



Center for
**LifeLong
Learning
& Design**

University of Colorado at Boulder

**Wisdom is not the product of schooling
but the lifelong attempt to acquire it.
- Albert Einstein**

Design = The Sciences of the Artificial

**Gerhard Fischer and Leysia Palen
Spring Semester 1999**

February 8, 1999

The Sciences of the Artificial

- **The Sciences of the Artificial: a book by Herbert Simon**
 - 1st edition: 1969
 - 2nd edition: 1981
 - 3rd edition: 1996
- **who is Herbert Simon:**
 - a founder of Artificial Intelligence
 - a Nobel Prize Winner in Economics
 - a major figure in Psychology and Cognitive Science

The Unifying Themes in Simon's Work

- **bounded rationality** = there are limits on a human as a decision maker and a problem solver (especially limits in cognitive processing) ---> these limits are important for the behavior of humans
- **satisficing** = accepting solutions which are “good enough”
 - “The best is often the enemy of the good”
 - the concept of “*satisficing*” separated Artificial Intelligence from Operations Research

Some Famous Quotes from Simon

- “What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention, and a need to allocate efficiently among the overabundance of information sources that might consume it.”
importance: contextualizing information (beyond current “push” technologies, making information relevant to the task at hand, critiquing,
- “The smartest people in the world do not generally look very intelligent when you give them a problem that is outside the domain of their vast experience.”
importance: domain-orientation, human problem-domain interaction
- definition of design = “Everyone designs who devise courses of action aimed at changing existing situations into preferred ones. The intellectual activity that produces material artifacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that devises a new sales plan for a company or a social welfare policy for a state (Simon, “Sciences of the Artificial”, p 130)”

Some Systems and Concepts Developed by Simon

- **systems**
 - Logic Theorist
 - General Problem Solver
 - Bacon
- **concepts:**
 - ill-defined problems
 - *informational equivalence versus computational equivalence* of representations ---> “even if two representations contain exactly the same information, it may be far cheaper, computationally, to make some of this information explicit using one representation than using the other

Number Scrabble

- two person game
- numbers from 1 to 9
- players alternate and take one of the numbers
- the player who can add exactly three numbers in her/his possession to equal 15 will win

Design Deals with Wicked or Ill-Defined Problems

Rittel in Cross "Developments in Design Methodology"

- there is no definitive formulation of a wicked problem. For any given tame problem, an exhaustive formulation can be stated containing all the information the problem-solver needs for understanding and solving the problem.
- they have no stopping rule. In tame problems, problem solvers know when they have done the job. Problem solvers terminate work on a wicked problem, not for reasons inherent in the 'logic' of the problem.
- solutions to wicked problems are not “true-or-false”, but “good-or-bad”
- every wicked problem is essentially unique
- the aim of design is not to find the truth, but to improve some characteristics of the world where people live

The Shape of the Design: Hierarchy

The Problem of Modularity

- **claim:** to design a complex structure, one powerful technique is to discover viable ways of decomposing it into semi-independent components corresponding to its many functional parts. The design of each component can then be carried out with some degree of independence of the design of others, since each will affect the others largely through its function and independently of the details of the mechanisms that accomplish the function.
- **examples:**
 - functional programming
 - object-oriented programming
 - rule-based systems
 - nearly decomposable systems