given changes in-

- Component technology
- Payload
- Margins



m: mass, p: power, v: volume & area, h: heat, t: temp. constraints

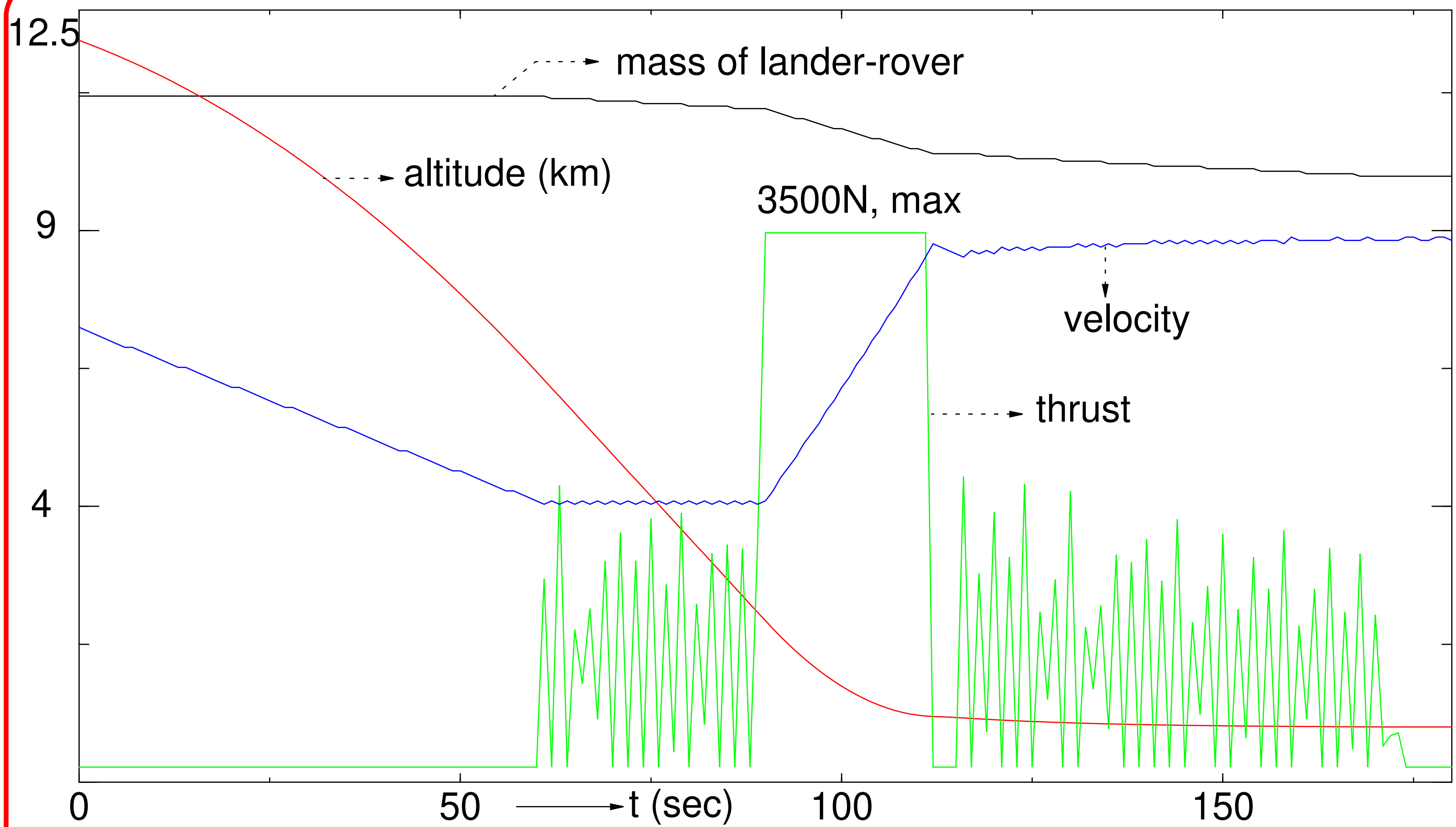
Carnegie
Mellon

simulation and software generation

Simulation and control software generation of Mars 98 lander:

- Landing technology is critical and expensive
- Mars 98 lander seems suitable, verification needed to proceed
- Typically a resource consuming task
- Estimated manual task time - 6 weeks of expert roboticist
- Synergy takes one day including designer interaction

multidimensional landing simulation



Carnegie Mellon

synergy insights and advantages

Advantages:

- Resizing and redesign time reduces from ~week to hours.
- Software, controller generation reduced from ~month to day
- Reuse of knowledge for different problems
- Exploration of vast design space > millions
- Language helps debugging catching formulation errors
- Auto-dependency flow generation provides insight to designer

Lunar rover design insights:

- Mars 98 lander suitable for lunar landing
- Volume and CG constraints important for rover craft
- Pole may be better suitable for mining rover operations
- Solar array is better than RTG for mining rover
- Battery temperature limits have significant effect

Carnegie Mellon

contribution

First general grammar and implementation of a language to comprehensively encode designs of general robots

First robot design software implementation and methodology to automate rational sizing, analysis, simulation, optimization and software generation, in a unified environment using a single representation of the robot

Carnegie Mellon

future directions

The goal is to have a design software where designers assemble configurations, and parts/ sizes are automatically optimized, simulated along with basic software. The end result is a robot design and a bill of materials, with a virtual prototype to assist development and integration

- Take it through entire robot cycle
- Incorporate geometric understanding
- Advance and refine agents
- Increase computational efficiency
- Generate uniform code for language tags
- Standardize component specification by manufacturers
- Effective user interface for both design and analysis
- Further automate language, stochastic variables, differential variables

Carnegie Mellon

where from here

System analysis and optimization is better left to software

Designers are best at creating configurations and micro behavior

Design tools will eventually encode necessary physics and knowledge to understand robot designs and behavior

Information fuels the design tool, accuracy and access are critical

Embedding synergy in the robot with appropriate inference and learning agents can create self-aware behavior, increasing robustness

Carnegie Mellon