Realistic Performance Simulations Using KEEL Technology

Compsim White Paper

Objective:

The desire to simulate real world behavior requires the development of systems capable of representing dynamic, analog, non-linear models that adapt to actions of the people and/or the equipment being tested. This paper proposes the use of Knowledge Enhanced Electronic Logic (KEEL®) Technology to generate and execute this type of behavior. A recognized requirement is that the simulations be repeatable. This paper suggests that this really means that the simulations must be explainable and auditable. This subsequently means that they can be modified and extended with relative ease.

Technical Concept:

The real-world is a dynamic, unstable environment, where every item is in balance with all other items that interact with it. At any point in time the balance may be upset with forces from one direction or another. The physics of the world controls the rate of change as does human response time and movement of information. Modeling the real world using conventional techniques cannot be effectively accomplished. The following list of seven requirements describes the characteristics that must be satisfied:

1. A methodology must be provided that allows a domain expert to define the capabilities of the environment with sufficient granularity so that it can be exactly translated into a form that can be explicitly executed by the target device.

   - The KEEL dynamic graphical language allows non-linear performance to be described as continuous curves that can be executed within simulations as if they were discrete formulas.

2. The methodology for describing the capabilities of the environment must support the efficient development of complex, non-linear scenarios.

   - The KEEL dynamic graphical language provides an easy way to document how information is to be processed without resorting to complex formulas.

3. The simulated functionality must be suitable for embedded real-time operation.

   - KEEL Engines that represent the domain expert's design are high-performance, small-footprint functions.

4. The methodology must be completely understandable so it can be efficiently tested before deployment.

- All functionality is displayed graphically. By tracing the wires one can see instantaneously how different and potentially conflicting data items are interpreted.

5. Target device performance needs to be audited after deployment.

   - Services are provided as part of the KEEL Toolkit such that real world data can animate the graphical language so that decisions and actions can be traced to their cause and justification.

6. The efficiency of the simulation life cycle must be considered (design, test, deploy, audit, extend).

   - The design is interactive, while it is being developed. This allows it to be tested during the development process.   Conventional code is created automatically, thus avoiding human typing errors. Deployment is handled by providing text files in the source code language of choice for easy integration into any IDE (integrated development environment). Auditing is provided with the ability to easily review the real-world interpretation using black box file reviews. Visually observing the importance of information and relationships allows complex scenarios to be reviewed with relative ease. Data is absolute so there is no human interpretation of the results required. Designs can easily be extended, by inserting new items and linking them into existing policies. There is no need to start over every time.

7. The methodology must be architecture independent so it can be deployed on a variety of platforms and in a variety of situations.

   - KEEL Engines are architecture neutral. They are simply cognitive engines created in the programming language of choice. The system architect has complete control over the marshaling of information and the scheduling of policy interpretation and execution.

Operational Concept:

Simulations and electronic training exercises are combinations of scripts that define the boundaries of the domain being tested and functions that describe the performance of the objects in the simulation. The visual and audible content of simulations has been the focus of attention for the last several years. It now provides a realistic view of the world.

The performance of the simulated items (devices, people), however, has not yet achieved the same level of realism. It cannot be achieved by just adding more processing power, because this alone does not address the lifecycle economics of the simulations. For example, if a system had infinite processing power and infinite memory, it may still require an infinite development effort to create the solution. So, while CPU and memory advancements are important, the development and execution models also need to be addressed.

This paper focuses on the analog nature of many of the items that need to be modeled in simulations. The industry has attempted to simulate the analog world by force-fitting the real world into models that satisfy the needs of computers, rather than challenging computers to process information in the analog domain of the real world. To simplify the task for computers, the industry has created a variety of mechanisms to represent approximations of the real world. This commonly means that the world is translated into IF | THEN | ELSE logic of some sort.

The real world could theoretically be modeled with one "big" formula with an almost infinite number of input variables, solving an equation with an almost infinite number of output variables. If that was even possible, then the processor with infinite resources would be required to execute it effectively.

While simulations do not need this infinite capability to be effective, they can benefit from a development and execution model that allows a designer to model the real world by identifying data items and functional relationships between data items without translating the real world to binary representations and writing complex formulas to define these relationships.

To further explain this issue, one might suggest that the act of writing a formula to describe a situation assumes that the author of the formula completely understands the situation and that the author has the capability to describe it in a formula.

An alternative view suggests it may be easier to describe how pieces of information are valued and how they interact. If an approach was available to get immediate feedback on how related information items interacted (without writing conventional code) then one could observe how a simulation would perform as the model is being developed. If it was easy to tune the model to get the appropriate representation, then this might be an alternative way to create the desired effects.

This is the fundamental concept behind KEEL (Knowledge Enhanced Electronic Logic) Technology. The basic idea behind KEEL is that everything is in balance. The forces of nature and human interaction are balanced to control the position and state of everything at an instant of time. For example, at one level an aircraft is balanced in space based forward thrust (lift), gravity and air resistance. At another level it is positioned by a pilot's desire to perform a task that interprets risk and reward (target value, weapons availability, personal risk, risk to equipment, fuel supply, assessment of the airplane status, personal mood, peer review, understanding of the mission…).

To simulate any scenario using KEEL, the domain expert identifies the output control variables that are to be exposed in the simulation. Then the domain expert identifies the inputs (data items) that impact the outputs. Then the domain expert identifies how the inputs impact the outputs. All of this is done graphically, without resorting to conventional coding (without writing formulas in the conventional sense). The "formulas" are derived from the model, not the other way around. Size is a representation of instantaneous importance of information. Wires define explicit functional relationships.

The "formulas" created with KEEL technology are not formulas in the conventional sense. They are packaged as functions (or class methods), depending on the computer source code language used for the simulation application. In KEEL terminology, these cognitive functions are called KEEL Engines. The execution of KEEL Engines is scheduled by the architect of the simulation application. There are no limits to the number of inputs and outputs that can be fused within a single KEEL Engine, although there may be some value in segmenting complex problems into multiple KEEL Engines from a simplicity standpoint.

Complete simulations that incorporate KEEL technology will be combinations of conventional logic and interpretive logic. KEEL provides the interpretive logic. This separation of functionality is similar to discussions of left-brain / right-brain functionality. Using this model, the left brain processes the scripts and the right brain interprets the situation and modifies the scripts. Using this model it is easy to map the concept to human behavioral modeling. If a human is given a task with a straight forward map to accomplish it, the human will follow the map until something unexpected occurs. At that point the human will make a judgmental decision about how to proceed by balancing risk and reward for each of the alternative options available. This judgmental decision will cause the original map (script) to be adjusted. The human will then continue on until the next obstacle is encountered. In some cases the adjustment process is almost continuous as the human is required to continuously adjust or adapt to the environment. Driving a car is an example of this scenario. The driver has a preconceived map (script) of how he/she will get from point A to point B. The driver (according to the script) knows how to start the car, manage the accelerator, brakes, and steering. The script will start according to a plan. But along the way, the driver will adapt to road conditions, obstacles, traffic, etc. Whenever there is an interrupt; something unexpected, the KEEL engine will be called to decide how to respond (modify the script).

Human behavior is not the only type of system that can be modeled using KEEL Technology. Equipment operated by humans can be modeled as can hardware-only systems. When modeling hardware systems it is easy to model equipment failure or equipment degradation or damage. Using this concept the equipment can exhibit almost human-like behavior.

KEEL models are created graphically using simple drag and drop techniques. The KEEL dynamic graphical language allows the designer to get visual feedback on the operation of the models as soon as the simple objects are dropped on the screen. By simulating inputs using simple scroll bars, the user can watch the system re-balance as the information propagates through the system. A KEEL Engine is created as it is defined on the display screen. When the model is complete is can be translated to conventional code for integration into the full simulation. This drag and drop programming technique allows domain experts, rather than software engineers and mathematicians, to create the complex models. The drag and drop technique itself greatly accelerates the development cycle. And by allowing the complete model to be developed solely by the domain expert, the economics of development process are greatly improved.

If applicable, an additional function can be integrated into the final application. This function supports aircraft "black box" functionality that allows the simulation to publish its view of the environment. This XML packet can be read back into the KEEL Toolkit where the simulation can be audited. This type of functionality satisfies the need to audit the authenticity of any simulation that is created. A "close to real time" version allows the KEEL Toolkit to be animated so the designer can "watch the simulation think" while it is in operation, and *trap* the operation when certain actions are taken.

Background (Compsim / KEEL Technology):

Compsim is a technology provider, not a solution provider. It can partner with domain experts to build simulations. In this manner Compsim will help the domain experts understand the KEEL dynamic graphical language so they can produce the simulations. KEEL Technology is proprietary and must be licensed from Compsim. KEEL Technology includes a set of tools which supports the dynamic graphical language, a model for fusing information, and a mechanism (KEEL Engine) for executing the model on a computer based device. KEEL Technology also includes a design mechanism for packaging the KEEL Engine as an analog circuit. This might be appropriate for some simulations where some components need very high performance.

KEEL Technology evolved from a human decision-making model that was implemented on a computer. Human decision-making was the starting point. This is different from other approaches that started with the capabilities of a computer and tried to make the computer function like a human. The result was that KEEL technology allows digital computers to perform as if they were analog computers (in the areas where KEEL technology is applied).

Because the real world is analog and not digital, realistic simulations must exhibit accurate analog relationships. Objects combine to create non-linear forces on other objects. Time and distance dictate non-linear adaptive reallocation of resources and challenge tactical and strategic initiatives. KEEL technology allows these complex relationships to be modeled with relative ease. The resulting KEEL Engines can be deployed into almost any architecture. Once embedded within a simulation, KEEL technology is invisible. It has no user interface, unless it is purposely being used to audit the performance of the simulation.

There are numerous interactive demonstrations, papers and application notes on Compsim's website: http://www.compsim.com that show KEEL Technology in simple simulations. There are also some examples that show the KEEL dynamic graphical language, where KEEL Engines are used to animate a rendering of the language.

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

Compsim LLC
PO Box 532
Brookfield, Wisconsin 53008

(262) 797-0418