Right Brain Programming

Case for a New Programming Paradigm

Tom Keeley
Compsim LLC

"It was at Caltech that psychobiologist Roger Sperry discovered that the left and right hemispheres of the brain are specialized for different capacities: the left brain for verbal thinking and language; and the right brain for spatial-visual thought. His work has had an incalculable impact in fields ranging from education to behavioral psychology to neurophysiology. Sperry was awarded the Nobel Prize in 1981."[1]

"The left inferior frontal lobe is activated during verbal association tasks where exact calculation is language linked, whereas approximate calculation is not."[2]

``Approximate arithmetic, in contrast, shows no dependence on language ... "[3]

One might suggest that the majority of computer languages are focused on left brain programming. Textual programming languages are defined as text based rules. Rules are defined with language terms like IF, THEN, ELSE, WHILE, DO, CASE, etc. Numeric values are key aspects of these conventional programming languages. The programmer uses structured techniques by building a sequence of instructions that follow a sequential path through the logic to accomplish the objectives of the program. Subroutines and Functions are optimization techniques that allow code segments to be reused. Object-oriented programming encapsulates the sequential instructions and methods in "objects", but they are still following the linear thought processes of the programmer.

Different programming languages have been developed over the years to focus on specific objectives.

- Ease of use
- Targeted solutions (operating systems, database)
- Emulate hardware construction where devices are made up of objects
- Reduction of errors with strict typing

The concept of a graphical programming language has been around for a long time. One of the original reasons was the suggestion that there was a need to provide a mechanism for non-programmers to create the same linear logic. Other approaches for graphical programming have focused on the encapsulation of the logic of physical objects or functions in a graphical context. The intent here is to enable object functionality to be linked graphically. These systems focus on the flow of data through the system. Graphical instrumentation allows the programmer to see how data is transformed at each device. Programmable controllers, in the industrial automation market, have used a graphical programming language called Relay Ladder Logic, which duplicates the hardware implantation of logic constructed of physical relays. This again, is a

representation of logical IF statements (IF the relay is closed, then Turn ON an operation).

Humans use tools to perform functions. An expert is a human whose knowledge and experience allows him/her to use the tools to perform at a superior level. The tools that the experts use come in the form of data or information and mechanical enablers. Computers have been used as tools for humans to manipulate data. They have had limited success in the form of "expert systems", because they have been data manipulators that perform functions without "reasoning".

Another view of an expert is that they exercise "reasoning" skills to understand how to interpret the information available to them and how best to apply their tools to solve problems. It is this "reasoning" that separates the expert from other humans when they have access to the same tools. The expert accomplishes the reasoning function by interpreting the importance of the information available and balancing potential actions to determine the best collective set of actions to address the problem domain. Using this balancing act, the expert can adapt to changes in the situation. As new data is provided, the optional actions are rebalanced.

This balancing act can be demonstrated by looking at a glass of water. Shake the glass and the water is mixed. It quickly rebalances and resumes its stable state. Drop something in the glass and the water rebalances again, but in a somewhat different formation. The experts develop their reasoning skills through training or experience. They construct mental rules for interpreting the importance of information. They use their ability to balance options in order to make judgmental decisions about courses-of-action.

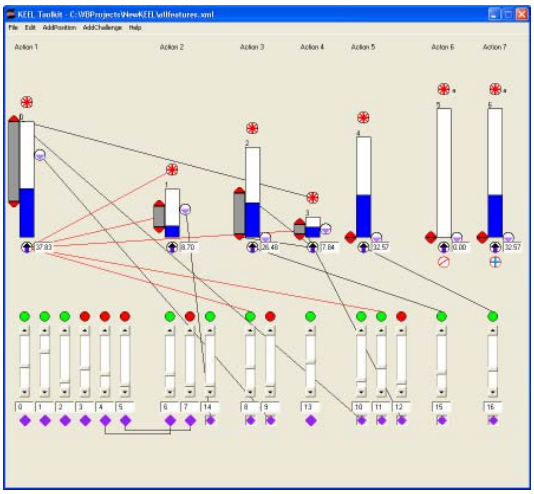
The balancing act performed by the expert (like the glass of water) is an iterative process of interpreting how one piece of information interacts with another, and then observing the results. Only when a stable environment is achieved is the solution complete. If one continually shook the glass of water, it would not stabilize. The same is true for the expert. If the expert was continually asked to interpret constantly changing, random data, the expert would not be able to address the problem.

It would be impractical to write conventional code to define the actions of all molecules in the glass of water to define the path that each would take as the glass was shaken and stabilized. It is also impractical to write conventional code to address the "reasoning" that the expert goes through to interpret the importance of information and balance the system to achieve a stable answer and choose the best set of actions to take. As stated initially, the "approximate arithmetic has no dependence on language".

A new paradigm for documenting human-like reasoning has been implemented as "Knowledge Enhance Electronic Logic" or KEEL®. KEEL focuses on viewing the importance of information and relationships between actions and decisions. It then processes the information and relationships by iteratively evaluating the system until a stable answer is achieved. The answer is a relative set of data items that can be used to

select options or perform relative actions. The source code for KEEL is a set of graphical information items that define importance by size. The graphical source code defines relationships between information items with wires.

Unlike conventional source code, KEEL source code starts executing as soon as it is created. The system begins balancing itself as soon as data items are entered or as inputs are manipulated graphically. This allows the programmer or domain expert to *immediately* see the system "think".



Because KEEL acts as the cognitive engine for systems that are created with conventional programming techniques, the KEEL Toolkit (the development environment for creating KEEL engines), packages the KEEL engine as conventional language routines (C, Java, C#, Visual Basic, VB.NET, Flash, Flash Objects, PLC Structured Text). The system designer is responsible for scheduling when to call the KEEL engine(s) to perform the cognitive process. At the scheduled time, a snapshot of input data is taken and the cognitive function is repeatedly called until a stable system is achieved.

KEEL engines have an extremely small code component. Most of the reasoning logic is maintained in tables. For this reason, the "same code" can solve a very simple problem or very complex problems. The table size for the complex problems would be larger and the time to obtain a stable answer would be longer.

To create a KEEL based system, the domain expert does not spend any time writing any IF, THEN, ELSE, CASE… instructions. There is no textual language used, except to identify data items by name or symbol. The "names" or "symbols" are used only for the programmer's benefit. The KEEL Toolkit will use the names as labels to assist the programmer in keeping track of the data items being used. The domain expert will develop applications by identifying 1) the outputs (actions the system will take), 2) the inputs to the system and 3) the relationships between data items. During the development process the domain expert will be simulating inputs and observing the system rebalance. This is the process of developing the rules that the KEEL engine will use to evaluate real data.

When the design is complete, the KEEL Toolkit can be used to create the conventional source code for the final system.

Problems that are addressed by KEEL technology are just like those addressed by human experts. The experts have one or more goals that they are trying to accomplish. They are

given a set of information and tools to work with. The solutions are obtained by interpreting the information and balancing the alternative courses of action. Relationships between data items are usually non-linear. The systems must handle time and distance aspects to the problem domain as they may be considering immediate issues or more strategic longer term issues. The decisions or actions of a human expert "should be" explainable. The results of KEEL decisions and actions are also completely explainable; perhaps even more so than the decisions of their human counterparts. Since the rules that the KEEL engine processes are completely visible, anyone can see how each data item interacts with others.

KEEL provides a new graphical paradigm for defining cognitive functionality. It can be applied to almost any activity where a human expert makes informed decisions to perform actions. It emulates the right brain function of interpreting varied inter-related pieces of information and balances the information to make judgmental decisions that can be explained. There are no conventional language constructs used in the KEEL source code, only graphical representations of importance and linkage.

There have been some recent studies that suggest that the "left brain / right brain" segmentation (where the left brain focuses on local detailed data and the right brain focuses on global subjective data) is incorrect. Whether true or not, one can still suggest that most conventional computer languages focus on sequential processing of information, while KEEL based solutions focus on balancing information through an iterative process to address objectives.

Compsim LLC is a technology company providing next generation cognitive technology for application in industrial automation, medical, military, governmental, enterprise software and electronic gaming markets. Compsim licenses its KEEL® technology for application in embedded devices and software applications. The website is: http://www.compsim.com.

Compsim LLC
PO Box 532
Brookfield, Wisconsin 53008
(262) 797-0418
Email: info@compsim.com

[1] "Scientific Milestones," Caltech website, http://www.admissions.caltech.edu/only/milestones/.
[2] McCrone, John, "'Right Brain' or 'Left Brain' Myth Or Reality?," The New Scientist, http://www.rense.com/general2/rb.htm.
[3] McCrone, John, "'Right Brain' or 'Left Brain' Myth Or Reality?," The New Scientist, http://www.rense.com/general2/rb.htm.

