Expectation-based Learning in Design

Dan L. Grecu, David C. Brown

Department of Computer Science
Worcester Polytechnic Institute
Worcester, MA 01609, USA
dgrecu, dcb@cs.wpi.edu

Design problems typically have a very large number of problem states, many of which cannot be anticipated at the onset of the design. Some design problem states are characterized by as many as hundreds of parameters. Given these amounts of uncertainty and information, AI design systems faced with learning tasks cannot know from the beginning what needs to be learned, and whether these needs will remain the same. In this abstract we describe how LEAD (Learning Expectations in Agent-based Design), a multi-agent system for parametric and configuration design, addresses these challenges in design learning.

LEAD design agents are *specialized* problem-solvers, having knowledge about a specific facet of a design domain. Each design agent also has a *partial* view of what the other agents are doing. As a consequence, during learning agents have to deal with a limited access to relevant information, because they either do not understand all the information, and/or because that information resides in other agents. An agent compensates for these limitations by using 'expectations.' An expectation is an assumption that a *fact* will be true under given *conditions*. For example, the price of a spring can be expected to be in the range (p_1, p_2), if it is made of material *m*, if its diameter is smaller than *d*, and if it is manufactured using technology *t*. In this case the expectation fact is the range of the price, while the material and diameter values, and the manufacturing technology are the conditions for the expectation. Expectations represent empirical knowledge that cannot be inferred in a deductive manner due to limited resources, such as time, or information. Expectations have been used with different meanings in other contexts, such as conditioned learning (Catania 1998), and language understanding (Schank 1982).

Since design problems have large number of constraints that agents are likely to violate, LEAD agents use expectations to evaluate the consequences of possible decisions before committing to one of them. This strategy aims to reduce backtracking and conflicts with other agents, and helps an agent find out whether its decisions will satisfy the design goals. To evaluate the consequences of a decision an agent inferences from facts that describe current values and events in the design system. The values may refer to properties in the design domain, such as material selections, or component dimensions, but they may also refer to what other agents are doing, e.g., whether agent *A* has made a decision before or after agent *B*. In this inferencing process, the agent evaluating decision consequences may also use expectation facts in addition to the facts that are currently true in the system. An expectation fact can be used in inferencing only if the expectation is active, that is, if its conditions are satisfied by facts or other active expectations. Expectations together with the existing facts determine how far ahead an agent can predict the results of its actions, i.e., how informed its decisions are.

In LEAD, expectations are learned when an agent decides that their associated facts would allow the agent to extend the evaluation of decision consequences further into the future. When designing, an agent keeps track of the missing pieces of information that prevent it from adding further conclusions to the chain of consequences of a decision. A design or design process property that has been repeatedly identified as potentially useful in evaluating decision consequences, and which is not available, initiates a learning process to acquire an expectation description. Expectation descriptions are learned as concepts. The concept features are represented by the expectation conditions, and the concept values (i.e. the expectations facts) are derived from the observed values for the piece of information that is needed.

Consider an agent building an expectation for the values of a parameter *P*. Once the agent detects the need for an expectation, the learning follows three phases. The agent first *collects information to be used in describing the expectation.* Based on its knowledge about *P*, such as dependencies and constraints in which the parameter *P* is involved, the agent sets up a set of *candidate* design and design process properties that might influence the values of *P*. During subsequent design sessions the values of these properties are recorded whenever the agent needs *P* for evaluation purposes. The value of *P* resulting from the design process is associated with the values of the candidate properties, and represents a training instance. The agent then *builds the description of the expectation*, by identifying which of the candidate properties that were selected in the first phase are relevant conditions for the expectation-concept. LEAD uses wrappers (Kohavi & John 1998) to determine relevant subsets of features for a concept and to generate a concept description. Finally, once an agent has acquired an expectation description, *the description remains open to revision.* Whenever the agent uses the expectation, it verifies whether the value predicted by the expectation matches the value that results in the design process. Mismatches are recorded, and used for retraining and updating of the expectation description.

References

Catania, A.C. *Learning*, Prentice Hall, Upper River Springs, NJ, 1998.

Kohavi, R., and John, G.H. 1998. "Wrappers for Feature Subset Selection," *Artificial Intelligence*, 97: 273-324.

Schank, R. 1982. *Dynamic memory*, Cambridge University Press, Cambridge, England.