



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**

Overview of Design

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

January 13, 1999

Design, Learning and Collaboration

	design	learning	collaboration
theory	reflection-in-action, problem framing and problem solving, domains	knowledge construction, distributed cognition	symmetry of ignorance, shared understanding, knowledge management
systems	DODEs, design rationale, decentralized construction	Intelligent Tutoring Systems, Interactive Learning Envrionments	organizational memories, social filtering
practice	participatory design, long-term indirect collaboration, product and process	self-directed learning, beyond lectures	communities of practice, communities of interest
assessment	context, usable, useful	beyond tests, self-assessment	ethnography

The Sciences of the Artificial

Understanding the Natural and the Artificial World

- **natural science: “how things are”**
 - knowledge about natural objects and phenomena
 - primary interest: analysis
 - examples: physics, chemistry
- **sciences of the artificial: “how things might be (and ought to be in order to attain goals and to function)”**
 - knowledge about artificial objects and phenomena
 - primary interest: synthesis
 - artificial things are as they are only because of a system's being molded, by goals and purposes, to the environment in which it lives
 - examples: engineering, medicine, business, architecture, painting, universities, cognitive artifacts, notations

Definition of “Artificial”

- **Definition of "artificial":** human-made as opposed to natural
- **questions:** where does mathematics / computer science / biology belong too?
- **claims by Simon:**
 - the world in which we live in today is much more a human-made, or artificial, world than it is a natural world
 - a plowed field is no more part of nature than an asphalted street - and no less
- **Alan Kay (Scientific America, Sept 84, p 57)**
“molecular biology has the advantage of studying a system already put together and working; for the composer of software the computer is like a bottle of atoms waiting to be shaped by an architecture he must invent and then impress from the outside”

Science of Design

- **Definition:**

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)
- **generic design — does it exist?**
 - design as an activity has a distinct conceptual and cognitive realization from non-design activities
 - it can be abstracted away from the particulars of the knowledge base of a specific task or discipline and studied in its own right

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

Complexity of Designs

from R. Dawkins: "The Blind Watchmaker"

- biology is the study of complicated things that give the appearance of having been designed for a purpose
- physics is the study of simple things that do not tempt us to invoke design
- treat complex human-made artifacts (e.g., computers, airliners, cars, books) as biological objects
- the behavior of physical, nonbiological objects is so simple that it is feasible to use existing mathematical language to describe it
- a complex thing is something whose constituent parts are arranged in a way that it is unlikely to have arisen by chance alone

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

Domain-Oriented Design Environments

- **goals:**
 - bring task to the forefront
 - analysis of work products
 - goal sharing (for agents, critics, task-based indexing)
 - information delivery
 - learning on demand
 - external simplicity through internal complexity
- **theory:**
 - collaborative problem solving
 - distributed cognition
 - integration of problem framing and problem solving
 - reflection-in-action
 - design-in-use
 - situational awareness
 - computational environments as “living” entities
- **users:**
 - skilled domain workers
 - stakeholders with different interest and different background knowledge

Examples of Domain-Oriented Design Environments

- user interface design — **Framer**
- floor plan design for kitchens — **Janus, KID**
- computer network design — **Network, Pronet**
- voice dialog design — **VDDE** (with USWest)
- lunar habitat design — **HERMES** (with NASA)
- multi-media design environment — **eMMa** (with SRA)

A Preview of Further Themes to Be Discussed in Design

- Design = The Science of the Artificial
- Three Generations of Design Methodologies
- Domain-Oriented Design Environments
- Critiquing
- Software Design
- Making Argumentation Serve Design (Reflection-in-Action)

Some References for Design

- Simon, H. A. (1996) *The Sciences of the Artificial*, 3rd edition, The MIT Press, Cambridge, MA.
- Norman, D. A. (1993) *Things That Make Us Smart*, Addison-Wesley Publishing Company, Reading, MA.
- Schön, D. A. (1983) *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York.
- Winograd, T. (Ed.) (1996) *Bringing Design to Software*, ACM Press and Addison-Wesley, New York.
- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977) *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, New York.
- Gamma, E., Johnson, R., Helm, R., & Vlissides, J. (1994) *Design Patterns - Elements of Reusable Object-Oriented Systems*, Addison-Wesley,
- Dawkins, R. (1987) *The Blind Watchmaker*, W.W. Norton and Company, New York - London.
- Rittel, H. (1984) "Second-Generation Design Methods," In N. Cross (eds.), *Developments in Design Methodology*, John Wiley & Sons, New York, pp. 317-327.
- Ehn, P. (1988) *Work-Oriented Design of Computer Artifacts*, Almqvist & Wiksell International, Stockholm, Sweden
- Fischer, G. (1994) "Domain-Oriented Design Environments," *Automated Software Engineering*, 1(2), pp. 177-203.