MLMI10
Designing Intelligent Interactive Systems

Lecture 4

Per Ola Kristensson

Lent 2022

Key Concepts

- Conceptual design
  - Function modelling
  - Morphological charts
  - Concept evaluation
  - Controllable and uncontrollable parameters
  - Envelope Analysis

Conceptual Design

Conceptual design

- The objective of the conceptual design stage is to arrive at a single conceptual design that links functions to function carriers
- It typically involves:
  - Determination of the overall function and its system boundary
  - Function decomposition of the overall function
  - Identification of candidate function carriers
  - The assignment of function carriers to functions to form concepts
  - The informed selection of a preferred conceptual design

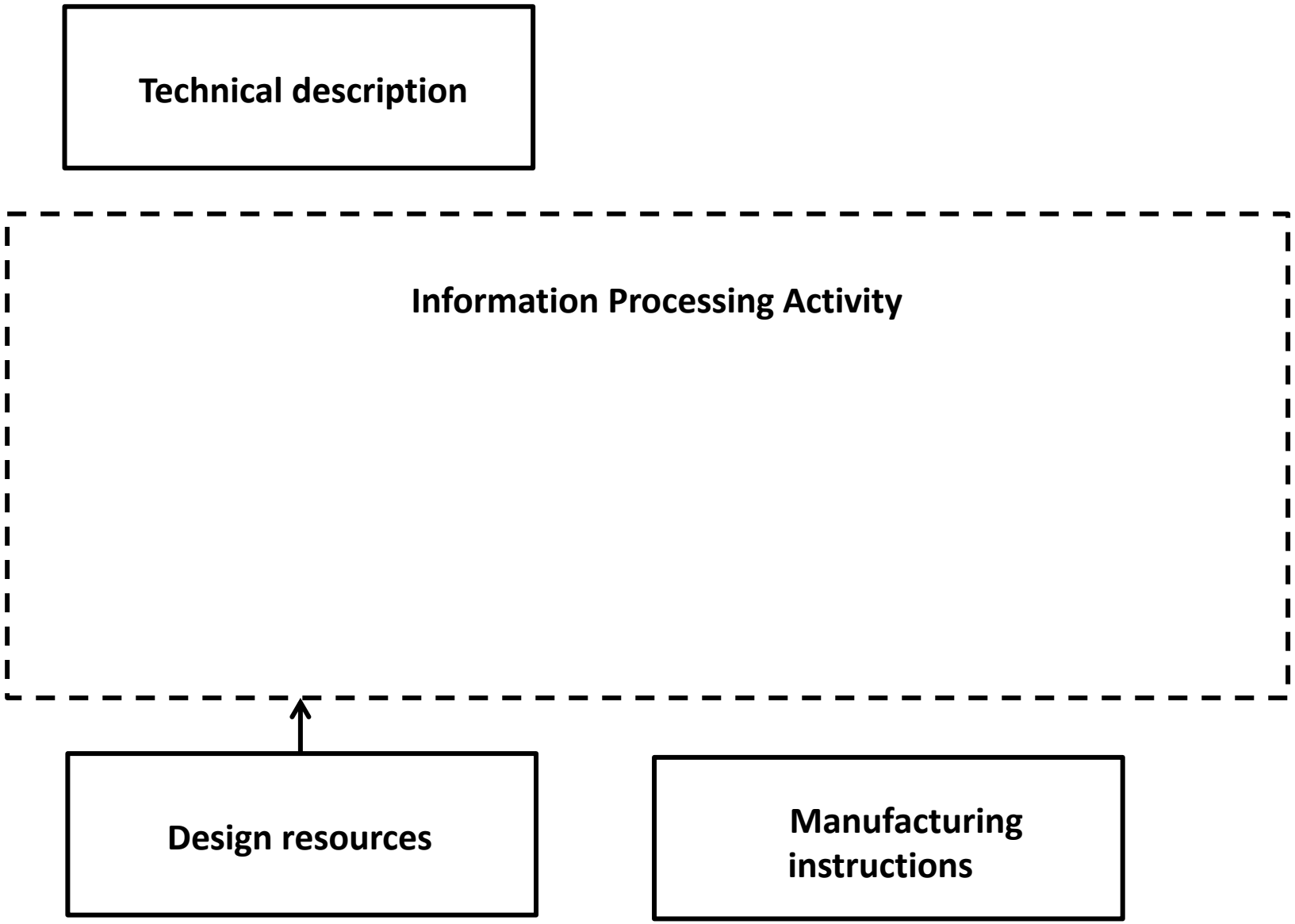
Conceptual design

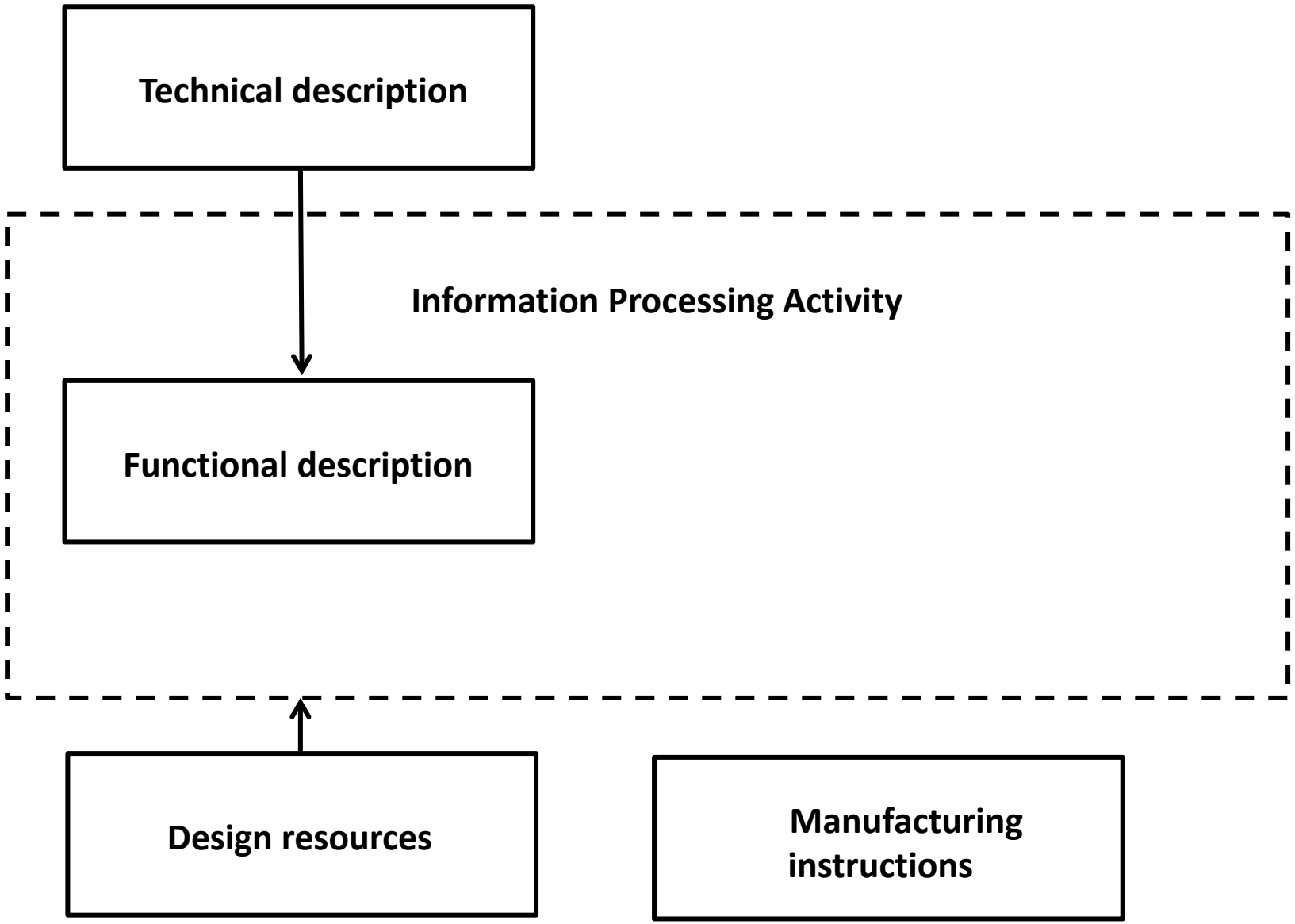
- Conceptual design is an integral part of design engineering
- The idea is to use an initial set of requirements and design know-how to elaborate the solution-neutral problem statement into a solution-neutral **function description** of the system
- We can then use this functional structure to arrive suitable **conceptual design** (sometimes just called **concept**), which proposes 1) a functional description; 2) links functions in the functional description to the most suitable function carriers for the overall design
- Conceptual design as described is typically not part of software engineering
- However, conceptual design is very suitable for non-trivial user interface technology which either:
  - Require non-trivial software/hardware interfacing; such as use of special-purpose sensors or actuators
  - Require elaboration of mechanistic emergent system behaviour due to high complexity in the interaction of underpinning design parameters

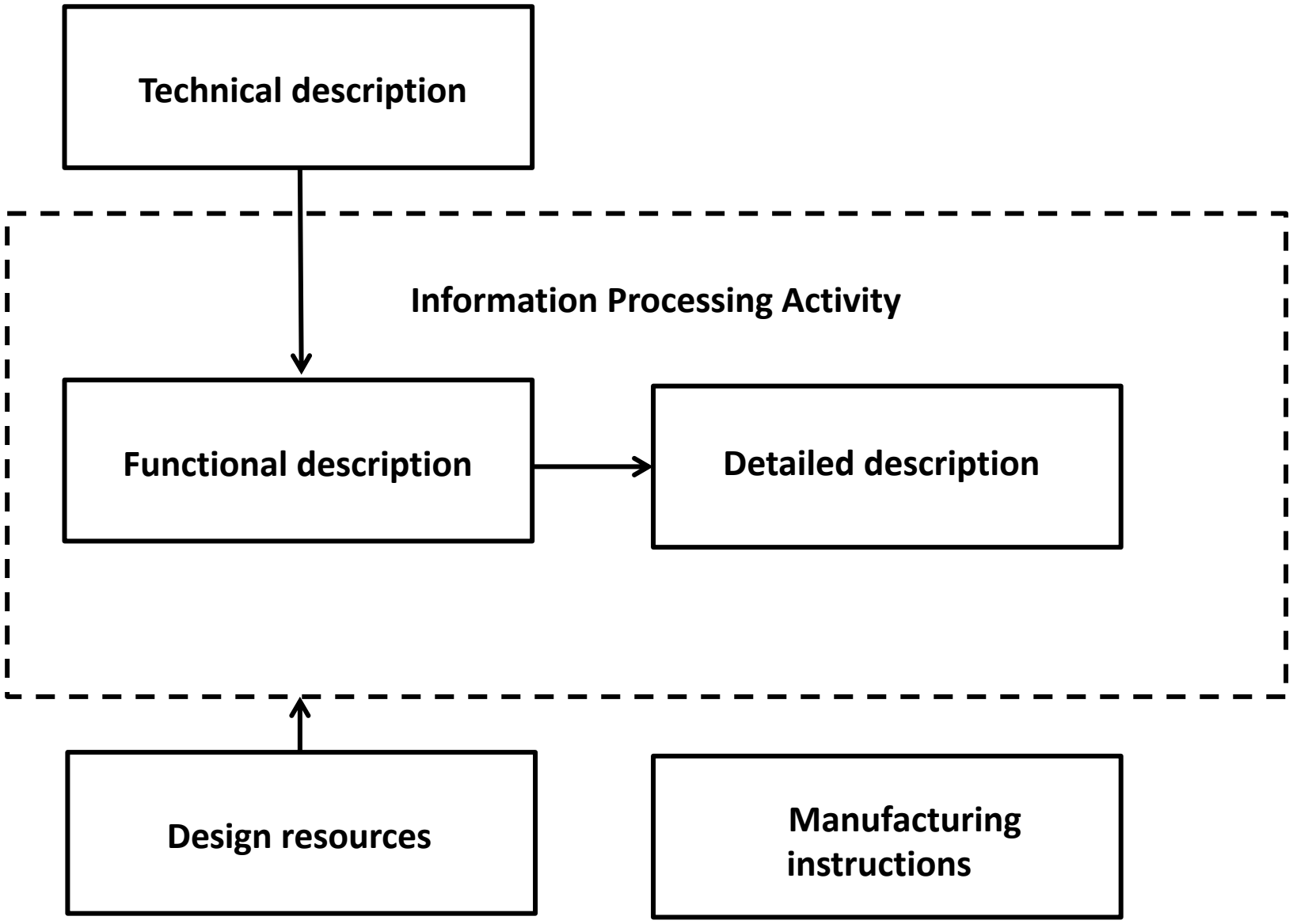
Technical description

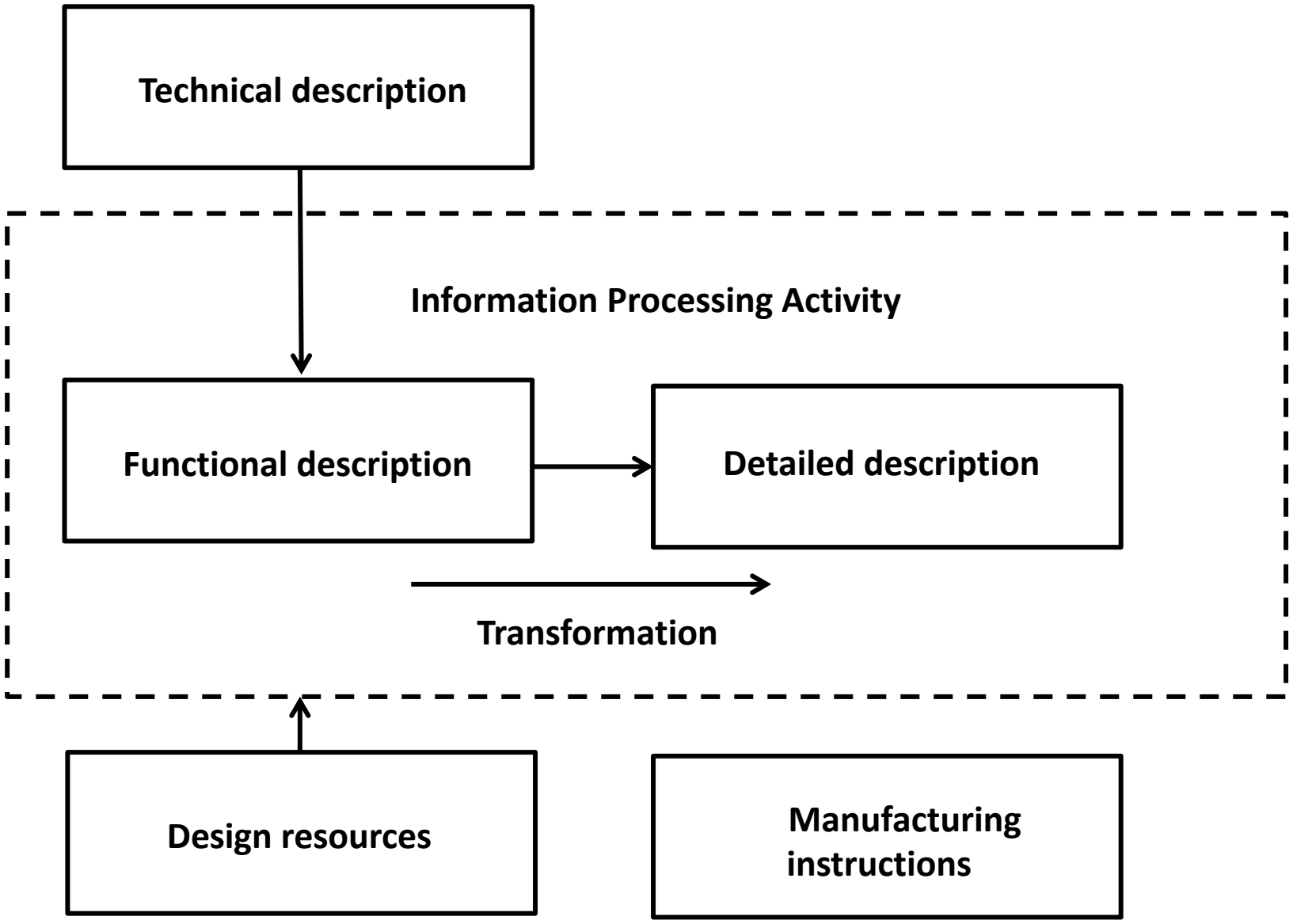
Design resources

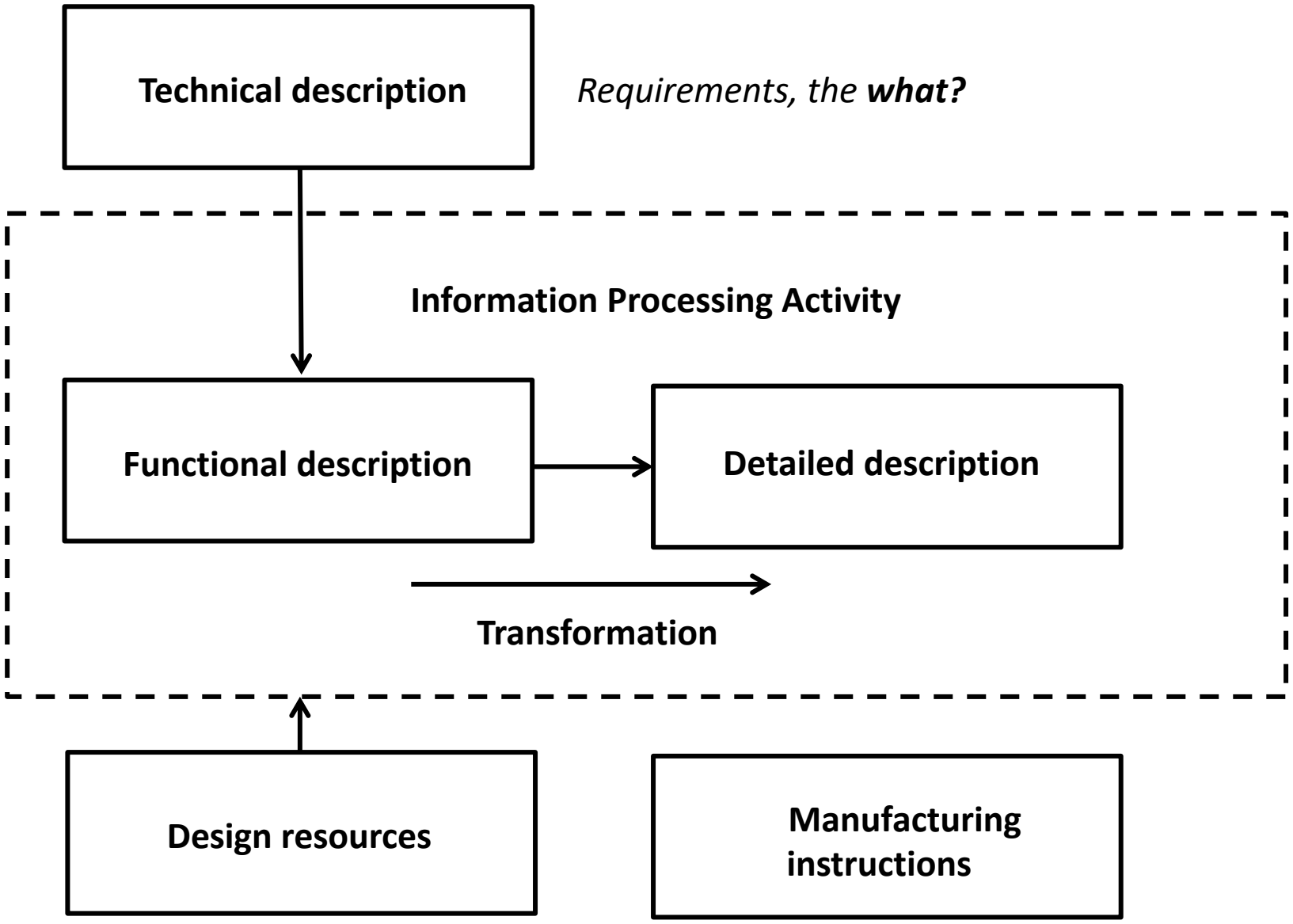
Manufacturing instructions

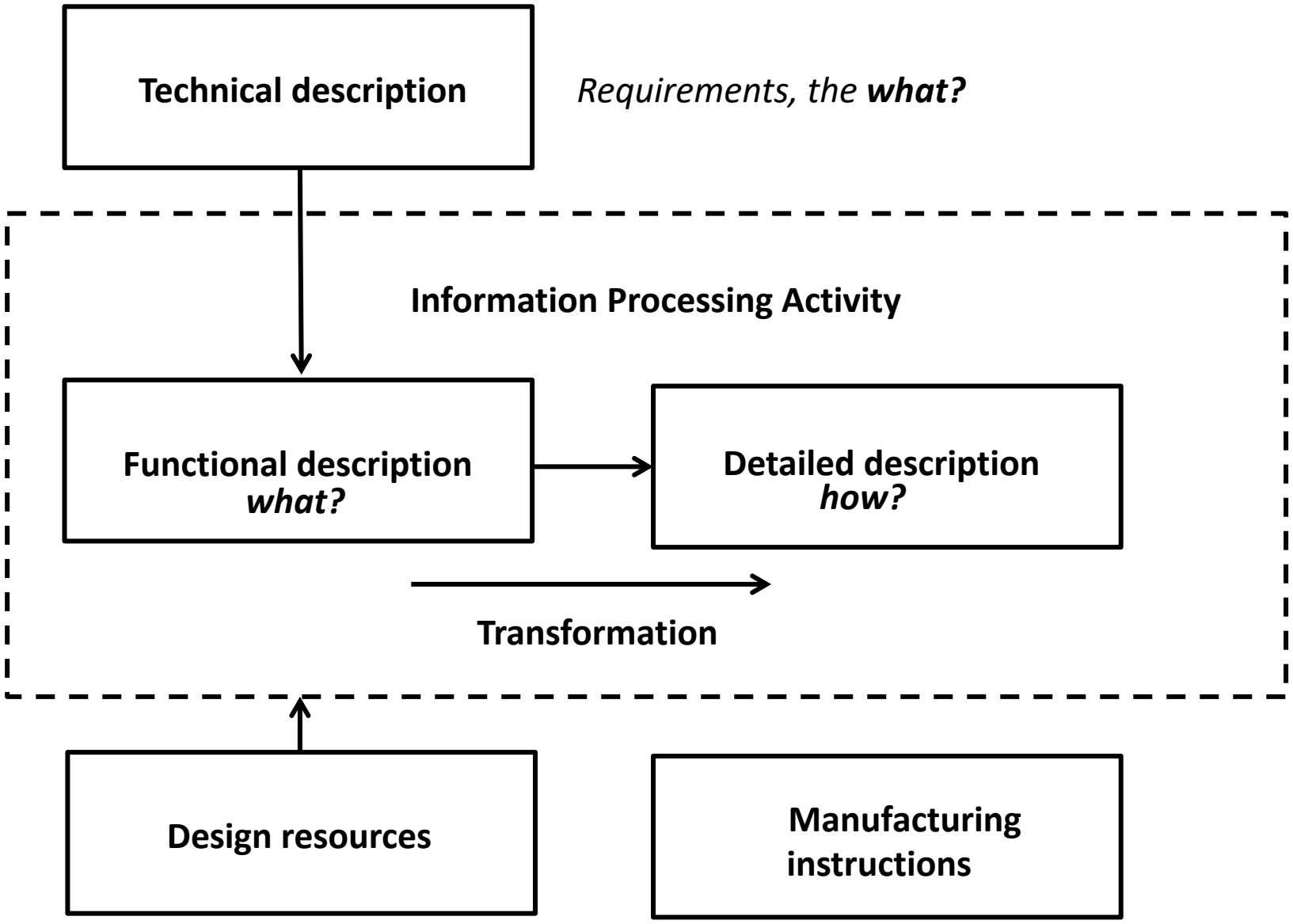

Technical description
Information Processing Activity
Design resources
Manufacturing instructions

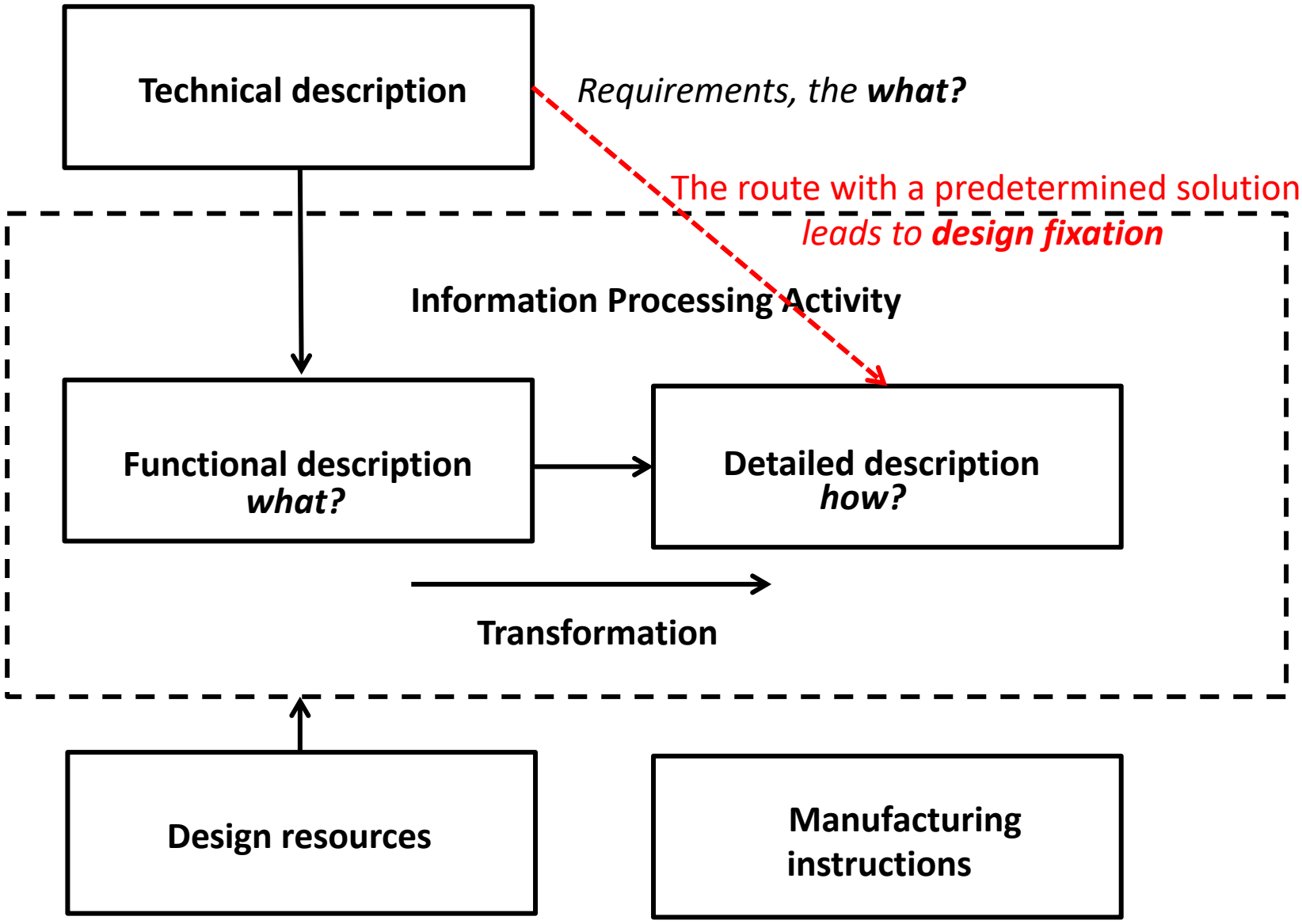

Technical description
Information Processing Activity
Functional description
Design resources
Manufacturing instructions


Technical description
Information Processing Activity
Functional description
Detailed description
Design resources
Manufacturing instructions

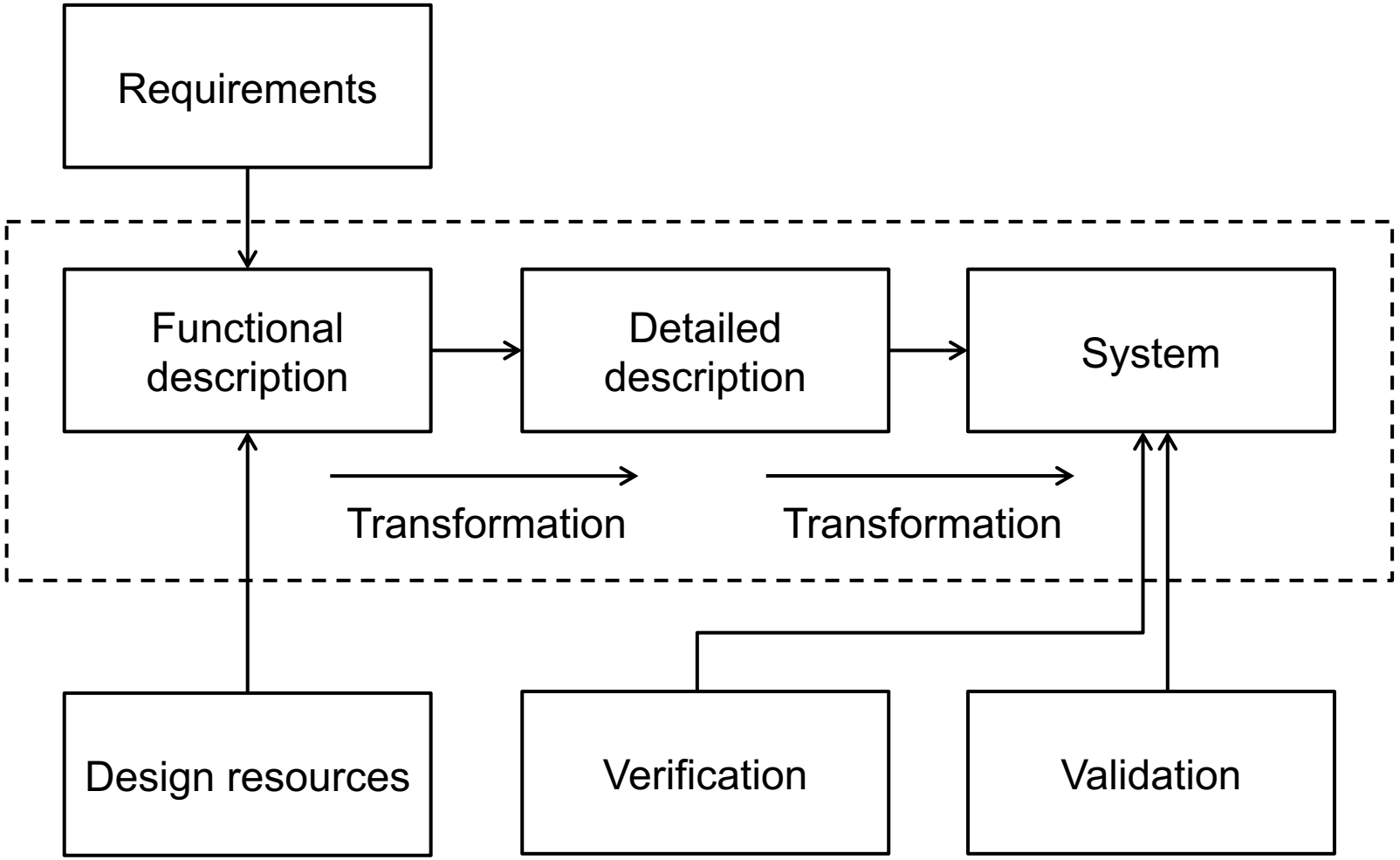

Technical description
Information Processing Activity
Functional description
Detailed description
Transformation
Design resources
Manufacturing instructions


Technical description
Requirements, the what?
Information Processing Activity
Functional description
Detailed description
Transformation
Design resources
Manufacturing instructions


Technical description
Requirements, the what?
Information Processing Activity
Functional description
what?
Detailed description
how?
Transformation
Design resources
Manufacturing
instructions

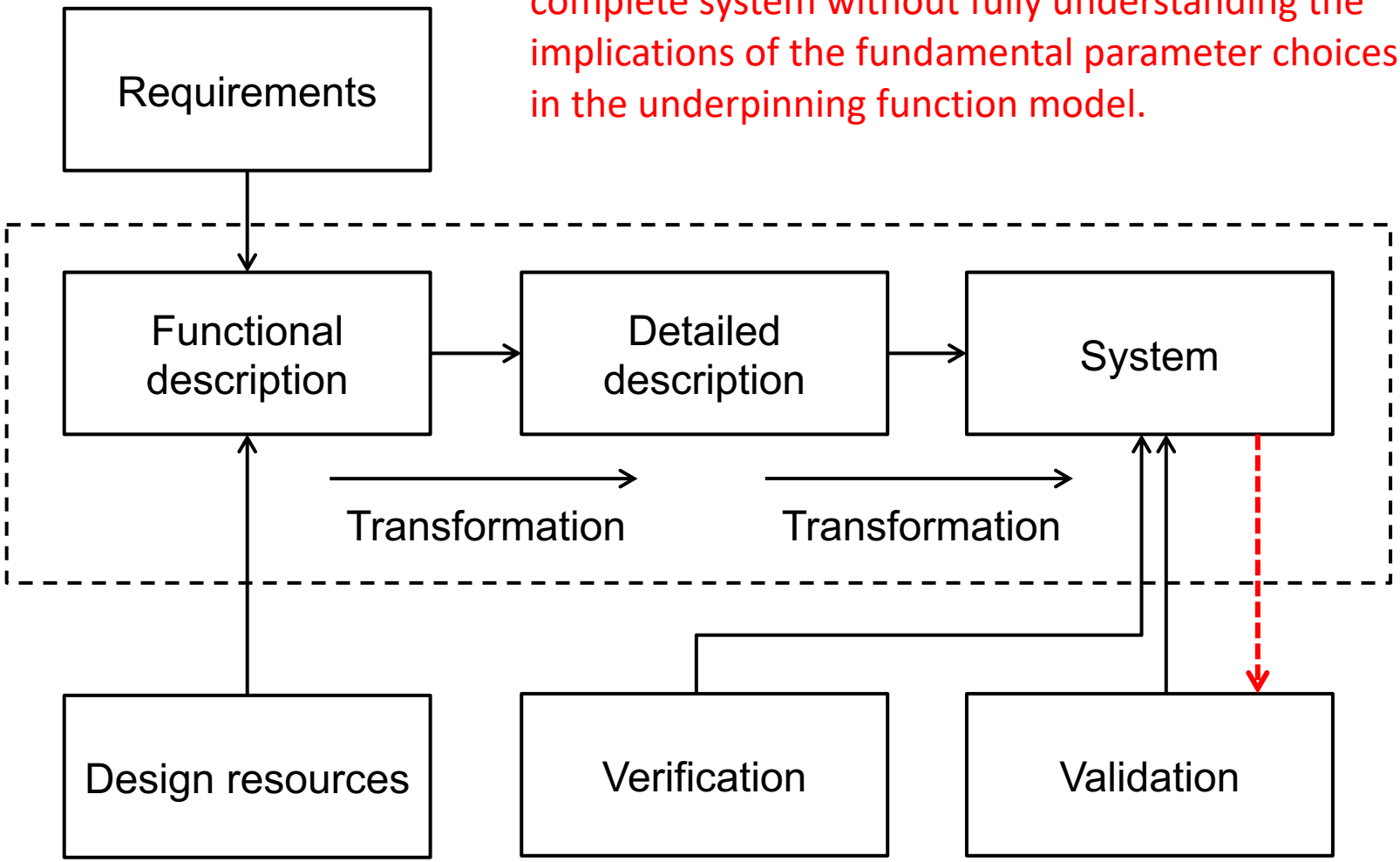

Technical description
Requirements, the what?
The route with a predetermined solution
leads to design fixation
Information Processing Activity
Functional description
what?
Detailed description
how?
Transformation
Design resources
Manufacturing
instructions

A modified model for intelligent interactive systems



Short-circuit evaluation

Evaluating a particular design choice using a complete system without fully understanding the implications of the fundamental parameter choices in the underpinning function model.



Function models

Function modelling

- Functional modelling is a means of determining and representing the functional description of a system

- Determination of the overall function
- Establishment of function structures
- Generation and selection of suitable combinations

Function modelling

- Functional modelling is a means of determining and representing the functional description of a system

- Determination of the overall function—**task clarification**
- Establishment of function structures
- Generation and selection of suitable combinations

Function modelling

- Functional modelling is a means of determining and representing the functional description of a system

- Determination of the overall function—**task clarification**
- Establishment of function structures—**elaborating on the functional design**
- Generation and selection of suitable combinations

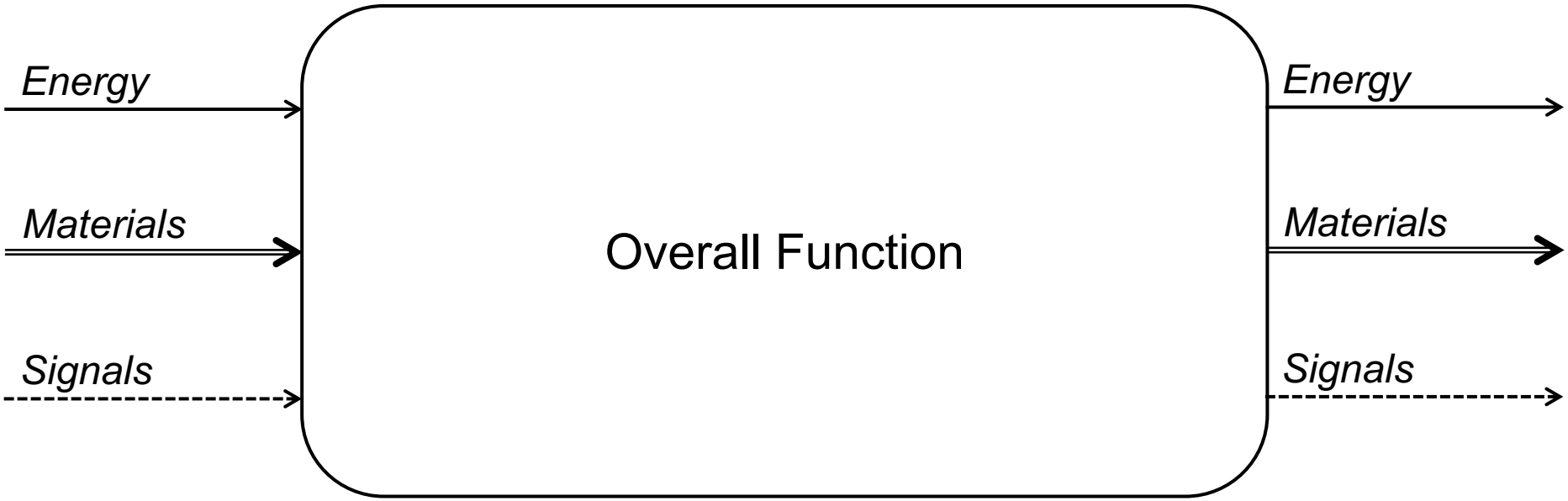
Function modelling

- Functional modelling is a means of determining and representing the functional description of a system

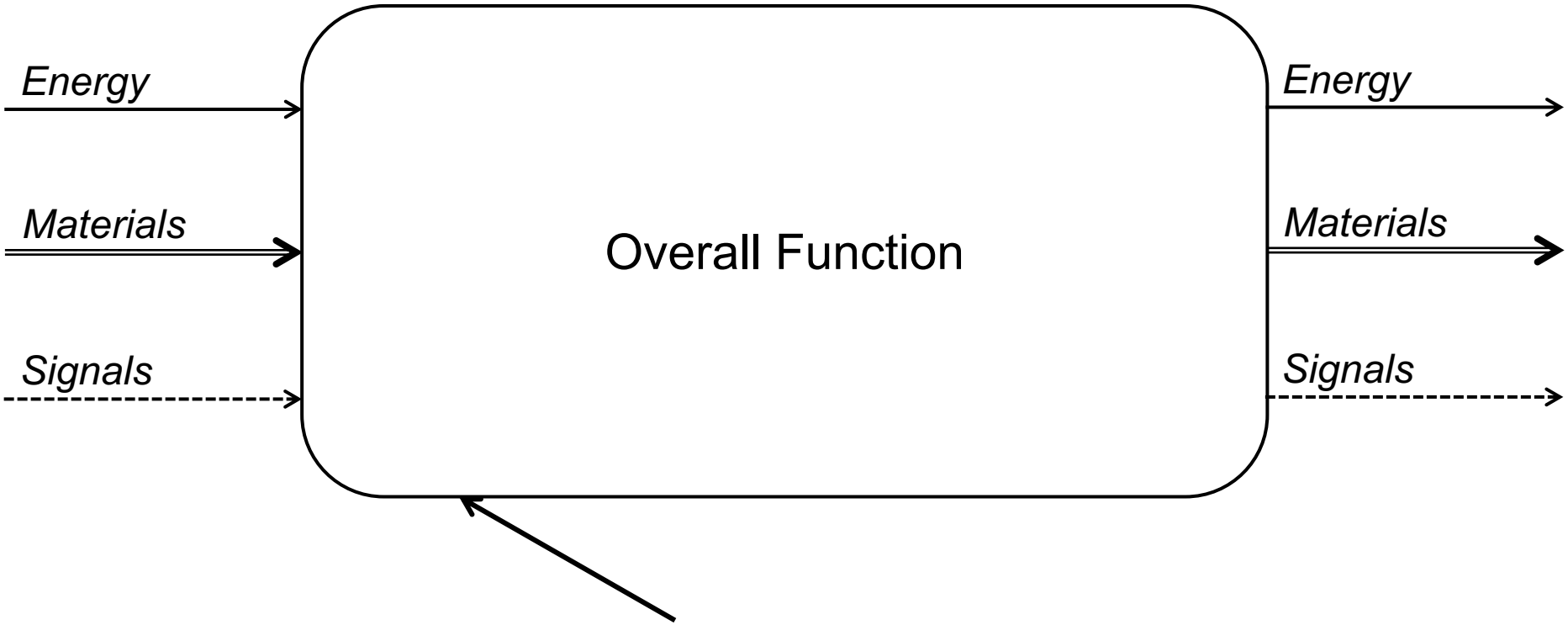
- Determination of the overall function—**task clarification**
- Establishment of function structures—**elaborating on the functional design**
- Generation and selection of suitable combinations—**identifying tradeoffs and exploring the design space**

Function Structures

Overall function

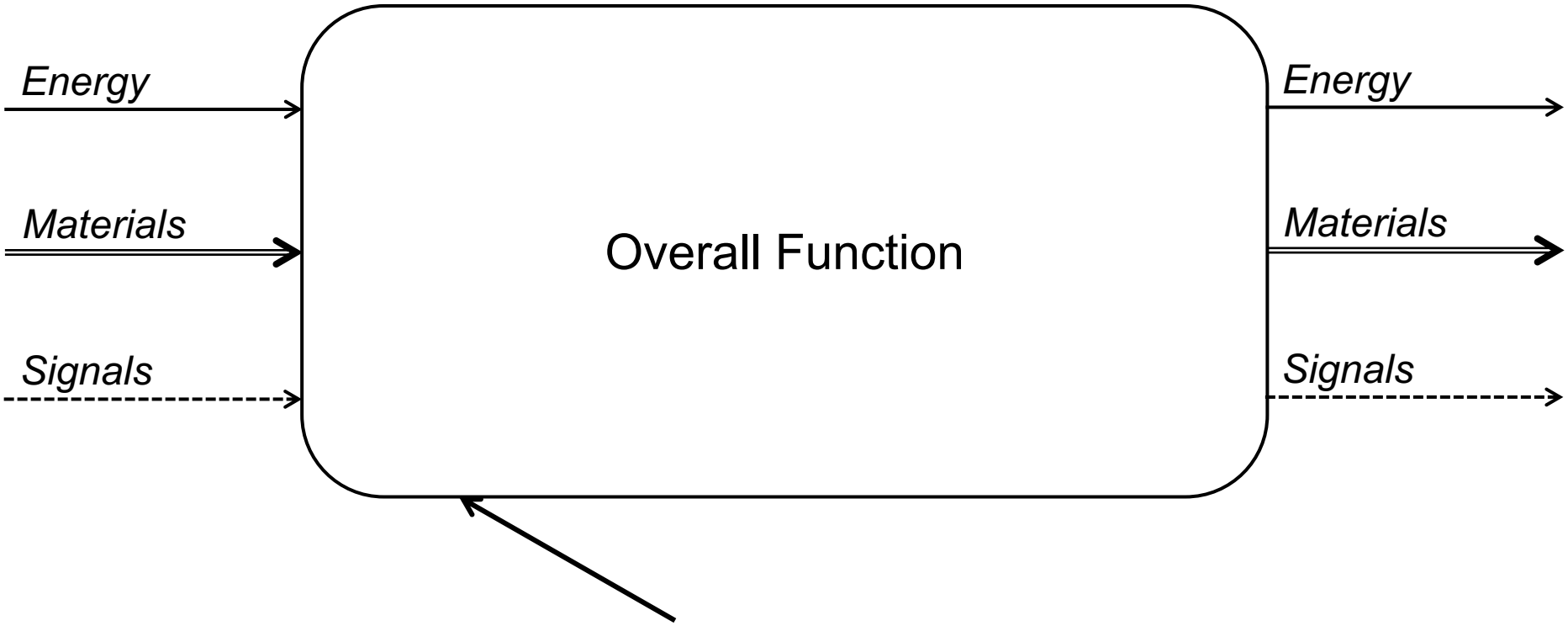


System boundary

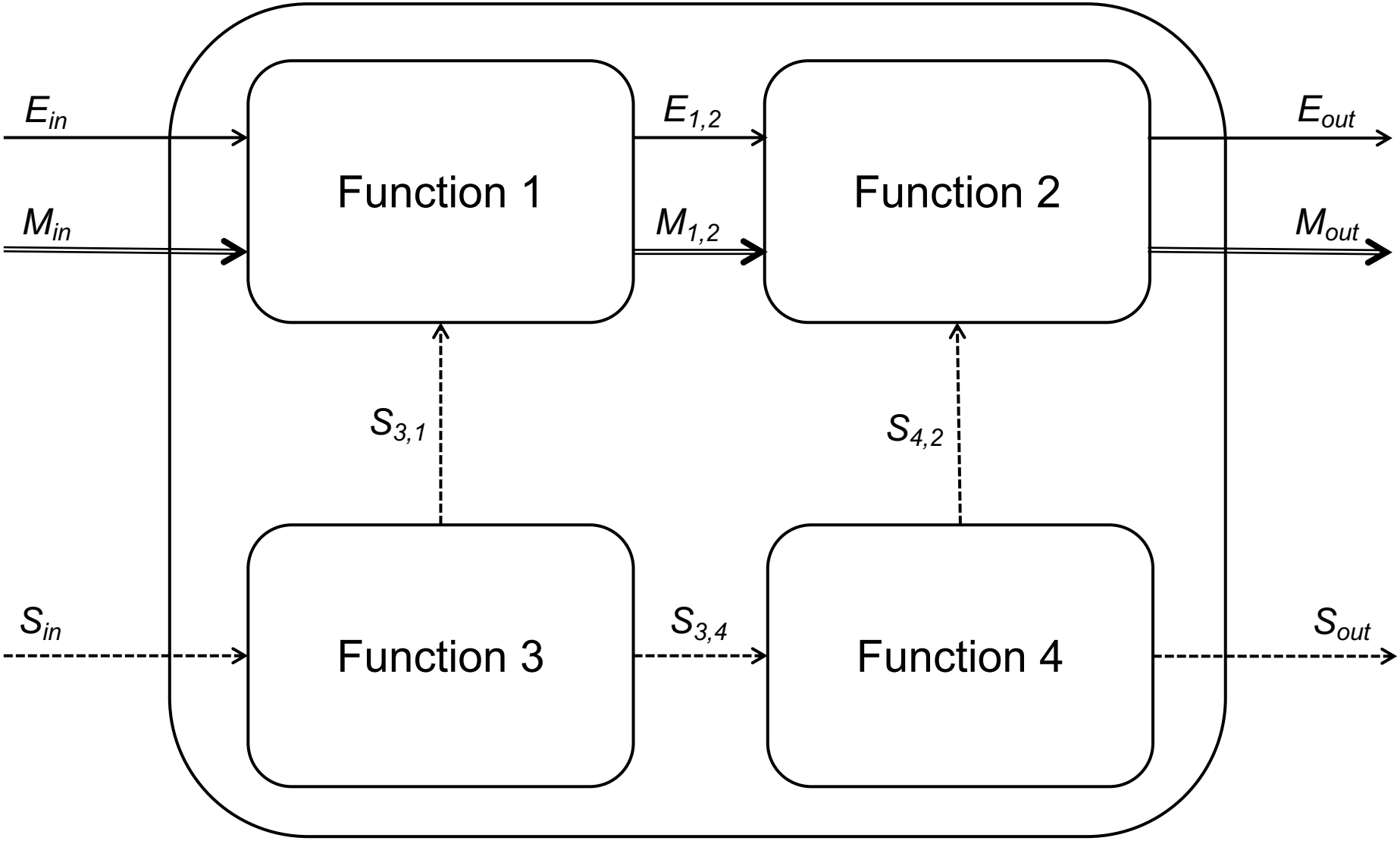


The system boundary describes the **technical boundary** of the system: anything outside the boundary is *not* going to be included in the functional description

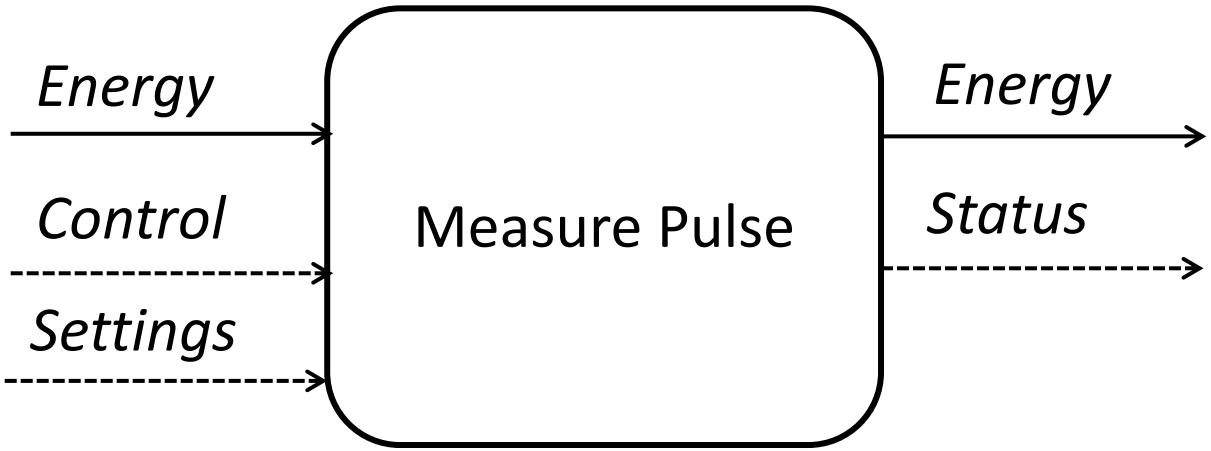
System boundary



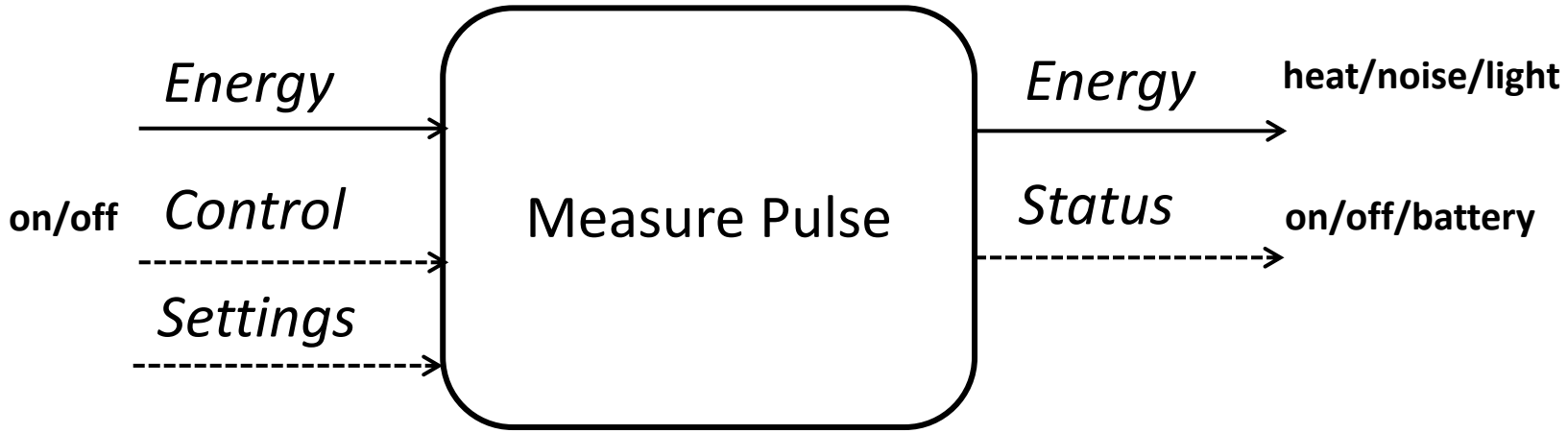
This boundary is immensely useful as it defines a **limit of concern** for the system. This can be used to, for example, apply a bound on risk analysis: risks outside the system boundary are not considered. The setting of this boundary thus also a tunable design parameter.


E_{in}
M_{in}
Function 1
$E_{1,2}$
$M_{1,2}$
Function 2
E_{out}
M_{out}
$S_{3,1}$
$S_{4,2}$
S_{in}
Function 3
$S_{3,4}$
Function 4
S_{out}

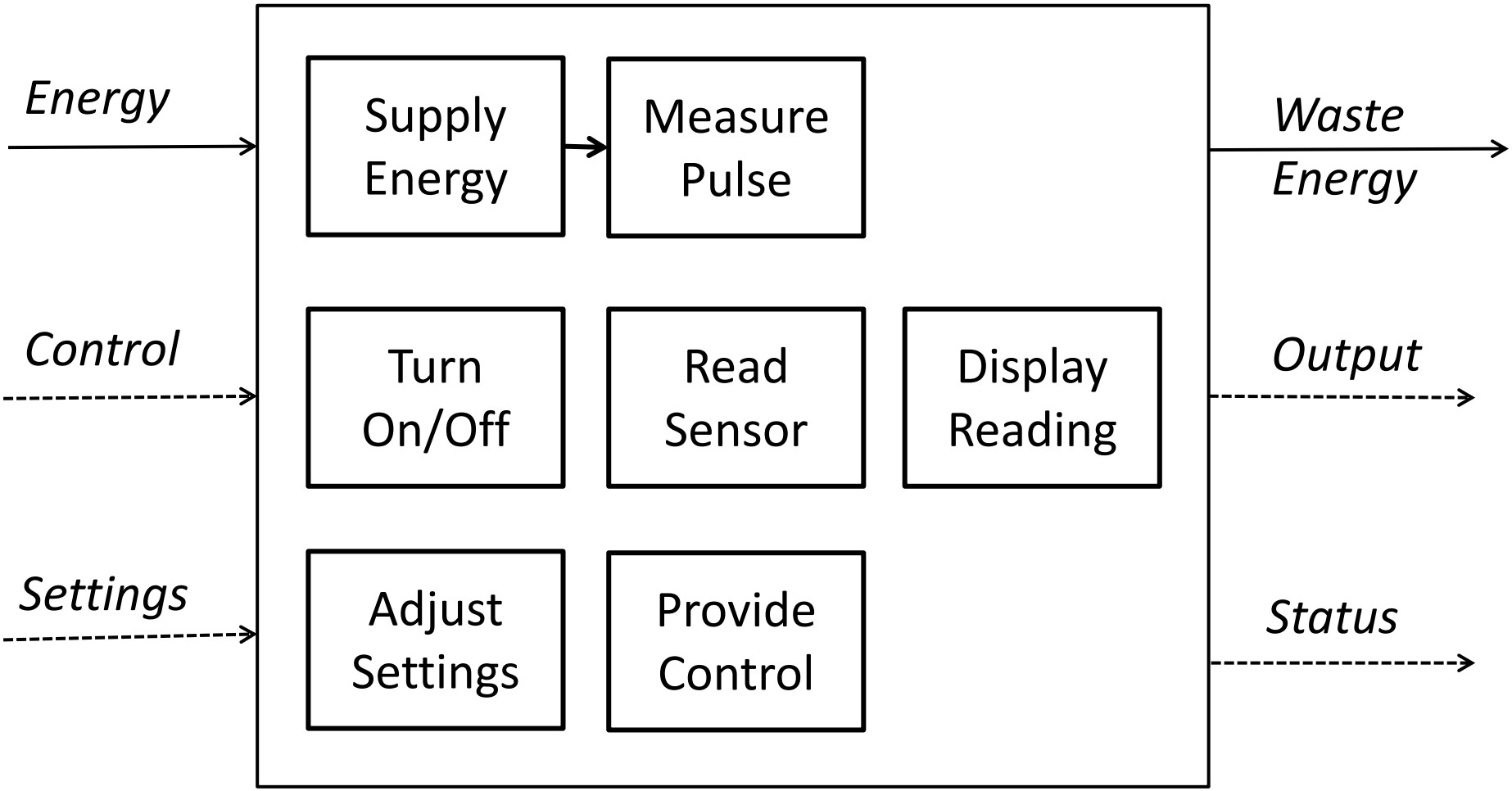
Example: pulse meter



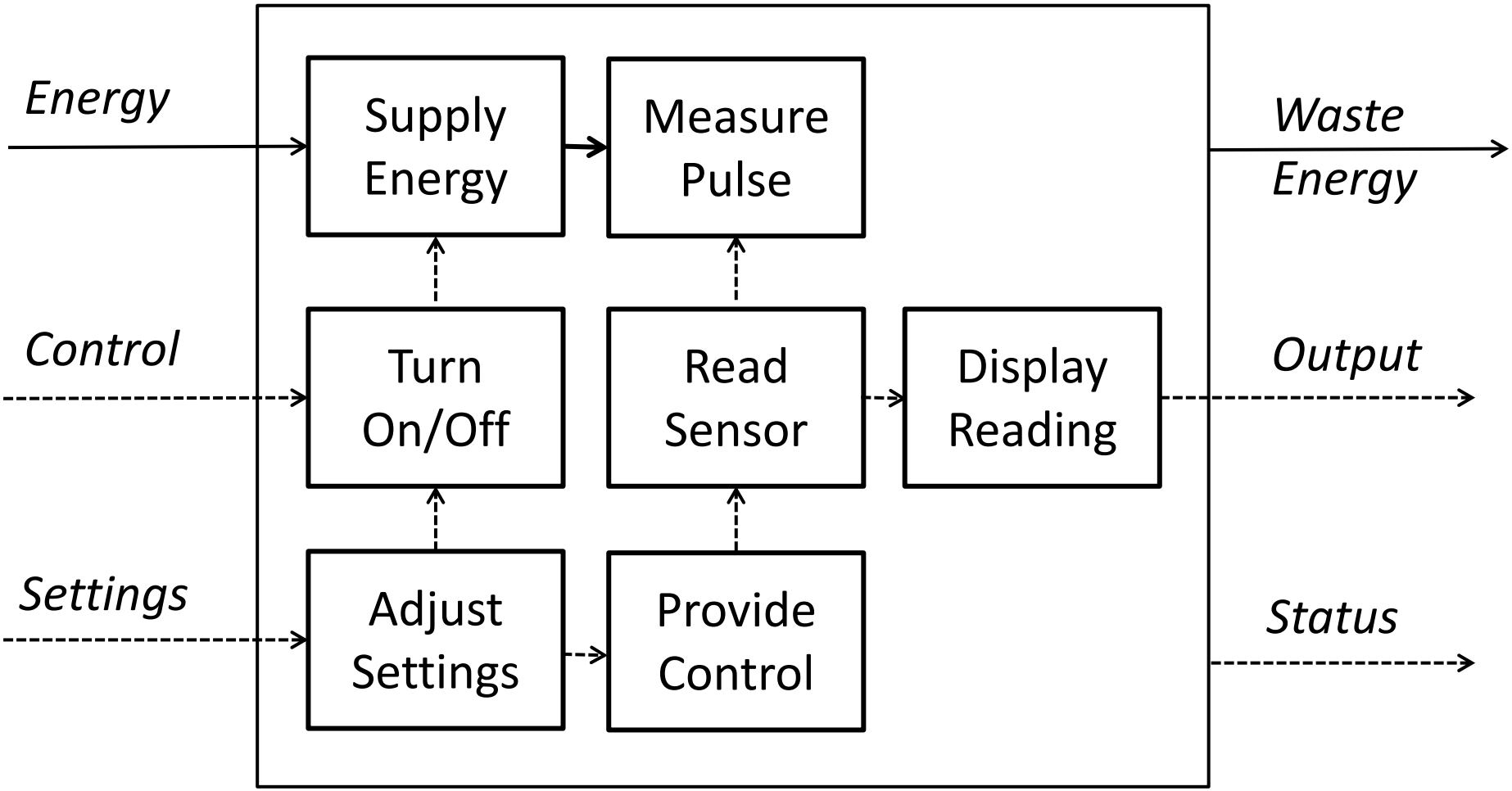
Example: pulse meter



Example: pulse meter

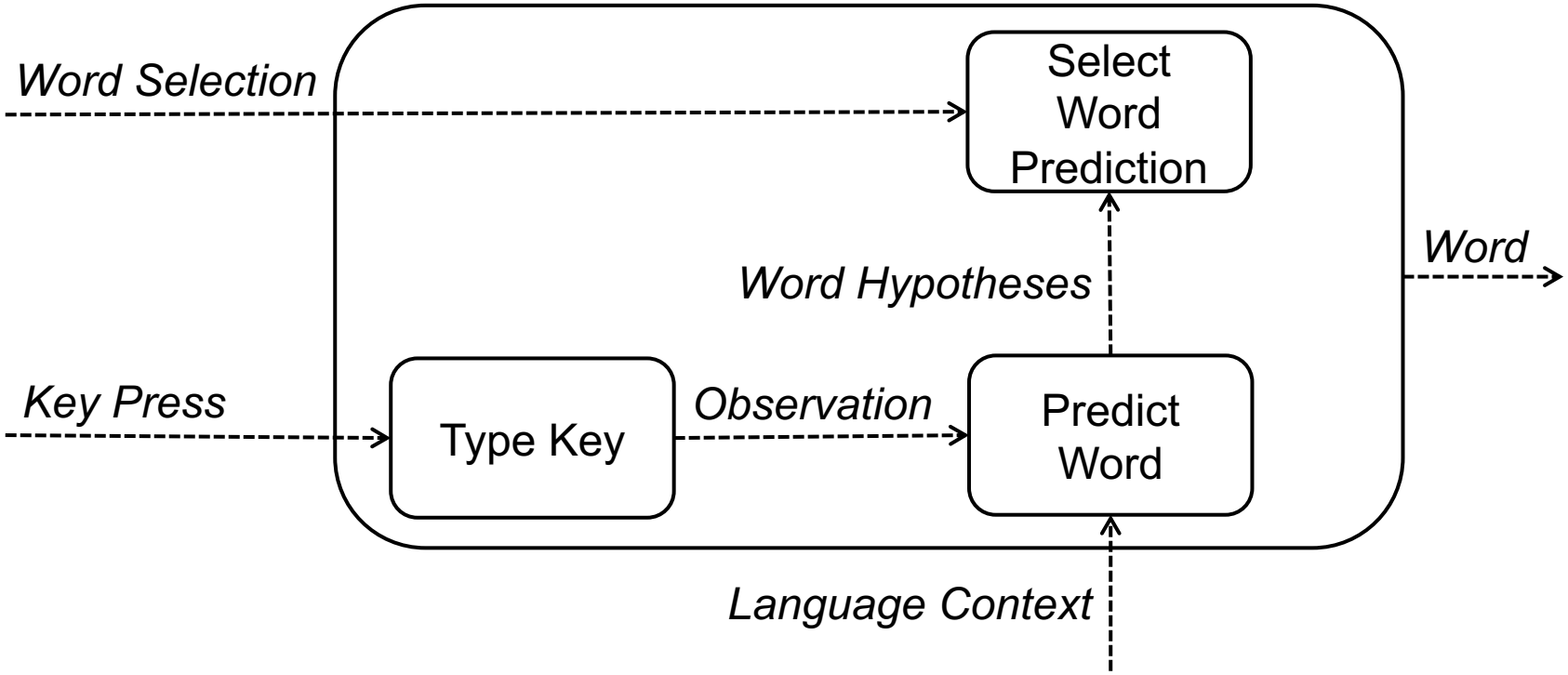


Example: pulse meter



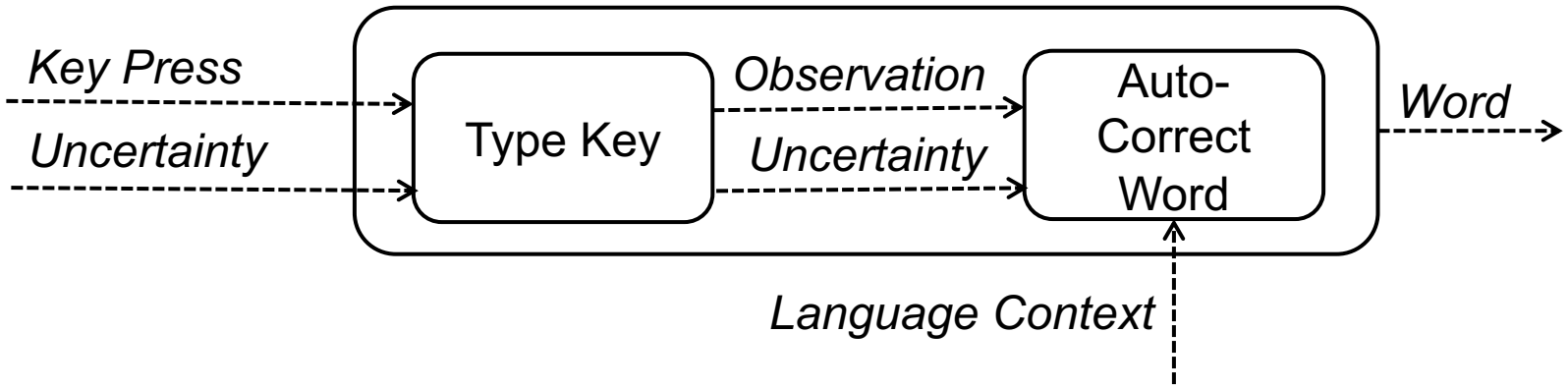
Function Models for AI-infused User Interfaces

Overall function: predict word



Kristensson, P.O. and Müllners, T. 2021. Design and analysis of intelligent text entry systems with function structure models and envelope analysis. In *Proceedings of the 39th ACM Conference on Human Factors in Computing Systems (CHI 2021)*. ACM Press: Article No. 584.

Overall function: regulate uncertainty



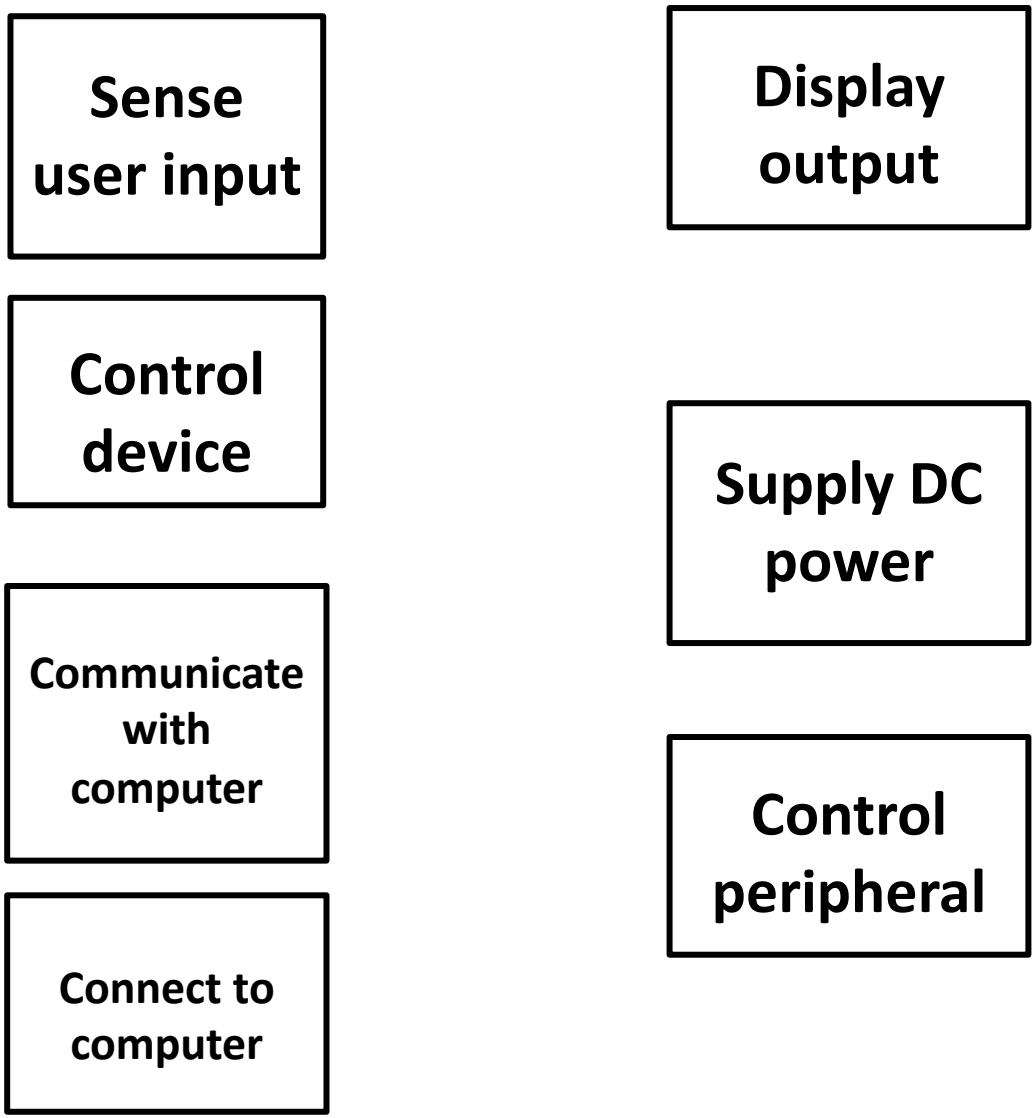
- Auto-correct function that allow users to regulate their uncertainty in their typing, for example, by pushing harder on keys they do not wish the system to auto-correct

Kristensson, P.O. and Müllners, T. 2021. Design and analysis of intelligent text entry systems with function structure models and envelope analysis. In *Proceedings of the 39th ACM Conference on Human Factors in Computing Systems (CHI 2021)*. ACM Press: Article No. 584.

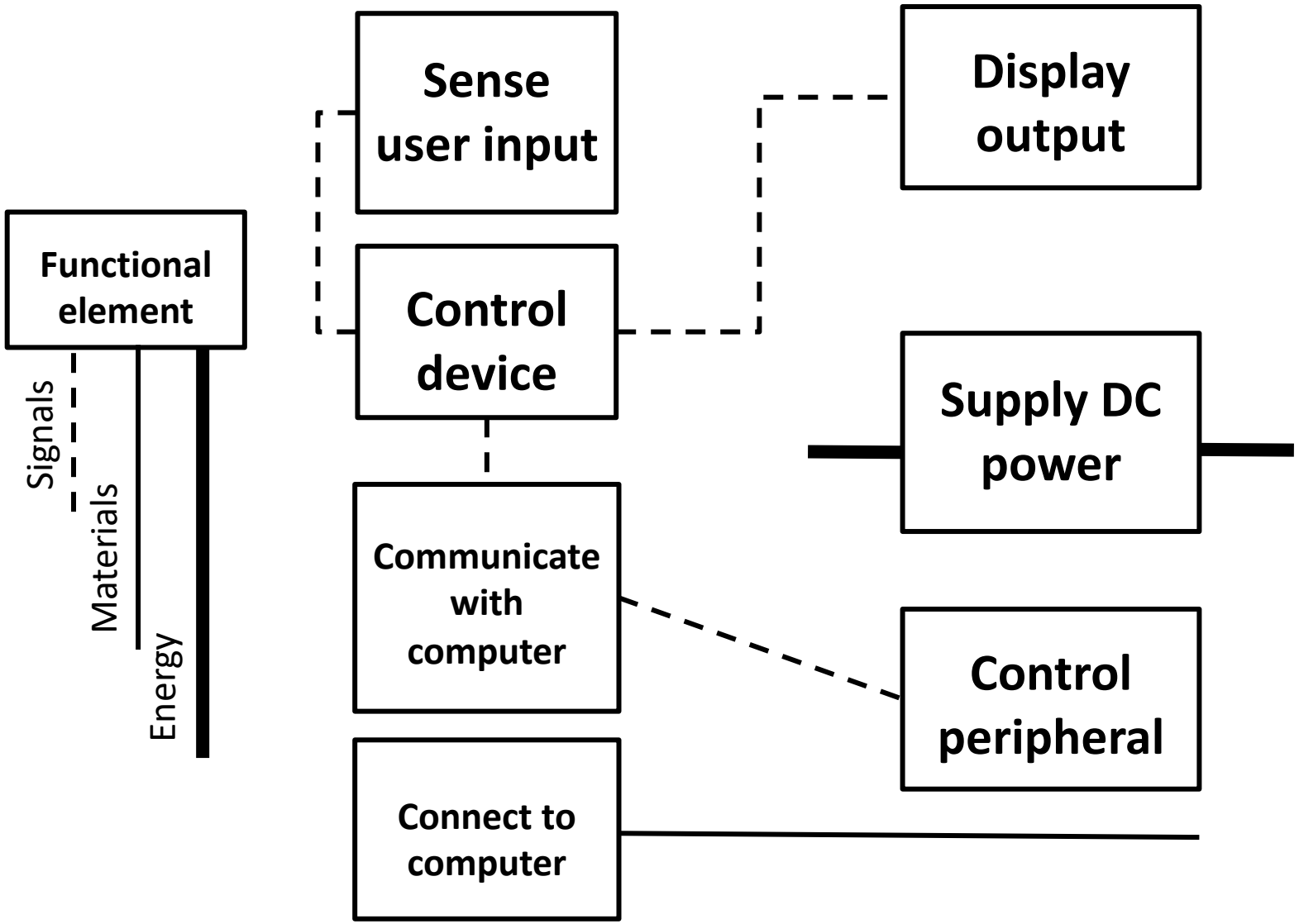
Modules

- To create a modular architecture:
  1. Start with a schematic overview of the product
  2. Select a desired level of product variety
  3. Create modules by clustering elements of the schematic overview

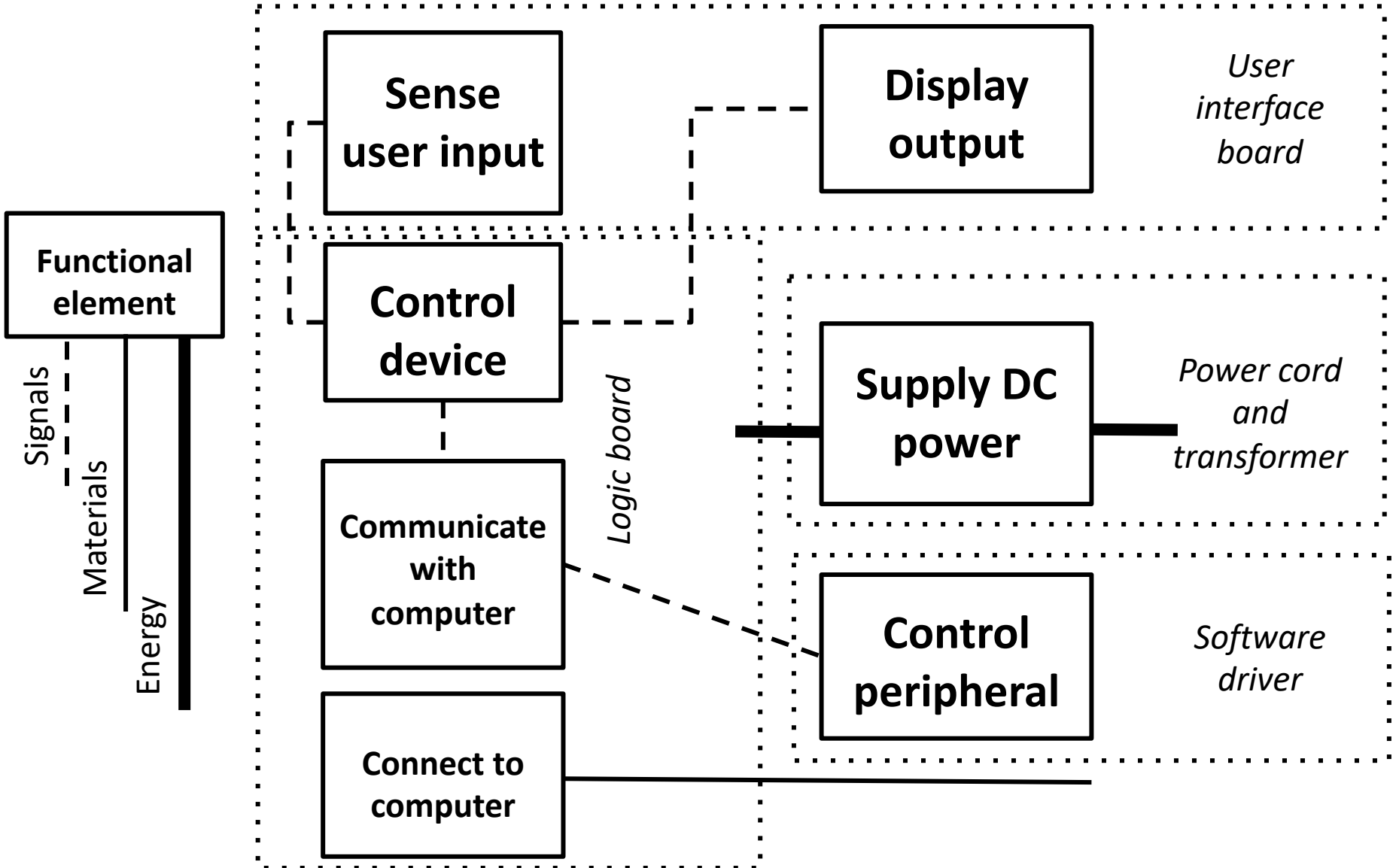
Example, subset of typical computer peripheral: functions



Example, subset of typical computer peripheral: functions+flows

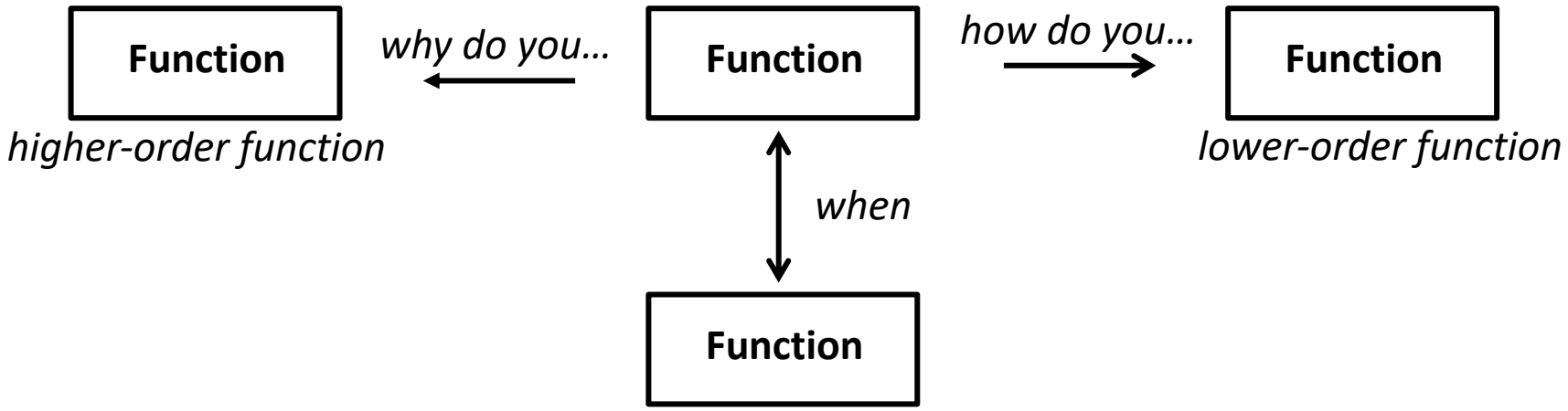


Example, subset of typical computer peripheral: functions+flows+modules



Functional Analysis System Technique (FAST)

FAST



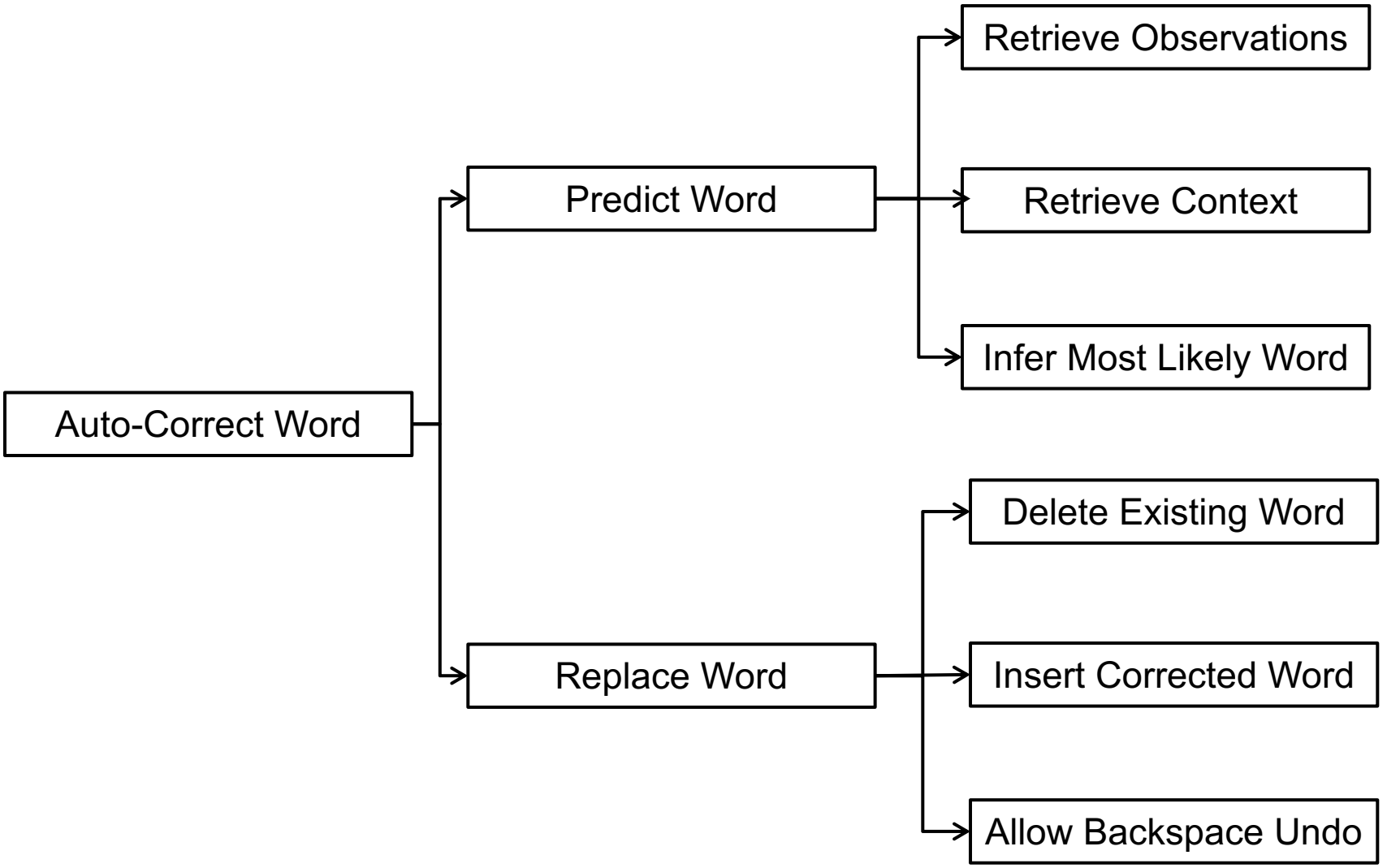
- A **function** is an active verb + measurable noun (e.g. “start motor”)
- The lateral axis models higher-order functions becoming lower-order functions
- Due to symmetry, a lower-order function explains *how* to carry out a higher order function, and a higher-order function explains *why* a lower-order function is carried out
- The vertical axis represent *time* (often at discrete steps)

FAST diagram

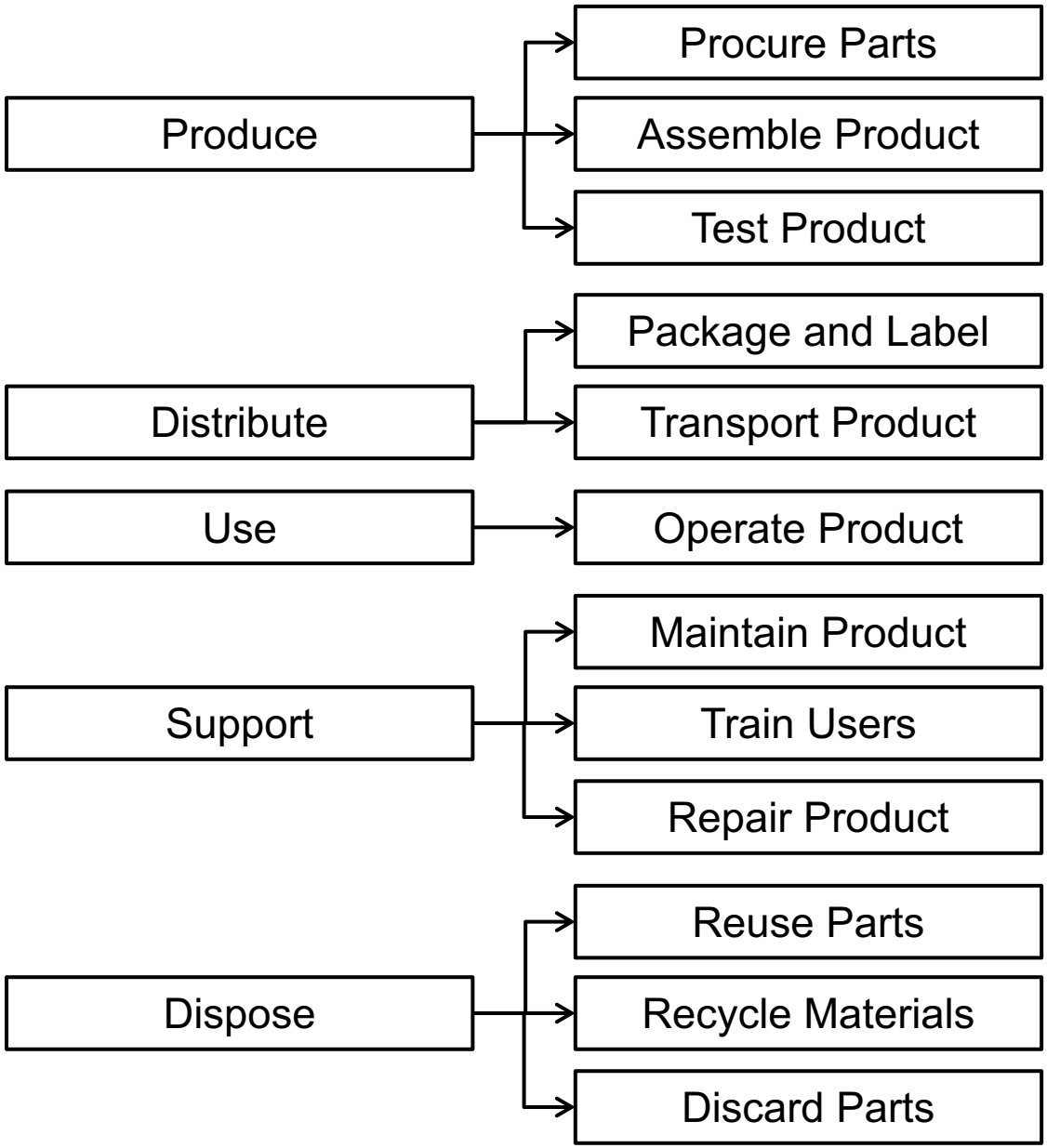
Constructing the FAST diagram has the following steps:

- Start with **general functions** and get progressively more specific
- Each function should be a **verb** and a **noun** (keep it as simple as possible)
- **Chronologically** trace through each function that must be accomplished
- Be sure to include all **special modes** of operation such as stand-by, run, cleaning, etc.
- **Avoid specifying** form, structure or solutions; describe behaviour, not embodiment
- **Customise** the FAST diagram; the more information that can be visualised, the more useful the diagram

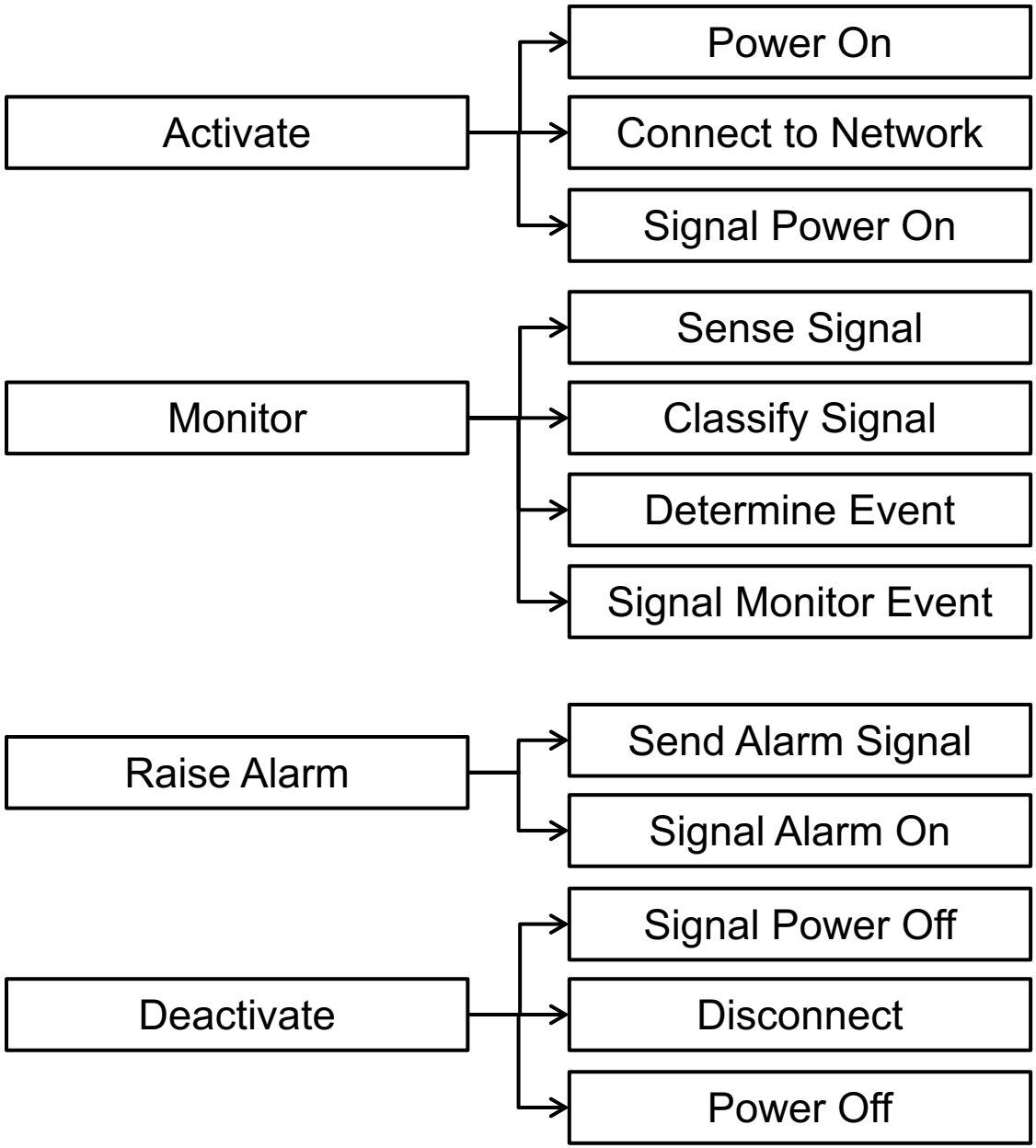
Example: auto-correct word (at a high abstraction level)



Life cycle analysis using FAST



Example partial workflow using FAST for a smart patient monitoring device in care home



Function structures vs. FAST

- Function structures
  - Allow gradual decomposition of the overall function (or functions) into subfunctions connected by signals, energy and material
  - Typically useful for understanding the conceptual design of **mechanisms**
- FAST
  - Creates a function model by being increasingly concrete (*how* do we carry out a function)
  - Typically useful for understanding **processes**, such as workflows
- Both types of function models allow for parameterisation

Morphological Charts

Morphological chart

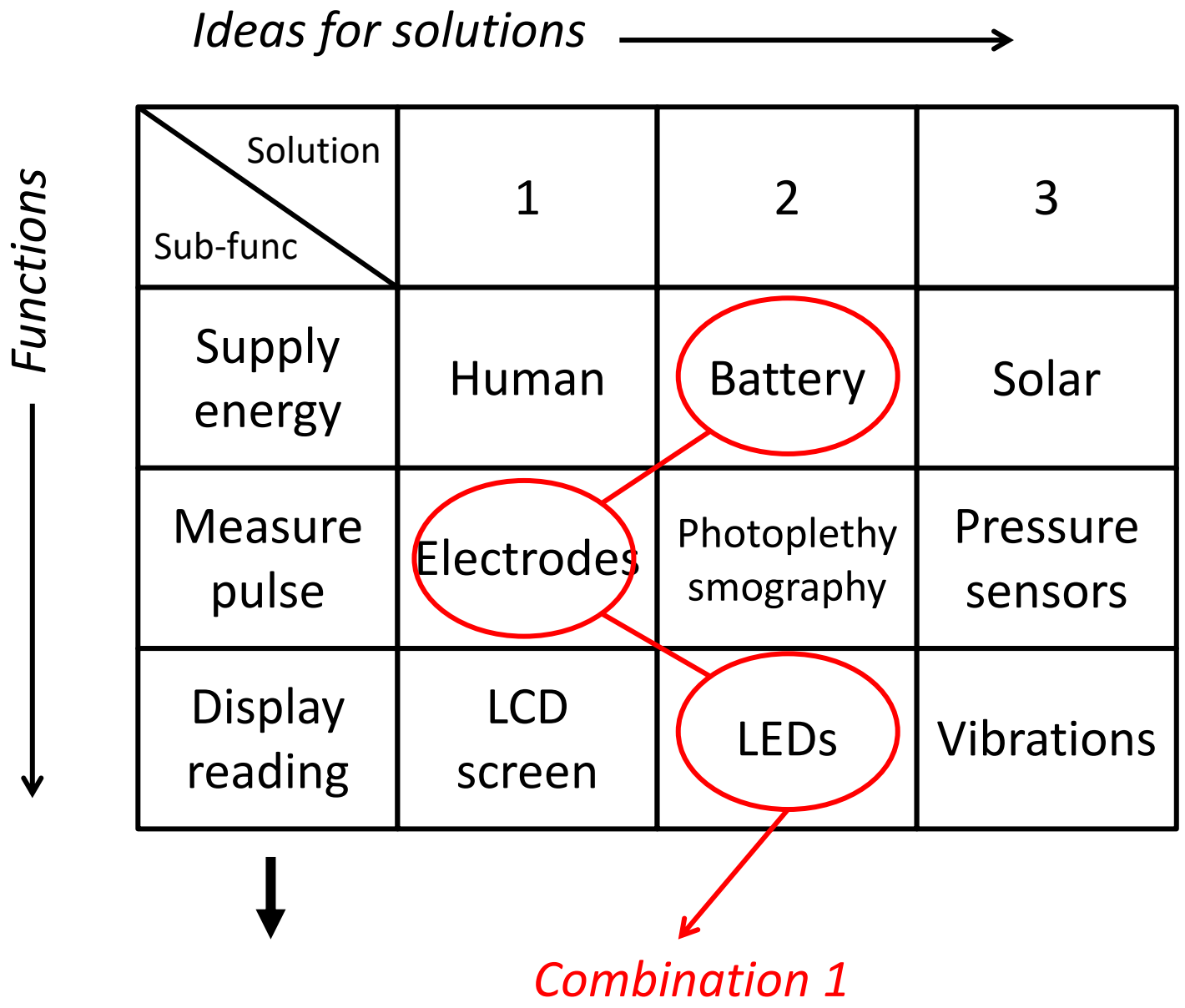
- Having identified a function model (or several), the next step is to translate **functions** into **function carriers** (sometimes called **solutions**)
- A **solution principle** is design know-how that can assist informed translation of a function to a function carrier
- By creating a mapping between all functions in our function model to relevant function carriers we create a **conceptual design** (often simply called a **concept**)
- A **morphological chart** is a design tool for generating candidate concepts

Ideas for solutions →

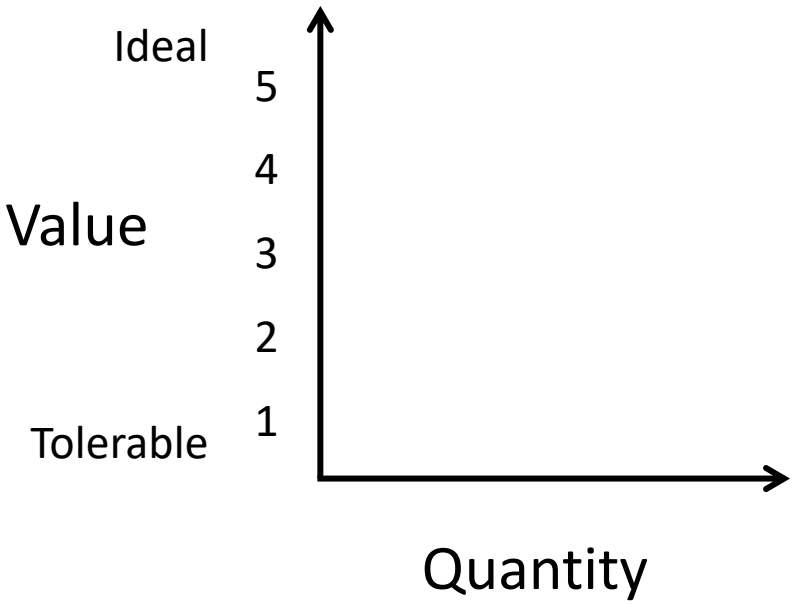
Functions ↓

Solution / Sub-func	1	2	3
Supply energy	Human	Battery	Solar
Measure pulse	Electrodes	Photoplethysmography	Pressure sensors
Display reading	LCD screen	LEDs	Vibrations

↓

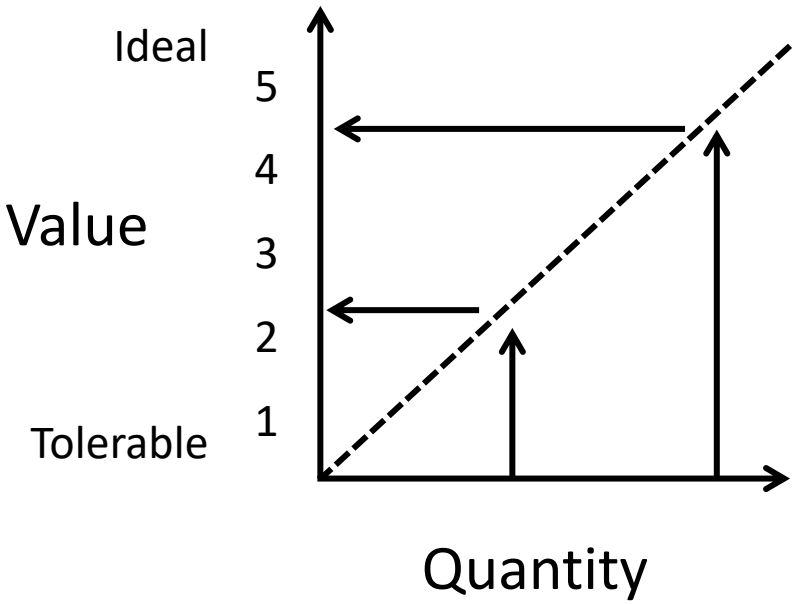

Ideas for solutions
Functions
Solution
Sub-func
1
2
3
Supply energy
Human
Battery
Solar
Measure pulse
Electrodes
Photoplethysmography
Pressure sensors
Display reading
LCD screen
LEDs
Vibrations
Combination 1

Concept Evaluation



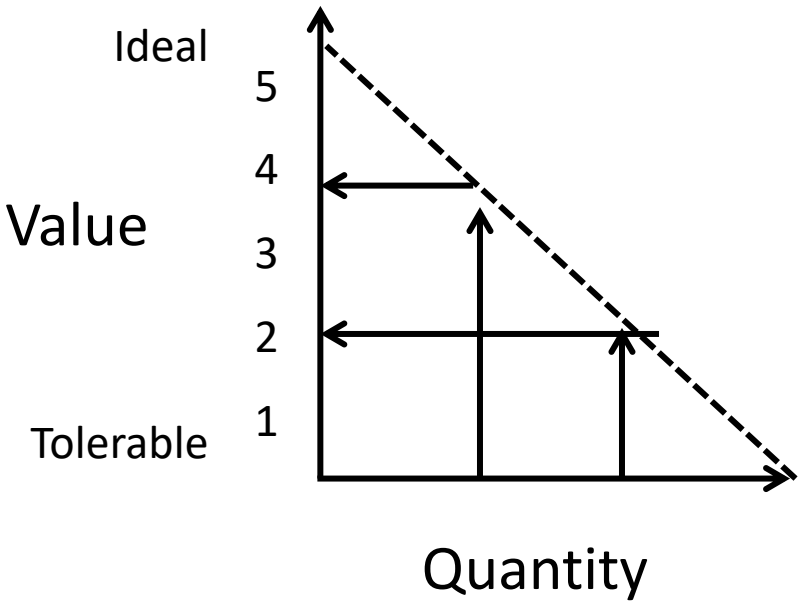
Battery life, weight, accuracy in sensing, etc.

estimates of parameters associated with concepts – may require some prototyping



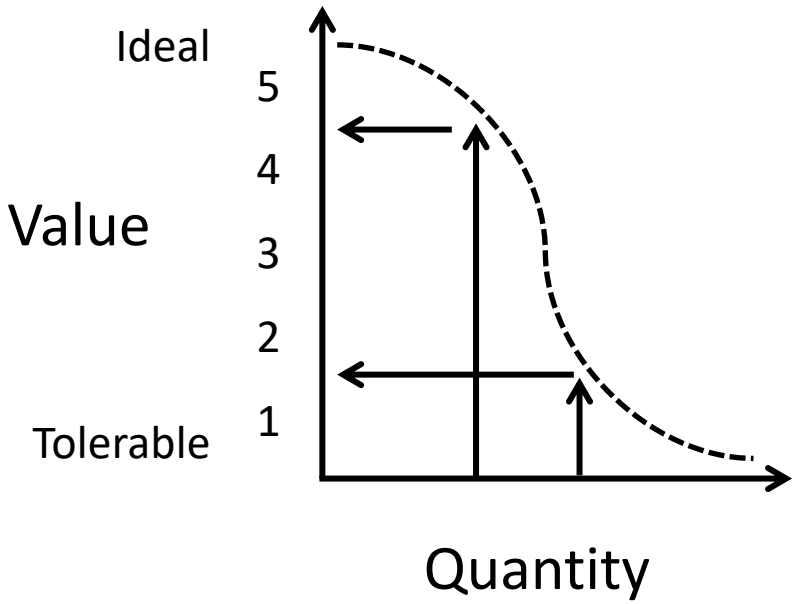
Battery life, weight, accuracy in sensing, etc.

estimates of parameters associated with concepts – may require some prototyping



Battery life, weight, accuracy in sensing, etc.

estimates of parameters associated with concepts – may require some prototyping



Battery life, weight, accuracy in sensing, etc.

estimates of parameters associated with concepts – may require some prototyping

Concept evaluation

Criteria	Weighting	Concept 1		Concept 2		Ideal
		Value	Wt val	Value	Wt val	Wt val
Weight	3	2	6	4	12	15
Appearance	3	2	6	4	12	15
Power	2	2	4	4	8	10
			16		32	40

linear combinations

rough estimates, precise numbers are misleading

the ideal is maximum, another benchmark could be a competitor or existing product (datum)

chosen concept must be motivated by a narrative

Envelope Analysis

Envelope analysis

- There is no formal term but I am going to call this **envelope analysis** (cf. flight envelope, back-of-the-envelope analysis, etc.)
- The idea is to extract design parameters from the functional description of the system and simulate potential performance by investigating a range of parameter choices
- Three common strategies:
  - KLM-GOMS and similar approaches
  - Wizard of Oz
  - Computational experiments

Function parameters

- Having identified a functional model it is possible to parameterise the model
- There are fundamentally two classes of parameters:
  - **Controllable parameters** govern function execution and can be set by the designer; they enable **optimisation towards design objectives**
  - **Uncontrollable parameters** govern function execution and cannot be explicitly set by the designer; they enable **sensitivity analysis**
- The effects of these parameters can be investigated in various ways
  - Analysis (such as envelope analysis)
  - Prototyping and measurements
  - Experimentation
  - Etc.

Keystroke Level Model (KLM)

GOMS

- **Goals**
  - Aims of the user
- **Operators**
  - Actions that can be done in the interface, such as clicking, dragging and typing
- **Methods**
  - Sequences of sub-goals and operators that can be used to achieve a particular goal
- **Selection rules**
  - The rules by which a user chooses a particular method (from a set of methods) to achieve a goal
- A GOMS task analysis breaks down tasks into a hierarchy of goals (unit tasks) and sub-tasks using the four GOMS elements: goals, operators, methods and selection rules

KLM-GOMS: Keystroke-level model

- A subset of GOMS that only includes operators and methods
- KLM predicts task completion times
- All operators have a specific execution time
- Task completion times are calculated by summing up the execution times for the different operators that need to be used to perform the task
- **KLM-GOMS** is the name of this particular GOMS model

KLM-GOMS standard operators (standard time estimates in parentheses)

K	Press a key on the keyboard (0.28 s)
T(n)	Type a sequence of *n* characters on the keyboard (*n* × K s)
P	Point the mouse to a target on the display (1.1 s)
B	Press or release the mouse button (0.1 s)
BB	Click mouse button (0.2 s)
H	Move hands between mouse and keyboard (or vice versa) (0.4 s)
M	Mental act of routine thinking or perception (1.2 s)
W(t)	Wait time for system response (*t* s)

Example from Cairns and Cox (2008)

- Deleting a file using the previously defined methods
- Operator sequence: **drag object [trash]**
  - Point to file icon (**P**)
  - Press and hold mouse button (**B**)
  - Drag file icon to ‘Trash can’ icon (**P**)
  - Release mouse button (**B**)
  - Point to original window (**P**)
- Operator sequence: send **object [trash]**
  - Point to file icon (**P**)
  - Press and hold mouse button (**B**)
  - Point to ‘Move to Trash’ item on pop-up menu (**P**)
  - Release mouse button (**B**)
  - Point to original window (**P**)
- Both involves 3**P** + 2**B** = 3 × 1.1 + 2 × 0.1 = 3.5 s

Limitations of KLM-GOMS

- Assumes error-free expert behaviour
  - Few interfaces are only used by users who interact using error-free expert behaviour
  - Ignores learning curve effects
- Assumes reliable fixed time estimates are available for all operators
  - We know, from for example Fitts' law, that not all tasks, such as pointing tasks, take an equal amount of time

Checklist when carrying out a KLM-GOMS analysis

1. Define the higher level activity (for example, “delete file”) precisely, so that it can be defined as a sequence of clearly defined tasks (for example, “select icon, drag icon to trashcan, release mouse button)
2. If possible, create “subroutines” for repeat tasks. This is possible because time predictions are constant.
   - For example, the “delete file via trashcan” operation we previously defined can be defined as a subroutine DeleteFileViaTrashcan that takes 3.5 seconds.
3. Carefully review your analysis, do you include all low-level steps (carried out via KLM-GOMS operators) that are required for the task?
   - It is easy to forget steps in a KLM-GOMS analysis

Wizard of Oz

Wizard of Oz

- Basic idea: A human operator **simulates** responses of an AI system
- Advantage: Ability to understand user issues without having to build an elaborate working system
- Uses:
  - Understanding emergent user behaviour at an early stage of the design, such as understanding how users may respond to a dialogue system
  - Collecting data necessary to train, tune or optimize an intelligent interactive system (boot-strapping)
  - Analysing the effect of design parameters or understanding the human performance potential at an early stage of the design process
- Challenges:
  - Representative tasks
  - Difficulty of human operator to simulate AI responses (latency, type of responses, etc.)
- Classic review: Dahlbäck, N., Jönsson, A. and Ahrenberg, L. 1993. Wizard of Oz studies — why and how. *Knowledge-Based Systems* **6**(4): 258-266

Computational Experiments

Oracle

- An oracle in user interface simulation is a simulation of a user that performs optimal
- It is also possible to create an imperfect oracle (I call it a “stochastic oracle” but there is no widely used or accepted term for it)
- Simulation by an oracle is a very useful way to carry out rudimentary envelope analysis to tease out trade-offs or prioritise features

Function Parameterisation

Mobile Speech Recognition Error Repair

Probabilistic error correction

- For any probabilistic text entry method…
  - Capable of assigning posterior probability distributions to words
- …there exists a **hypothesis space**
- The best result is the maximum probability path in this hypothesis space
  - However, it need not be the one the user intended
- By exposing part of the hypothesis space to users, high efficiencies can be gained when users correct words

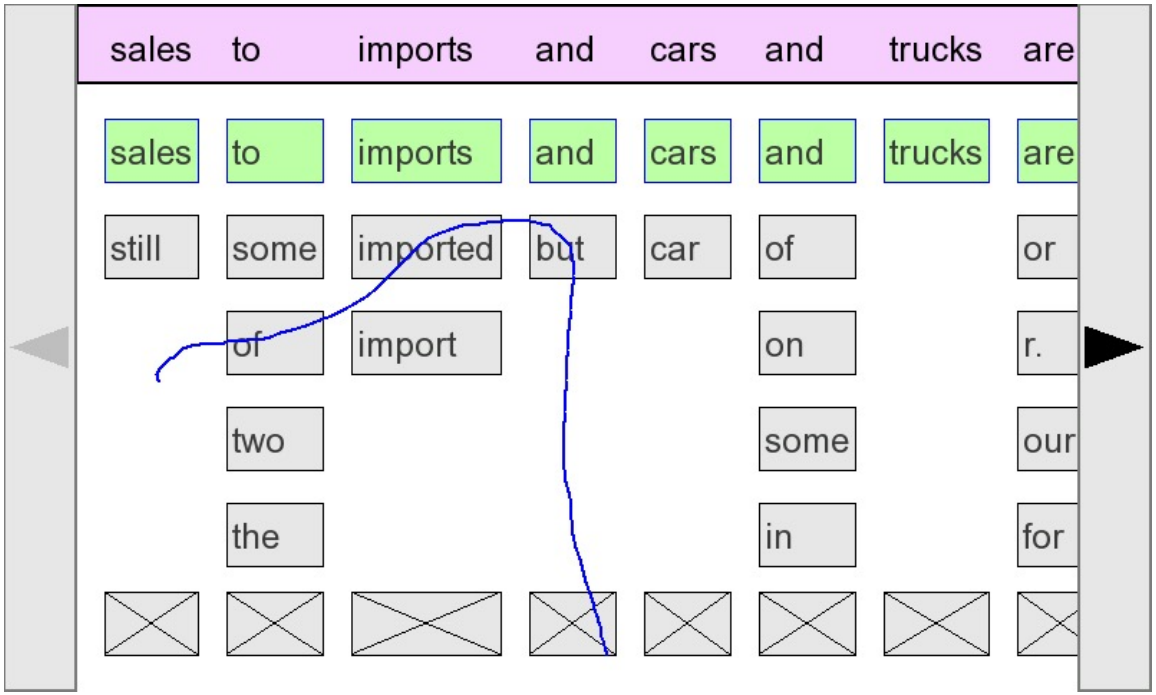
Example: Parakeet

- Built on PocketSphinx
- Runs on Nokia N800 devices
- Continuous speech recognition on the device

- Avoid cascading errors
- Exploit the speech recognition hypothesis space
- Efficient and practical interaction by touch
- Support fragmented interaction

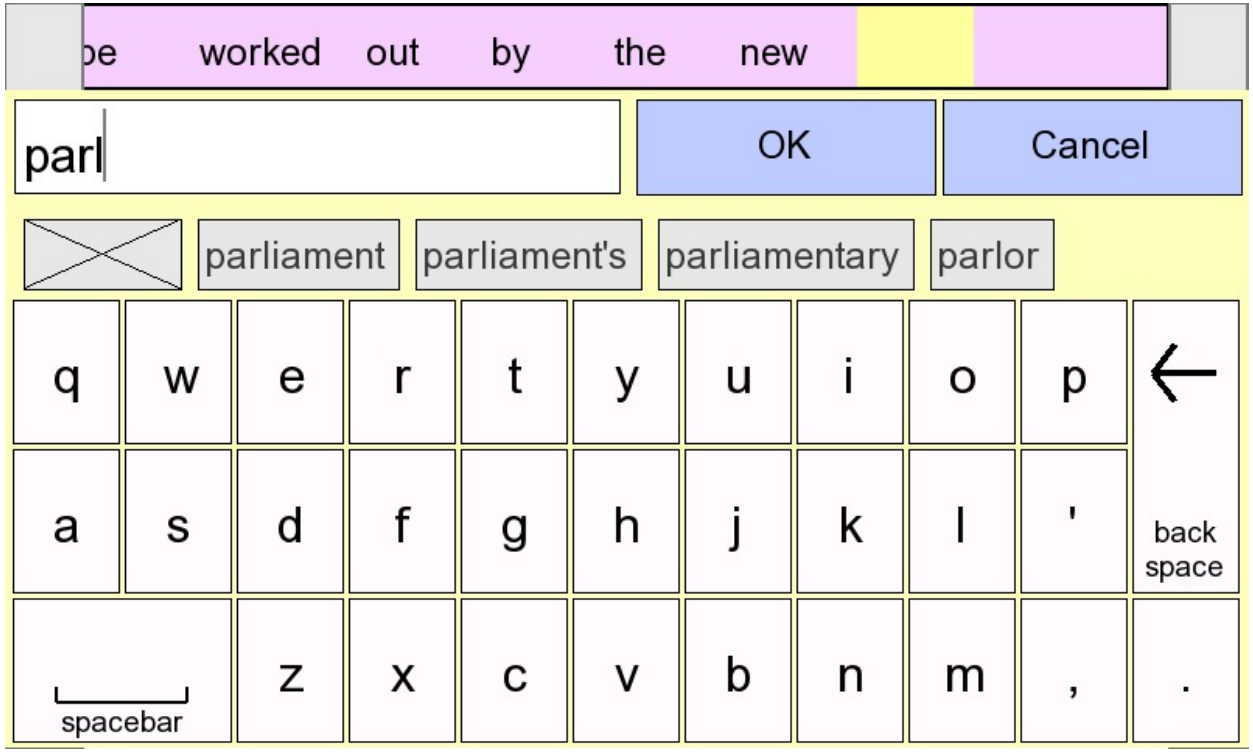
Vertanen, K. and Kristensson, P.O. 2009. Parakeet: a continuous speech recognition system for mobile touch-screen devices. In *Proceedings of the 14th ACM International Conference on Intelligent User Interfaces (IUI 2009)*. ACM Press: 237-246.

Word-confusion network (WCN)



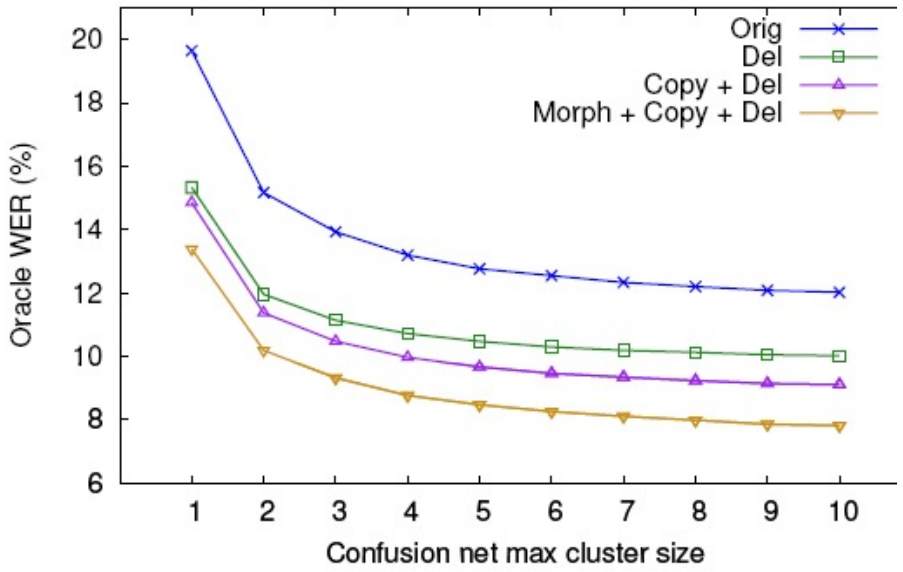
Vertanen, K. and Kristensson, P.O. 2009. Parakeet: a continuous speech recognition system for mobile touch-screen devices. In *Proceedings of the 14th ACM International Conference on Intelligent User Interfaces (IUI 2009)*. ACM Press: 237-246.

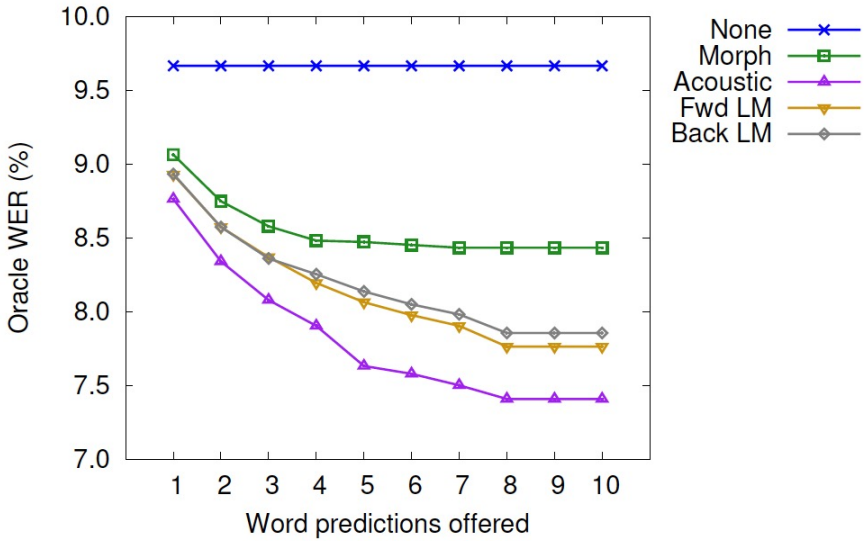
Predictive touch keyboard



Vertanen, K. and Kristensson, P.O. 2009. Parakeet: a continuous speech recognition system for mobile touch-screen devices. In *Proceedings of the 14th ACM International Conference on Intelligent User Interfaces (IUI 2009)*. ACM Press: 237-246.

Design optimisation



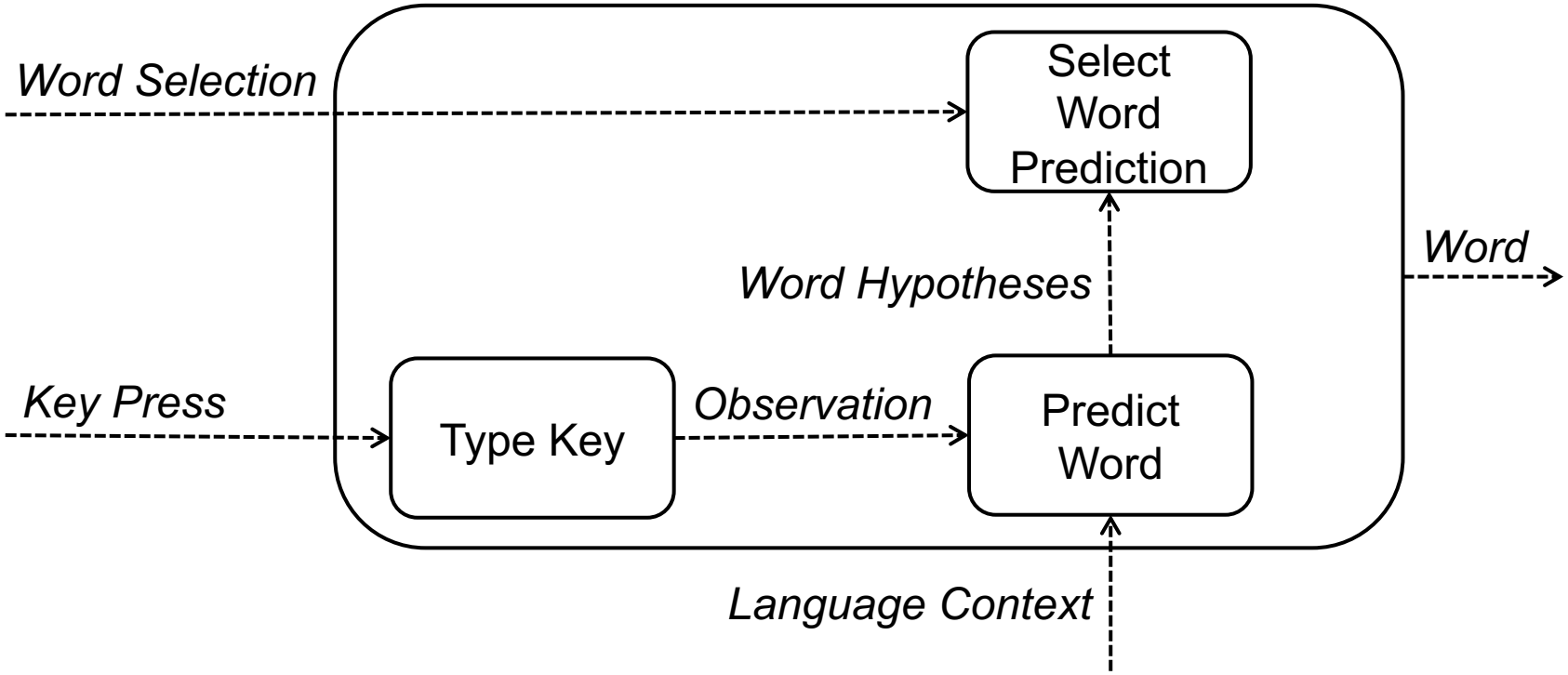


- Simulated a perfect (oracle) user
- Identified features which reduced the oracle word error rates the most

Function Parameterisation

Word Prediction User Strategy

Function parameterisation: word prediction example

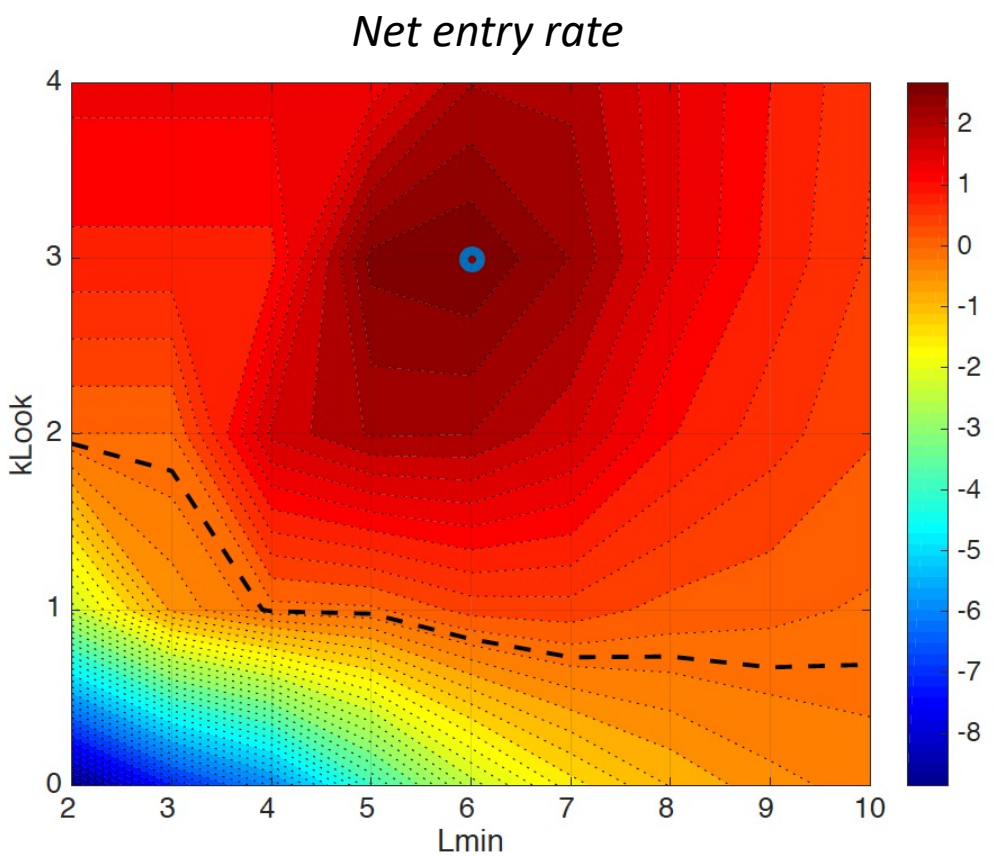


Kristensson, P.O. and Müllners, T. 2021. Design and analysis of intelligent text entry systems with function structure models and envelope analysis. In *Proceedings of the 39th ACM Conference on Human Factors in Computing Systems (CHI 2021)*. ACM Press: Article No. 584.

Function parameterisation: word prediction example

- We have two input signals and they can be parameterised as follows:
  - T_{key}: the time it takes the user to hit a key
  - T_{react}: the time it takes a user to react to a word prediction suggestion
- A latent set of parameters, *strategy*, generates process interaction between the design parameters:
  - L_{min}: restrict word prediction usage to words of a minimum length
  - k_{look}: type k keys then look
  - p_{max}: only attempt to check word predictions up to p_{max} key strokes per word
- Note that all these parameters are uncontrollable
- Other strategies are of course possible and in reality a user is likely using a mix of strategies

Kristensson, P.O. and Müllners, T. 2021. Design and analysis of intelligent text entry systems with function structure models and envelope analysis. In *Proceedings of the 39th ACM Conference on Human Factors in Computing Systems (CHI 2021)*. ACM Press: Article No. 584.



- Performance is strongly tied to the choice of typing strategy.
- The black dotted line is the zero-crossing and marks the boundary between performance gain and loss due to predictions.
- Any strategy that looks at word predictions without typing at least one letter is shown to slow the user down
- Performance gains are only realized if the user types at least the first two letters of each word before looking at predictions
- The reliability of predictions increases with the number of letters typed
- Net entry rate can range from -8.8 to +2.9 words per minute solely as result of the choice of typing strategy
- The highest point identifies the optimal typing strategy: $L_{min} = 6$; $k_{look} = 3$

Kristensson, P.O. and Müllners, T. 2021. Design and analysis of intelligent text entry systems with function structure models and envelope analysis. In *Proceedings of the 39th ACM Conference on Human Factors in Computing Systems (CHI 2021)*. ACM Press: Article No. 584.

Function Parameterisation

Context-Sensitive Sentence Retrieval for Nonspeaking Individuals with Motor Disabilities

Sentence retrieval

- Complementary completion assistance
- Fundamental idea based on observations of high-level of sentence re-use
- Rather than explicit look-up by user, integrate sentence retrieval with usual typing workflow

Kristensson, P.O., Lilley, J., Black, R. and Waller, A. 2020. A design engineering approach for quantitatively exploring context-aware sentence retrieval for nonspeaking individuals with motor disabilities. In *Proceedings of the 38th ACM Conference on Human Factors in Computing Systems (CHI 2020)*. ACM Press: Paper 398.

Sentence retrieval

Positive qualities

1. Potentially a very large increase in keystroke savings
2. Compatible with existing workflow (minimal learning)
3. Predictable outcomes for user allow for optimised retrieval after extensive use
4. Performance improves automatically the more it is used
5. Modular by its nature and allow for performance boosts by improving various sub-functions in the system

Negative qualities

1. Performance gains limited to sentence cache
2. Difficult to A/B evaluate system without expensive and invasive longitudinal study
3. Potential invasion of privacy
4. Many technical and non-technical design parameters and their individual and collective impact on eventual performance are unclear
5. As in all AAC design, representative data is extremely scarce

Kristensson, P.O., Lilley, J., Black, R. and Waller, A. 2020. A design engineering approach for quantitatively exploring context-aware sentence retrieval for nonspeaking individuals with motor disabilities. In *Proceedings of the 38th ACM Conference on Human Factors in Computing Systems (CHI 2020)*. ACM Press: Paper 398.

Quantitative analysis: basic idea

- Use AAC surrogate set consisting of 5,000 **distinct** AAC sentences
- Draw 500, this forms the sentence cache
- Generate context tag families (such as **Location**) and context tags within context tags families (such as **Location1**, **Location2**, etc.)
- Simulate typing by creating a **typed sentence** that will contain:
  1. Context tags immediately assigned to the sentence (with assumptions on correct inference of context tags)
  2. Gradually build-up of words, either words completely typed out by simulated user, or auto-completed (with certain probability)
  3. Both context tags and generated words (either typed or auto-completed) are added to a bag-of-words
  4. The system uses this bag-of-words to rerank all sentences in the sentence cache
  5. A subset of top-ranked sentences (default: 4) are "shown" to the user
  6. If any of these match the intended sentence, there are observed keystroke savings

Kristensson, P.O., Lilley, J., Black, R. and Waller, A. 2020. A design engineering approach for quantitatively exploring context-aware sentence retrieval for nonspeaking individuals with motor disabilities. In *Proceedings of the 38th ACM Conference on Human Factors in Computing Systems (CHI 2020)*. ACM Press: Paper 398.

Quantitative analysis: setup

- Objective: Study emergent system performance envelope by varying controllable and uncontrollable parameters
- Implemented three standard information retrieval algorithms
  - IDF
  - BM25
  - Unigram
- The simplicity of the algorithms ensure the analysis is conservative

Kristensson, P.O., Lilley, J., Black, R. and Waller, A. 2020. A design engineering approach for quantitatively exploring context-aware sentence retrieval for nonspeaking individuals with motor disabilities. In *Proceedings of the 38th ACM Conference on Human Factors in Computing Systems (CHI 2020)*. ACM Press: Paper 398.

Quantitative analysis: main results

- Under very simple and conservative assumptions, context-aware sentence retrieval provides good gains in terms of keystroke savings
- Assuming two perfect context tags, we typically obtain a keystroke savings range of 94–97% when the word prediction accuracy is assumed to reside at around 80%
  - This forms an important requirement on this particular sub-system
- A context tagging error of 50% results in average keystroke savings at around 70%

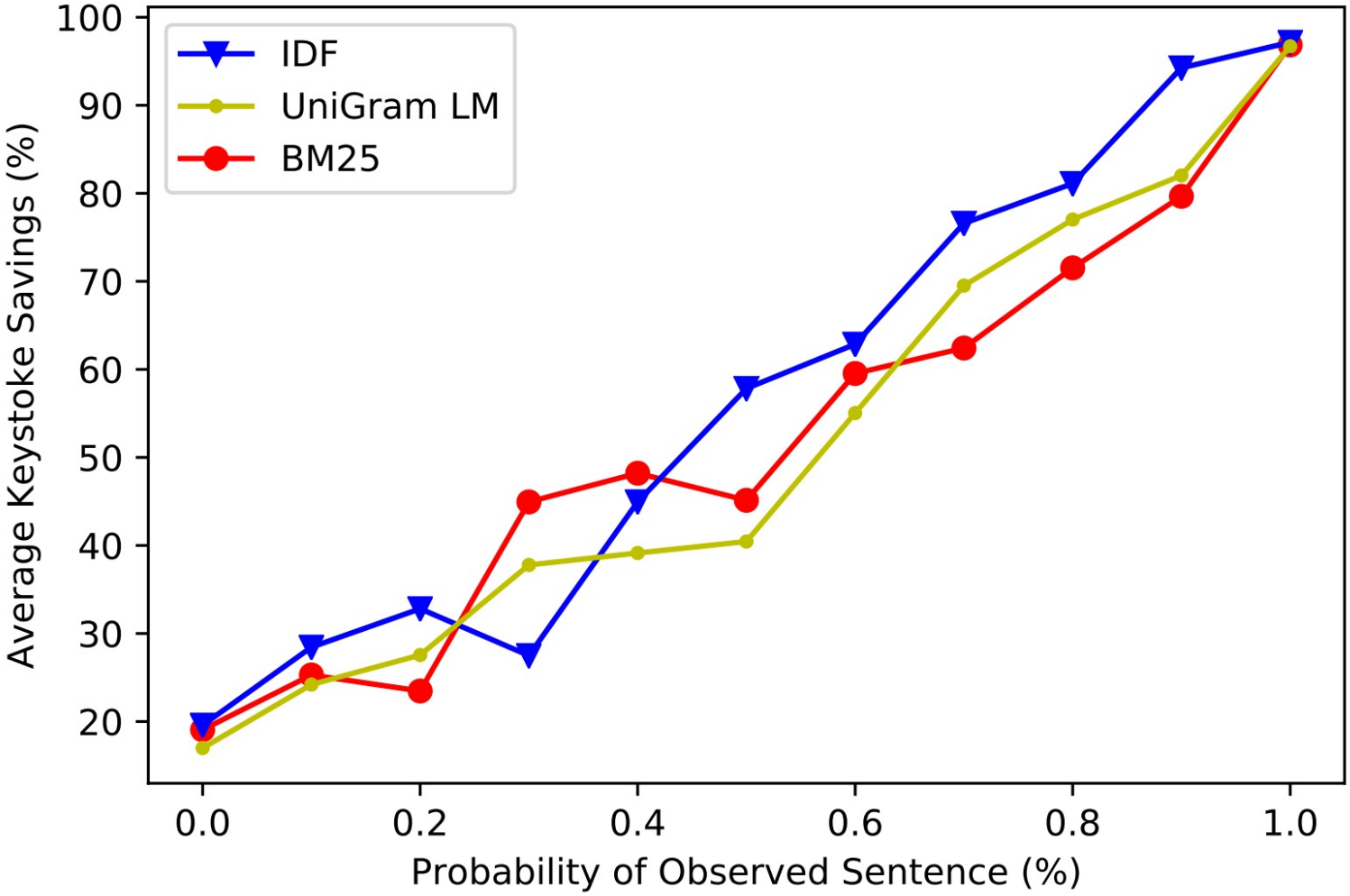
Kristensson, P.O., Lilley, J., Black, R. and Waller, A. 2020. A design engineering approach for quantitatively exploring context-aware sentence retrieval for nonspeaking individuals with motor disabilities. In *Proceedings of the 38th ACM Conference on Human Factors in Computing Systems (CHI 2020)*. ACM Press: Paper 398.

Quantitative analysis: design implications

- What does this mean:
  - For sentences **inside the cache**, context-aware sentence retrieval provides very large gains under even modest performance of context tagging and auto-complete
  - Auto-complete is not a major factor if context tagging is moderately accurate
  - The more context tags, the more accurate they are, and the more well-distributed they are across sentences, the better the performance
  - More sentences help…

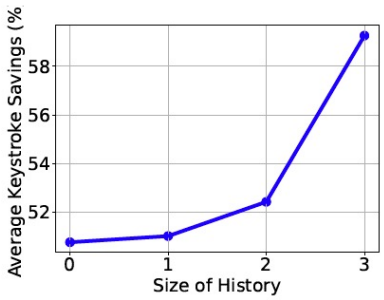
Kristensson, P.O., Lilley, J., Black, R. and Waller, A. 2020. A design engineering approach for quantitatively exploring context-aware sentence retrieval for nonspeaking individuals with motor disabilities. In *Proceedings of the 38th ACM Conference on Human Factors in Computing Systems (CHI 2020)*. ACM Press: Paper 398.

Sentence outside cache, worst-case analysis

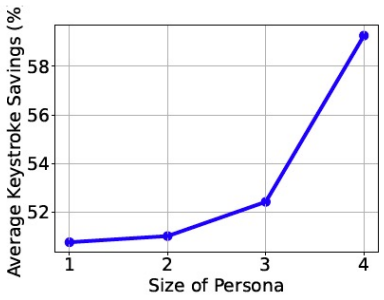


Kristensson, P.O., Lilley, J., Black, R. and Waller, A. 2020. A design engineering approach for quantitatively exploring context-aware sentence retrieval for nonspeaking individuals with motor disabilities. In *Proceedings of the 38th ACM Conference on Human Factors in Computing Systems (CHI 2020)*. ACM Press: Paper 398.

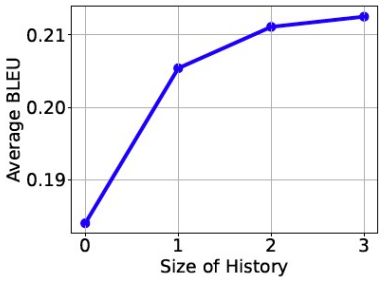
Example: GPT-2 model for sentence generation



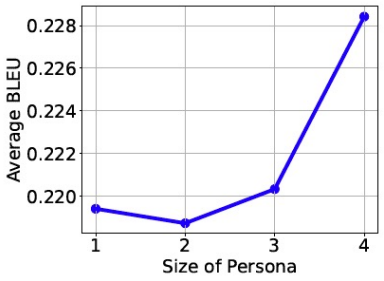
(a) Keystroke savings versus the size of history.



(b) Keystroke savings versus the number of persona tags.



(c) BLEU score versus the size of history.



(d) BLEU score versus the number of persona tags.

Fig. 7. Benefits of increasing the size of history (number of exchanges in conversation history) and persona tags. More history and persona tags help the language generation model to output higher quality responses.

- Estimating benefits of persona (tags describing the user) and history (prior exchanges in a dialogue)
- Allows estimation of benefits and determination of design parameters without elaborate user studies (which in this case are also very difficult to carry out)

Shen, S., Yang, B., Dudley, J.J. and Kristensson, P.O. 2022. KWickChat: a multi-turn dialogue system for AAC using context-aware sentence generation by bag-of-keywords. In *Proceedings of the 27th ACM International Conference on Intelligent User Interfaces (IUI 2022)*. ACM Press: forthcoming.

Example: GPT-2 model for sentence generation

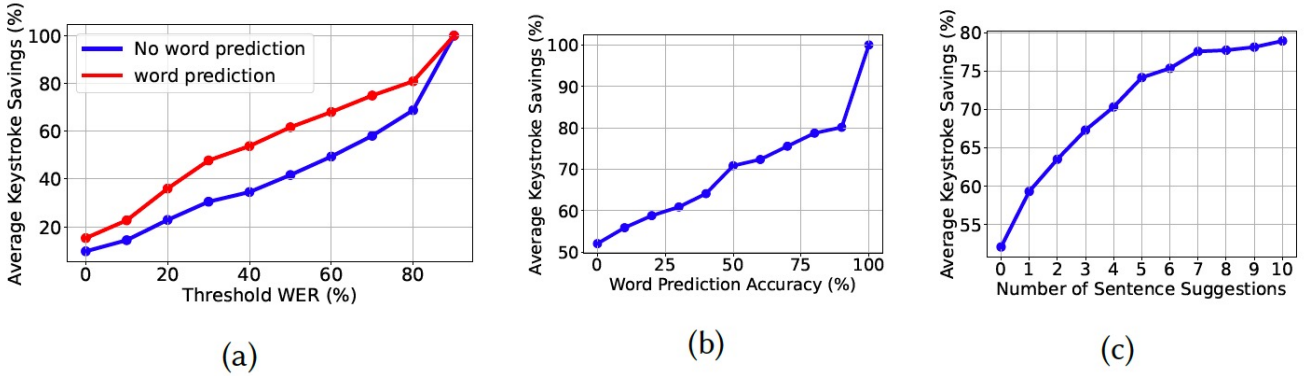


Fig. 12. Envelope analysis on word prediction and sentence display. Both word prediction and sentence display play an important role in improving the performance of the KWickChat system. a) WER scores change as the threshold WER changes for word prediction and no word prediction. b) Keystroke savings versus word prediction accuracy. c) Keystroke savings versus the number of sentences suggested.

- Estimating benefits of persona (tags describing the user) and history (prior exchanges in a dialogue)
- Allows estimation of benefits and determination of design parameters without elaborate user studies (which in this case are also very difficult to carry out)

Shen, S., Yang, B., Dudley, J.J. and Kristensson, P.O. 2022. KWickChat: a multi-turn dialogue system for AAC using context-aware sentence generation by bag-of-keywords. In *Proceedings of the 27th ACM International Conference on Intelligent User Interfaces (IUI 2022)*. ACM Press: forthcoming.

Summary

- Conceptual design is the process when a designer arrives at a functional description of a system and then translates functions to function carriers to generate conceptual designs
- Analysis of controllable and uncontrollable design parameters allow the generation of performance envelopes that can both help shape requirements and inform further design