第08章
智能生成设计

宋超阳
南方科技大学

Design, AI & Robotics

Reachy by Pollen Robotics

The Natural vs. The Artificials

The Natural

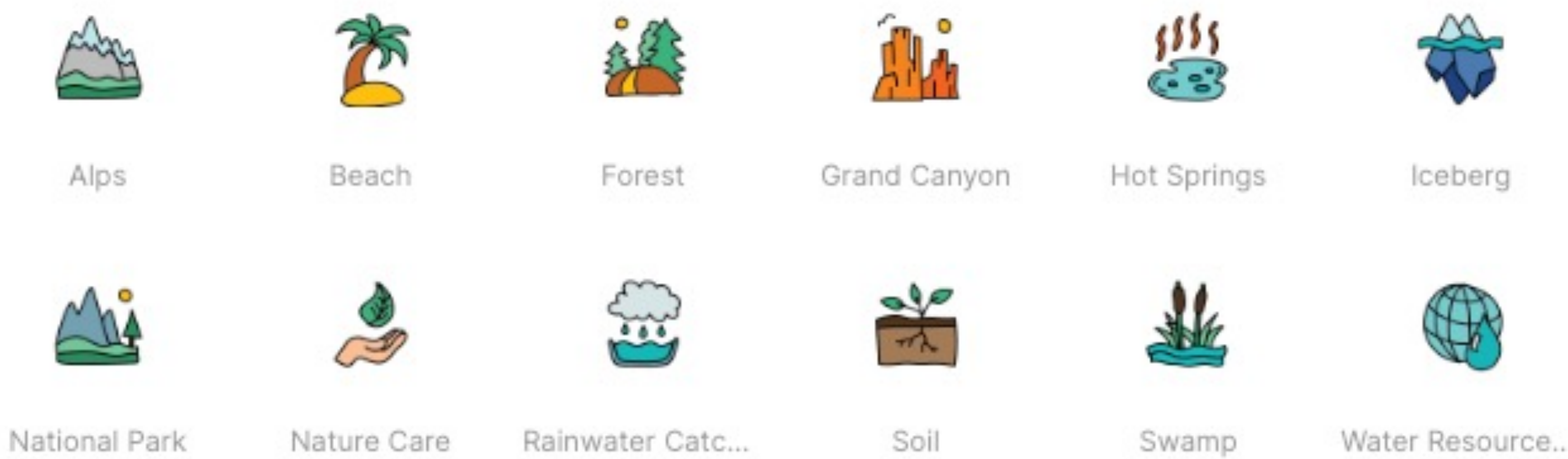
Let's begin by asking a simple question:

what is the natural, and what about the artificial?

The Artificial

https://icons8.com/

The Natural vs. The Artificials



The Natural

We can generally describe ***the natural*** as anything that already existed on earth.

Or anything that is not made by the human, including the human

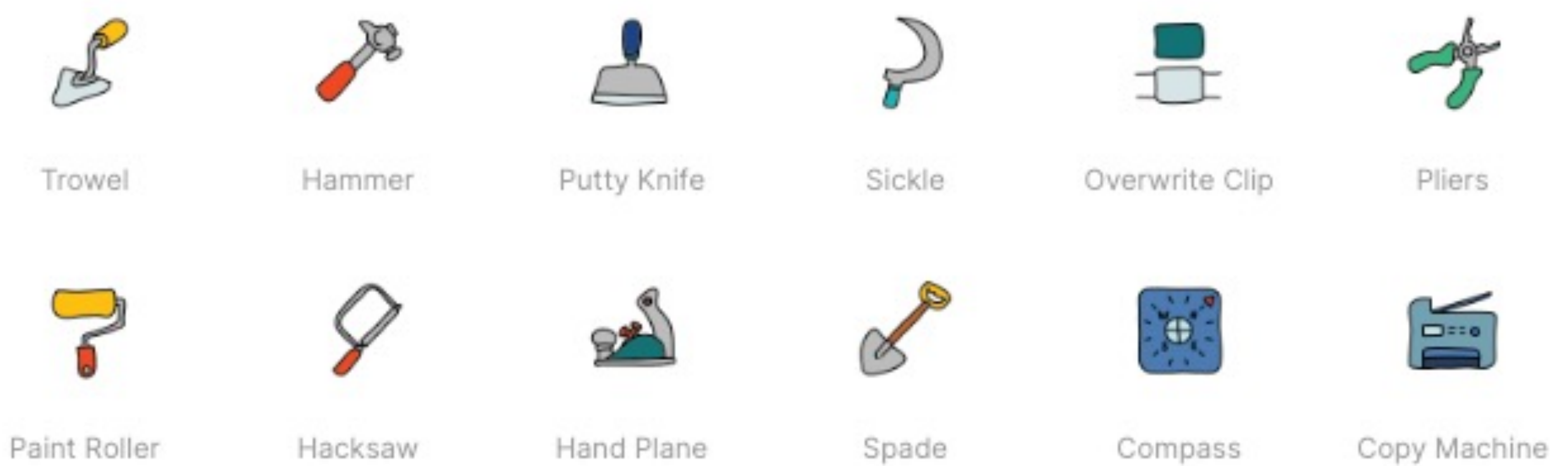
https://icons8.com/

The Natural vs. The Artificials

We can generally describe ***the artificial*** as everything other than the natural.

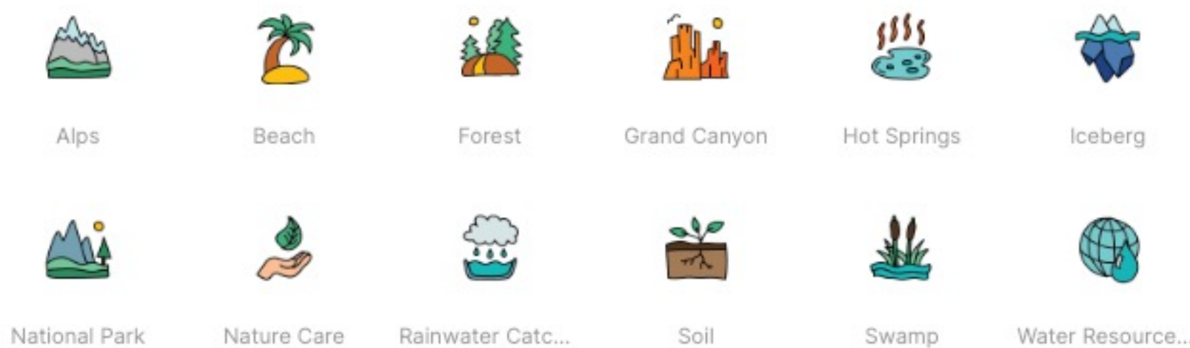
Or anything that is made by the human

The Artificial



https://icons8.com/

The Natural vs. The Artificials



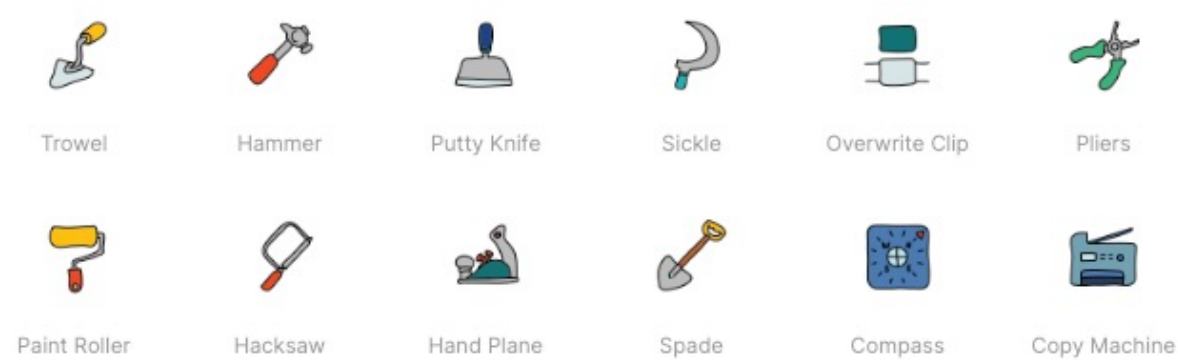
Natural

↑

Human

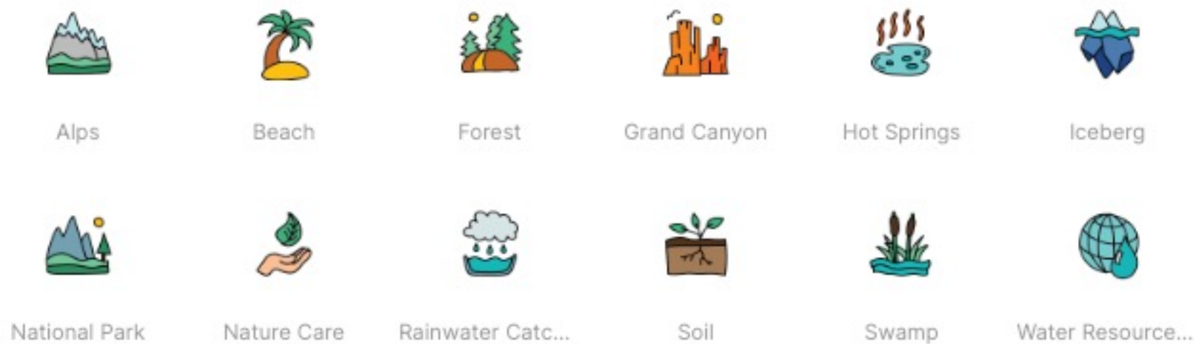
↓

Artificial



https://icons8.com/

The Natural vs. The Artificials



Unknown Unknowns → Human

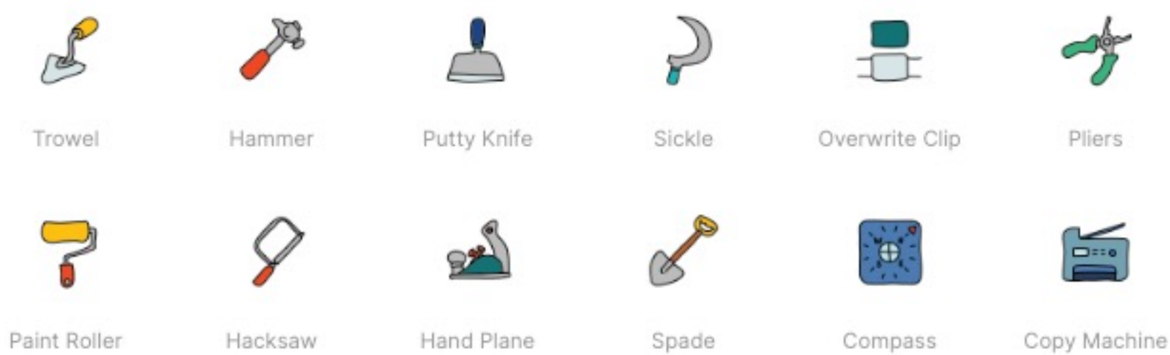
Natural

Human → Natural

Human → Artificial

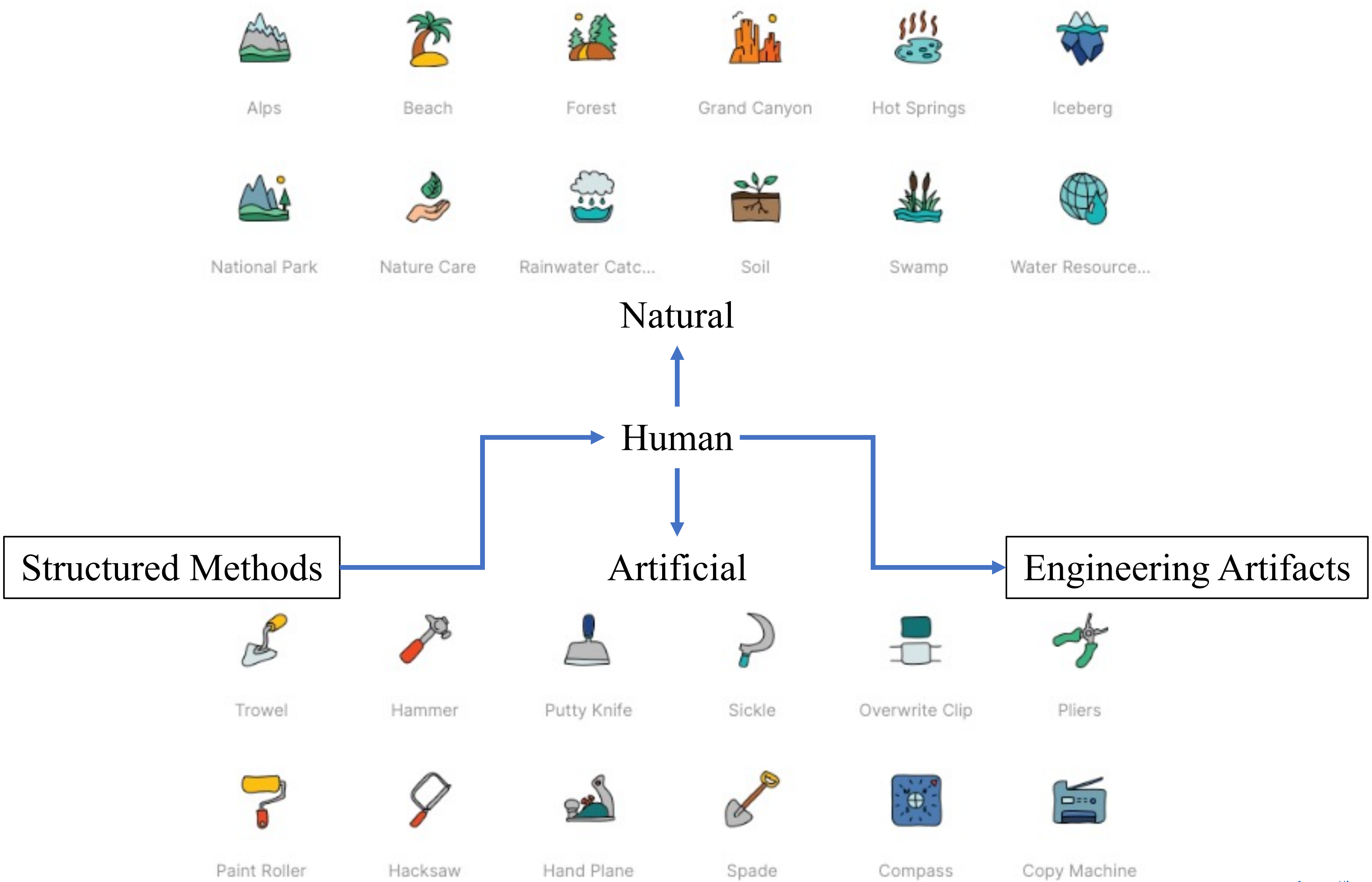
Human → Scientific Discoveries

Artificial



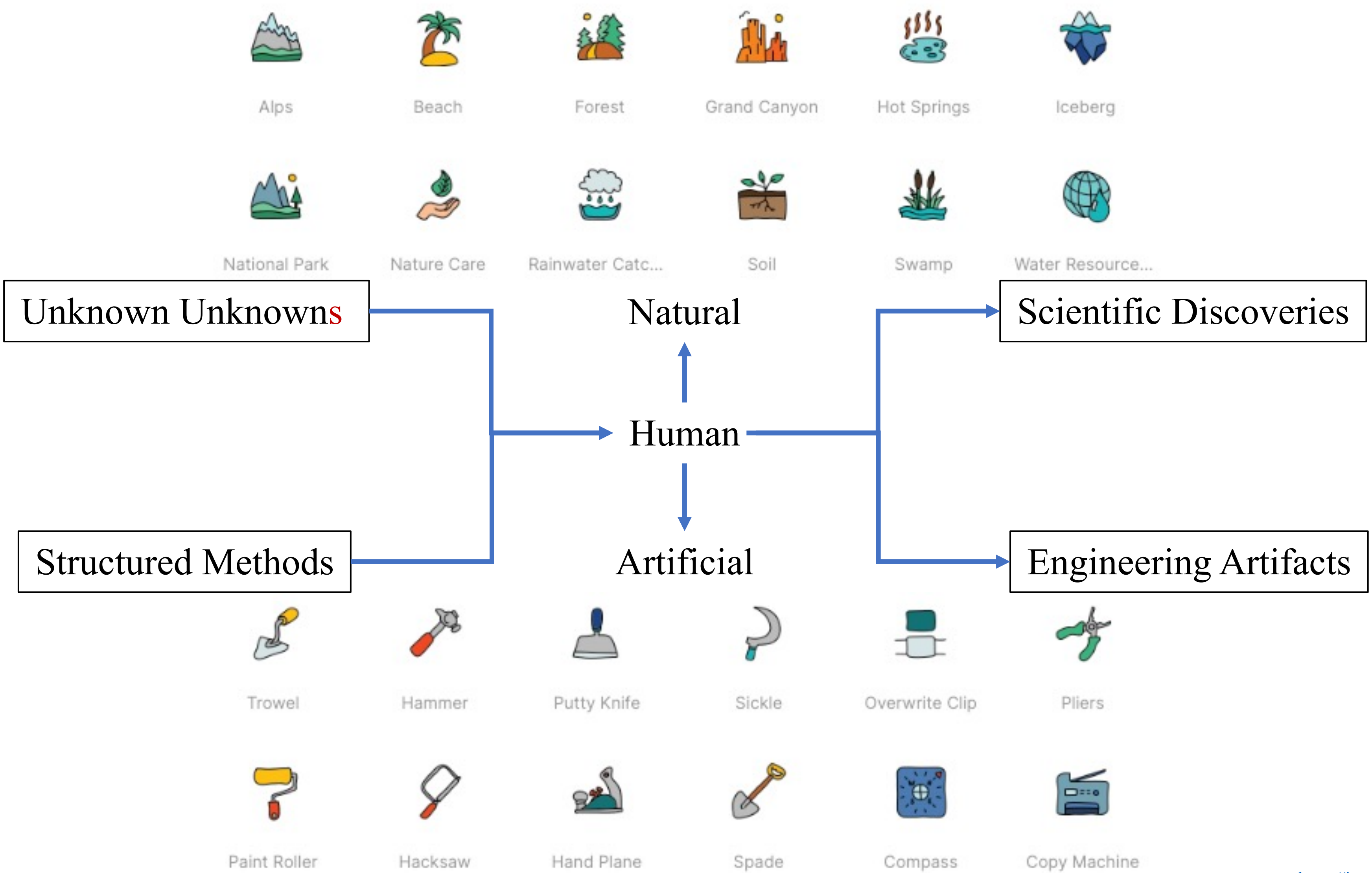
https://icons8.com/

The Natural vs. The Artificials



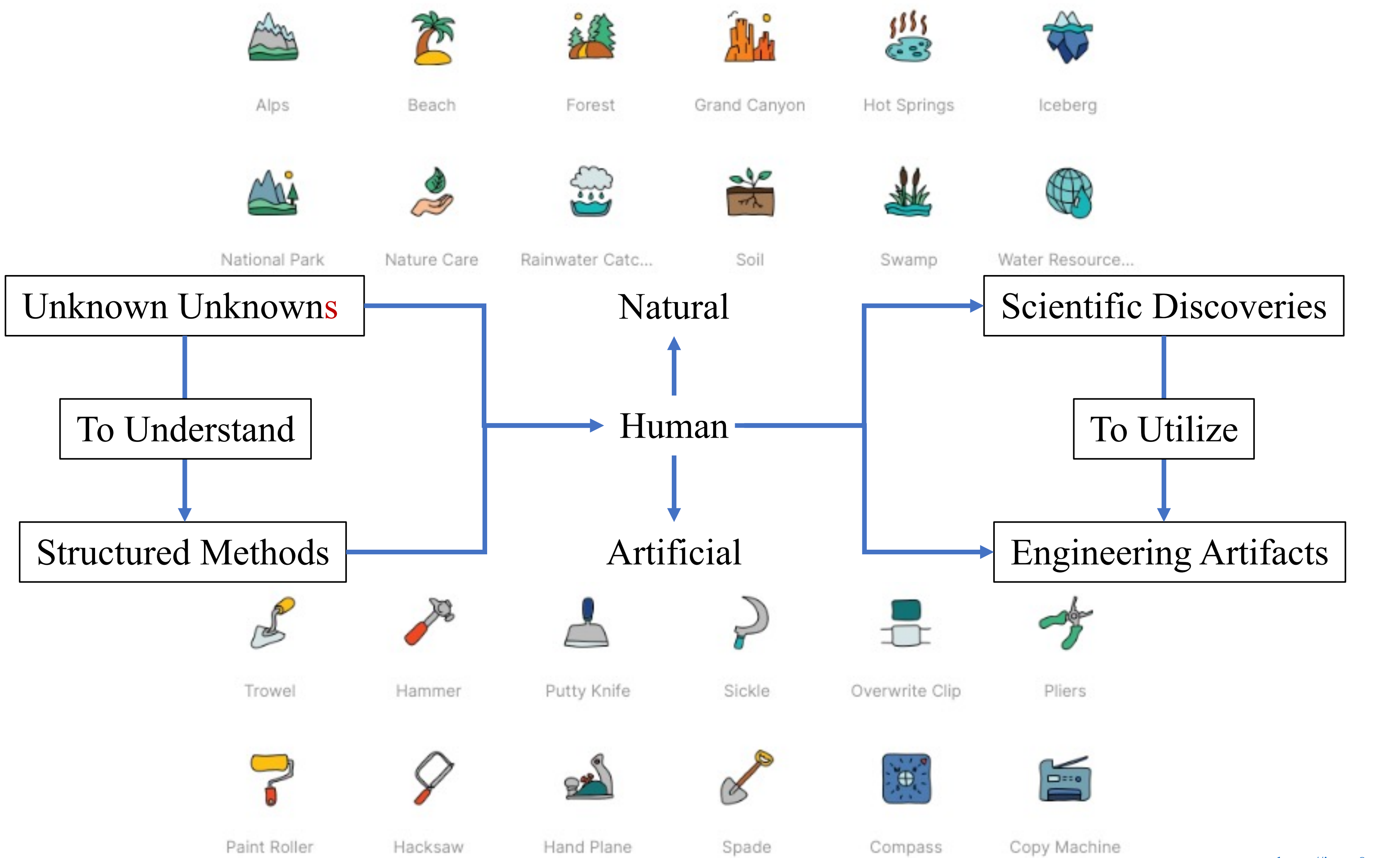
https://icons8.com/

The Natural vs. The Artificials



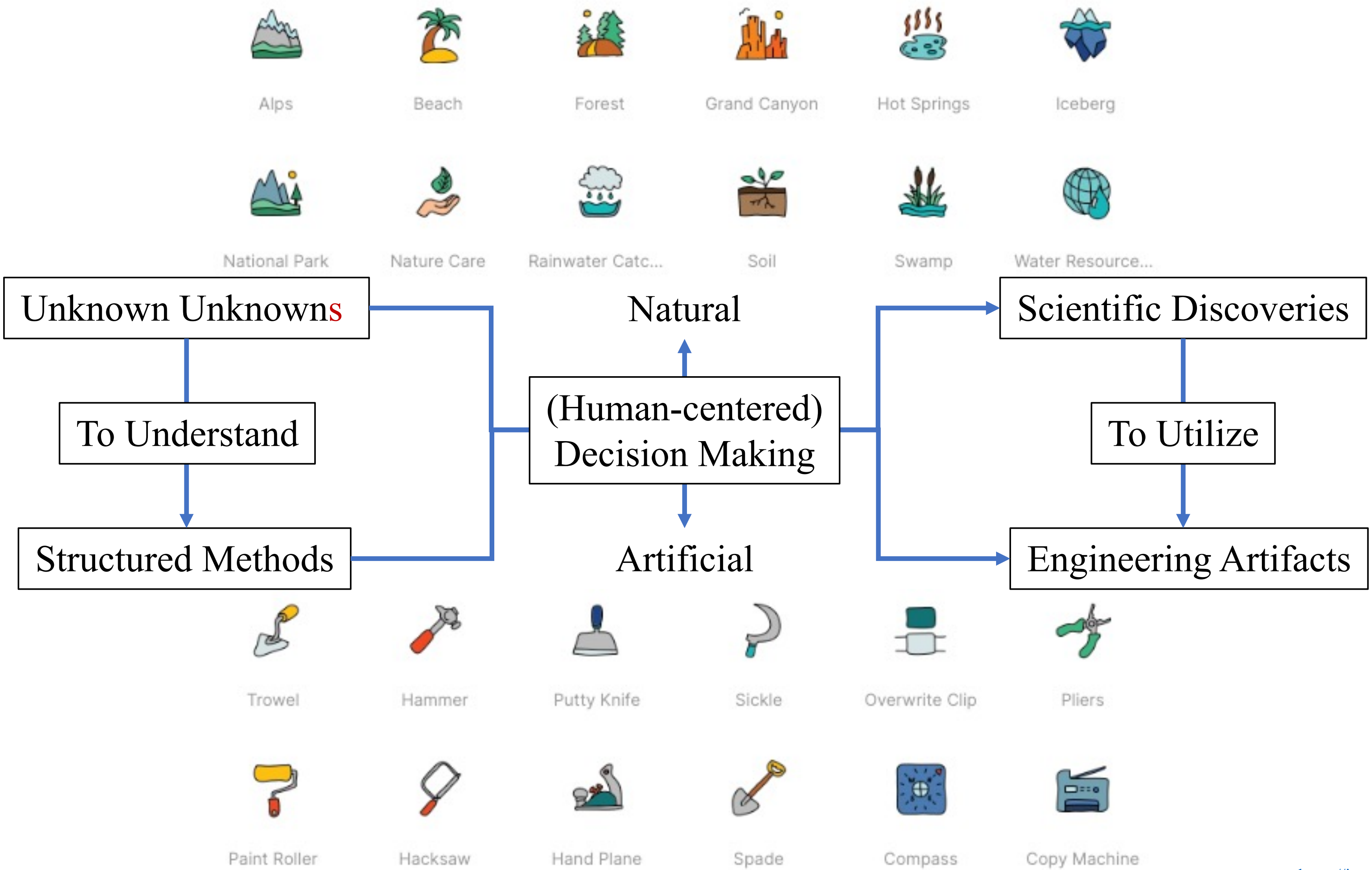
https://icons8.com/

The Natural vs. The Artificials



https://icons8.com/

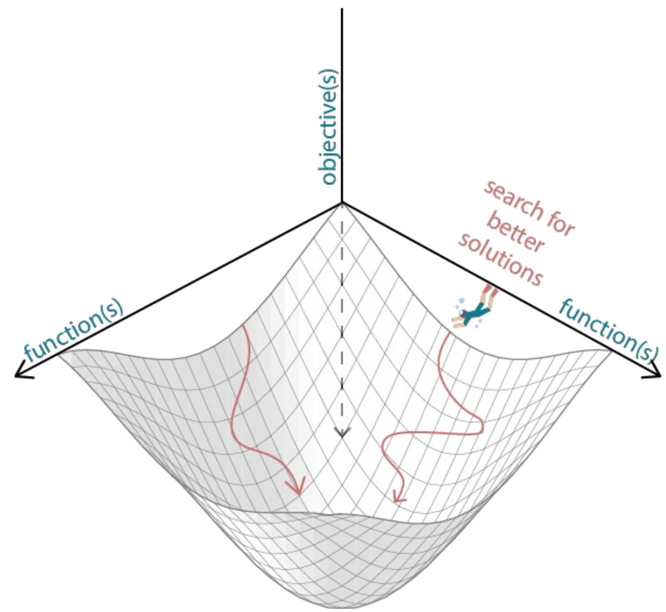
The Natural vs. The Artificials



https://icons8.com/

Decision Making as an Optimization Problem

- Optimization is an inherently mathematical subject.
  - It is about maximizing or minimizing some mathematical function to arrive at the best possible solution to a problem, and involves creating design options that are shaped by certain outcomes as they are being created.
- Optimization problems arise in all kinds of fields, from aerospace engineering to architectural design.
  - However, regardless of domain, every optimization problem has three features:
    - An objective function.
    - Constraints.
    - Data.



Decision Making as an Optimization Problem

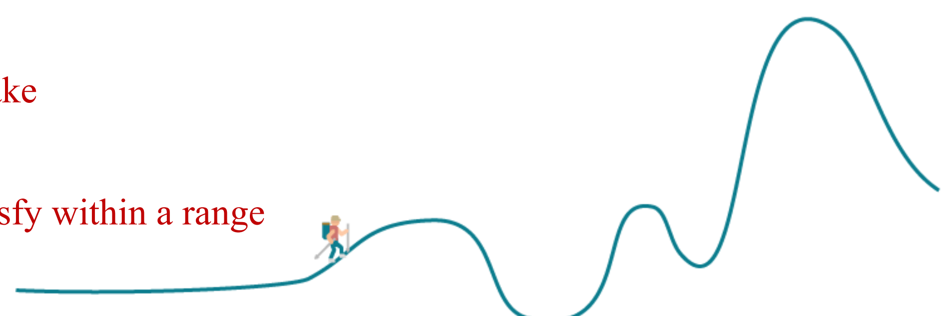
- Without any loss of generality an optimization problem can be defined by:

If to max f_m, then use $-f_m$ instead.

s. t. means subject to

$$\boxed{\min \quad f_m(x)} \qquad x_i^L m = 1, \dots, M$$
$$\boxed{\text{s.t.}} \quad g_j(x) \leq 0 \qquad x_i^L j = 1, \dots, J$$
$$h_k(x) = 0 \qquad x_i^L k = 1, \dots, K$$
$$x_i^L \leq x_i \leq x_i^U \qquad x_i^L i = 1, \dots, N$$
$$x \in \Omega \qquad x_i^L$$

- where
  - x_i represents the i-th variable to be optimized, x_i^L and x_i^U its lower and upper bound,
    - The variables that defines describes the problems, more if the problem is complex, less if simple
  - f_m the m-th objective function,
    - Your goal, or the decision you want to make
  - g_j the $j\text{–}th$ inequality constraint and
    - A type of constraints that you need to satisfy within a range
  - h_k the k-th equality constraint.
    - Another type of constraints that you MUST/WANT TO satisfy for sure



Decision Making as an Optimization Problem

- Be aware of the complete optimization problem helps you
  - to identify the challenging facets of your optimization problem and, thus,
  - to select a suitable algorithm

Refer to the text if interested

Variable Types

Number of Variables

Number of Objectives

Constraints

Multi-modality

Differentiability

Evaluation Time

Uncertainty

Variable Types.

- The variables span the search space Ω of your optimization problem. Thus, the type of variables is an essential aspect of the problem to be paid attention to. Different variables types, such as continuous, discrete/integer, binary, or permutation, define the characteristics of the search space. In some cases, the variable types might be even mixed, which increases the complexity further.

Number of Variables.

- Not only the type but also the number of variables (N) is essential. For either a very small or large number, different algorithms are known to work more efficiently. You can imagine that solving a problem with only ten variables is fundamentally different from solving one with a couple of thousand. For large-scale optimization problems, even the second-order derivate becomes computationally very expensive, and efficiently handling the memory plays a more important role.

Number of Objectives.

- Some optimization problems have more than one conflicting objective (M>1) to be optimized. Before researchers have investigated multi-objective optimization, single-objective problems were the main focus. Single-objective optimization is only a particular case where M=1. In multi-objective optimization, the solution's domination relation generalizes the comparison of two scalars in single-objective optimization. Moreover, having more than one dimension in the objective space, the optimum (most of the time) consists of a set of non-dominated solutions. Because a set of solutions should be obtained, population-based algorithms have mainly been used as solvers.

Constraints.

- Optimization problems have two types of constraints, inequality (g) and equality (h) constraints. From an end-user perspective, constraints have a priority over objective values. No matter how good the solution's objectives are, it is considered infeasible if it turns out to violate just a single constraint. Constraints can have a big impact on the complexity of the problem. For instance, if only a few islands in the search space are feasible or a large number of constraints ($|J|+|K|$) need to be satisfied. For genetic algorithms satisfying equality constraints can be rather challenging. Thus, this needs to be addressed differently, for instance, by mapping the search space to a utility space where the equality constraints are always satisfied or injecting the knowledge of the equality constraint through customization.

Multi-modality.

- Most aspects discussed so far are most likely known or to be relatively easy to define. However, the nature of the fitness landscape is less obvious bet yet essential to be aware of. In the case of multi-modal fitness landscapes, optimization becomes inevitably more difficult due to the existence of a few or even many local optima. For the solution found, one must always ask if the method has explored enough regions in the search space to maximize the probability of obtaining the global optimum. A multi-modal search space quickly shows the limitation of local search, which can easily get stuck.

Differentiability.

- A function being differentiable implies the first or even second-order derivative can be calculated. Differentiable functions allow gradient-based optimization methods to be used, which can be a great advantage over gradient-free methods. The gradient provides a good indication of what direction shall be used for the search. Most gradient-based algorithms are point-by-point based and can be highly efficient for rather unimodal fitness landscapes. However, in practice, often functions are non-differentiable, or a more complicated function requires a global instead of a local search. The research field addressing problems without knowing their mathematical optimization is also known as black-box optimization.

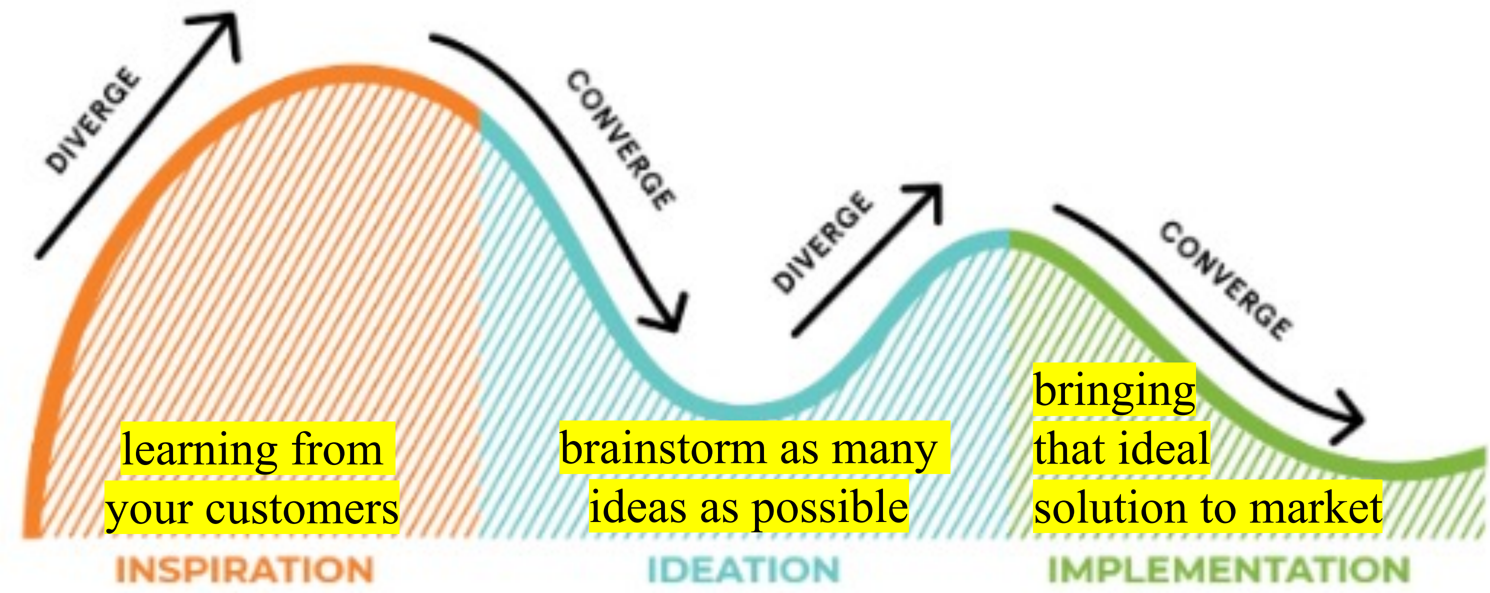
Evaluation Time.

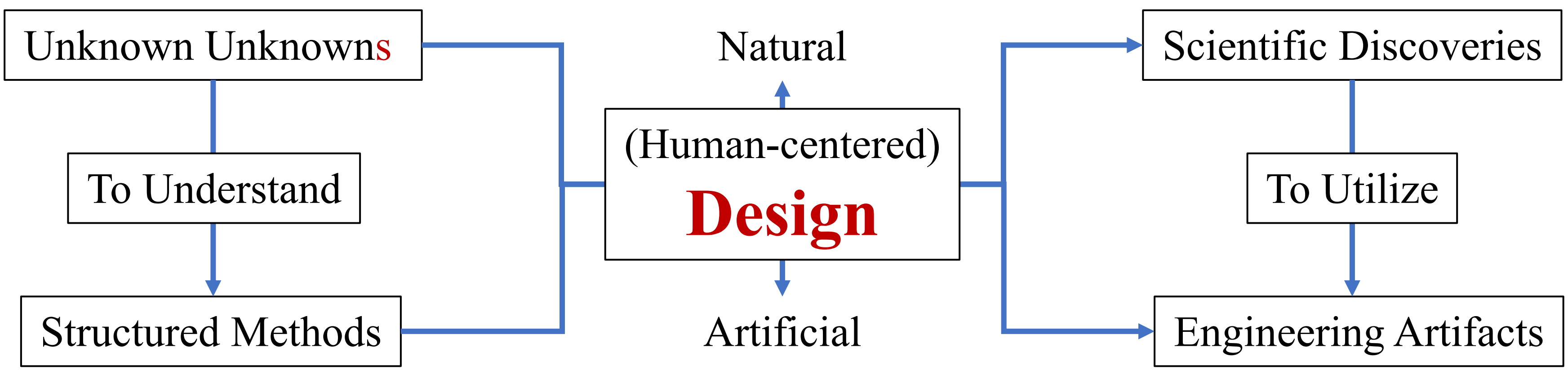
- Many optimization problems in practice consist of complicated and lengthy mathematical equations or domain-specific software to be evaluated. The usage of third-party software often results in a computationally expensive and time-consuming function for evaluating objectives or constraints. For those types of problems, the algorithm's overhead for determining the next solutions to be evaluated is often neglectable. A commercial software performing an evaluation often comes with various more practical issues such as distributed computing, several instances to be used in parallel and software license, and the software's possible failure for specific design variable combinations.

Uncertainty.

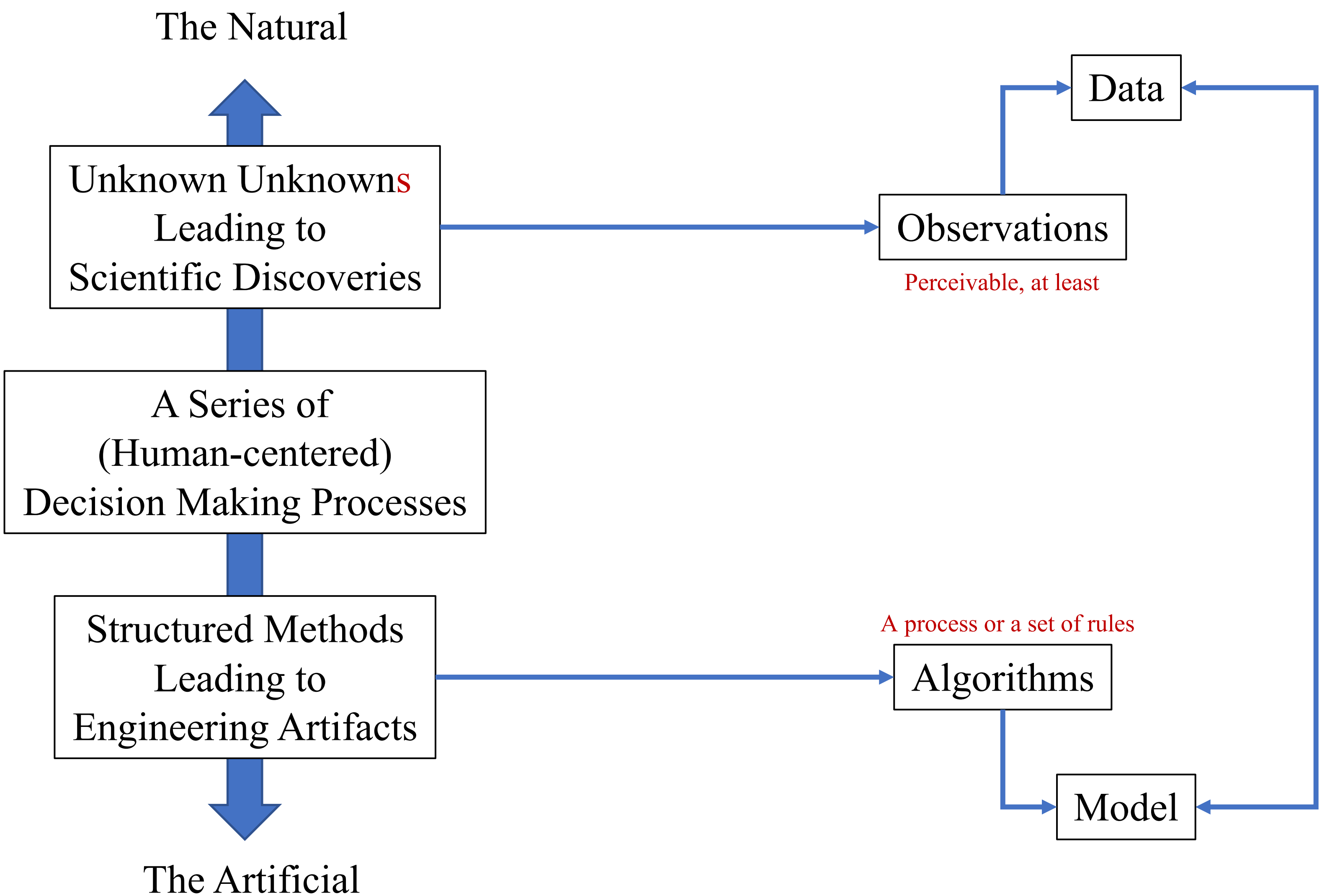
- Often it is assumed that the objective and constraint functions are of a deterministic manner. However, if one or multiple target functions are nondeterministic, this introduces noise or also referred to as uncertainty. One technique to address the underlying randomness is to repeat the evaluation for different random seeds and average the resulting values. Moreover, the standard deviation derived from multiple evaluations can be utilized to determine the performance and the reliability of a specific solution. In general, optimization problems with underlying uncertainty are investigated by the research field called stochastic optimization.

Design as Decision Making

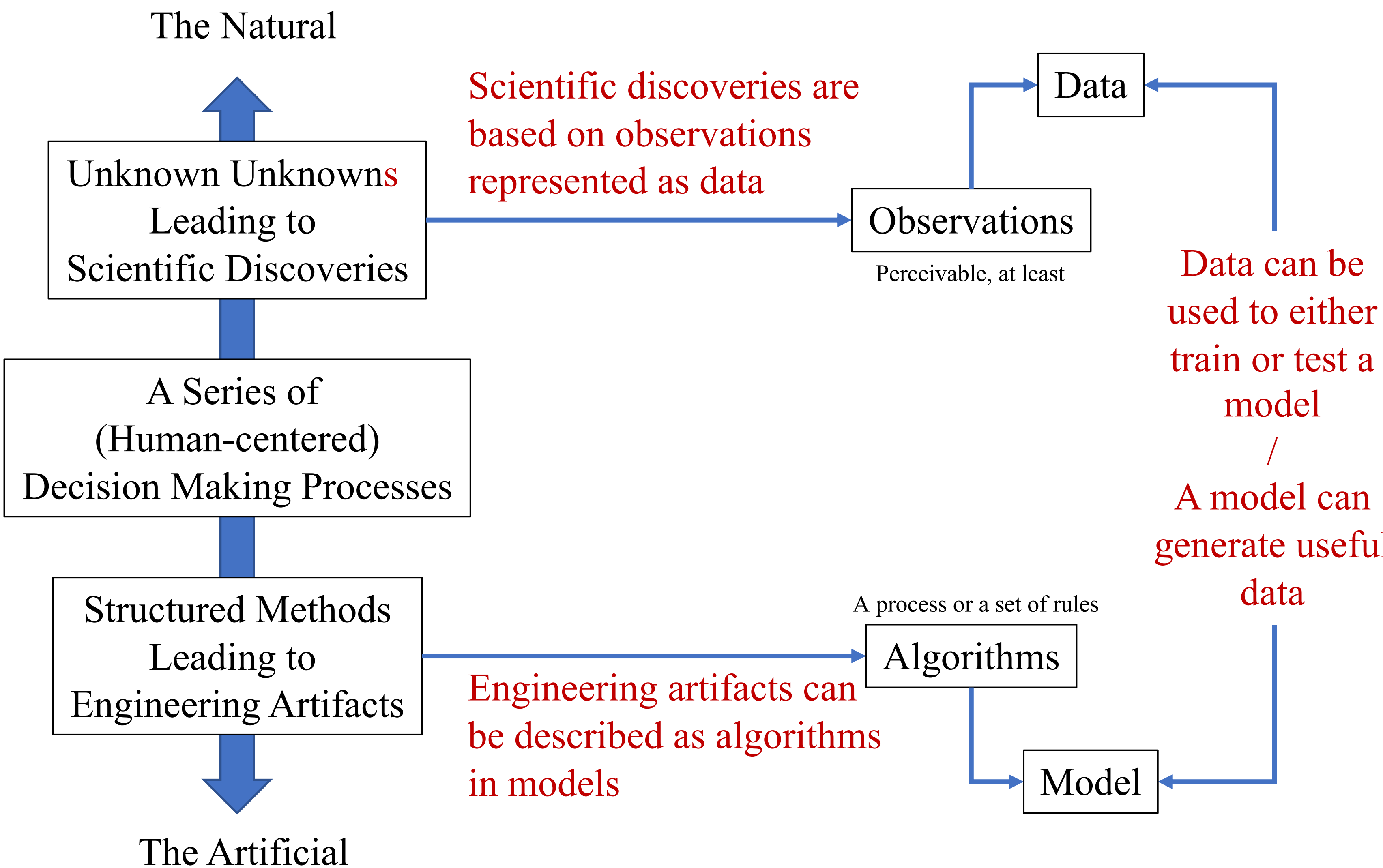




Principals of Data-driven Technology



Principals of Data-driven Technology



The ML process

To get acquainted with terms and understand how a model arrives at a prediction, it can be helpful to draw an analogy with a process we're familiar with: baking a cake.

Input	Rules	Hardware	Learning	System	Output
Data +	Algorithm(s) +	GPU +	Training →	Model →	Prediction
Ingredients	Recipe	Utensils	Practice →	Chef →	Cake

Data is the raw material you feed to the algorithm as input to produce a ML model.

An algorithm is a set of rules or step-by-step instructions to solve a problem.

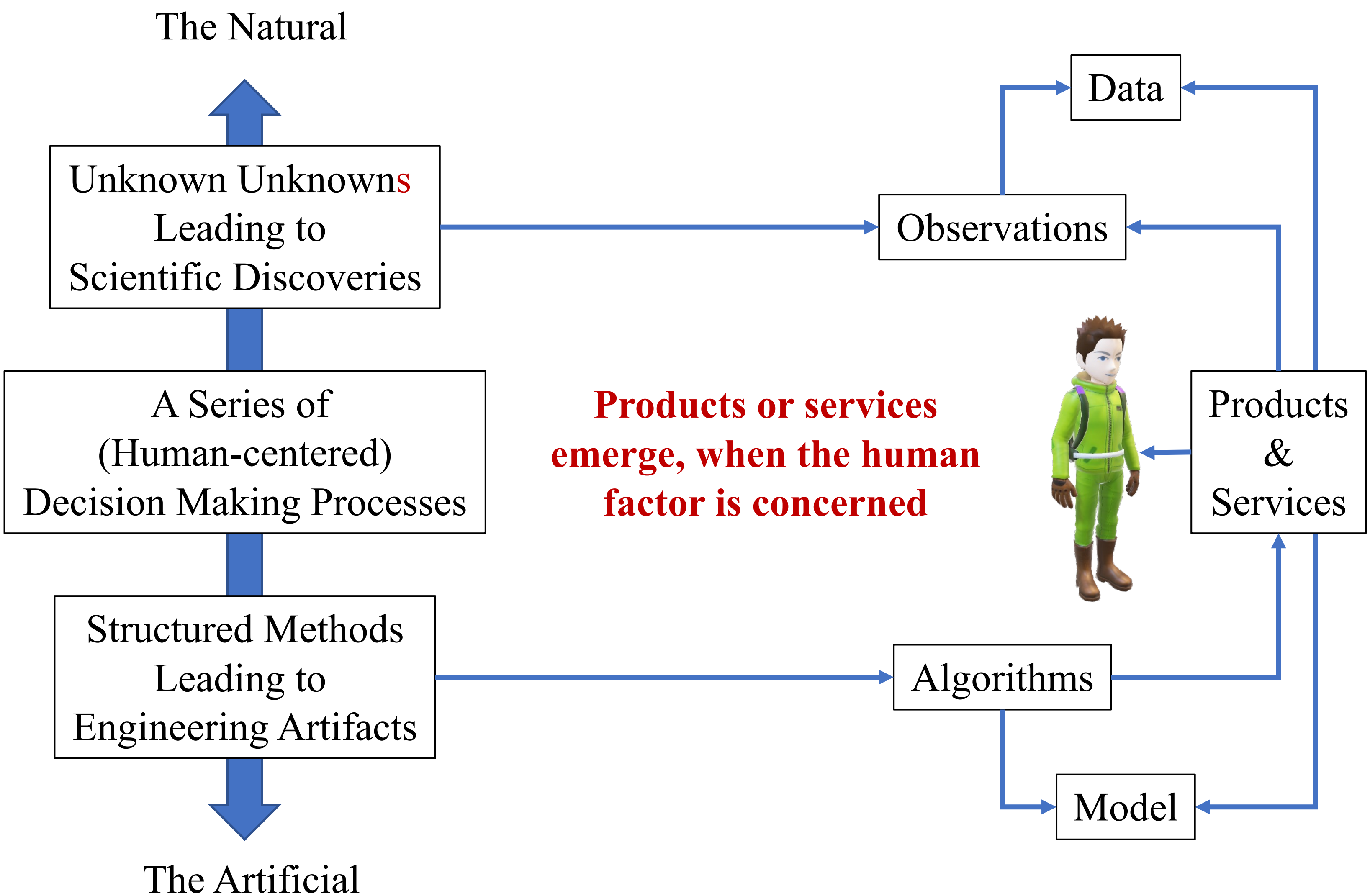
The model requires GPU, and sometimes other resources, to run on.

Training process taking time and tweaking to learn, create and improve its model.

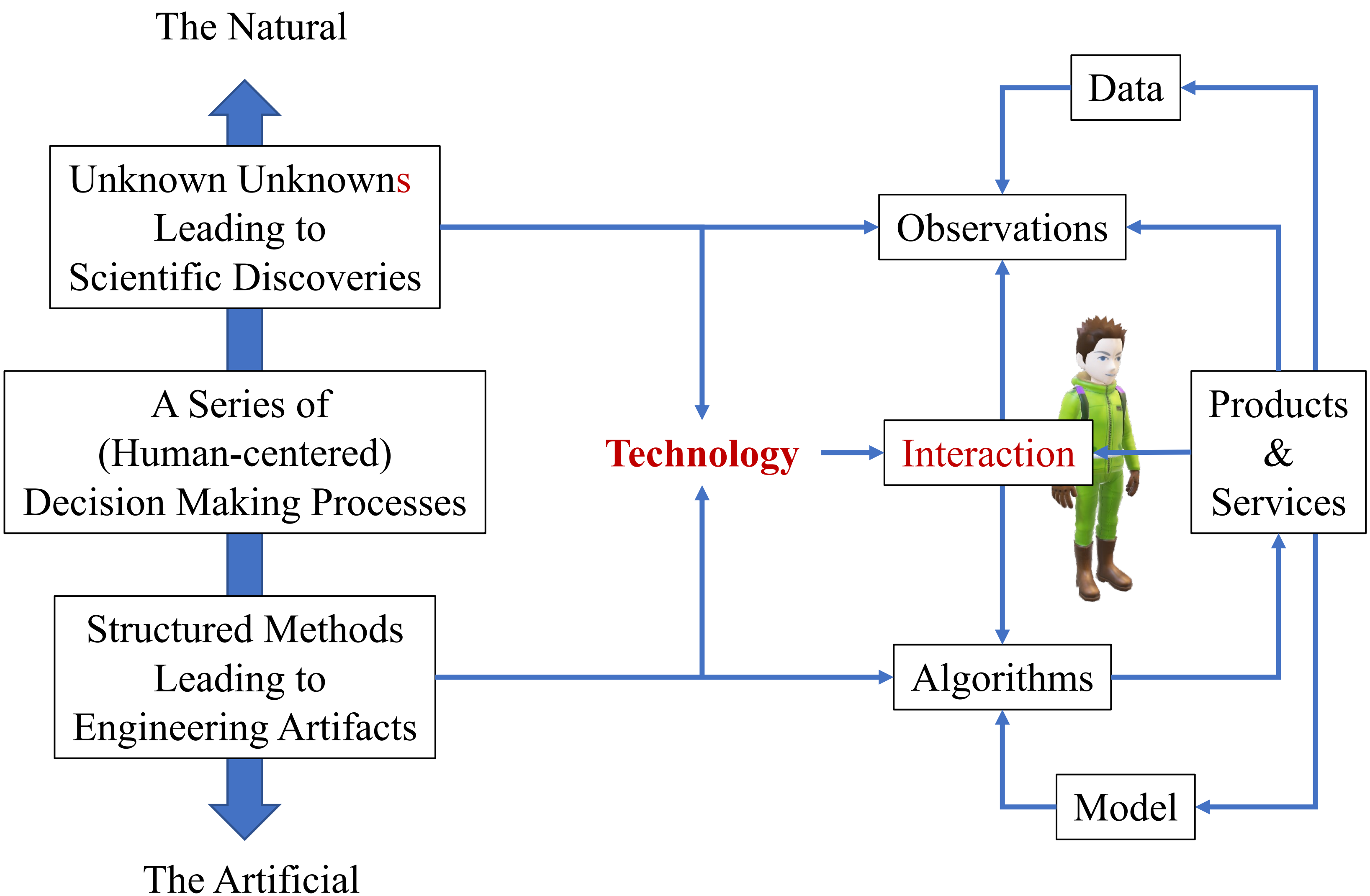
A model is a mathematical representation based on the algorithm(s) and data that is able to predict or produce an output and continues to learn over time.

Disclaimer: Please note this is a highly simplified representation of the real process which is a lot more complex and consists of plenty subtasks.

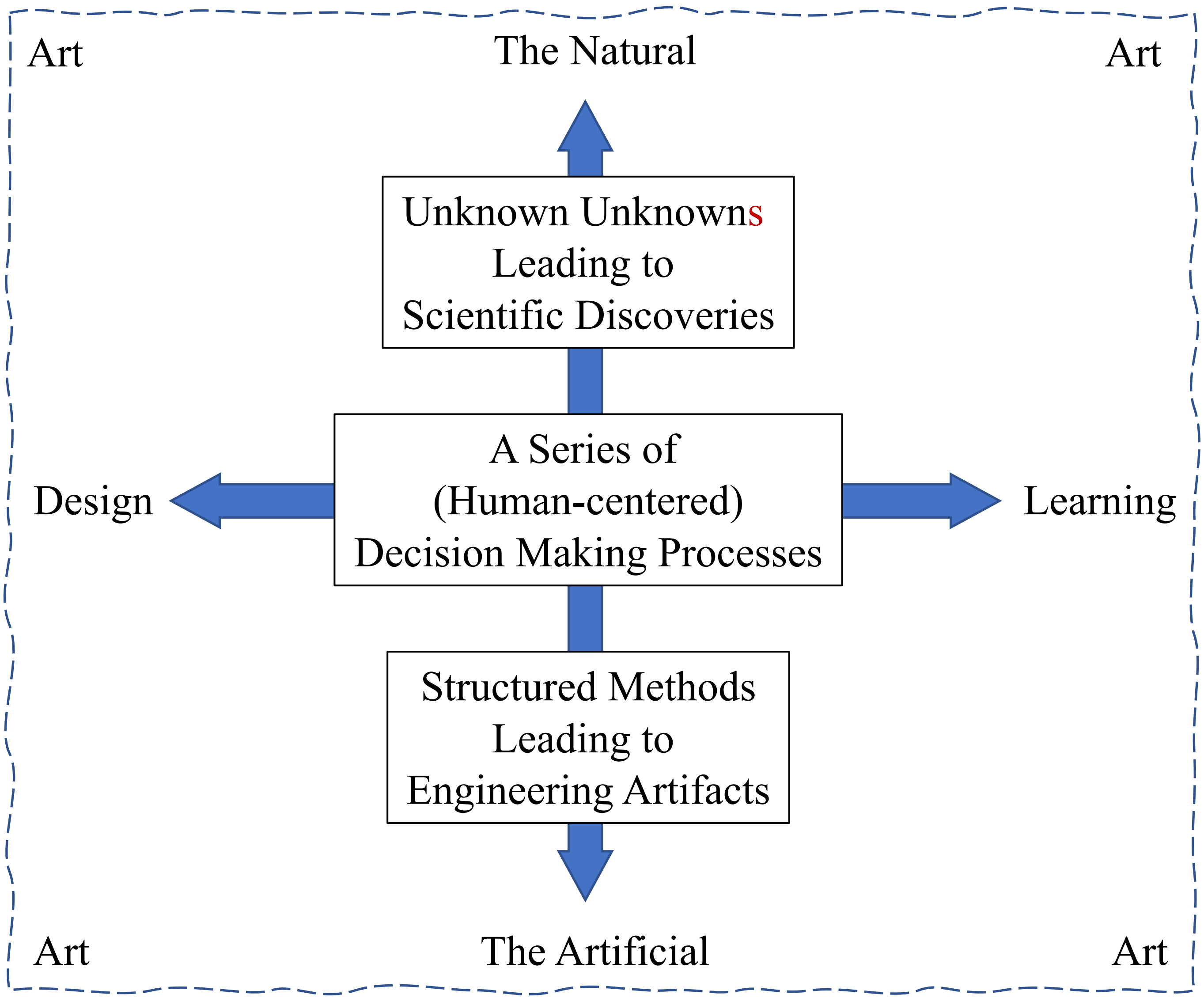
Principals of Data-driven Technology



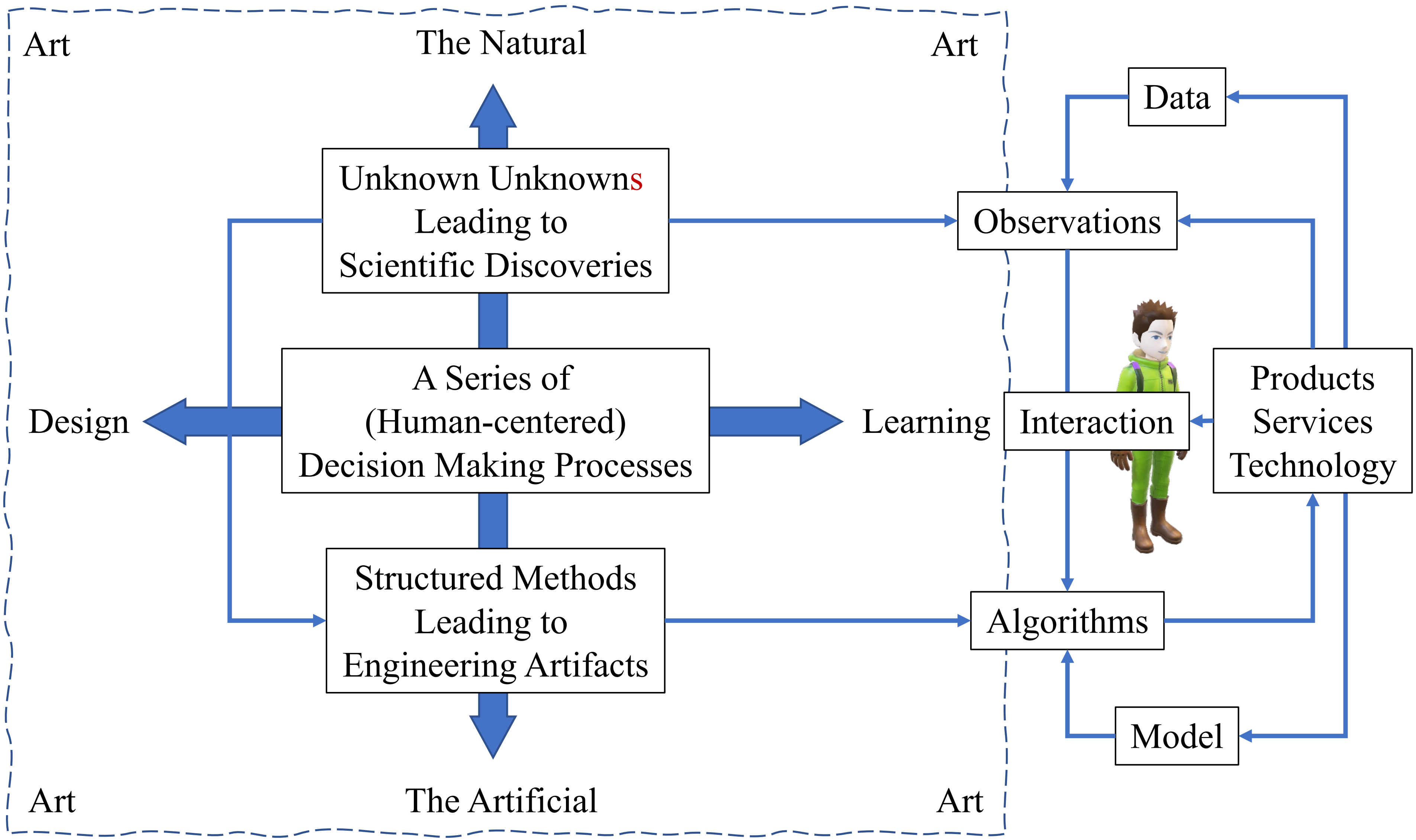
Principals of Data-driven Technology



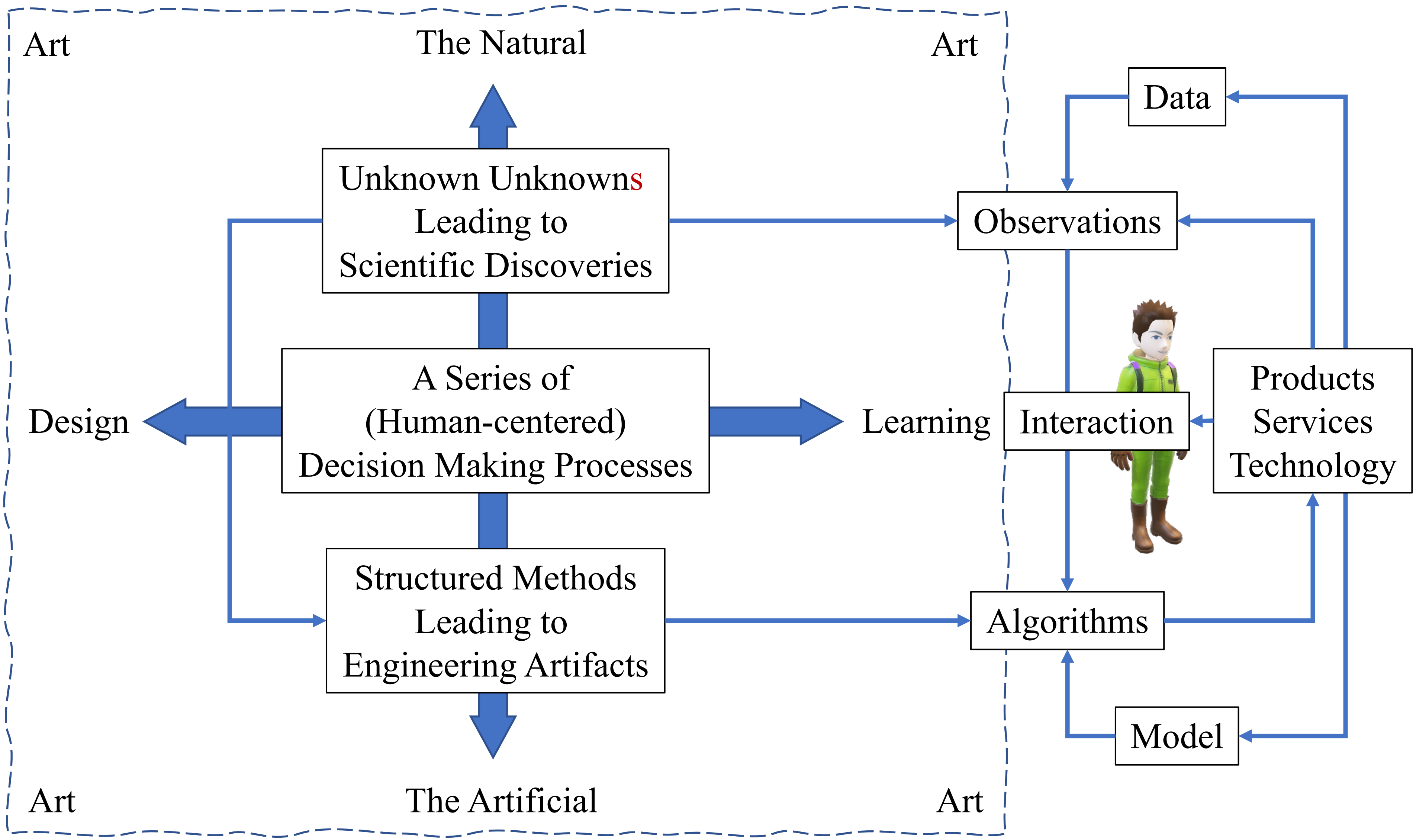
The Science and Engineering of Design and Learning

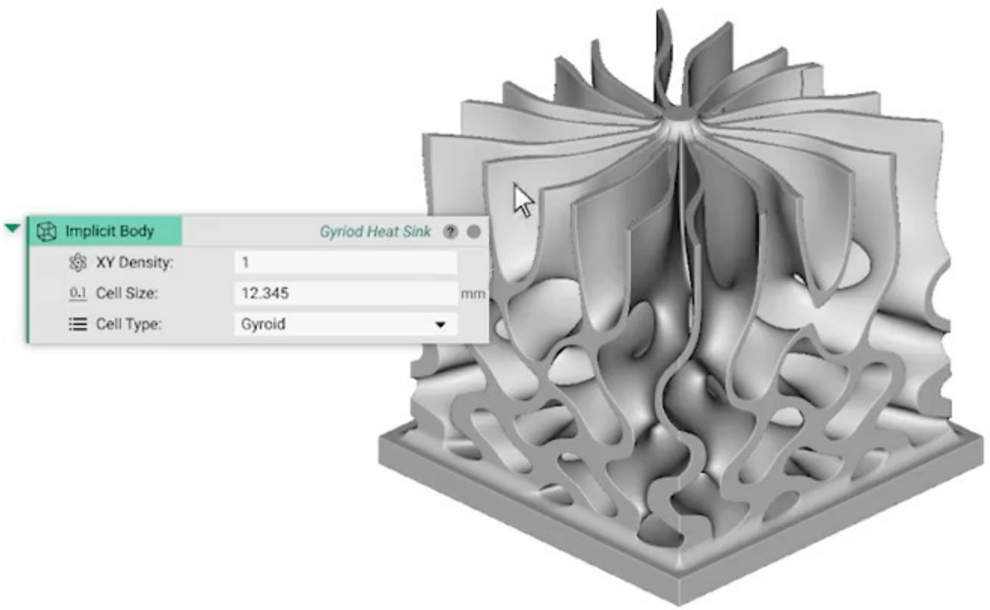


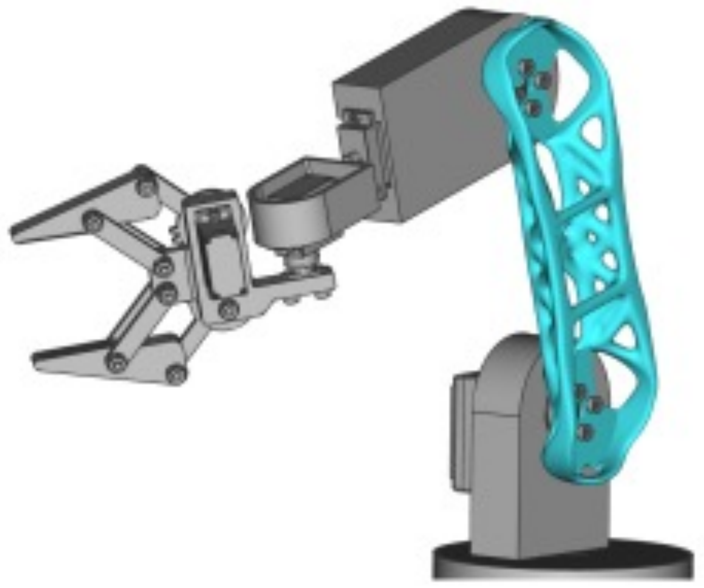
The Science and Engineering of Design and Learning



The Role of AI in Design





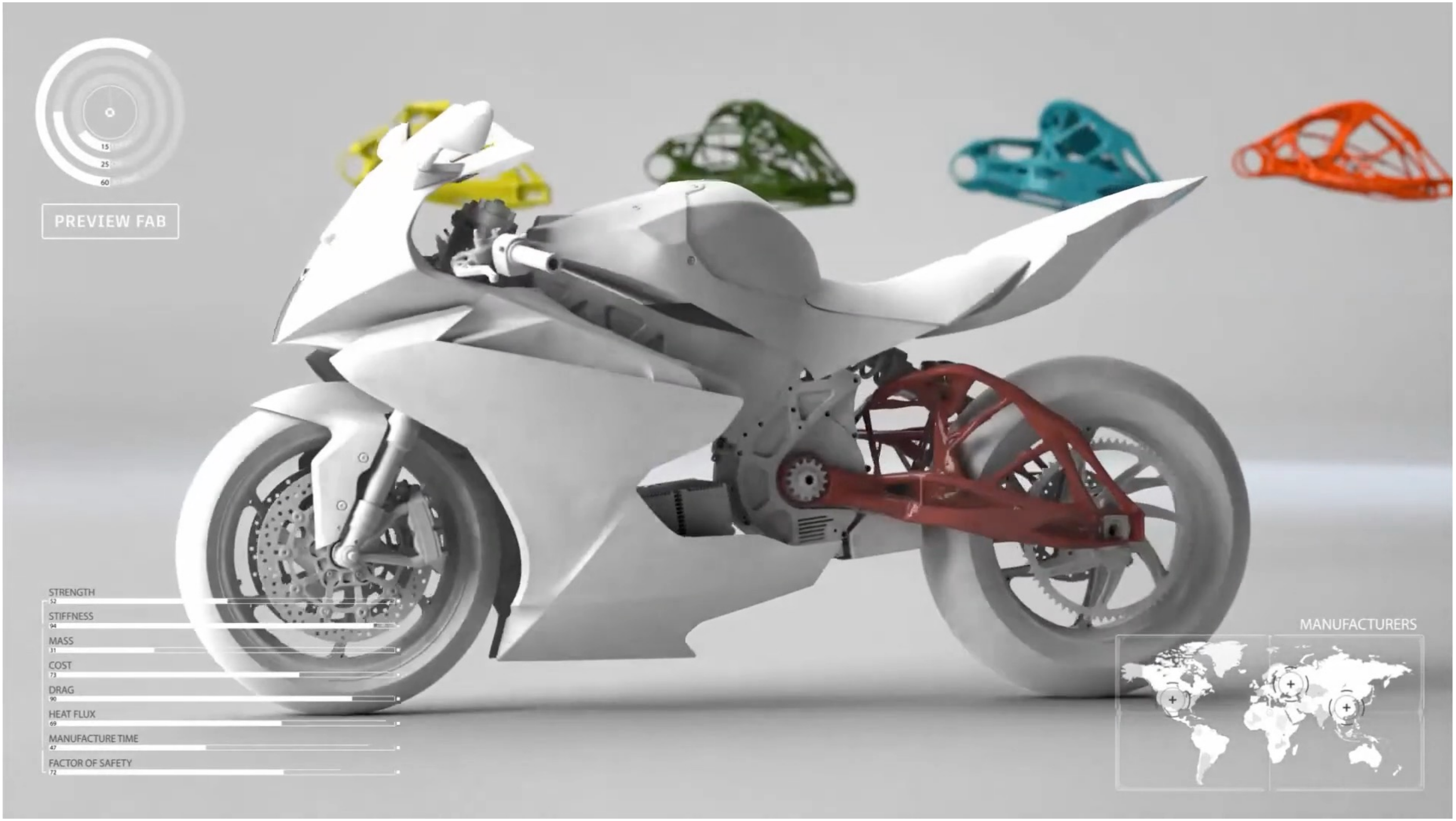


Generative Design

an advanced engineering methodology that combines geometry generation, simulation, and design automation

nTop

Introduction to Generative Design



https://www.youtube.com/watch?v=hk4TN8xkSQM

Computational Design

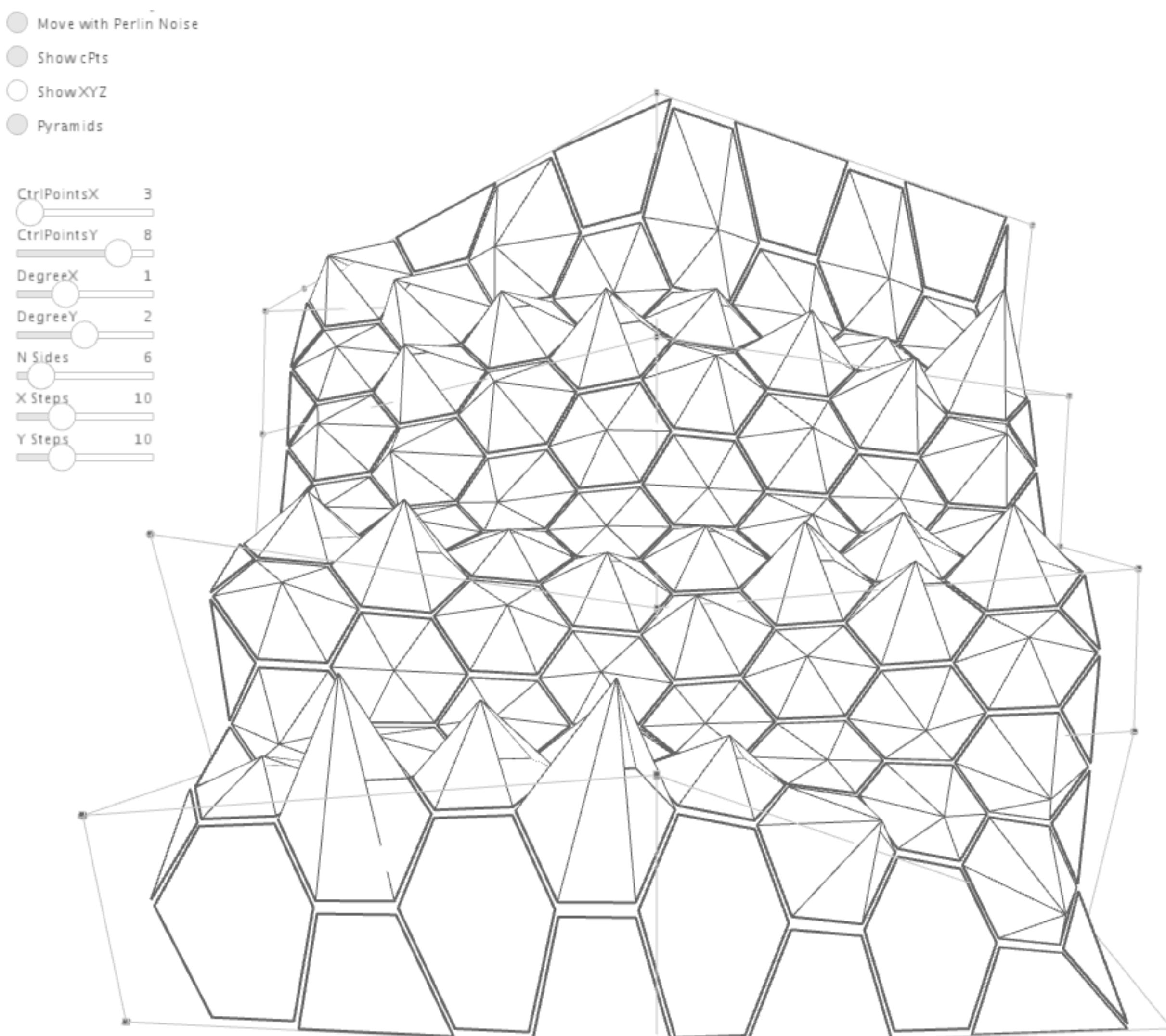


Image of an NURBS manipulations from Martin Stacey - UCL

- Computational design is **<u>NOT</u>** any one algorithm or off-the-shelf process you can utilize.
- Rather, we describe it as an approach whereby **<u>a designer defines a series of instructions, rules and relationships that precisely identify the steps necessary to achieve a proposed design and its resulting data or geometry</u>**.
- Crucially, these steps must be **<u>computable</u>**, meaning they can be understood and calculated by a computer.

Computational Design

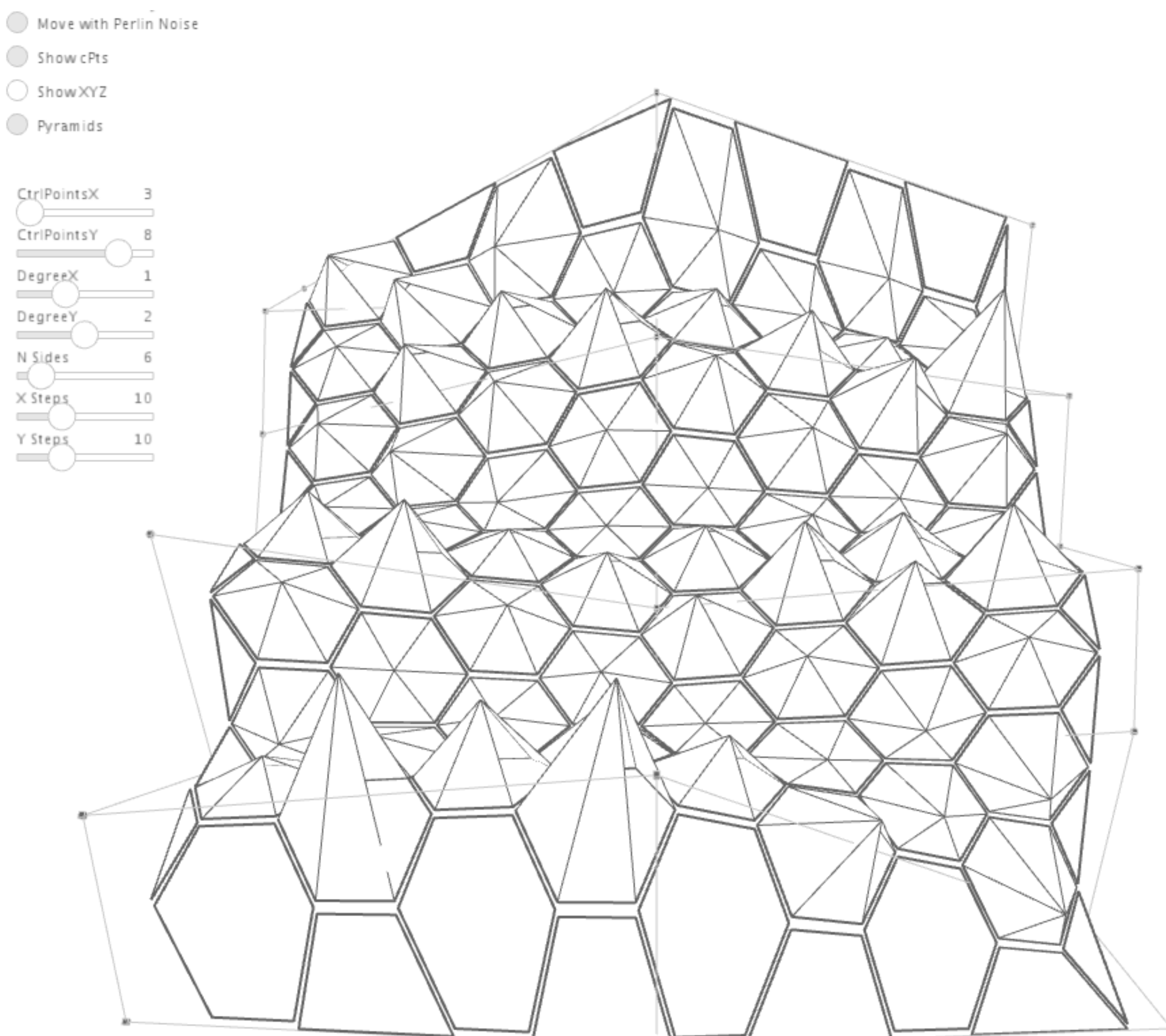
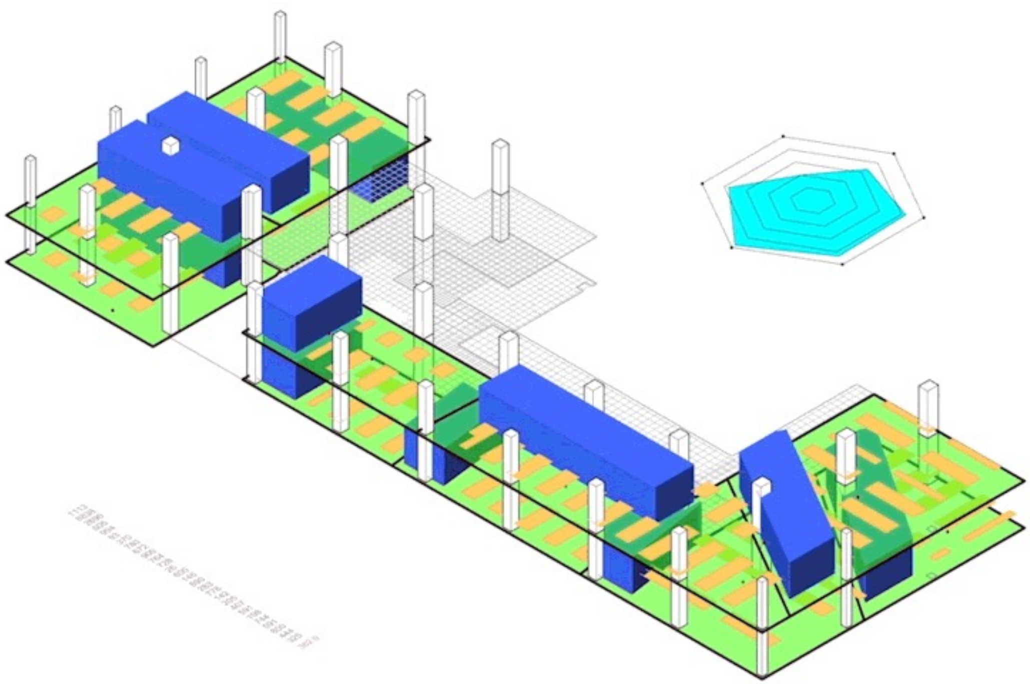


Image of an NURBS manipulations from Martin Stacey - UCL

- When approaching a design computationally, the designer would
  - **focus on developing the procedure that would create a design** - not the design itself.
- The process of iterating through options and data are **offloaded to a computer**.
  - Saves time, money and effort
  - Lets the designer focus on the creativity of the design process

What is Generative Design?

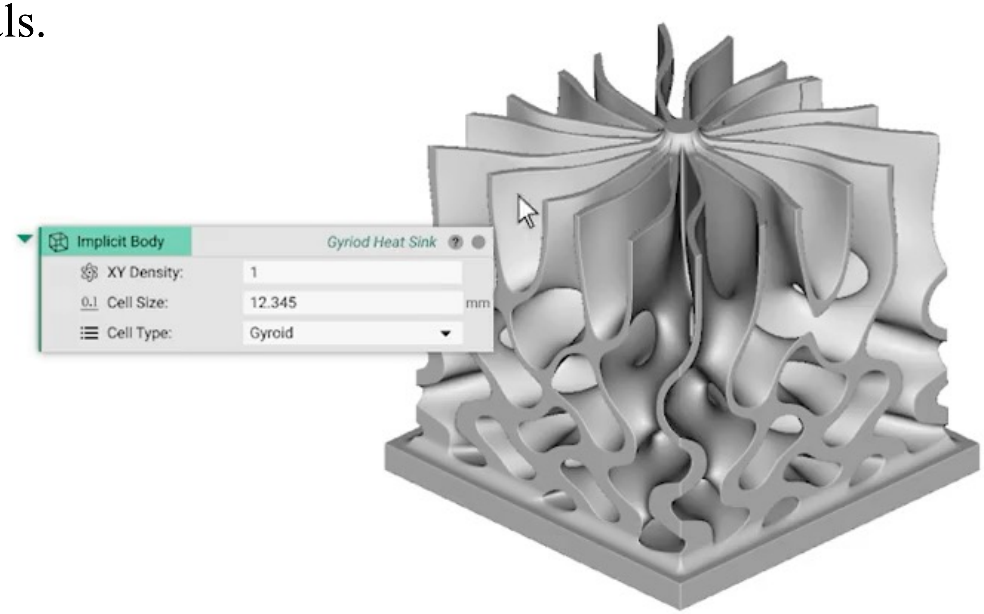
- A collaborative design process between humans and computers.
- During this process, the designer
  - **defines** the design parameters and the computer produces design studies (alternatives),
  - **evaluates** them against quantifiable goals set by the designer,
  - **improves** the studies by using results from previous ones and feedback from the designer, and
  - **ranks** the results based on how well they achieve the designer's original goals.



Some generated alternatives - Mars Innovation District - The Living

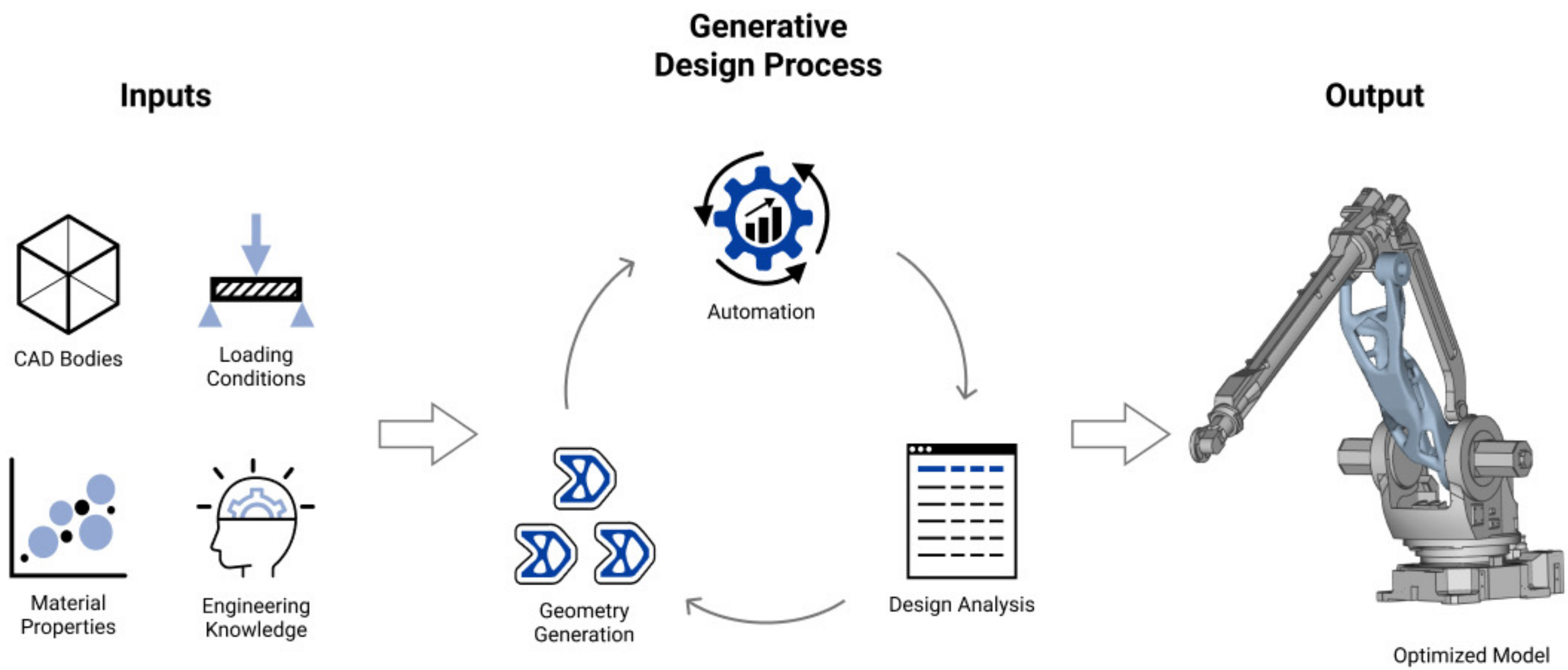
What is Generative Design?

- Generative design is a specific application of the computational design approach, with the following distinctions:
  - The designer defines goals to achieve a design (rather than the exact steps).
  - The computer helps the designer to explore the design space and generate multiple design options (not just one).
  - The computer enables the designer to find a set of optimal solutions that satisfy multiple competing goals.
  - The designer compares multiple design scenarios to find a set of design options that fits the design goals.



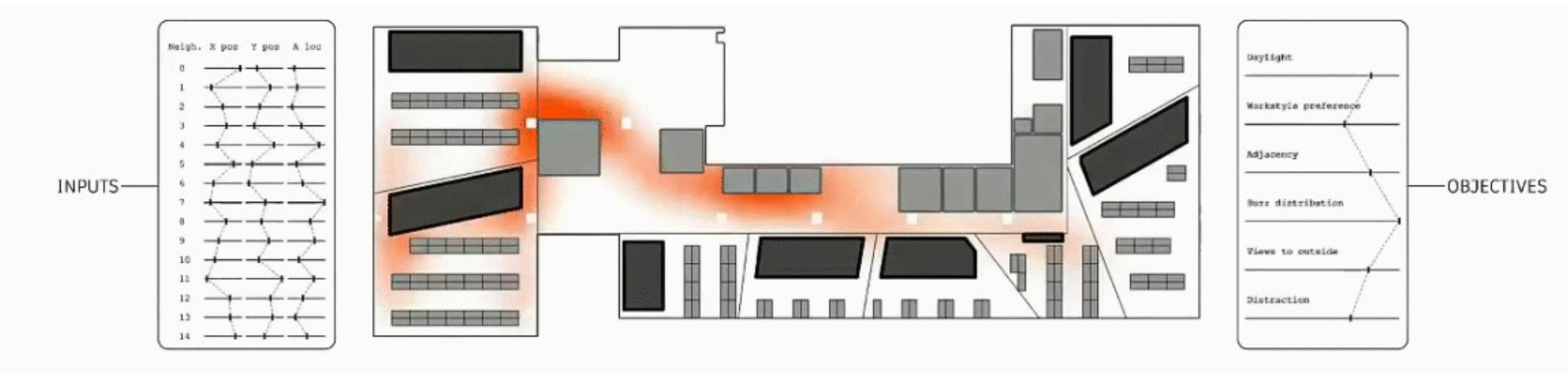
What is Generative Design?

- In a nutshell, generative design is a goal-driven approach to design that leverages automation so that designers and engineers can:
  - have better insight into their designs;
  - make faster, more informed design decisions; and
  - explore more options using the power of computers.



Why should I use Generative Design?

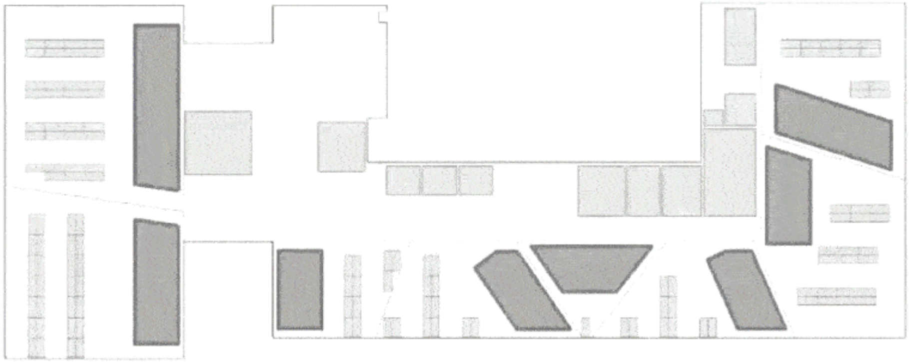
- **Better Outcomes and Insight**
  - As the designer, you specify which outcomes you want to achieve for your design and how they are measured. With your guidance, the computer produces sets of optimal designs, along with the data used to prove which design performs best against your goals. By analyzing how the generated designs measure up against the set goals, you can gain valuable insight into which design aspects impact the outcome and how.



Maximization of active shared spaces - Mars Innovation District - The Living

Why should I use Generative Design?

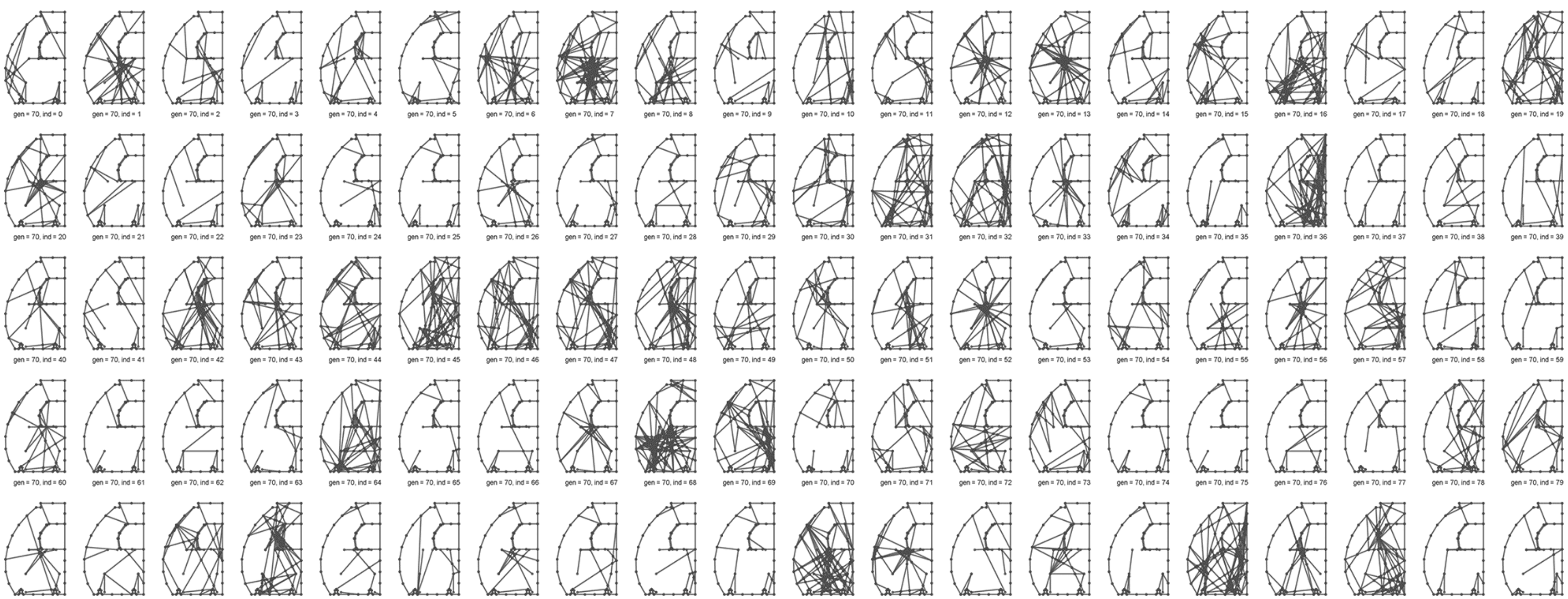
- **Faster, More Informed Design Decisions**
  - Generative design can help you find better designs for your project more quickly by leveraging what computers are good at: computation and repetition.
  - Computers can generate and evaluate a huge number of design variants in only a fraction of the time it would take an individual designer, allowing you to learn what does and doesn't work at an accelerated pace.



Design options generated - Mars Innovation District - The Living

Why should I use Generative Design?

- **A Greater Variety of Options**
  - With a generative design approach, the initial design parameters you input are used to generate your potential design solutions, with the only limitation being how much computer power and time you have.
  - For example, using traditional computational design techniques, it's feasible for you to explore ten variants (or more, perhaps). However, using generative design, an algorithm can generate thousands of variants in mere minutes.



Design options generated - Bionic Partition for Airbus - The Living

A Short History of Generative Design

'70s

- Generative Design has been the holy grail of CAD and CAE since their inception. **The earliest mentions in the late '70s focused on shipbuilding and architecture.**

'80s

- With the proliferation of CAD in the '80s, the interest in generative design increased. The results were still **limited by the computing power of the time**.

'90s & '00s

- In the '90s and early '00s, simulation-driven design, such as **topology optimization**, started to gain traction. The first structural optimization software hit the market.

'10s

- In the '10s, advancements in digital and additive manufacturing pushed companies to **accelerate the development** of commercial generative design solutions.

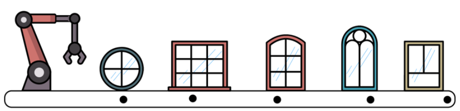
Today

- Generative design **finds applications beyond structural optimization**, enabled by the increased computational power and advanced engineering design software.

nTop

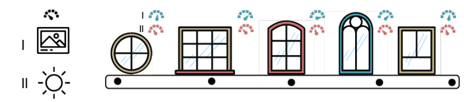
What goes into a Generative Design Process?

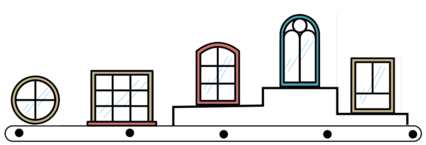
- A generative design approach allows for a more integrated workflow between human and computer



Generate
- The design options are created or generated by the system, using algorithms and parameters specified by the designer.

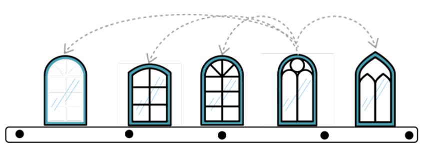
Analyze
- The designs generated in the previous step are now measured or analyzed based on how well they achieve goals defined by the designer.

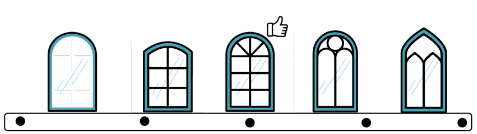




Rank
- The design options are ordered or ranked based on the results of the analysis.

Evolve
- The process ranks the design options to figure out in which direction they should be further developed or evolved.





Explore
- The designer compares and explores the generated designs, inspecting both the geometry and evaluation results.

Integrate
- The designer chooses a favorite design option and integrates it into the wider project or design work.

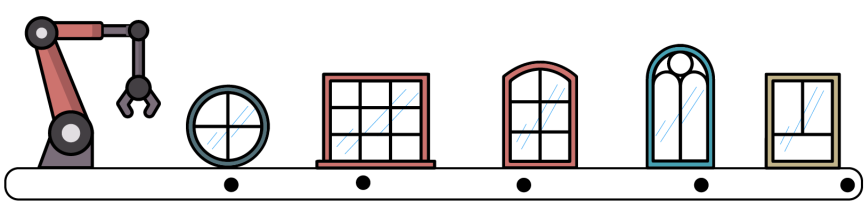
Anatomy of Each Stage

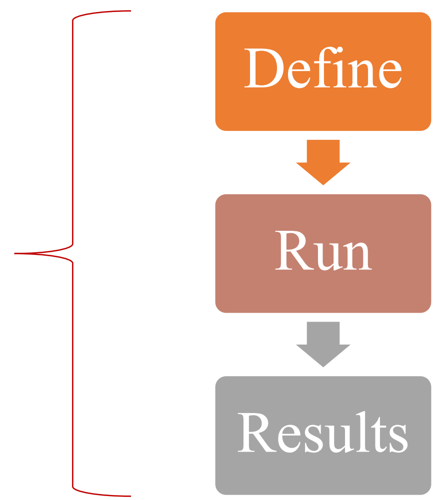
- Each of these stages can be further broken down into define, run and results steps.
  - The ***define*** step is the responsibility of the designer,
  - while the ***run*** and ***results*** steps are performed by the computer.

- Take the **Generate Stage** for example

Generate

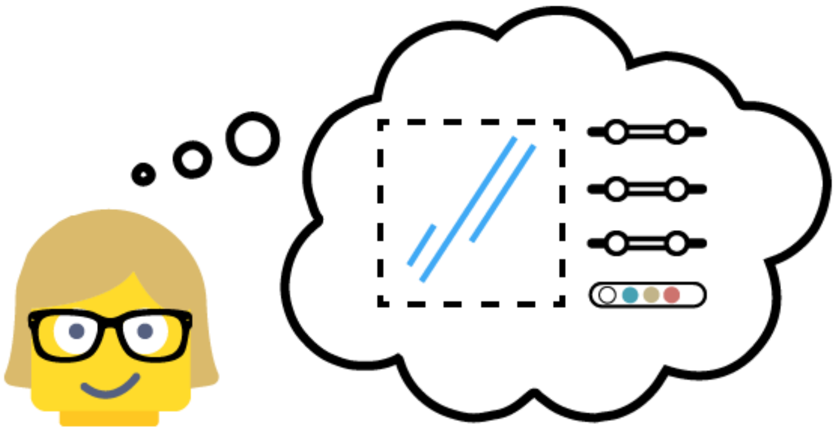
The design options are created or generated by the system, using algorithms and parameters specified by the designer.





Anatomy of Each Stage: Define

- For the define step, the designer will need to do the following:
  - Establish the generation algorithm - this is the logic that defines how designs are generated, which may include things like constraints and rules.
  - Provide the generation parameters - these are the variables or inputs needed for the previously-defined algorithm.

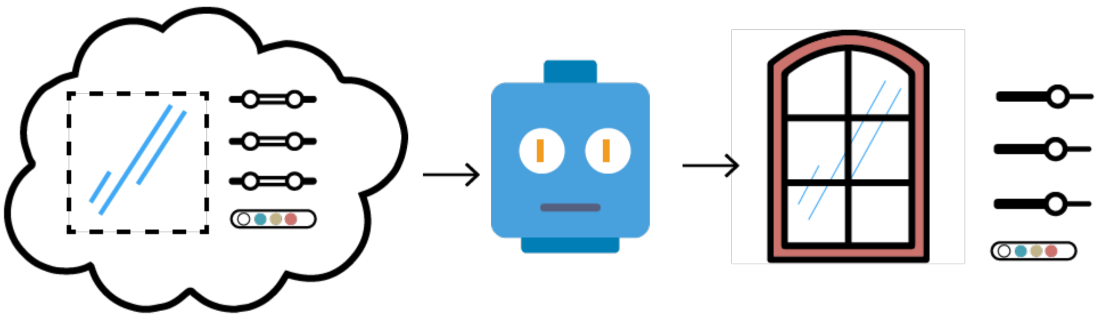


- This define step is present and vital for all stages of the generative design process, as the validity of outputs relies on the quality of the designer's contribution in this step.

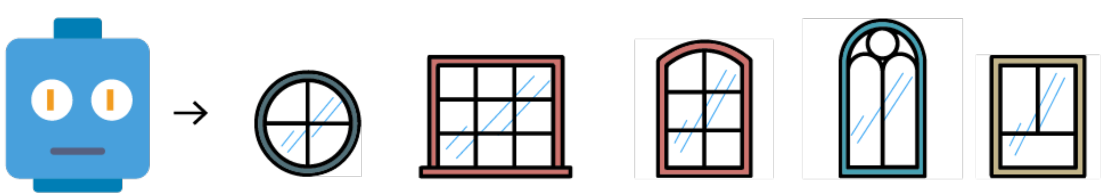
- With clear and concise logic, the computer can provide suitable outputs.

Anatomy of Each Stage: Run & Results

- Run
  - Once everything is defined in the algorithm and its accompanying parameters, the computer begins to run, meaning it starts to generate different design options.
  - This process might happen locally on the designer's computer or, for more intensive calculations, it may happen using cloud computing.



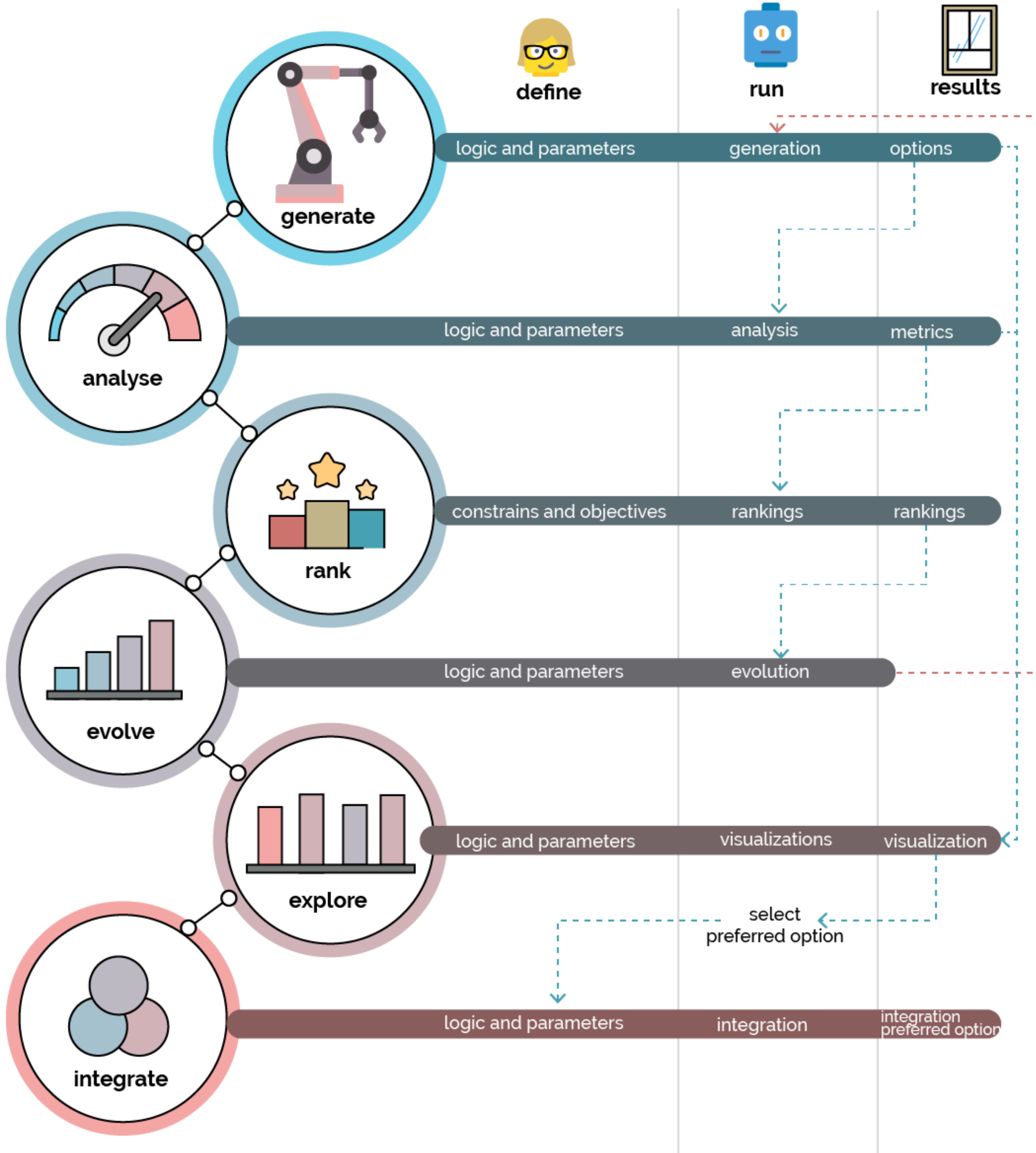
- Results
  - The things that are generated during the run step are the final outputs from each stage. These are then used as inputs or parameters in subsequent phases.
  - For example, the designs created in the generate phase will be used as one of input parameters in the analysis phase.



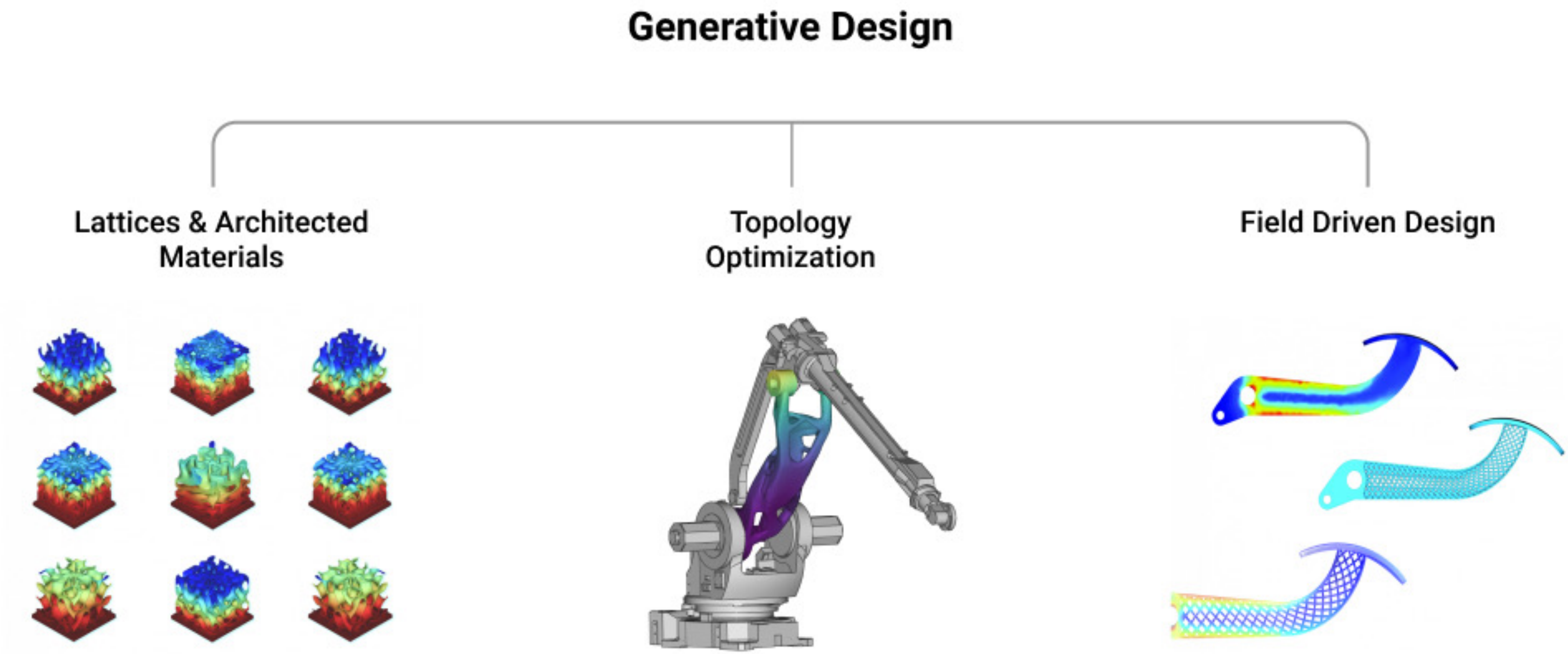
Generative Design

Overall Process

- The diagram shows
  - Each stage and step is **dependent** on the previous one.
  - The entire study process is **repeatable**, as each iteration learns from the previous results.



Generative Design Vs. Topology Optimization



- Often (erroneously) used interchangeably. Both are valuable simulation-based engineering design terms, but they have distinctly different meanings.
  - **Topology optimization** is a simulation-driven structural optimization tool. Designers define the technical requirements, and the software removes material from the designated design space through iterative simulation steps.
  - **Generative design** is a broad design methodology that allows engineers and designers to build both technical and non-technical requirements into their models.

Remove

vs.

Generate

Generative Design Vs. Topology Optimization

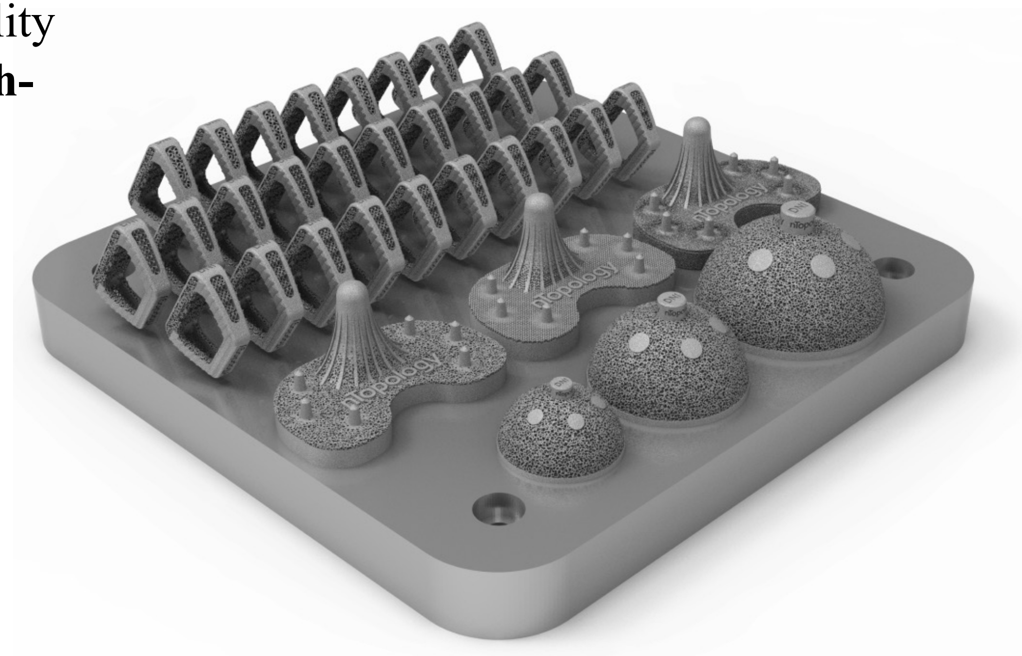
https://www.youtube.com/watch?v=QLA92V_85_I

Generative Design & Additive Manufacturing

- Generative design enables the development of high-performance 3D printed products and is a near-necessity for any DfAM workflow.

DfAM = Design for Additive Manufacturing

One of the key benefits of industrial 3D printing is that it gives engineers the ability to manufacture **highly complex** and **high-performance parts** that are either impossible or prohibitively expensive to produce using traditional techniques.



However, modeling these complex and optimized geometries manually in traditional CAD software is **a near-impossible task**. The digital toolset of generative design enables engineers to manage the complexity of additive manufacturing and use it to their advantage.

Benefits & Limitations

Benefits of Generative Design

High-performing products

The digital capabilities of generative design can unlock a previously inaccessible design space. Using tools such as topology optimization, advanced lattice structures, and field-driven design, you can build lighter, higher-performing parts with increased functionality. Generative design has applications in every field of product development; from improving thermal management in electronic devices to developing more efficient rocket propulsion systems to reduce the cost of shooting payload into orbit.

Faster time to market

Generative design accelerates all stages of product development, from concept design to manufacturing. Using its digital tools, engineers can quickly generate geometries with high complexity, from organic, freeflow parts to repetitive patterns with millions of elements. Since manufacturability can be taken into account early in the design process, the probability that time-consuming revisions are needed later on is much lower.

Unbiased engineering solutions

While designing new products, engineers tend to draw inspiration from their past projects and experiences. While this is exceptionally valuable, an algorithmic approach (such as a well-designed generative process) can produce unbiased results that may contradict preconceived notions. Combining these results with the engineer's experience leads to faster and more radical product innovation.

Limitations of Generative Design

(Potentially) Non-transparent workflows

Engineers frequently need to know just as much about the process as the resulting solution. Due to the complexity of generative algorithms, many software solutions operate using a "black box" approach. The engineer gives inputs and then is asked to evaluate the outputs without having visibility or control over the process that was followed in the backend. For mission-critical applications where design outputs must produce repeatable and reproducible results, this significantly hinders the adoption of specific generative design implementations.

Limited range of optimization requirements

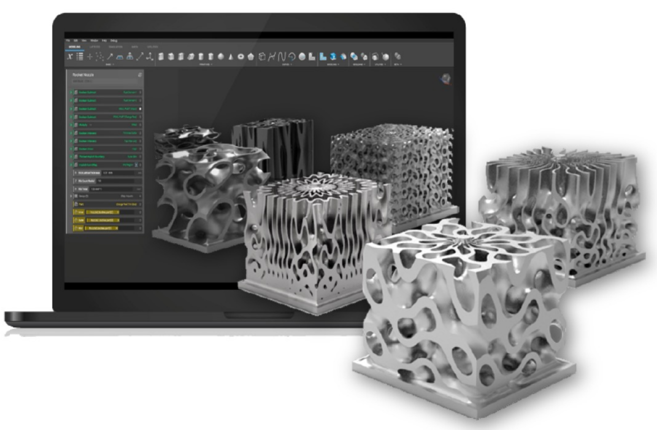
When performing generative design, it is essential to remember that your solution is only as accurate as the simulations used to produce it. Many physical phenomena aren't supported by most generative design software. This means that the result is optimized only for the limited set of design requirements that the software can handle. It is crucial to recognize that there are often many design requirements that may have not been taken into account during optimization.

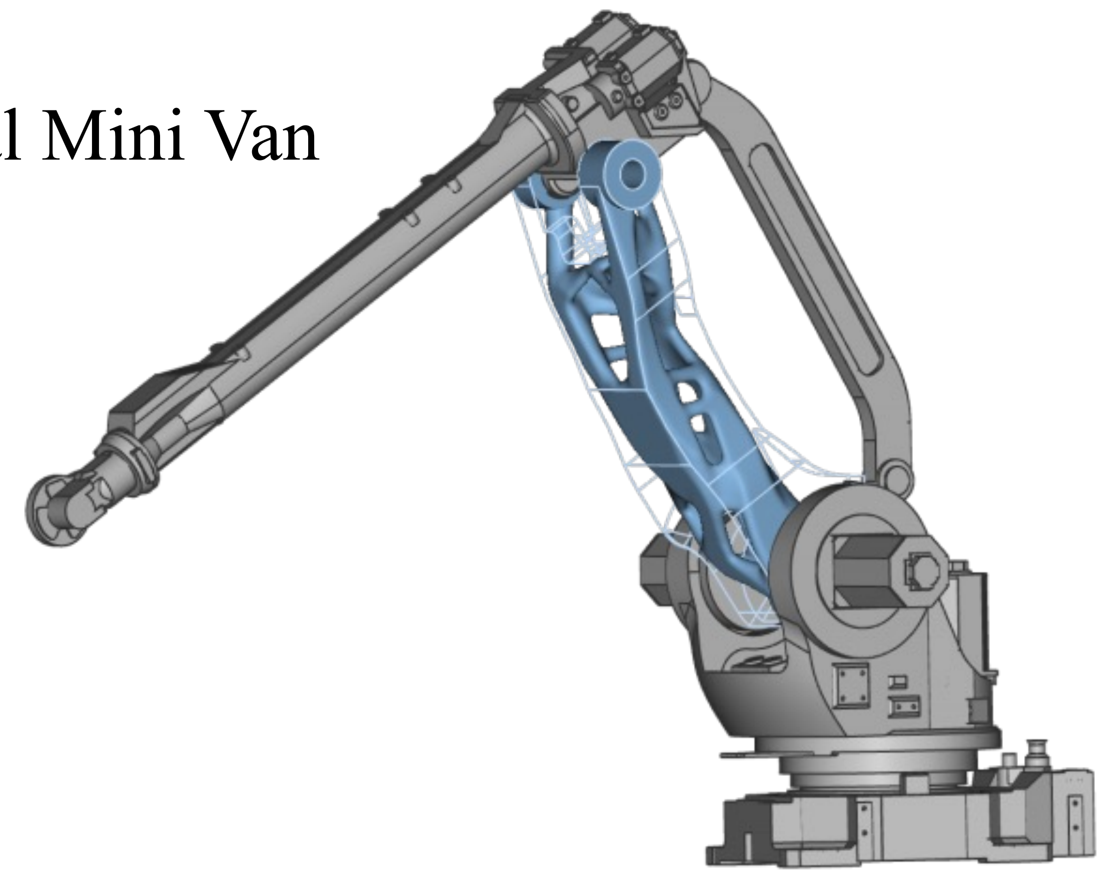
Output quality depends on input quality

Generative design still relies on the quality of information that an engineer can supply. Generative Design has two main input components: the design space and the loading conditions. To get optimized output, both the problem and the inputs need to be defined accurtaly. It is the job of the engineer to define the input parameters and the goal. For this reason, you should think of generative design as a collaborative process between the engineer and the design software.

Examples of Generative Design

- MaRs Innovation District of Toronto
- Furniture Design
- A Further Analogy
- VolksWagen’s Classical Mini Van

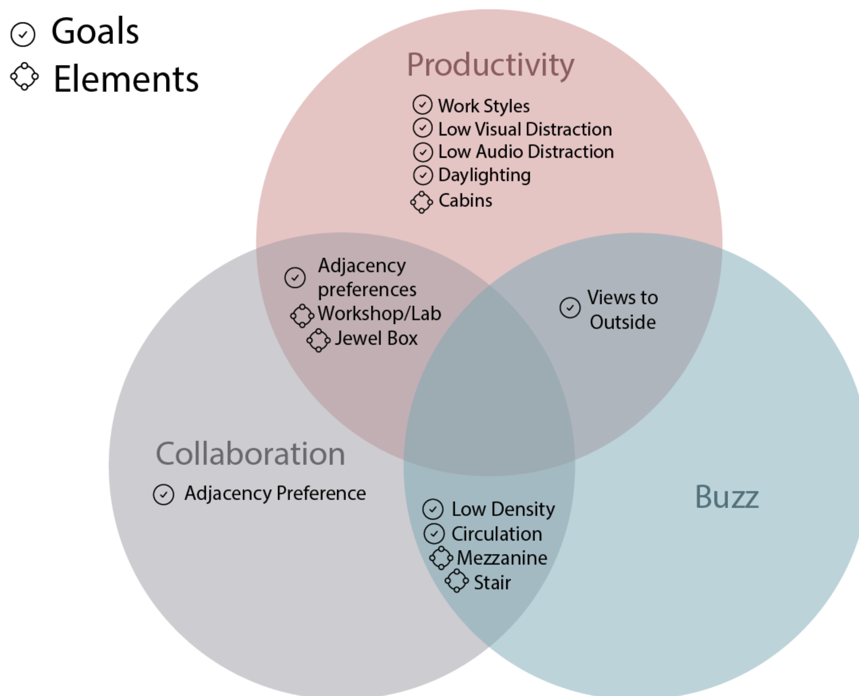




MaRs Innovation District of Toronto

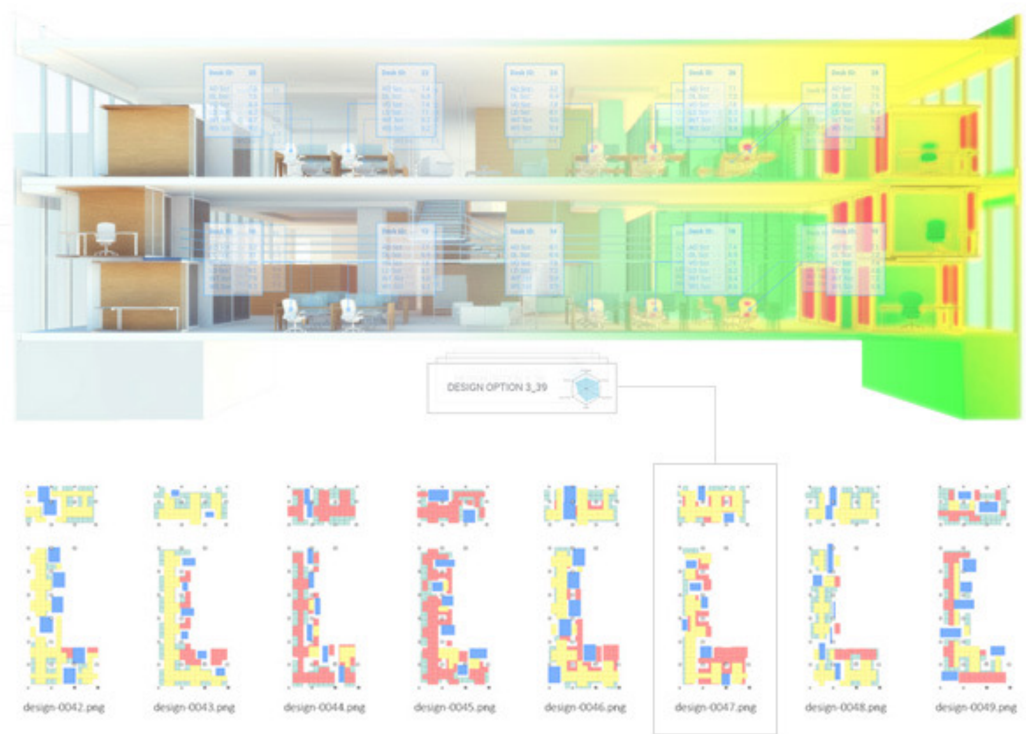
- To design the new office and research space in the MaRs Innovation District of Toronto, Autodesk used generative design processes.
  - Starting with high-level goals and constraints, the design team used the power of computation to generate, evaluate and evolve thousands of design alternatives. The result was a high-performing and novel work environment that would not have been possible without this approach.

- **Generate**
  - The designers created a geometric system that meant the computer could explore multiple configurations of work neighborhoods, amenity spaces and circulation zones.
  - This work represents the *define* step of the *generate* phase.
  - Using this algorithm, the computer varied the parameters to produce thousands of design options.



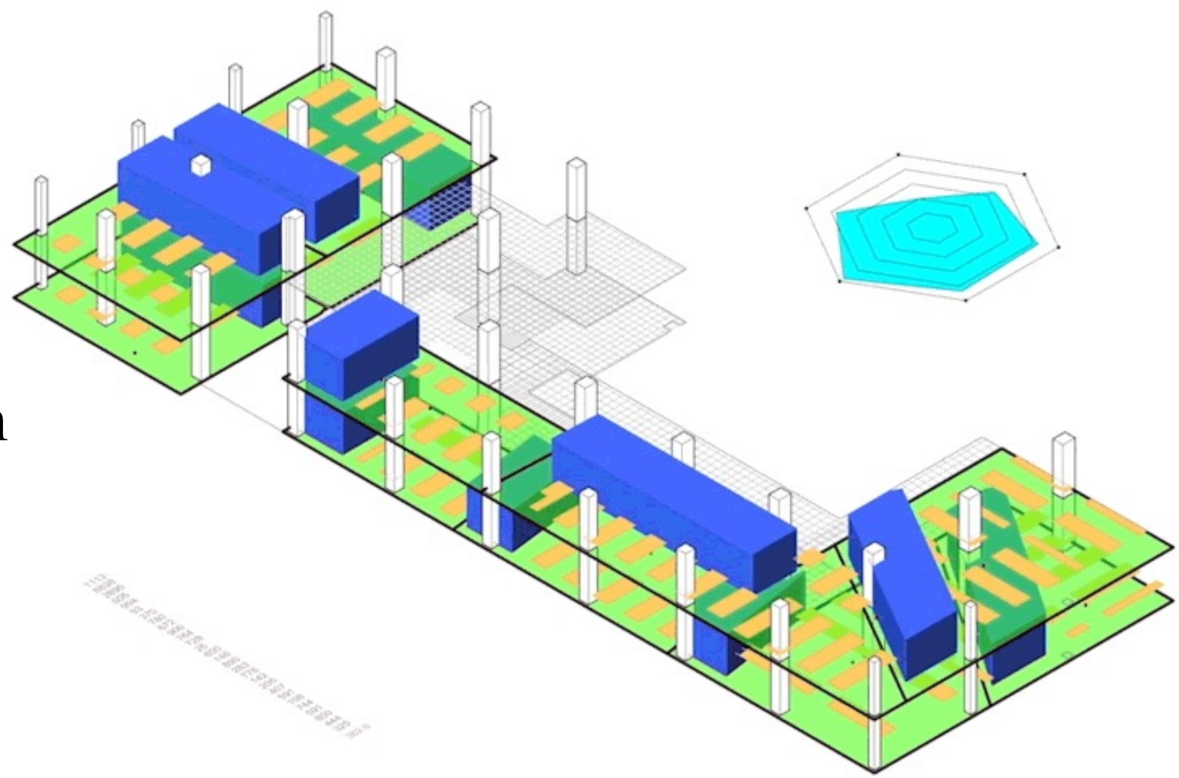
MaRs Innovation District of Toronto

- **Evaluate**
  - For this stage, information was collected from employees and managers about work styles and location preferences. Based on this data, six primary and measurable goals were defined:
    - work style preference
    - adjacency preference
    - low distraction
    - interconnectivity
    - daylight
    - views to the outside



MaRs Innovation District of Toronto

- **Explore**
  - After the designs were evaluated, the designers looked at the solution space to explore the generated designs together with their evaluation results.
  - Considering each defined goal, they identified the design that best achieved the goals overall.

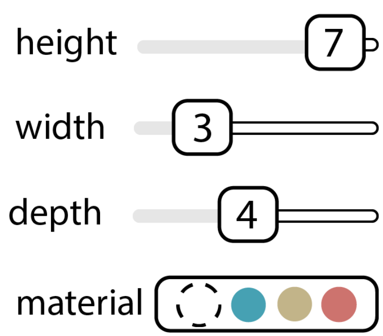


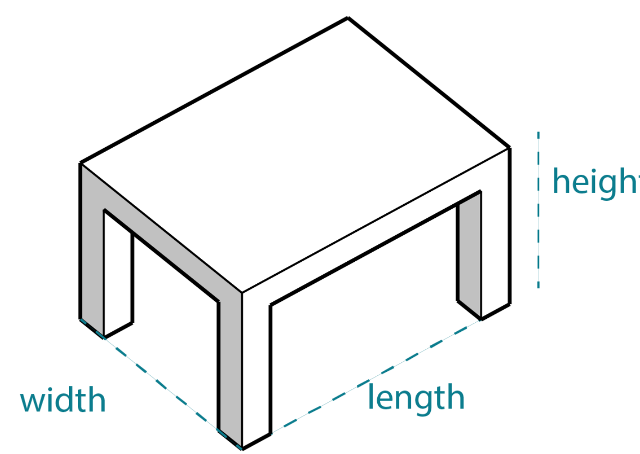
Furniture Design

- Looking at a simpler example, let's consider the process of designing a typical, four-legged table.
- **Using a standard approach**, you as the designer would manually define the length, width, height and material of the table.
    - The resulting output is **a single, physical object with a fixed, immutable form**.
    - Here, you have the option to test several distinct sets of dimensions and material combinations to end up with three or four prototypes (or however many iterations you wanted).

In a generative design approach, you would instead create an algorithm that specifies:

- a range of permissible values for each dimension;
- a series of available materials and their properties (such as cost/m²); and
- a set of goals that measure how successful a table design is.



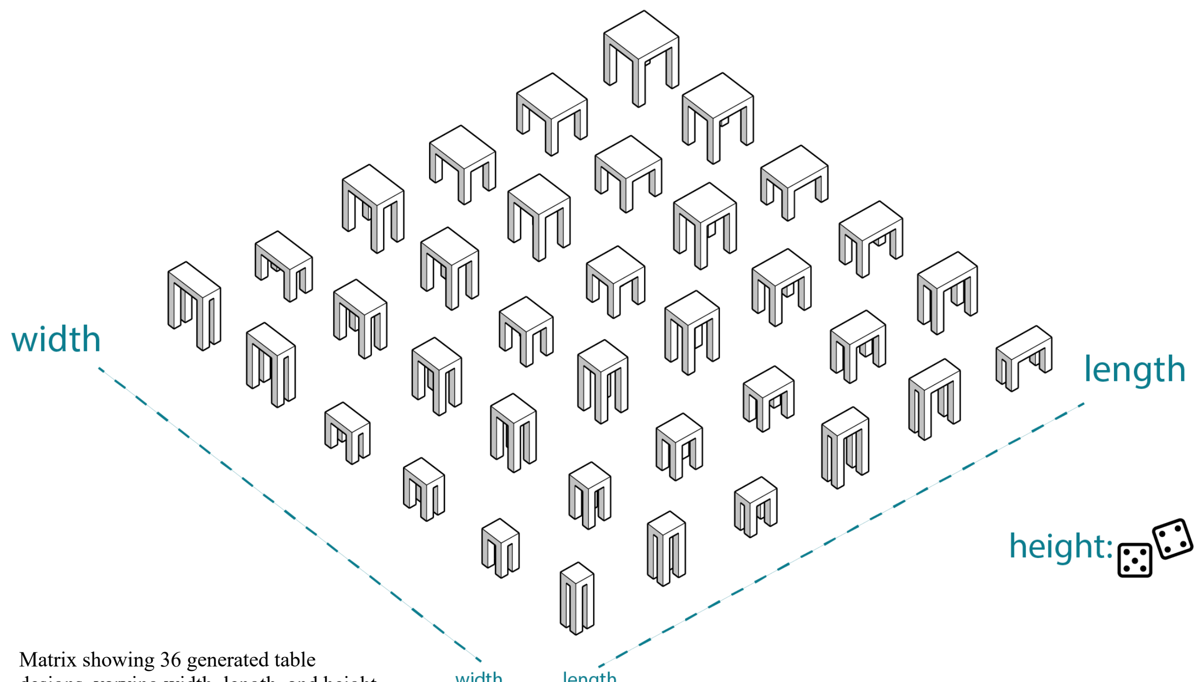


Furniture Design

- Generate
  - Then, you would use a computer to run the algorithm and generate a series of designs that fall within the ranges you previously specified.
  - Some designs will be short and wide, others will be tall and thin, but each will satisfy the user-defined constraints. This is key, as many designs can be generated very quickly, much more than any human could feasibly examine.

Let's imagine the computer looked at 20 different values for each of: length, width, height, table/leg material combinations.

The resulting solution space would be 20*20*20*20 = 160,000 designs, which is way too many options for a person to reasonably evaluate.



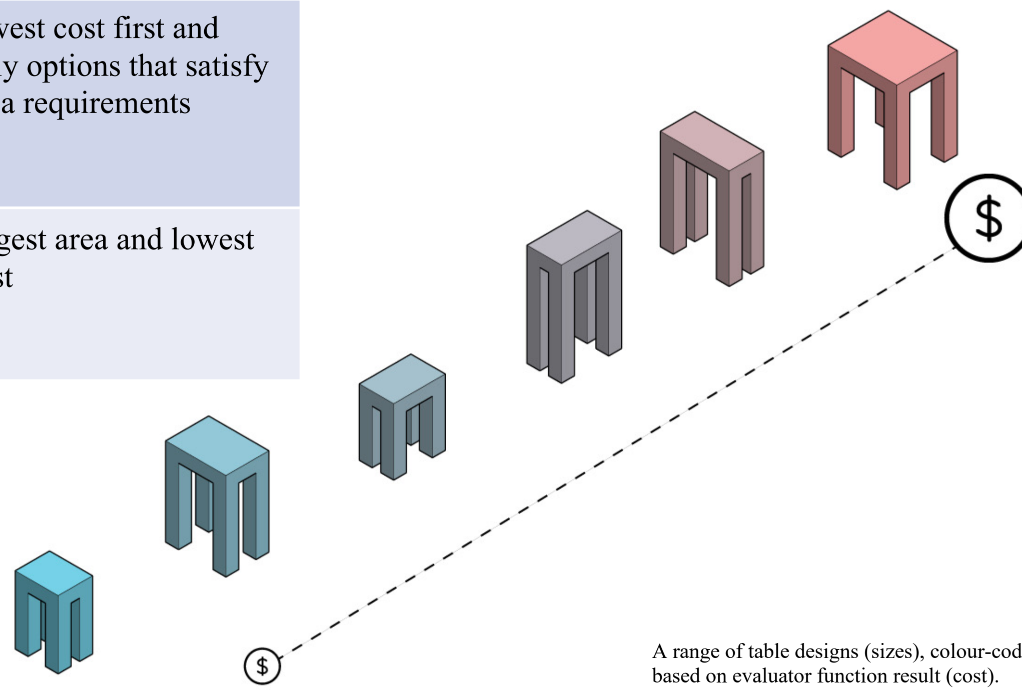
Matrix showing 36 generated table designs, varying width, length, and height.

Furniture Design

- Evaluate
  - The next step is to define how the generated designs are evaluated. This is your opportunity to clearly express your design goals.
  - Let's see how different design goals could be expressed in this *evaluation* stage:

Design goal	Analysis method	Ranking method
lowest cost per desk, with minimum 800mm x 600mm size	desk size: at least 800mm x 600mm in size = *yes/no* and desk cost: area * material cost/m² = currency *$* value)	lowest cost first and only options that satisfy area requirements
most profitable (largest desk area with lowest material cost)	desk area = outputs m² and unit cost (area * material cost/m²) = currency *$* value	largest area and lowest cost

The matrix above exemplifies how you can use this stage in the generative design process to design for wildly different scenarios.



A range of table designs (sizes), colour-coded based on evaluator function result (cost).

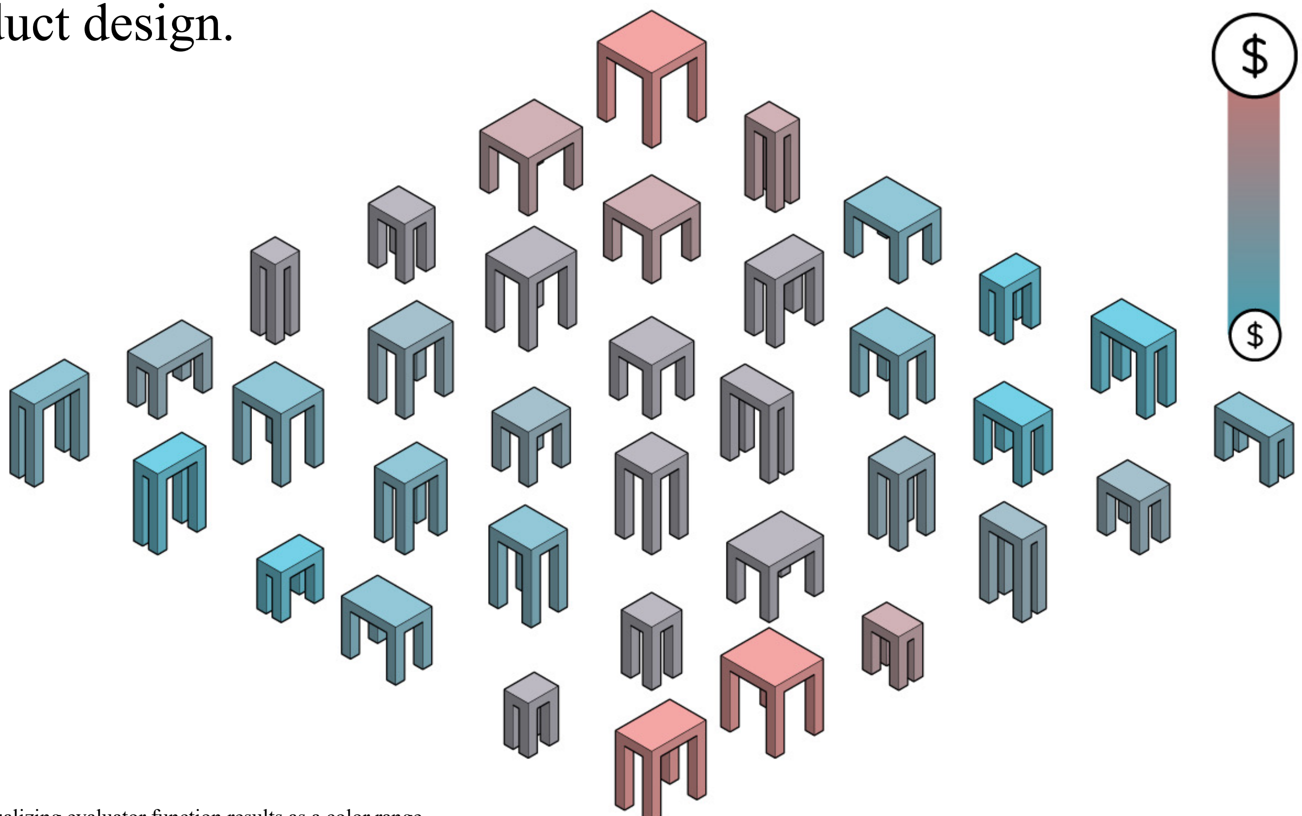
Furniture Design

- Evaluate
  - **In the first scenario**, lowest overall cost is the driving goal, so we can expect small desk sizes and cheap materials while still satisfying the size requirement.
    - This scenario would be relatively simple for humans to replicate, so generative design would only come in handy when the variation or complexity of material costs is high.
  - **For the second scenario**, we're aiming to maximize return on investment (ROI) for each desk.
    - This means that we can expect larger, more expensive desks than the first scenario, but that still have the best overall ROI. It wouldn't be unexpected for this process to identify a desk with cheap legs and more expensive tabletop materials as a viable option.
    - ***A good illustration*** *of using generative design to work towards multiple and competing goals, which is very hard for humans to replicate.*

Design goal	Analysis method	Ranking method
lowest cost per desk, with minimum 800mm x 600mm size	desk size: at least 800mm x 600mm in size = *yes/no* and desk cost: area * material cost/m² = currency *$* value)	lowest cost first and only options that satisfy area requirements
most profitable (largest desk area with lowest material cost)	desk area = outputs m² and unit cost (area * material cost/m²) = currency *$* value	largest area and lowest cost

Furniture Design

- Results
  - As you can see, both of these examples follow the same fairly generic process, which is why there are so many possible applications of generative design in areas as diverse as aviation, automotive and building design, manufacturing, and product design.



Visualizing evaluator function results as a color range.

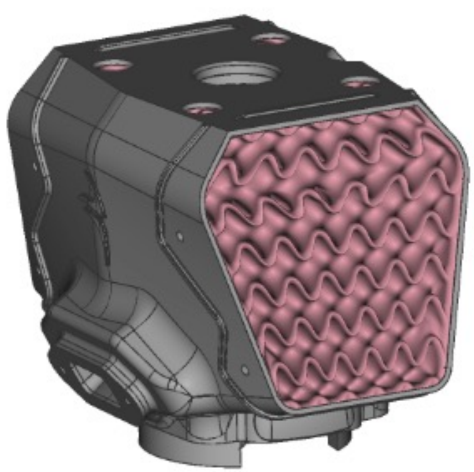
VW's Mini Van


R
S
T
X
Y
Z
ENTER
HELP
RECALL
ESC
10:24:40
10:25:16
10:25:34
11:56:22
12:12:04
07:48:20
10:41:50
14:39:08
15:20:26
15:32:28
16:16:14
11:39:38
14:30:32
16:12:38
10:17:38
bytes

Applications of Generative Design

Aerospace

Aerospace companies are applying generative design to shape tomorrow's greener, lighter and more efficient aircrafts, rockets, satellites and drones.

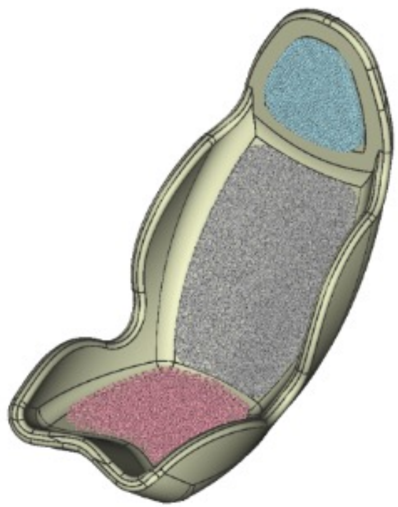


EXAMPLE APPLICATIONS

Heat Exchangers, Hydraulic and Pneumatic Systems, Landing Gear, Doors, Fuselage, Nacelles & Pylons

Automotive

With objectives centered around weight reduction, safety, and style, the automotive industry is already using generative design to develop parts for both performance and aesthetics.

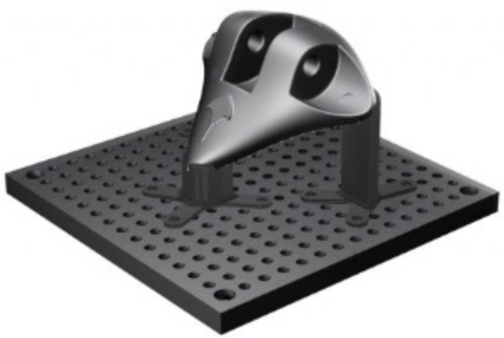


EXAMPLE APPLICATIONS

Uprights, Brake Caliper, Hydraulic Manifold, Seat Cushioning, Car Grilles, Customized Accessories

Manufacturing

Lightweighting and design automation, can enhance the efficiency any manufacturing process, from jigs & fixtures for large-scale assembly lines to customized 3D prints.



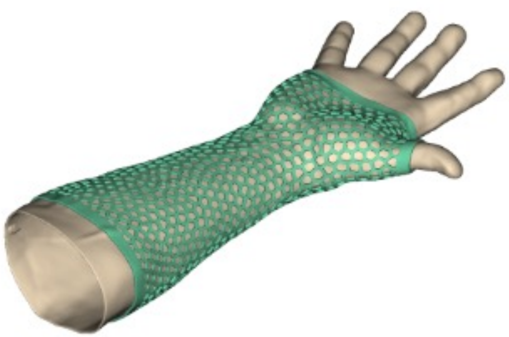
EXAMPLE APPLICATIONS

Jigs & Fixtures, Molds & Dyes, AM Build Preparation, Robotic End of Arm Tooling

Applications of Generative Design

Medical Devices

With automated design analysis and geometry generation, biomedical engineers can design a wide variety of patient-specific medical devices with unrivaled speed and customization options.



EXAMPLE APPLICATIONS

Orthopedic Implants, Prostheses, Orthotics, Casts, Dental implants

Consumer Products

Generative design software gives engineers the ability to generate manufacturing-ready design candidates, saving valuable design time and giving you a differentiation advantage.

EXAMPLE APPLICATIONS

Sports Equipment, Luxury Products, Footwear, Protective Gear

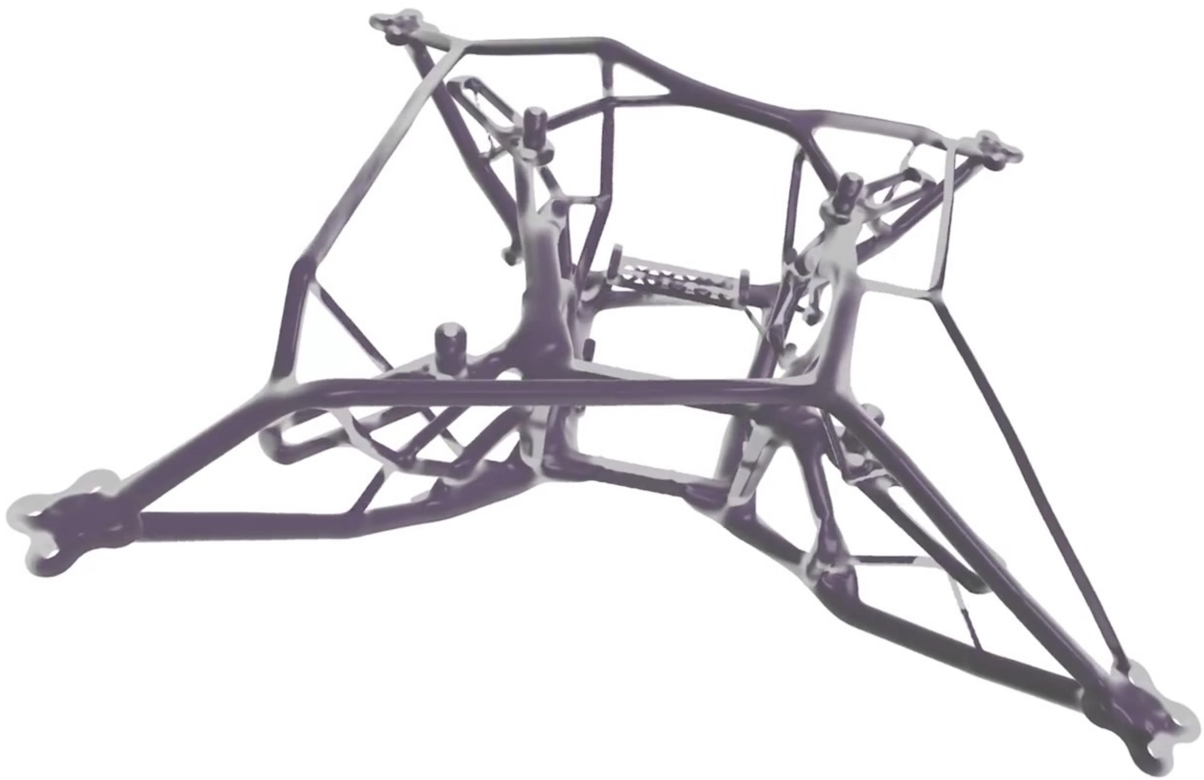
Heavy Industry

Weight reduction of heavy machinery through generative design enables engineers to minimize cost, improve safety and reduce energy consumption during both assembly and operation.

EXAMPLE APPLICATIONS

Trucks, industrial machinery, large metal casts, and forgings

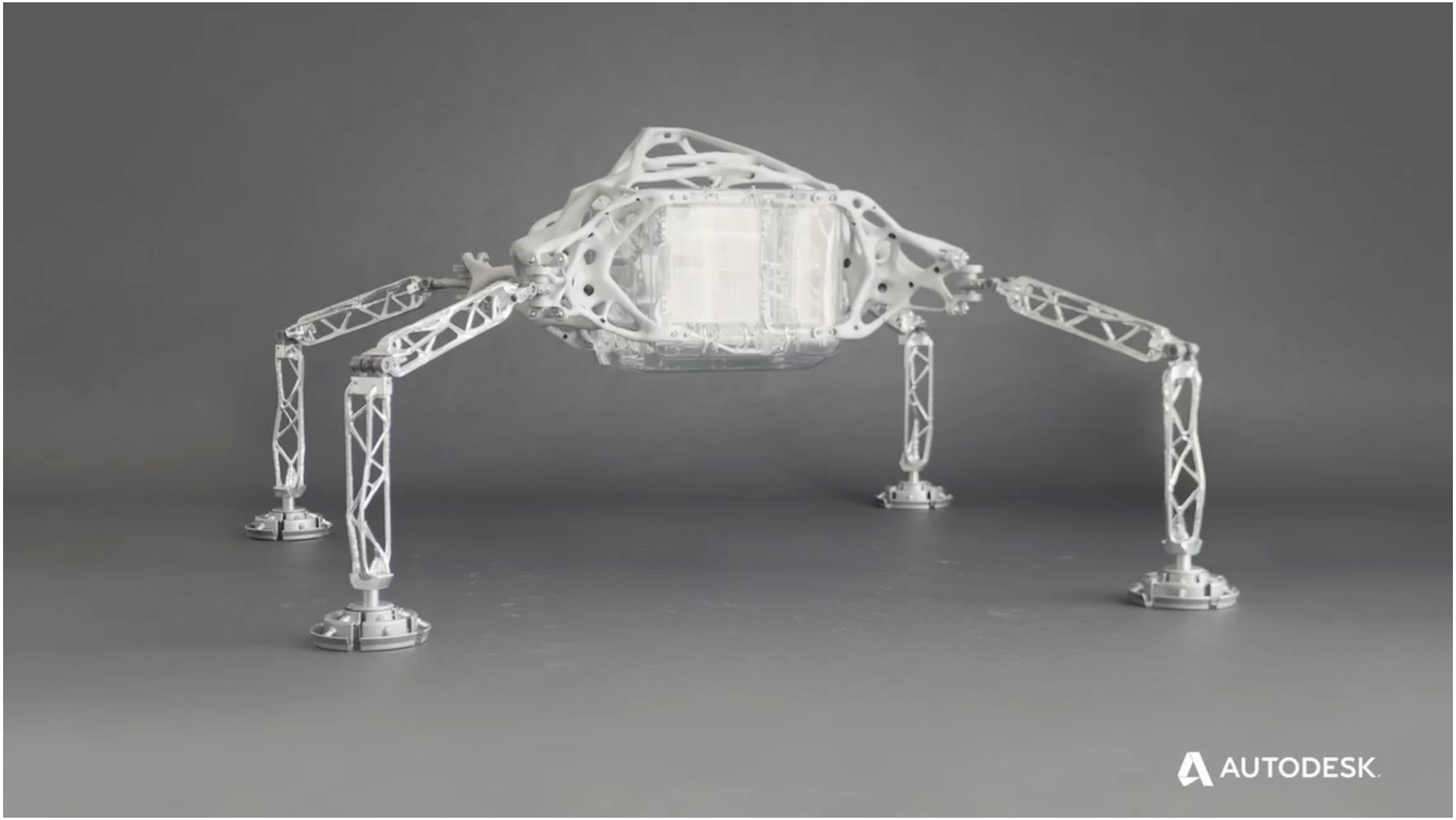
Autodesk Generative Design



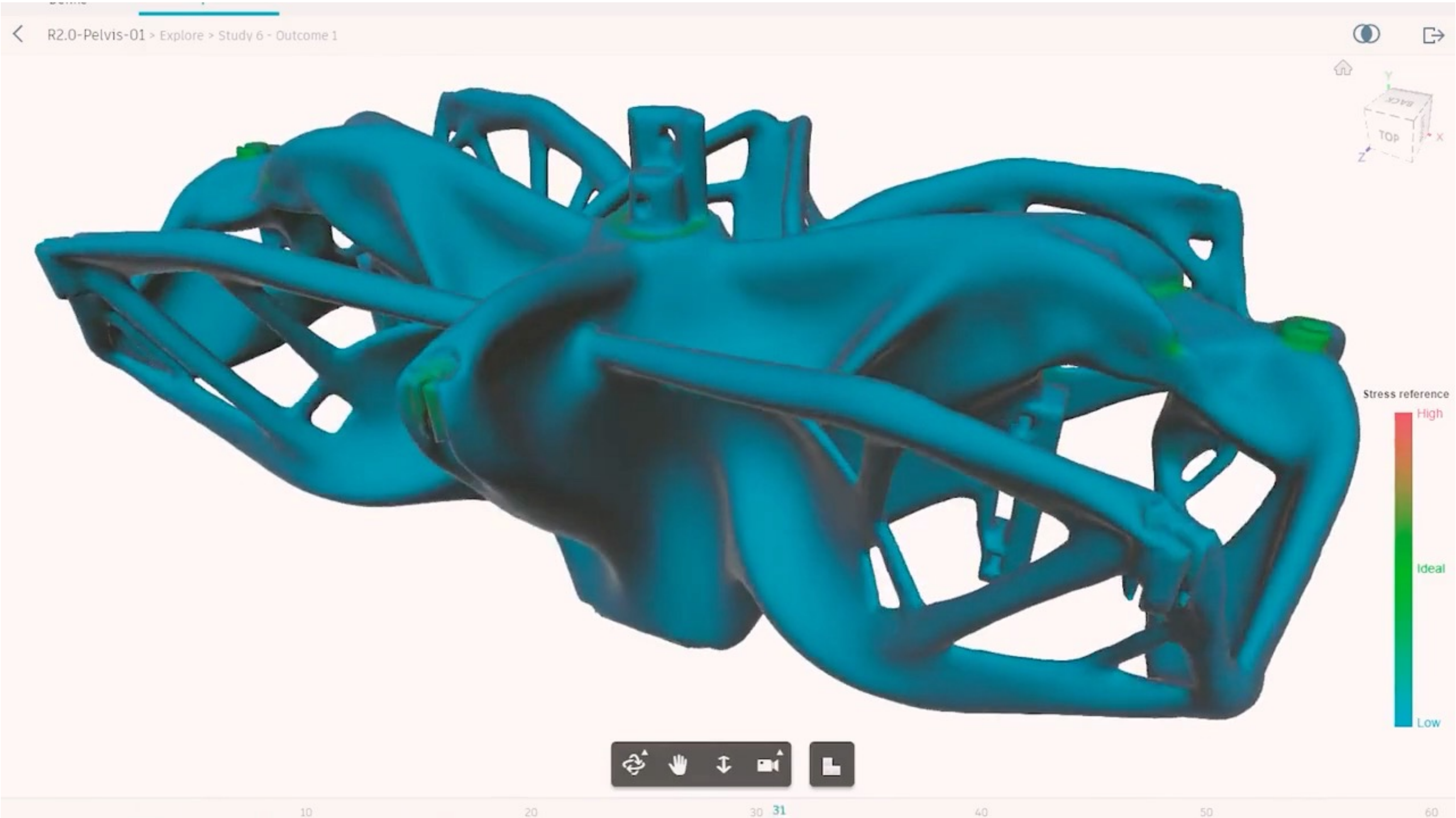
AUTODESK

https://www.youtube.com/watch?v=CtYRfMzmWFU

Generative Design NASA’s Lander

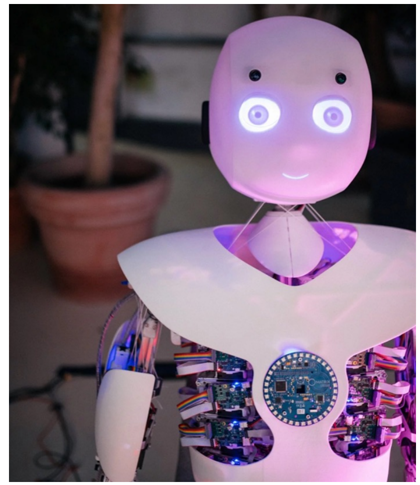


Roboy 2.0

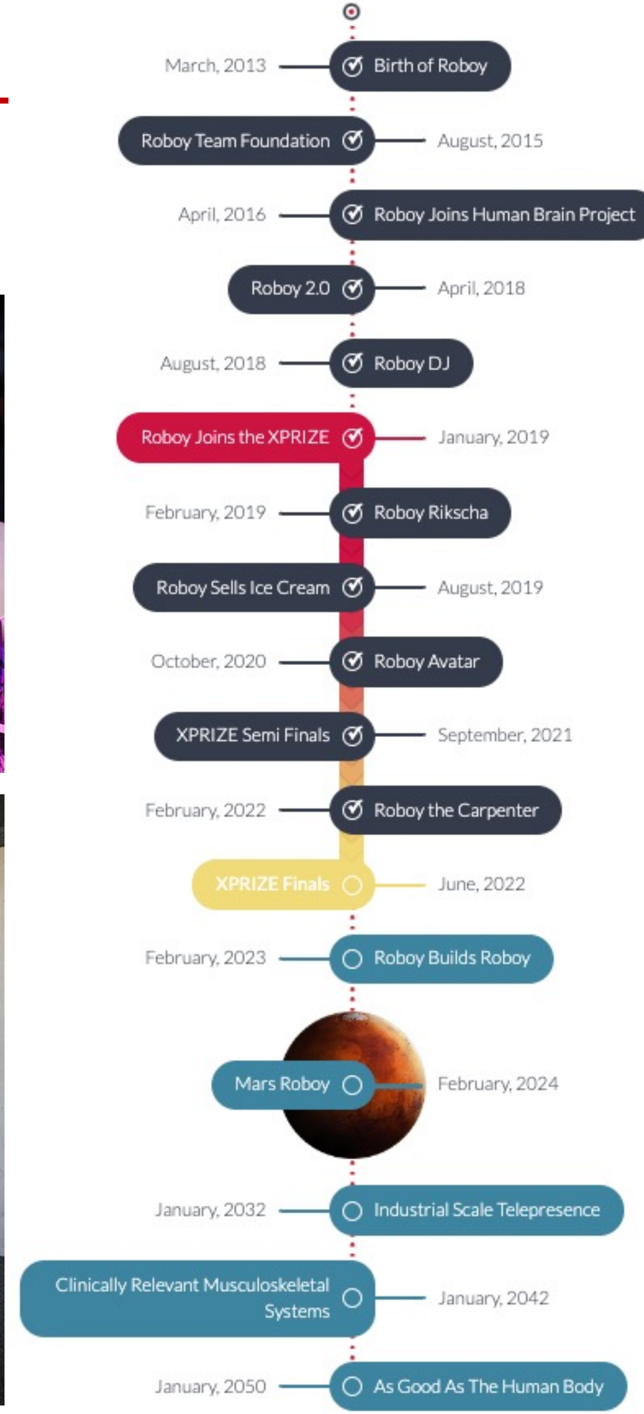


https://redshift.autodesk.com/articles/humanoid-robot-design

Roboy 3.0




2.0


March, 2013
Birth of Roboy
Roboy Team Foundation
August, 2015
April, 2016
Roboy Joins Human Brain Project
Roboy 2.0
April, 2018
August, 2018
Roboy DJ
Roboy Joins the XPRIZE
January, 2019
February, 2019
Roboy Rikscha
Roboy Sells Ice Cream
August, 2019
October, 2020
Roboy Avatar
XPRIZE Semi Finals
September, 2021
February, 2022
Roboy the Carpenter
XPRIZE Finals
June, 2022
February, 2023
Roboy Builds Roboy
Mars Roboy
February, 2024
January, 2032
Industrial Scale Telepresence
Clinically Relevant Musculoskeletal Systems
January, 2042
January, 2050
As Good As The Human Body

A.I. Chair by Philippe Starck

https://www.youtube.com/watch?v=M22NyJW8Hfo

A.I. Chair by Philippe Starck

A.I. Chair by Philippe Starck

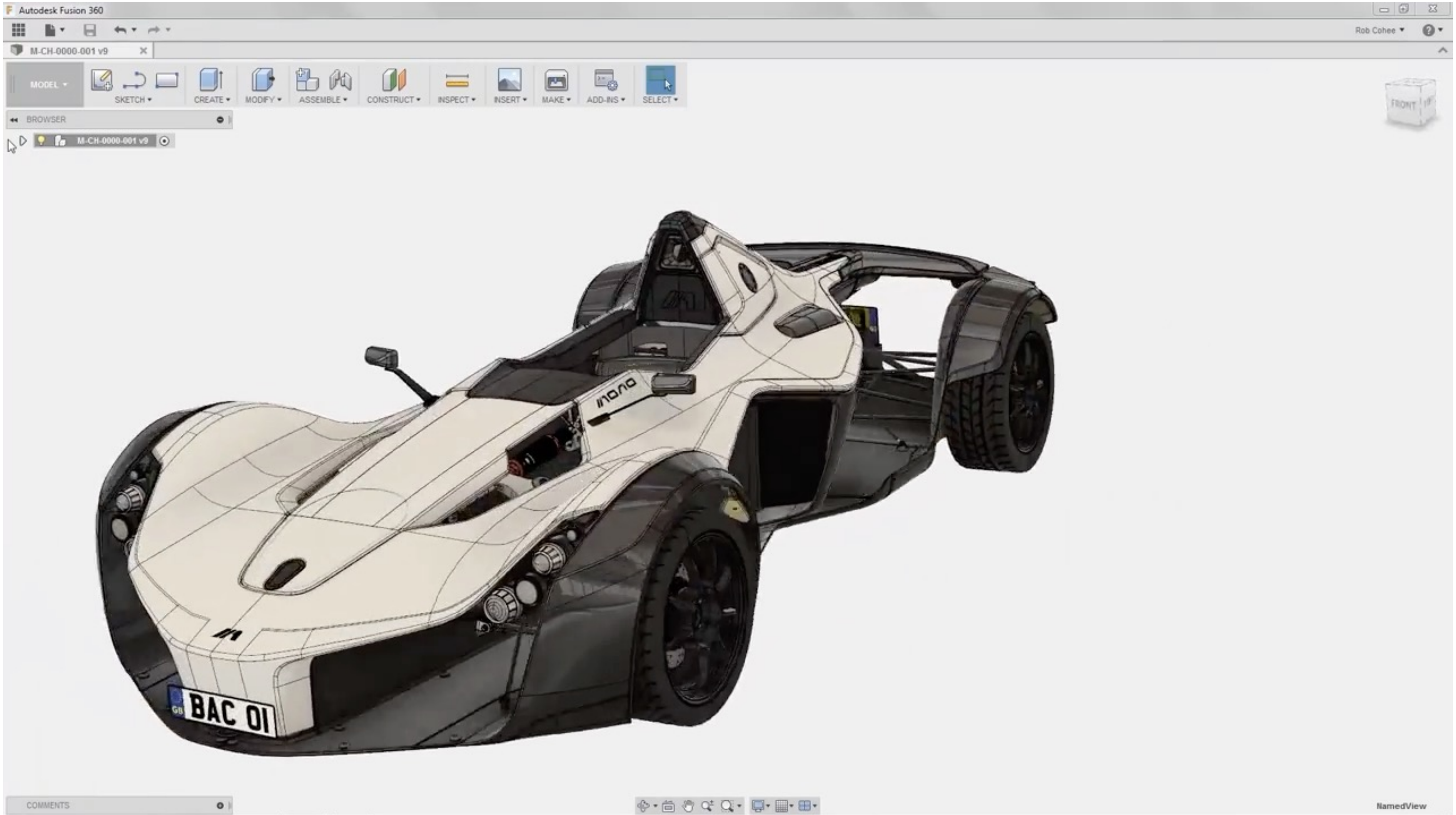
- Philippe Starck is a French industrial architect and designer known for his wide range of designs, including interior design, architecture, household objects, furniture, boats and other vehicles.

  --Wikipedia

- *... design has no future, because matter has no future. we enter now the era of dematerialization and bionism, that is to say the alliance of the body with integrated high technology. in the upcoming years, all the useless things around us will disappear, they will directly integrate our environment and our body ...*

  --philippe starck.

Fusion 360 Generative Design Technology



https://www.youtube.com/watch?v=a6bDLMWlS98

谢谢~

宋超阳
南方科技大学