Learning to Cope with an Open World

Boi Faltings
Artificial Intelligence Laboratory (LIA)
Swiss Federal Institute of Technology (EPFL)
IN-Ecublens
1015 Lausanne, Switzerland
faltings@lia.di.epfl.ch

October 10, 1995

Design as an open-world problem Science has developed very detailed and well-founded theories for *analyzing* the behavior of artifacts. For example, Boeing was able to correctly verify an entirely new airplane, the Boeing 777, before any prototype was even built. However, there are few theories, and no computer systems, that would allow us to *design* structures with a similar degree of automation.

Physical theories are formulated as deductive inferences: given that I drop an object of a certain shape and weight from a certain height, they can predict exactly with what speed the object will hit the ground. On the other hand, design is an *abductive* problem: given that an object should hit the ground at a certain speed, what weight and shape should it have? From what height should it be dropped? This problem has infinitely many answers.

While a computer can very well solve deductive problems, where one answer is computed from initial data with a fixed sequence of steps, it has much more difficulty to solve an abductive problem. The only general algorithm for abduction is to systematically search the space of all possible structures to find one for which deduction shows that it satisfies the specifications. But how can we enumerate all possible object shapes?

The problem is that design, in particular design involving geometric structures, is fundamentally an open-world problem: there will always be structures which the current knowledge is unable to generate. An intelligent design system will never have a static knowledge base, but has to evolve with experience to cover more and more ground. Thus, learning is essential for creating powerful intelligent design systems.

I distinguish two forms of learning in design:

- in learning for *customization*, the goal is to adapt a general system to a specific user, i.e. to provide a more convenient way to integrate his or her knowledge than programming

- in learning for extending *coverage*, the goal is to learn new knowledge that extends the range of designs the program is able to produce.

Learning for customization One aspect of learning in design is that of customization: a general design system is adapted to the specific needs and preferences of a particular user. As an example, the `FAMING` system ([1],[2]) is a program for innovative design of mechanism part shapes. The user can provide cases of earlier devices along with interpretations which indicate the desired functions. Using techniques of qualitative physics, `FAMING` constructs an explanation that links the structure provided in the case to the function given in the interpretation. This explanation is then used to generalize the case into a parameterized prototype. Thus, `FAMING` generalizes examples given by the user using explanation-based learning techniques.

Figure 1 shows an example of using `FAMING`. Here, the designer starts with a familiar device - a ratchet - but intends to use it in a novel way: rather than regulating the motion of the wheel, it is used to drive the wheel by applying force to the lever. This dramatically changes the constraints on the device, requiring learning. `FAMING` uses a theory of qualitative kinematics to *explain* why the ratchet device can be used in this way, and under what constraints the function remains valid. This results in a new prototype mechanism which can be reused in different contexts, for example to design a novel forward-reverse device.

The importance of this learning process in `FAMING` is that in contrast to existing intelligent design aids such as `ICAD`, it makes it very easy for a user to add new knowledge to the system. Where existing technology required detailed programming, `FAMING` can be provided with an individual designer's knowledge by simply giving cases as examples. However, `FAMING` requires a human users's guidance. Could one also imagine systems which *discover* new knowledge by themselves?

Learning for extending coverage Another aspect of learning is to systematically look for missing knowledge by proposing experiments that will lead to new design knowledge. Such systems for *discovery* were pioneered in work on `AM` and `EURISKO` ([3]). This kind of learning is based on the difference between knowledge that can only be used deductively, such as finite-element analysis, and knowledge which can also be used abductively, such as qualitative models. It generalizes knowledge of the first kind to discover new knowledge of the second kind which can then be used in design. This process

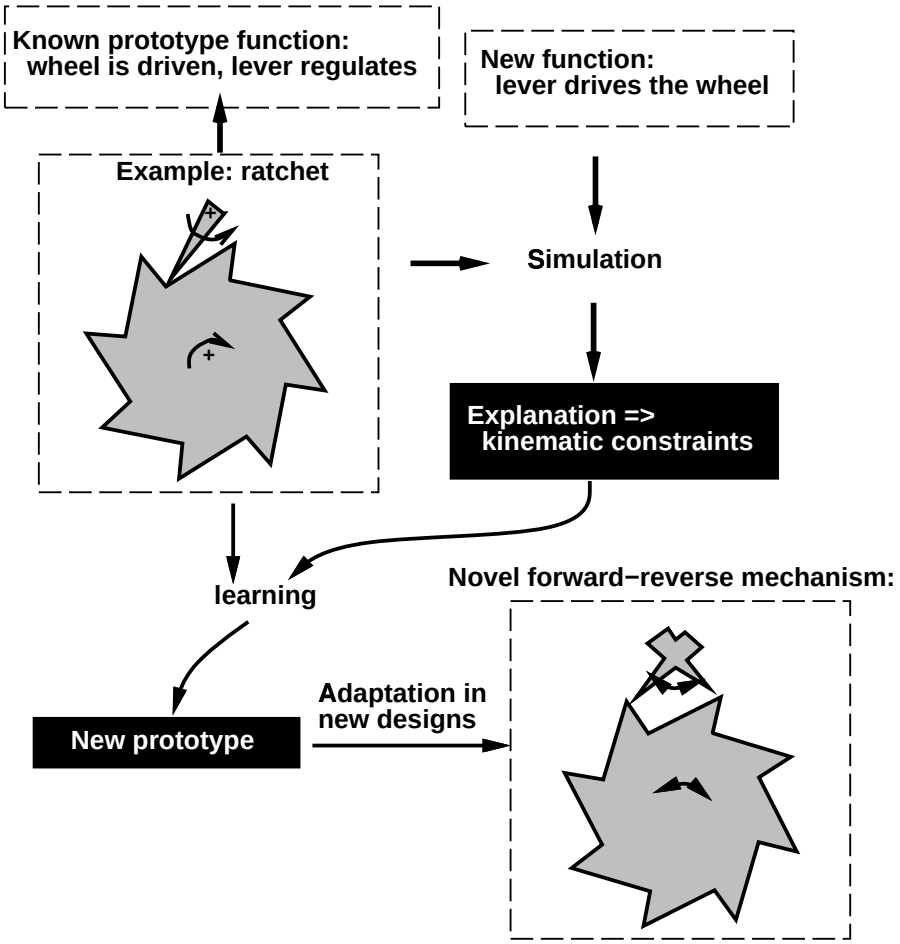

Figure 1: FAMING uses explanation-based learning to make shapes reusable in new contexts.

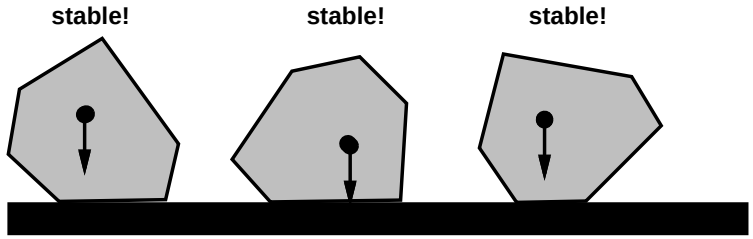

Figure 2: *Stable positions of a polygon-shaped object found by abductive reasoning using a qualitative rule.*

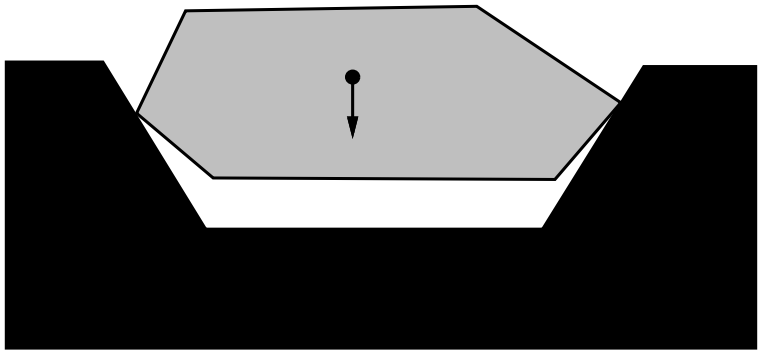

Figure 3: *A new discovery which also turns out to be stable.*

can apply to a single design; learning may lead the system to produce better solutions at each iteration.

In its simplest form, the design system would propose random structures, use its deductive knowledge to analyze them, and keep those whose behavior was interesting and different from earlier design solutions known to the system. The process can be made more efficient by proposing structures which are explicitly different from the known ones.

In [4], I describe an example of such a system which designs stable placements for blocks. It starts with a certain number of qualitative rules for predicting behavior. Only one of these rules predicts a stable placement:

$$\text{support}(x,O) \wedge \text{above}(\text{cg}(O),x) \Rightarrow \text{stable}(O)$$

It states that if polygon O is supported by edge x and the center of gravity cg(O) is above x, then O is stable. This rule can be used in abductive reasoning to generate positions which are known to be stable (Figure 2)

When these prototypes are insufficient to solve a particular problem, the program searches for ways of combining the elements for which the qualitative rules fail to give a concrete prediction.

For example, two rules can be applied to the left and right contact points of the position in Figure 3:

support(x,O) $\wedge$ left-of(cg(O),x) $\Rightarrow$ turn(O, ccw)
support(x,O) $\wedge$ right-of(cg(O),x) $\Rightarrow$ turn(O, cw)

which predict that the object would both turn counterclockwise (ccw) and clockwise (cw). This contradiction means that the rules are insufficiently precise to apply to this situation. A numerical analysis is called to determine the actual behavior - in this case, stability - and a new rule is added to cover this case. This rule is derived from the numerical analysis using explanation-based learning techniques. This new knowledge now enables the system to systematically design positions of the type shown in Figure 3.

Conclusions Design is an open-world problem where knowledge is never complete. Like people, design systems will have to continuously expand their knowledge in order to keep up with the requirements.

Design knowledge is an "inverse" of analysis knowledge in that it can be explained by analyzing the results it produces. In most domains, this analysis knowledge is closed and can be formulated once and for all as a scientific theory. This makes it feasible to automatically construct explanations for designs. The technique of explanation-based learning can then be applied to learn new design knowledge in an automatic and reliable manner.

I have shown two examples of such a process. The first, `FAMING`, applies this process to practical examples and supported invention of novel devices. However, it only learns what a human user already knows. In the second example, I apply explanation-based learning to simpler problems but show how a computer program itself could be capable of independent discovery. I believe that many other useful applications of the explanation-based learning paradigm to design can be found between the two extremes.

References

[1] B. Faltings, K. Sun: "FAMING: Supporting Innovative Mechanism Shape Design," *to appear*, *Computer-Aided Design*, 1995

[2] B. Faltings, K. Sun: "Computer-Aided Creative Mechanism Design," International Joint Conference on Artificial Intelligence, 1995, Video Track

[3] D. Lenat: "The Role of Heuristics in Learning by Discovery: Three Case Studies", Machine Learning: An Artificial Intelligence Approach, Springer, 1984

[4] B. Faltings: Supporting Creativity in Symbolic Computation, 2nd Conference on AI and Creativity, Heron Island, 1992