2006-894: ENGINEERING PROBLEM-SOLVING USING PATTERN MATCHING: A NEW COURSE

Murray Teitell, DeVry University-Long Beach

Engineering Problem-Solving Using Pattern Matching: A New Course

Methods of Problem-solving

There are a number of ways to solve a problem. [1, 2] Engineers, however, tend to learn and use a limited set of structured methods.[3] The most popular is the deductive method [2]: analyze the problem and apply a series of generalized steps that have worked before on similar problems leading to a solution. Conversely, using the inductive method (reverse engineering), the solution is assumed and the engineer works backwards to find the original starting point.[4] If the starting point is found, then the backward steps are implemented to achieve the solution.

Sometimes the steps are well documented and in other cases the details of the steps need to be worked out. Each step is described verbally and visualized in pictures and diagrams. The requirements are specified as to design including materials, measurements, and tolerances. At each step of the solution, the requirements are then implemented. These well-worn methods work best on small problems or when dividing the original problem into a number of small problems (decomposition).

Most engineering curriculums have the students use and practice some variation of the deductive and inductive methods described above. Students invariably have a final project in many courses where they are assigned a design problem. In their senior year, they often have a senior project which entails designing and building a unique prototype. The students are required to submit a proposal documenting the requirements, the proposed design, the materials, the schedule, and the test plan. After review by the professor, the proposal may require changes. Further modifications may be required at any point as unforeseen events occur. In the end the prototype is expected to pass all tests and demonstrate on delivery that it meets the original requirements. Many of these same steps occur on the job when graduate engineers pursue engineering projects, so the project course is justified.

Bigger Problems and a New Course to Train Engineers to Tackle Them

Engineering problems have gotten bigger and more complex. Likewise, driven by the growth in scope of the problems, the technology to handle big problems has blossomed. The capacity of disk drives continues to expand as does the data density: the ability to store more data in a smaller space. Search algorithms are increasing in speed and allow retrieval of more data in less time. Overall storage space and processing is readily available at low cost.

Courses in structured engineering design teach students to analyze existing products and designs (literature search).[3] Students taking more advanced training in Computer-aided Engineering (CAE) learn to merge design and analysis in order to optimize the design.[5] When designing for bounded product lines such as living room furniture or warehouse handling equipment, engineers often use templates that allow them to specify alternative features.[6] However, it is questionable if engineers are trained to fully utilize the new capability: faster, more expanded, and more detailed analysis. Engineers need to be taught to utilize these new methods in their designs. Computers allow for the storage and retrieval of volumes of detailed plans and processes. In designing a new product (system) the engineer should be able to find the closest matching stored plan and then add changes and then apply the process. How do we organize previous designs so they are readily available to the engineer? Below we suggest a model for this organization and a synopsis of a course to train engineers to create it and use it.

The model includes creating a depository for previous designs. Each stored design has pre-conditions. The hierarchical depository is organized around the pre-conditions. When an engineer begins a new design, the pre-conditions (requirements) are the trigger for the search for a closely matching previous design (candidate design). The next step is to fine-tune the candidate design to take care of the differences. Finally there is modification of the pre-conditions so the new design is correctly located in the future. These pre-conditions should be able to suggest features and characteristics that were not in the original specifications—fill in the gaps.

Models of the Brain

A model of the cortex of the human brain suggests that the cortex stores complete solutions to problems.[7] It stores the complete sequence of actions that are used to solve a problem in the past. Every time a person faces the same situation, the brain pulls up a stored sequence that was used successfully in the past and applies it to the current situation. This Cortex model suggests that the brain predicts a situation before it happens and is ready to react based on that prediction. Certain clues provide the trigger. Your boss is coming towards you and you know that you will be asked to work on Saturday. Already you are ready with an excuse why you can't do it. If the prediction is wrong, the brain modifies its model and adjusts it accordingly for the next time the situation appears. For example, the steps in driving a car are so engrained in our brain that a person can drive from one point to another and not remember how they got there. But what happens if an American driver goes to Great Britain. Many of the expectations are wrong and have to be modified. In the case of driving, one trigger is the act of getting behind the wheel.

Simulating the way the brain (Cortex model) solves problems is only one of many models that can be applied to problem-solving. Familiar models include the case-based method, the algorithmic method, the precompiled partial solution method, the decomposition method and a host of others.[8] However, the Cortex model lends itself to the abstraction[7] of designs into classes using a hierarchical organization so they can be easily retrieved by the engineer.

Model for Engineering Problem-solving

The Cortex model can be applied to engineering problem-solving. The technology is available to employ this problem-solving method on a corporate level or on the internet level. Some companies already employ versions of this method. Industry will be able to take greater advantage of this method as the internet expands.

Aerospace and automotive manufacturers store their old designs, manufacturing plans, and testing and maintenance plans and retrieve them when they begin redesigning a new vehicle. Writers pull out old templates to write new articles.

Templates are utilized quite frequently in software engineering.[9] But where are engineers taught to make templates for their designs so they can be applied to the next or similar or a whole set of products. Templates must not only have specs, but designate variables that would change from application to application and pre-conditions that would trigger the use of the template.

The Smoke Alarm Example

Triggers:

What triggers a smoke alarm design? [10, 11] There is a request to design a new smoke alarm with certain pre-conditions. The pre-conditions fall into categories that help determine the repertoire of design templates available. For example there may be four pre-conditions that determine the key templates:

.

Pre-Conditions
Type of detector
Type of location
Type of power
Type of alarm

A suggested format for the template follows: Each template comes with ready-made drawings. There is a parts list. There is a manufacturing plan. There are programs for the assembly line. There is a test set with acceptable outcomes. There is a ready-made user manual with maintenance instructions. With 4 different pre-conditions and a number of choices for each pre-condition, a number of different templates are offered. To generate the varying templates from the pre-conditions requires the engineer to utilize automation (write programs or use ready-made programs).

Higher Order (Abstracted) Template for a Sensor-Alarm System

The smoke alarm template promotes a higher order template for the generic Sensor-Alarm System. This template has the more basic plan for sensor alarm systems that include sensors for carbon monoxide, carbon dioxide, ozone, nitrogen monoxide, nitrogen dioxide, chlorine gas, and other toxic gases.

In the Cortex model [7], the brain stores its representation from the grainiest parts of the representation (solution) up to the classification for the whole. It even suggests that the Cortex is organized this way in layers and that incoming sensory data is matched to the levels up to the top level and then down to lower level to predict the situation. In recognizing someone’s face, the brain absorbs a lot of data about the different parts of the face. This starts with individual blots of light (color) received by the retinal sensors which coalesce into face parts like cheeks, nose, eyes, ear, forehead, hair and then onto the highest abstraction level to make the identification as a person named Rachel. Then the brain begins to predict a lot of other things that it expects of Rachel such as how she walks, how she talks and how she will respond to a certain question.

Borrowing from this model, the design templates are arranged in a hierarchical directory from the more specific to the more generic. When the most generic level is reached the gaps in the design should be filled in automatically according to design rules. Automated database management systems are employed to accomplish this.

Some engineering curriculums include instruction in database design.[12] It is important that the engineering student understand how to organize engineering data. As part of teaching engineers about database design, it could include a course to teach the student how to use a simple database management system like Microsoft Access, how to program in a database language like SQL, and how to design a database system. [12]

Template Course for Engineering Students

The primary purpose of the course is to teach students to create designs from an organized series of templates. The prerequisite for this course is presumed to be a conventional course in engineering design.[3] In the course, the student learns to create a system of templates that are classified by pre-conditions. Starting with his/her own design, the student learns how to make the design a function of a series of variables. The student is instructed on how to store the design function as a kernel of a database. Applying his/her knowledge of programming languages and database management systems, the student is prompted on how to generate a system which includes variations of the design. From the series of variations, the student gains practice in creating a more generic template and in continuing the process to the next hierarchical level until the system consist of all the products that a department, company, or firm will produce. Finally, the student learns how to create a series of operators that perform the transformations of each template into the next template. The course stresses the skills the student needs for the continued use and maintenance of the design template system.

In this course, the design template system is individually each student's own work. It starts with the student creating a design according to the principles learned in a prerequisite course. Any form of plagiarism or academic dishonesty is not tolerated in the course.

It is proposed that the topics and skills as described above can be integrated into one course in the engineering curriculum. This is an important course if the engineer is to be able to face the challenges of creating the competitive and innovative product of the future.

A synopsis of the list of topics for the course is:

1. Design Repertoire
2. Generating a template
3. Identifying the variables
4. Finding the pre-conditions
5. Organizing the template into a database
6. Programming the database to generate new templates
7. Filling in the gaps; predicting what is missing.
8. Generating a higher level generic template
9. Determining the boundaries of the system
10. Final Project

Examples of the variables and their pre-conditions are:

VARIABLES	PRE-CONDITIONS (values)
Shape	Folded newspaper
Size	12l X 9.25w X 4.75h
Material	Polypropylene
Feature	Grommet

Influence on Engineering

Many companies have database systems that store designs. By and large, the database system is enforced on the whole company. But the organization of the templates and the relationship to each other is part of the design. The engineer needs to be able to create his own system of designs. The key to the system is determining all the variables and pre-conditions. The system is cataloged by the pre-conditions.

It is felt that this way of doing engineering will lead to more innovative products, products that meet the requirements of the customers, and new products that adapt to changing conditions and can be created quickly.

References

1. Fensel, D., & Motta, E. (2001). Structured development of problem solving methods. *Transactions on Knowledge and Data Engineering,* 13(6), 913-932.
2. Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering Education,* 78(7), 674-681.
3. Pahl, G., & Beltz, W. (1996). *Engineering design: a systematic approach*. 2nd ed. New York: Springer.
4. Otto, K., & Wood, K. (2000). *Product design: techniques in reverse engineering and new product development*. Upper Saddle River, NJ: Prentice Hall.
5. Peak, R., Fulton R., Nishigaki I., and Okamoto, N. (1998). *Integrating engineering design and analysis using a multi-representational approach*. Engineering with Computers, 14(2), 93-114.
6. Pfefferkorn, C. (1975). *A heuristic problem solving design system for equipment or furniture*. Communications of the ACM, 18(5), 286-297.
7. Hawkins, J., & Blakeslee, S. (2004). *On intelligence*. New York: Henry Holt and Company.
8. Chandrasekaran, B. (1990*). Design problem solving: a task analysis*. AI Magazine, 11(4), 59-71.
9. Pressman, R. (2005). *Software engineering: a practitioner's approach*. 6th ed. New York: McGraw-Hill.
10. System Sensor. (2002). *System smoke detectors.* Applications Guide ed. St. Charles, Il: System Sensor.
11. Grosshandler, W. (1995). *A review of measurements and candidate signatures for early fire detection.* NISTIR 5555 ed. Gaithersburg, MD: National Institute of Standards and Technology.
12. Rob, P., & Coronel, C. (2004). *Database systems: design, implementation & management.* 6th ed. Boston: Thomson Course Technology.