

Lego logics

Description

default watermark

Thereâ??s been a lot on reddit lately about Lego bricks: How many Legos, stacked one on top of the ereâ??s lots to learn from toy construction sets, not



In the first flush of enthusiasm for artificial intelligence and automated knowledge-based systems back in the 1980s, some of us thought that at their most creative, people employ two kinds of intelligence. Thereâ??s (i) the *generative* and (ii) the *interpreting* kind of intelligence.

We thought that *generative* intelligence in architecture requires rules about how components go together. *Interpretive* rules are those from which you derive some kind of meaning, or a description of the way the building will perform. The trick is to use both kinds of rules (knowledge) to produce designs that are functional and meaningful. In a book on *Knowledge-Based Design Systems* published in 1990, we explained the two kinds knowledge in terms of Lego bricks:

Consider a childâ??s construction set (such as a set of Lego parts). Inside the box is a vocabulary of plastic parts which fit together according to a syntax of composition, determined by the plug and socket connections between parts. The designs that result can be interpreted as spaceships, moon buggies, cars and houses. The interpretation works both ways. If the child wants to play with a â??truck,â?• he or she appears to translate this â??meaningâ?• into a particular set of design charactersitics â?? four wheels, a platform, and a cab. Using knowledge of the vocabulary and syntax, he or she constructs a design with those characteristics (p.58).



default watermark

We also thought there was some practical advantage with this division, particularly if you want to create computer programmes that designers would use, or that could even automate some aspects of the design process: produce office floor plan layouts, or design stairs to fit in small spaces.

If design is a rule-based activity, then you might have different rules for each kind of knowledge: rules for generating and rules for interpreting. An automated â??design systemâ?• would need both *generative* and *interpretive* knowledge.

It sounds plausible, but there are major problems with Lego logics. Here are just a few.

- Thereâ??s a lot of research into the automated production of designs using knowledge as rules, constraints, algorithms, networks, and schemas. As anyone working in the area knows, the process is bedevilled by the so-called â??combinatorial problem.â?• There are many millions of ways of organising even a handful of Lego blocks. (See interesting online article: â??A Lego

counting problem.â?•) Testing them all beats the capabilities of any computer, and clever short cuts are never generalisable to all design conditions.
- To think of human intelligence, capability or aptitude in terms of rules or algorithms is very limiting. Knowledge is *tacit* rather than *explicit*. Itâ??s not as easy to identify design rules in real-world design situations as it is to fit Lego bricks together. The goal of any design task is ill-defined in any case.
- The idea of knowledge bases makes it difficult to accommodate the social, political, cultural and contextual, ie the contingent. Put simply, it leaves out the people.
- Design approaches that focus on generative knowledge put a lot of emphasis on making forms and shapes, as if designing is just a matter of manipulating Lego bricks. Critics attribute some of the least human, even if starkly beautiful, environments to a configurational approach to building cities, eg Oscar Niemeyerâ??s National Congress complex in Brasilia.
- The dual-knowledge idea (generative v interpretive) perhaps reinforces the idea that there are two cultures, or even two kinds of people. There are those who are good at designing buildings (the creatives), and there are those highly adept at criticising, reviewing, evaluating, and writing about buildings (the interpreters). This division between the creators and the critics has a long history.
- Some colleagues and I think itâ??s more useful to now think of making and interpreting as part of the same process. In *Interpretation in Architecture*, Adrian Snodgrass and I argued that ***to design*** is ***to interpret***. See What does it all mean?

default watermark

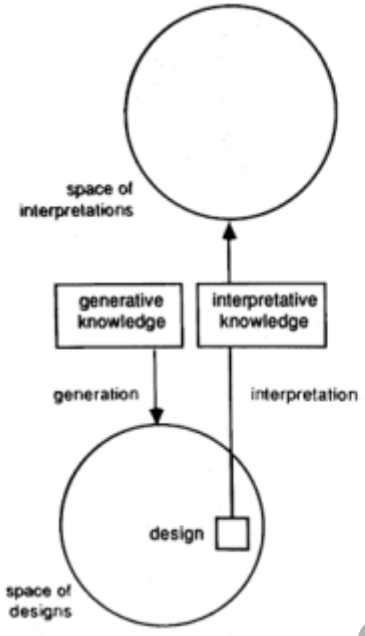
As it happens, whatever you say about *making* can be applied to *interpreting* and vice versa. So there arenâ??t really two types of knowledge. Perhaps there are thousands â?¦ or just one.

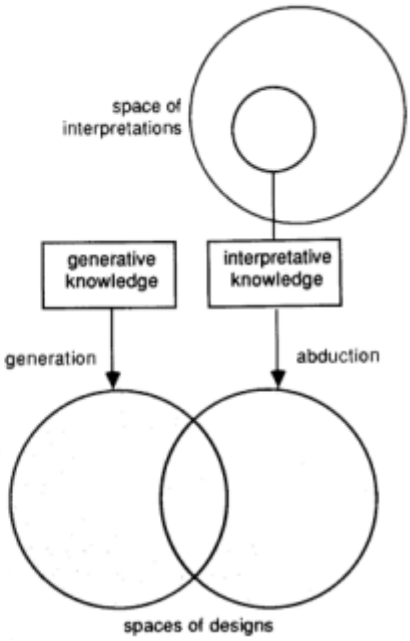
Bibliography

- Coyne, Richard, Michael Rosenman, Anthony Radford, M Balachandran, and John Gero. 1990. *Knowledge-Based Design Systems*. Reading, Mass.: Addison-Wesley.
- Mitchell, William J. 1990. *The Logic of Architecture: Design, Computation, and Cognition*. Cambridge, MA: MIT Press.
- Stiny, George. 1980. Kindergarten grammars: designing with Froebelâ??s building gifts. *Environment and Planning B: Planning and Design*, (7)409-462.
- Vale, Brenda, and Robert Vale. 2011. Lottâ??s Bricks, The Arts and Crafts movement and Arnold Mitchell. *Architectural Research Quarterly*, (15) 2, 119-130.

Notes

- Also see De-generation, Neuroscience eclipses AI, Wicked problems revisited, and Brain scans and creativity.
- In *Knowledge-Based Design Systems* we thought of spaces of designs and spaces of interpretations. The design task is to derive designs that satisfy the constraints implied by both kinds of knowledge. To reason â??backwardsâ?• from functions/meanings to define a space of designs required reasoning about *constraints*, or a kind of reasoning known as *evidential reasoning,* or *abduction*. I now think this model and its variants are outdated and limited. However, such quasi Venn diagrams can take on a surreal cast, which is appealing. See for example Graham Shawcrossâ?? blog post on Visual Thinking.





default watermark

Category

1. Artificial Intelligence

Tags

1. AI
2. artificial intelligence
3. creativity
4. design
5. hermeneutics
6. interpretation
7. knowledge-based systems

Date Created
December 15, 2012
Author
rcoyne99