Device ontology

An enabling technology for consistent modeling of physical systems

Riichiro Mizoguchi

JAIST

(Japan Advanced Institute of Science and Technology)

Agenda

- Modeling in Engineering
- What is Device ontology and why do we need it?
  - Underlying philosophy
  - The key concept with examples
- Applications
  - Function definition
    - Artifact functions and Biological functions
  - Function decomposition
  - etc.

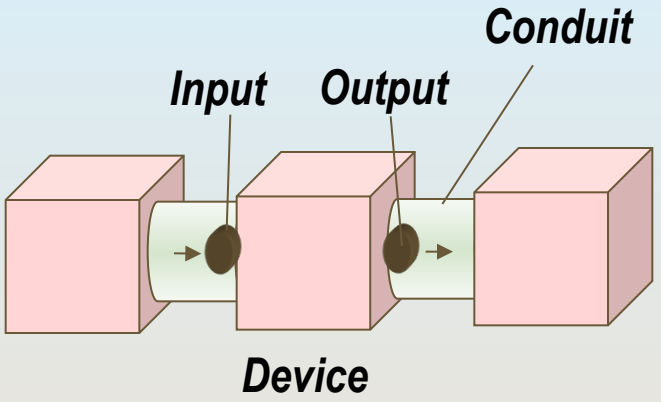
Modeling the targets

- Quantitative VS. Qualitative models
- Domain ontology
- How to **capture essentials** of the domain in a domain-independent manner

Capturing essentials

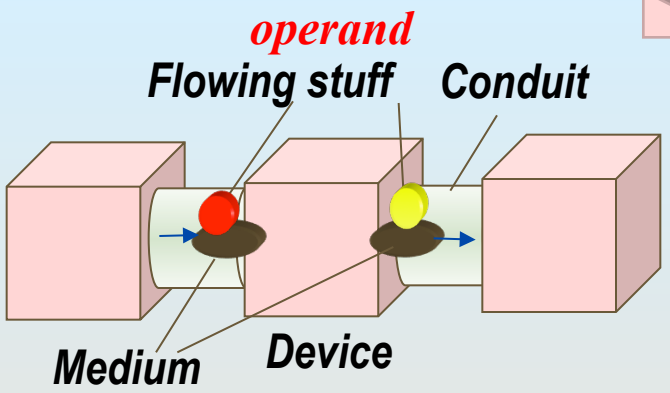
- Domain-specific terms/concepts
  - Fine-grained and expert-friendly
  - Small generality with large specificity
  - Hard to capture essentials
- Domain-independent terms/concepts
  - Coarse-grained and abstract
  - Large generality with less specificity
  - Sometimes too general to be useful
- Another way of abstraction/generalization ??
  - **Moderate generality with links to domain specificity**

Device ontology

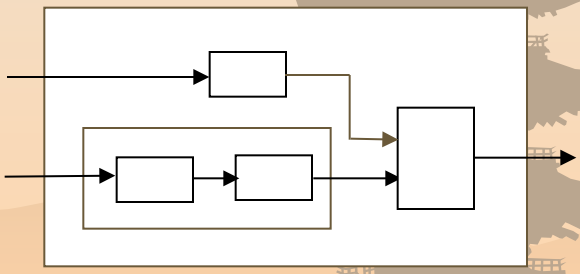


- Basic elements
  - **Device**
    - It receives input, operates on it, and outputs it
    - Black box assumption
  - **Connection**
    - A device has Input/output ports and connected with one another
  - **Conduit**＝Semi-device
    - Ideally, it can be considered that it changes only location of the input to output.
      E.g., pipe, wire

Extended device ontology



- Major elements
  - **Device (*agent*)**
    - It processes the flowing thing (***operand***). **Black box** assumption
  - **Conduit** ＝Semi-device
    - Ideally, it can be considered that it changes only location of the flowing thing.
      E.g., pipe and shaft
  - **Operand** (which is processed by the ***device***)
    - It flows through devices
    - The change of its state is interpreted as **the behavior of the device**.
      E.g., energy, fluid, motion & force, information
  - **Medium**
    - It keeps/carries ***operand*** to enable it to flow
      E.g., fluid (carrying heat) and shaft (carrying torque)
    - It flows, but not always (case by case)
  - Connection, Input/output



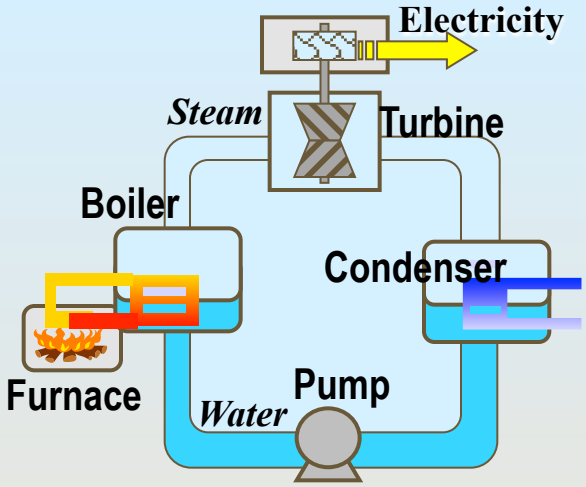
Roles

- A role is a dependent entity played by an entity in a context
  - **Teacher** in a school context, **wife/husband** in a marriage context
- Any object and occurrent can be a context of a role
  - Part role (*teacher*) and participant role (*singer, destination*)
- An entity becomes a role-holder when playing a role
  - A role-holder inherits properties inhering in the role by playing it
  - So, the **player** and the **role-holder** are **different entities**
  - Domain entities are potential players for a role(s)
  - Depending on the context, an entity can plays multiple roles

Device ontology is a Role assignment system in modeling phenomena in reality

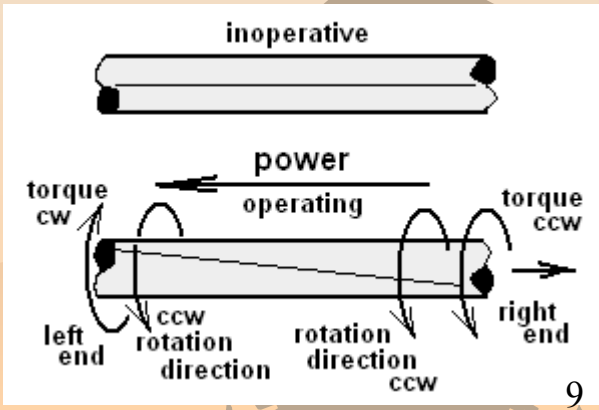
- The world is full of **Roles** **depending on contexts**
- **Role vs. Players (domain entities)**
- **We need a guiding principle to enable us consistent interpretation of the domain/target domain-independently**
- **Device ontology is a candidate**

Examples

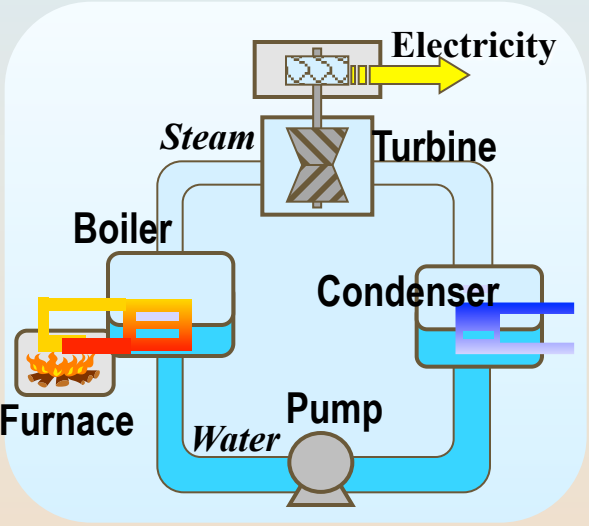


- In the power plant domain (e.g., a pipe, a boiler)
  - A pipe plays a ***conduit*** *role* to allow the water/steam to flow
  - Heat energy plays an ***operand*** *role*, and is
  - carried by the flowing water/steam which plays the ***medium*** *role*
  - The ***medium*** (liquid) flows through the devices/conduits

- In the mechanical engineering(e.g., a transmission)
  - A shaft plays a ***conduit*** *role*
  - Torque plays the ***operand*** *role*, and is
  - carried by a rotating shaft which allows **torque** to flow.
    A shaft plays the ***medium*** *and* ***conduit*** *roles*
  - The ***medium*** (shaft) does NOT flow



Two use contexts

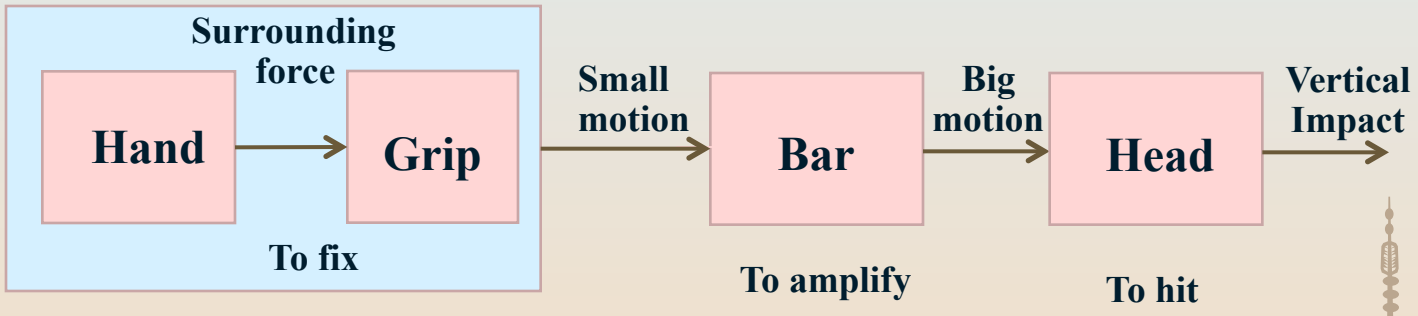


- Used as a component in a system
  - Imagine a pump
  - Who uses it?

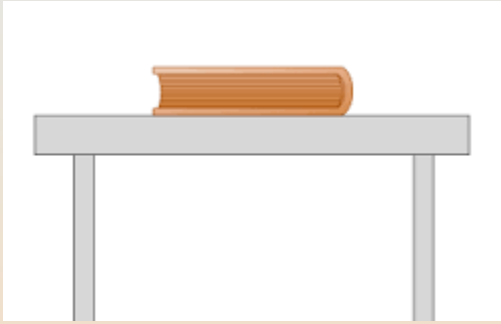
Systemic context

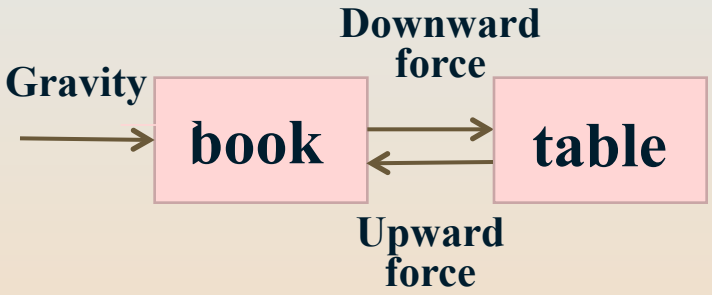
- Used as a tool
  - Hammer or screwdriver
  - A human ***uses*** it

The hammer case



Another case

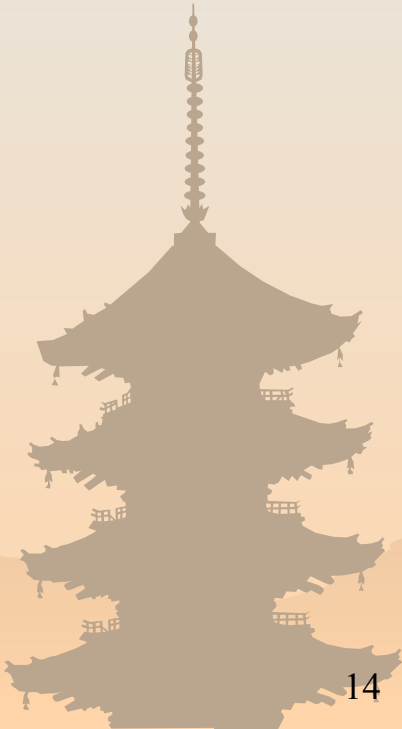




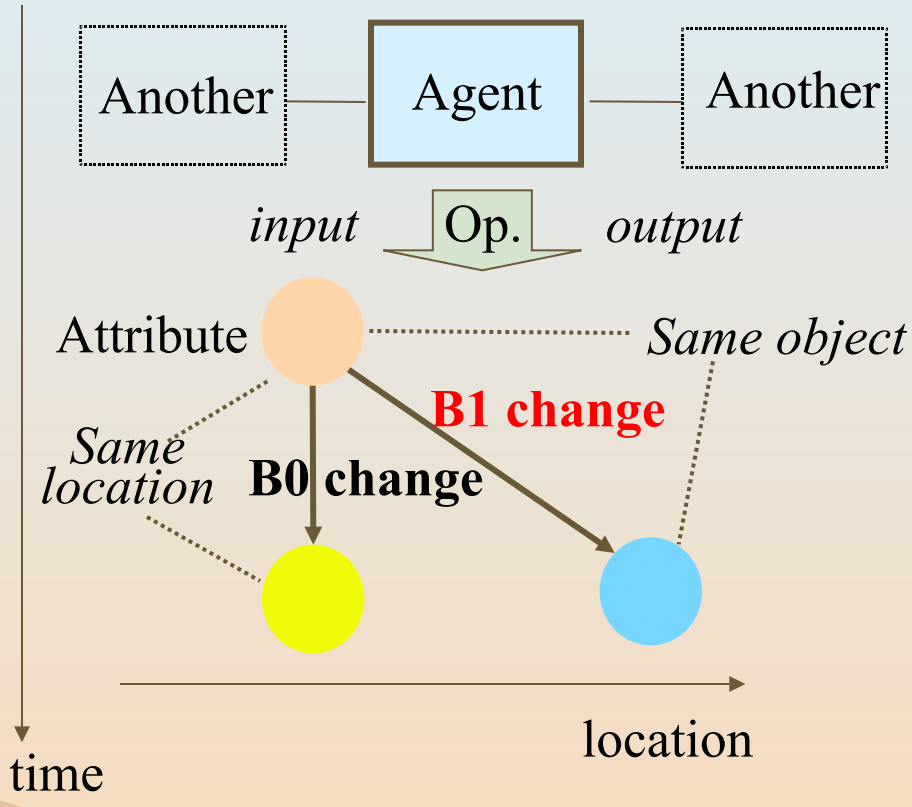
Essentials of Device ontology

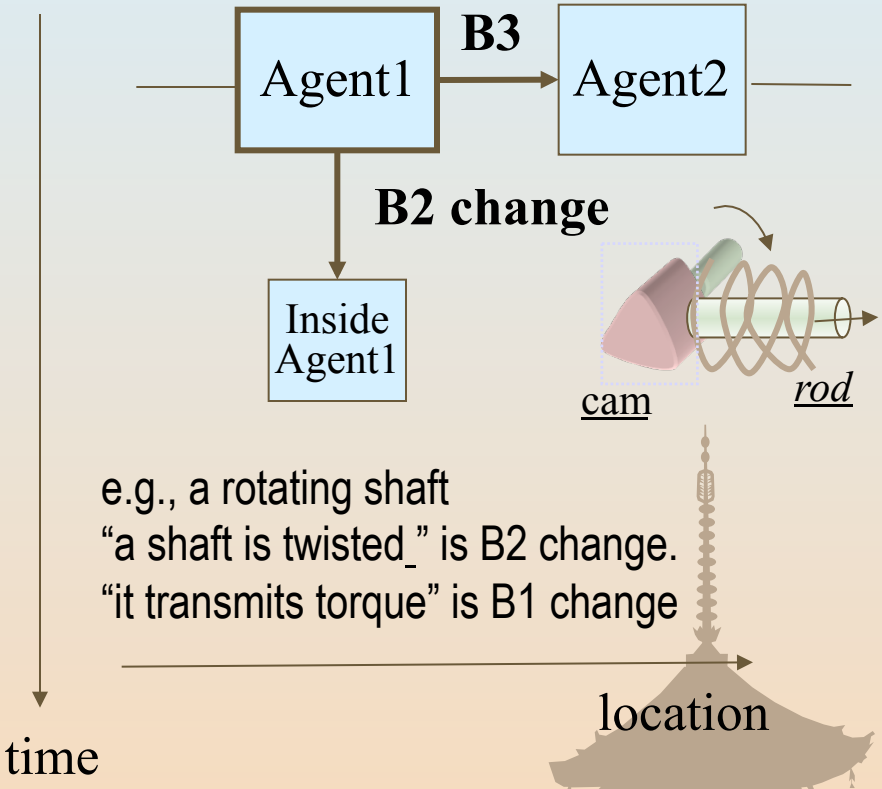
- Any phenomena can be interpreted as a behavior of a system composed of multiple components/parts/features
- A device is a role played by an object/system whose behavior is defined **as a pair of input and output**
- A pair of Input and output of a device is determined according to the context set by the modeler
- Any object can be interpreted as playing a device role by identifying behaviors of interest
- **Any device is a black box** at a level of the **granularity** and hence you are not allowed to look into inside to know how the behavior is realized at the level of granularity
- Any device except at the least granularity level is composed of finer-grained devices connected each other, that is, **a device is modeled in a nested structure** in device ontology
- Every pair of role and role player keeps links between domain entity and role
- This guarantees **consistent modeling** in a **domain-independent** framework

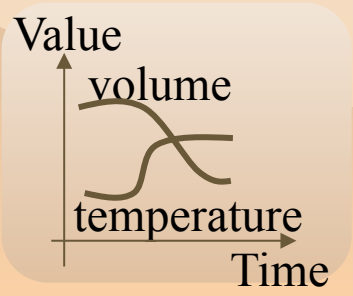
Applications



Four kinds of behavior



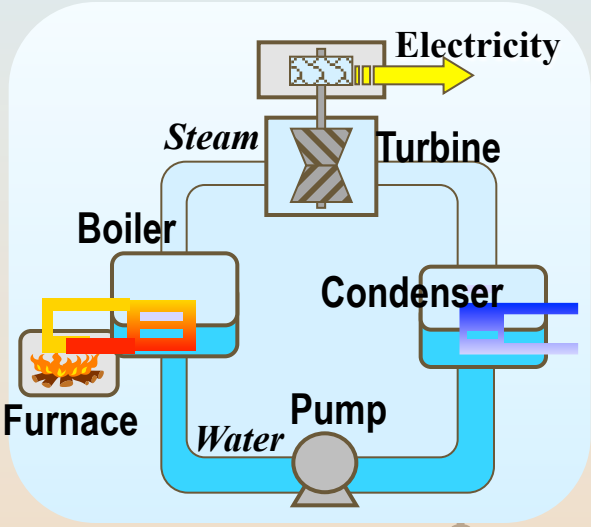




B0: the temperature increases
B1: A heater heats the water
B2: Moving air by rotation
B3: A cam pushes up a rod

Basic characteristics of function

- Context-dependence
  - Systemic context
    - **Internal**/parts' goal
    - **External** goal (teleological context)
- Implementation-independent
  - Black box
- Composability
  - Localization is a key
  - So, to cool is not a function of an electric fan



Distinction (1): Detail

Our definition of function

- *A function performed by a device is a role played by the behavior of the device to achieve a specific goal under a systemic context (including context of use), based on a certain capacity*
- *inherent in the device.*

- Role is context-dependent entity defined with external entities.
  - Our theory of role [Mozoguchi 00, 06, 15].
  - **Founded** (i.e., extrinsic property essentially dependent on an external entity)
    - Function of a device refers to the systemic context.
  - **Anti-rigid** (i.e., contingent (non-essential) property)
    - The role isn't essential to the behavior, it loses the role in a different context
  - **Dynamic** (i.e., temporary and multiple)
    - A behavior can play multiple roles and the role can be played by different behaviors.

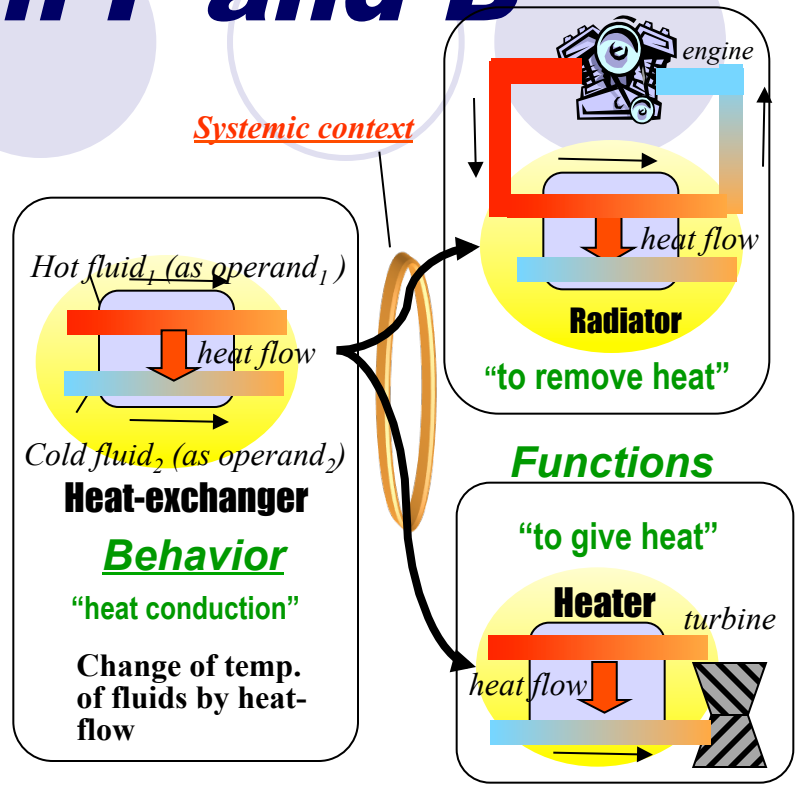
Functional ontology

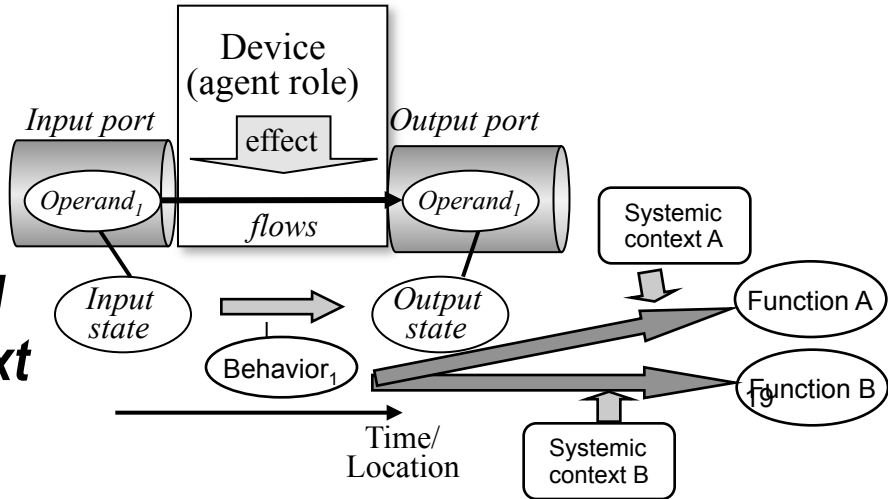
- Functional concepts
- Meta-function, Function type
- Function achievement way
- Functional decomposition

Distinction between F and B

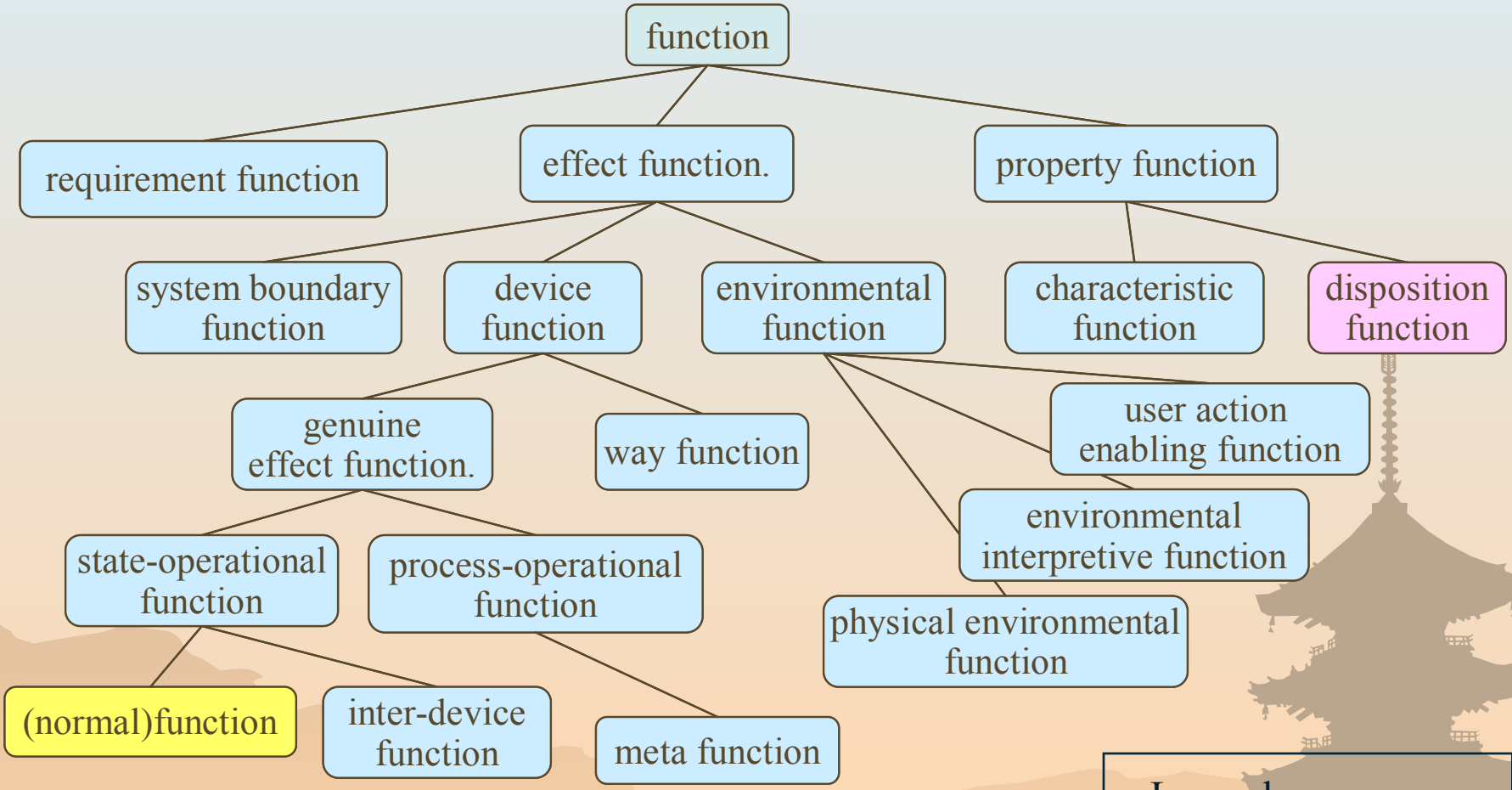
- *Behavior of a device*
  - Temporal changes of physical quantities of *operands* as input and output of the device
  - Independent of the context
  - **B1-behavior** for **composability. Black-box**.

- *Function of a device*
  - Dependent on "systemic context"
    - A behavior can perform different functions under different contexts.
    - Systemic context (a system configuration)
      - External context (direct user's purpose)
  - A function can be performed by different behaviors and agents.
  - ***Function is defined as a "role" played by a behavior under a systemic context***



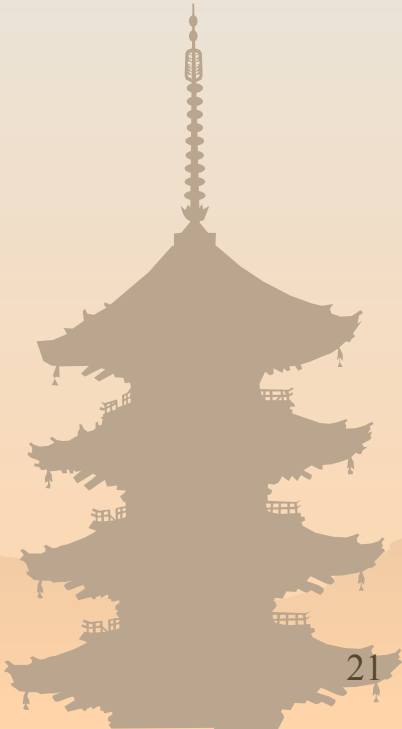


Reference ontology of function



Function decomposition

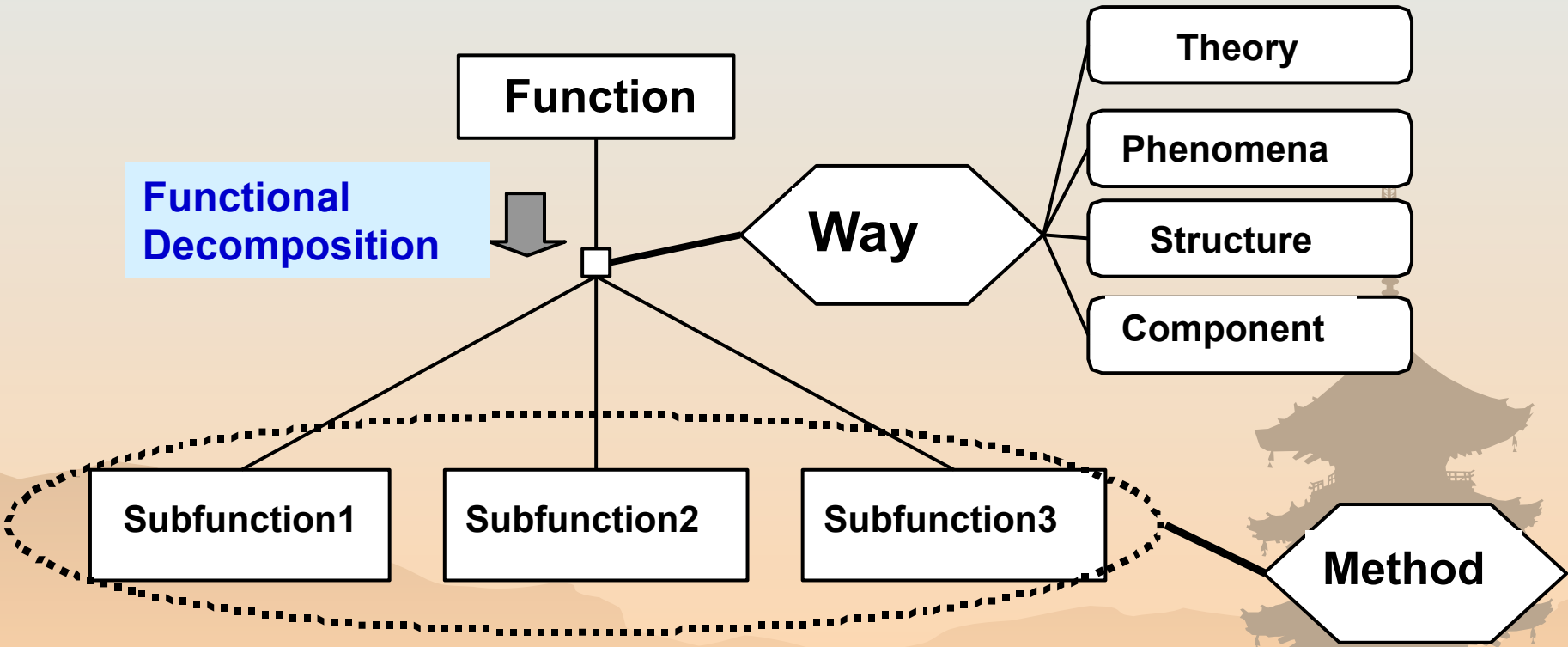
- Another way of capturing **essentials**



Functional decomposition

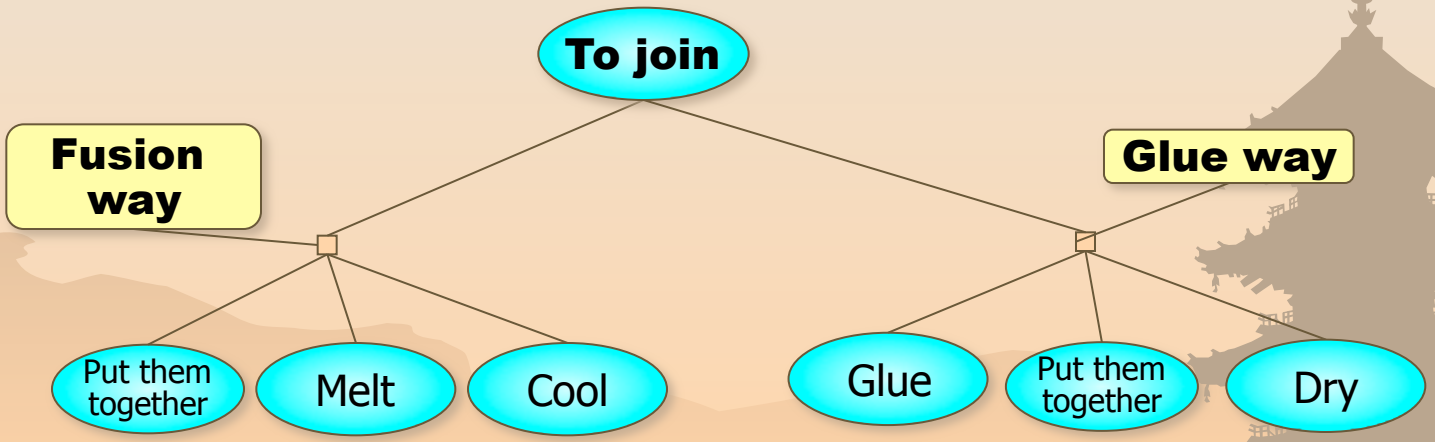
Two kinds of decomposition:

1. In terms of Granularity
2. **What to achieve** and **how to achieve**



Examples of decomposition

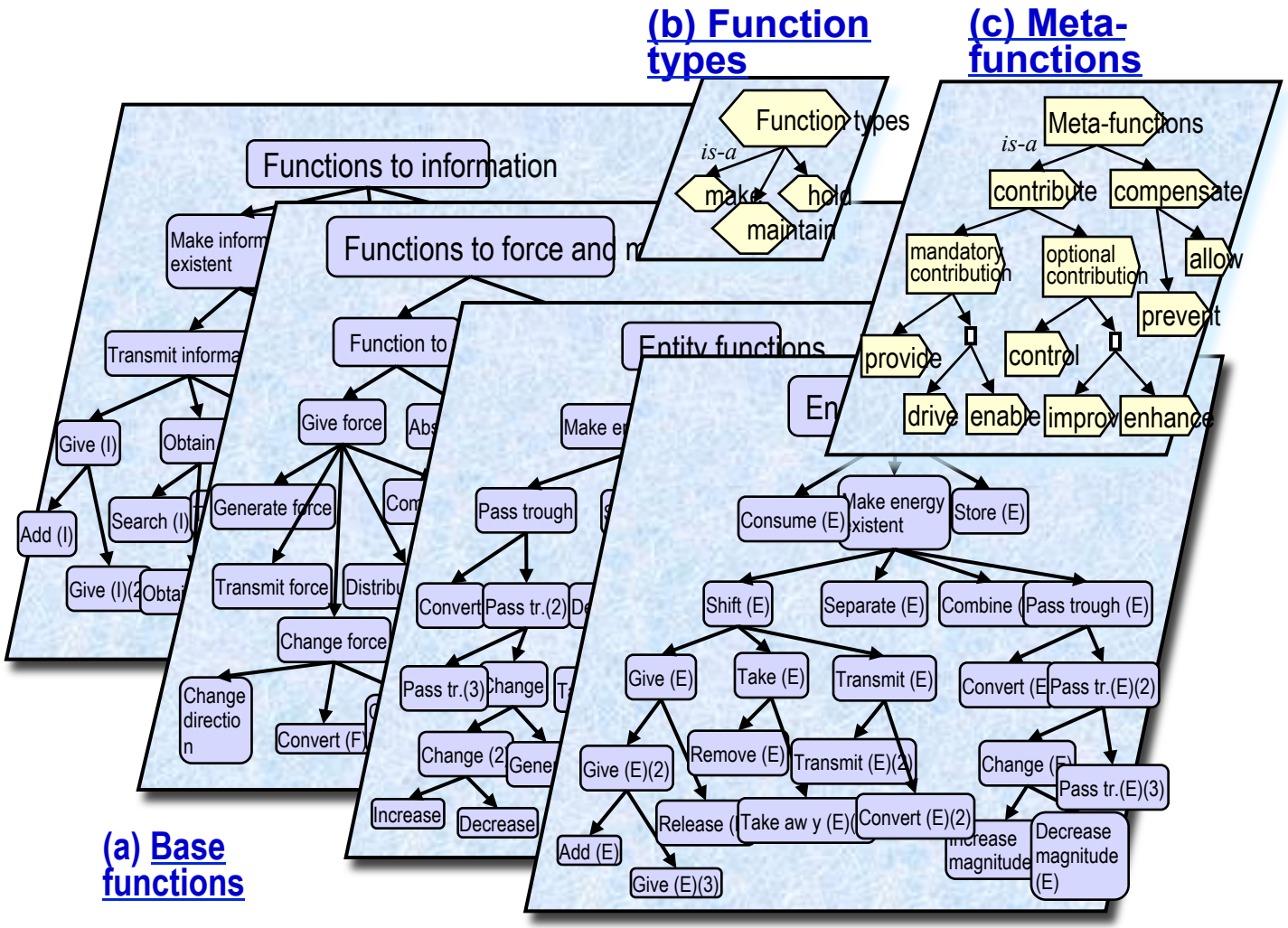
- To **weld**
  - What to achieve = To ***join*** sheets of metals
  - How to achieve = By fusion way
- To **glue**
  - What to achieve = To ***join*** two things
  - How to achieve = By glue way



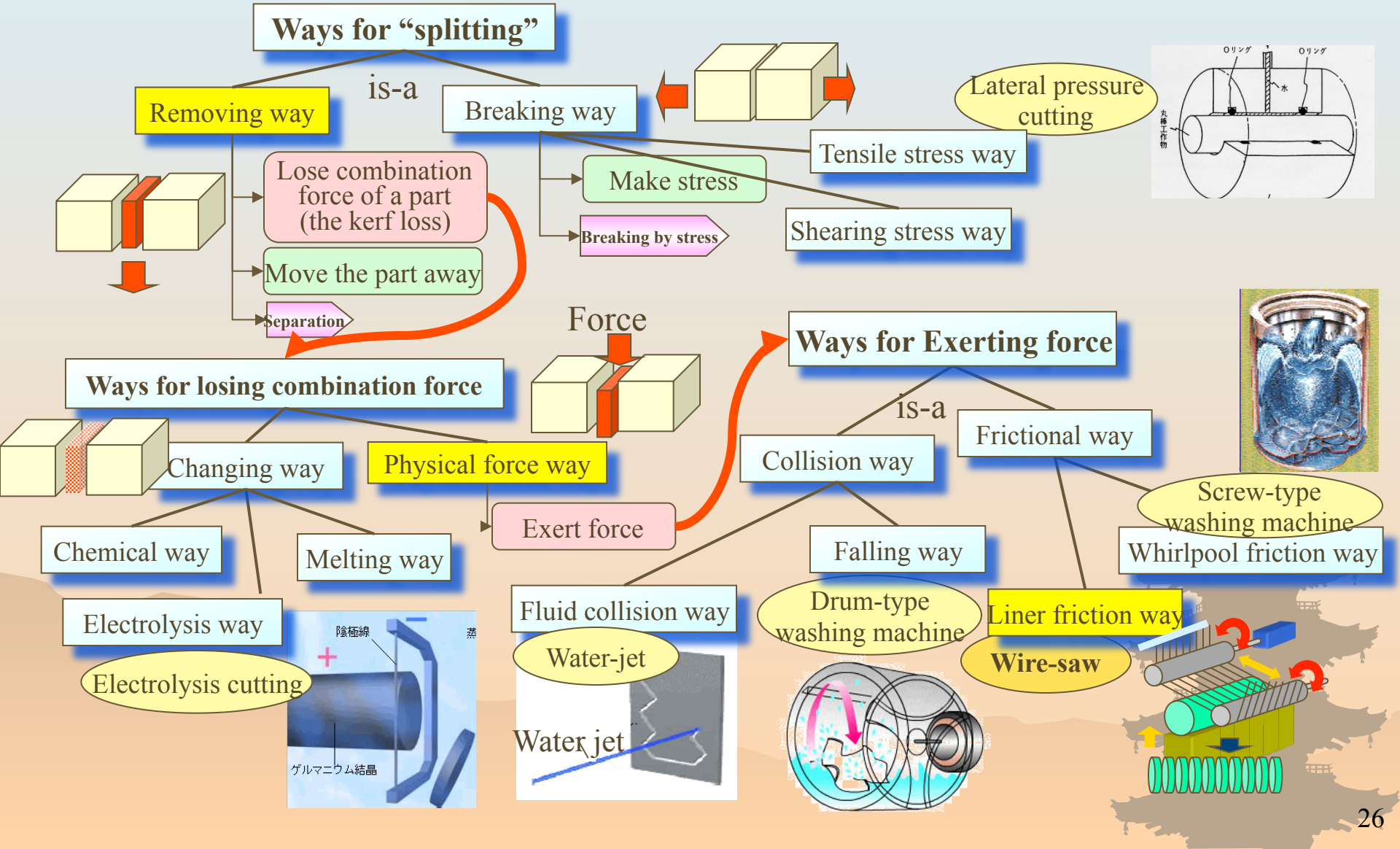
toloGear SE
(New Document)New Project(*)
e Edit View Operate Reference Tool Help
Whole View
Tree View
FDT1
coffee mak...
Function Property
Subject
Basket
Description mix water and coffee powder
Function
mix
Search
Object
water
and
coffee powder
Kind
Basic Function
Attribute
Function
Subject
Object
0
<Extend>
Filter
remove coffee powder
Jug
store coffee
Filtering way
Filtration
Melior
move water and coffee powder
consume movement of coffee powder
High pressure way (espresso)
Pressure
Dripping way
Vaporization
Air pressure
Air pressure
pressure
move up water
control quantity of water
move down water
water
of water

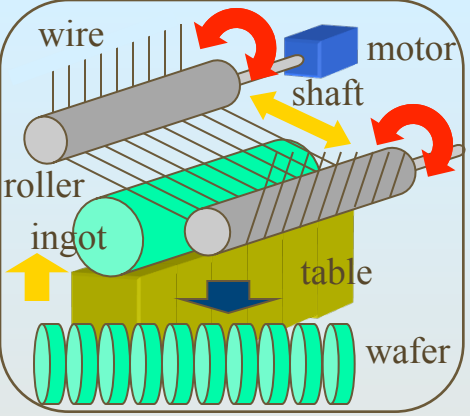
Functional concept ontology

- # of functional concepts = about **100**

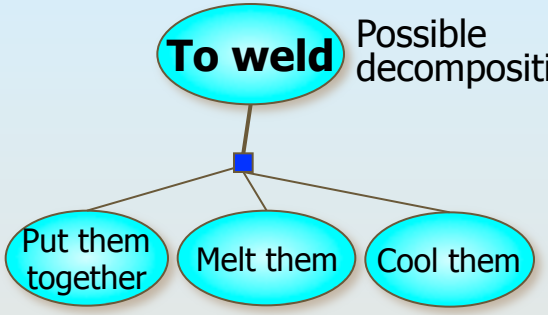


Organization of generic ways of function achievement for “splitting”

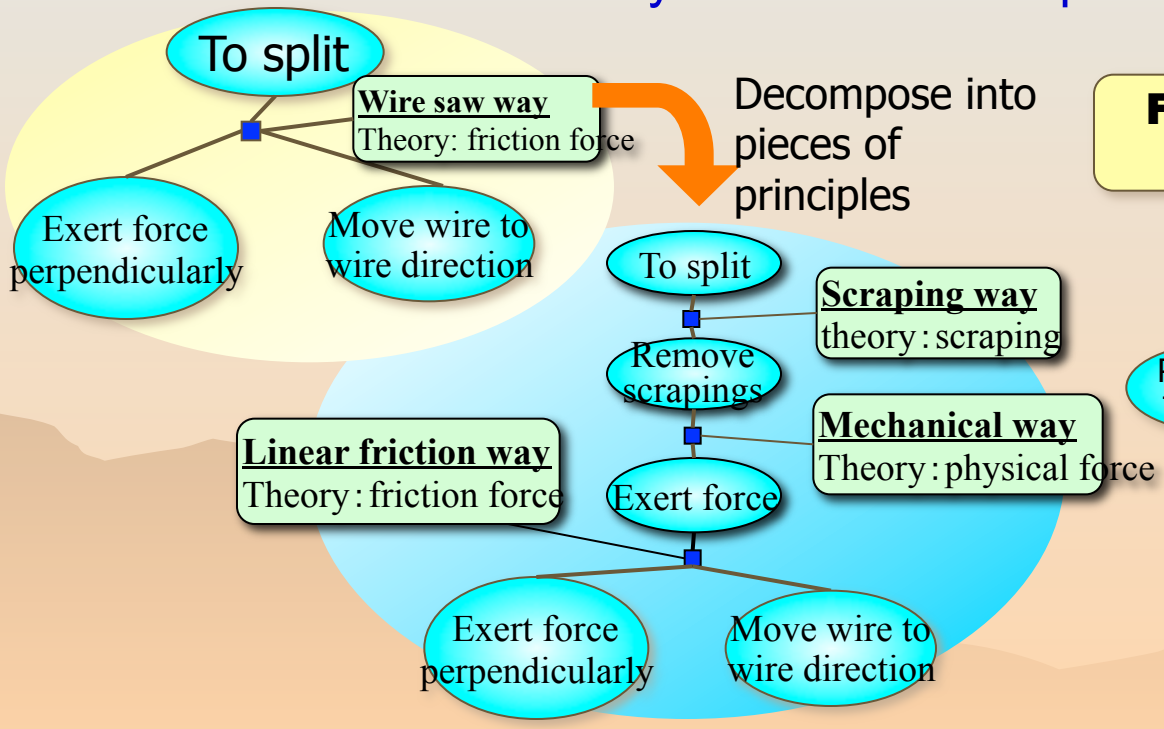


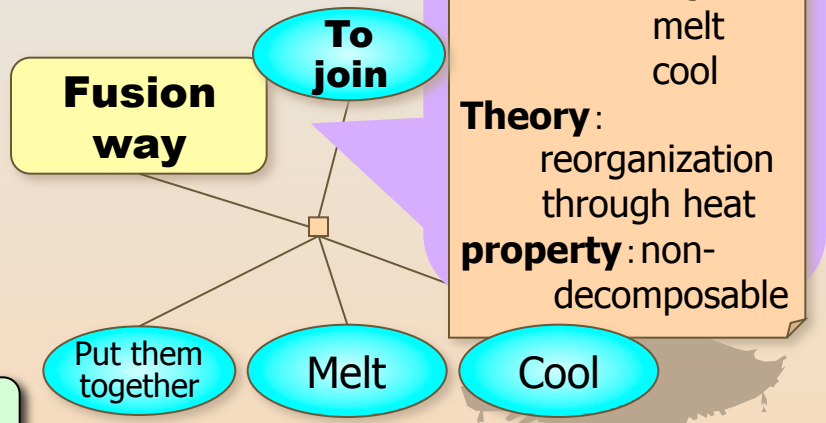


Secrete of way description



- **Detach "how to achieve" from the function**
  - To weld = to join by fusion way
- A way consists of one principle





Way knowledge

Sup.Func: Connect
Sub.Func: together
melt
cool
Theory: reorganization through heat
property: non-decomposable

Concluding remarks

- Device ontology is a conceptual tool
- It enables you to model dynamic systems domain-independently
- It has contributed to defining function

- We have proposed a theory of Causation
  - Function grounds causation
    - function exists ⇔ causation exists