Application Note: Application of KEEL Technology in Electronic Games (9/14/2003)

Objective:

Compsim's KEEL technology has the potential to revolutionize the electronic gaming industry. This Application Note is intended to describe how that can happen.

Today most games are scripted by one means or another.
By scripting, we mean that the functionality is commonly driven by IF, THEN, ELSE or CASE statements.

The complexity of the game and its ability to keep the player involved is dependent completely on the author of the game.

While the animatrons can gain and loose skills during the game, the game author is in total control of the situation.

Using KEEL technology, the games will be able to evolve on their own.

The ability to differentiate the games of one company from another company can have a great deal to do with their profitability.

The ability play an electronic game that evolves on its own can drive continued interest in its use, thus reducing development costs required to reach the same revenue stream.

Market:

Published reports indicate that last year the game industry exceeded the motion picture box office revenue, reaching $20B in global revenue. It is projected to grow to over $100B in the next decade.

Victorian computer game industry - taking a large step globally
"Few people would know that in the last 12 months the electronic game industry exceeded motion picture box office revenue for the first time, reaching more than US$20 billion in global revenue for sales of software and hardware.

The potential for growth is enormous, with increased uptake in the Asian market expected to help boost turnover to US$100 billion in the next decade. Even fewer people would be aware that game development companies employ more than 300 people in Victoria or

that the industry earns $A30 million (Australian Dollar) a year in exports.

The latest industry figures from Inform show that the national computer games industry has boomed over the last year, with sales of software and hardware up 43 per cent to $A684 million.

The extent to which the industry has grown was undreamed of 20 years ago, when Melbourne's Alfred Milgrom took Beam Software to the world stage with the legendary game, The Hobbit. Beam built on its initial success to become the first publicly listed game company on the Australian Stock Exchange. Beam Software - now Infogrames Melbourne House - continues to be a force in the lucrative global computer game market. Since then, Victoria has become home to more than half of Australia's game development industry, with companies such as Tantalus Interactive, Infogrames Melbourne House, IR Gurus, Torus Games and Blue Tongue Software all working on big projects for many of the world's leading entertainment software publishers. Running through the computer games developed in Melbourne, one quickly recognises that the city truly is a global player.

Games including Le Mans 24 Hours, AFL Live 2003, South Park Rally, Looney Tunes Space Race and the game based on the hit movie Jurassic Park, have all been developed in Melbourne.

Adam Lancman, Managing Director and CEO of Infogrames Melbourne House, says that Victoria is equal to the best that the US, Japan, and Europe have to offer when it comes to cutting-edge game development.

He says some of the reasons why multinational entertainment software publishers and developers such as Acclaim, Infogrames and THQ have set up Asian regional headquarters in Victoria include access to retail distribution, strength in game development, an excellent education system, a significant local market and easy access to Asian markets. Lancman's view is reiterated by Bruno Bonnell, Chairman and CEO of one of the world's top five computer game companies, Infogrames, "When we decided to extend our development capacity to Asia, placing the regional headquarters in Melbourne was not a difficult decision to make. The culture of our company is imagination and passion, and that is to be seen there. Of all our locations, Melbourne is one of the best, if not the best," he says.

Making sure Melbourne remains a world centre for computer game development is the reason why the Victorian Government has moved quickly to support the industry, says the Victorian Minister for

Information and Communication Technology, Marsha Thomson. "We recognise that computer game development is an emerging industry, which is generating valuable jobs and export revenue for the state," she says. "For example, each new game developed in Victoria represents an investment of up to $A3 million and could create up to 30 additional jobs." Thomson says the computer game industry was also creating new opportunities for other Victorian companies that were developing multimedia products and visual content. "The computer game sector dovetails perfectly with Victoria's other creative industries, sharing the benefits of the computer game industry all round."

The Victorian Government's support of the industry came initially in the form of 'Game Plan: A statement of support for the computer game industry', which was released in 2000. It was followed last year with 'Game Plan: The Next Level', which built on the initial statement by committing the government to a number of initiatives. "'Game Plan: The Next Level' is a blueprint for making the industry in Victoria even stronger by improving infrastructure, growing local businesses and developing skills," Thomson says.

The first of these measures is a program being implemented by the Victorian branch of the Game Developers Association of Australia (GDAA), with Victorian Government funding, to make Sony PlayStation 2 developer kits available at no cost to approved companies. "The lucrative console market is fundamental for the continued success of the Victorian industry. "Promoting IT skills to school students is an investment in the future of Victoria's IT industry, ensuring there will be computer game developers next year and in 10 years' time," he says.

While the computer game industry in Victoria has been positioning itself globally for more than 20 years, awareness of the huge potential of the game market is relatively recent. Continuing to find ways for the industry to take advantage of the global appetite for games is a key challenge both for industry and government.

One of the challenges, identified by Adam Lancman who is also president of the GDAA, is supplementing Victoria's strength in game development with increased capabilities in publishing. "The cost of creating the next generation of games is continually rising, with demands for new graphics, new sound and greater levels of interactivity. "One of the ongoing challenges for us is finding enough capital to work with and that involves finding ways to entice more overseas publishers to invest in the cost of game development here in Australia," he says.

Lancman also says that the support of the Victorian Government is fundamental to the computer game industry's long-term future. "We've got to where we are because of the imagination of the Victorian Government and the vision they've had for the contribution of the game industry to Victoria's economic prosperity," he says. "With support we've been able to increase the visibility of what we are doing in the game industry in Victoria, which is now attracting overseas players keen to utilise the enormous capabilities that already exist here."

Lancman says the foundations for the future growth of the game industry in Australia is well established and Victoria, in particular, is well placed to become one of the world's top three computer game centres.

"Victoria has all the ingredients for what is needed, all we need now is to just make it happen," Lancman says.

FURTHER INFORMATION
The Victorian Computer Game Industry, contact:
Mark Bishop
Manager, ICT Industry Development
Multimedia, Victoria
E-mail: mark.bishop@mmv.vic.gov.au mark.bishop@mmv.vic.gov.au mark.bishop@mmv.vic.gov.au
Phone: (03) 9651 9479
. Victoria is home to more than 50 per cent of Australia's game development industry.
. The Victorian game industry employs more than 300 people.
. National hardware and software sales of $A684 million in 2001-02, up 43 per cent on previous year. (Source: Inform)
. The Victorian game industry earns $A30 million a year in exports.
. Infogrames (France) has its Asian regional headquarters for development and publishing in Melbourne.
. US publisher Acclaim Entertainment has its Asia-Pacific headquarters in Melbourne.
. US publisher THQ has its Asia-Pacific headquarters in Melbourne.
. Electronic Arts (US) has distribution, sales and marketing operations in Victoria.
. Nintendo Australia (Japan) has distribution, sales and marketing operations in Victoria.
. Examples of games developed in Victoria include: Le Mans 24 Hours, South Park Rally, Jurassic Park, Looney Tunes Space Race, Kevin Sheedy's AFL Coach 2002, and Minority Report."

The use of KEEL technology to differentiate a game from its competitors, (even if it impacts only a small segment of the market), can provide a major revenue stream.

Scripting of today's games requires a significant software development effort. The ease of developing complex relationships in the KEEL development environment allows more cost effective game development.

Considerations:

The schedule for evaluating the data
When complex cognitive systems are being created, the designer needs to consider the time sensitivity of input and output data.

In human systems, humans may try to postpone decisions until all the relevant data is available. When pressed, however, humans can react to demands that drive them to make instant decisions. In a KEEL system, the designer may need to have the system "adapt" to changing demands.

The flexibility of the KEEL system design allows for normal conservative operation and still adapts to emergency situations.

The system designer needs to consider when the data should be evaluated. Options could be to trigger the evaluation on a scheduled basis; it could be polled by a higher level authority; it could operate on a change of state from one of its sensors; or it could be continuously running.

Human interaction with the solution
In cases where KEEL engines participate with humans as part of an overall system, consideration should be given to the human interface characteristics of the system.

KEEL engines can trigger the dialog, or the dialog can be initiated by the human.

KEEL engines can provide continuous analog output which can be translated into human readable form or as stimulation to HMI tools.

KEEL engines can accept input from any source, as long as it is translated to the normalized format required by the KEEL engines.

The system designer will consider how and when the dialog between the human and the KEEL engine will be scheduled.

The ability to create complex cognitive solutions in a timely manner

Using the KEEL toolkit, it is possible to develop complex cognitive solutions in a short timeframe.

Software features are built into the toolset that allow the cognitive design to be structured in a manner that makes it easy to integrate the KEEL engine(s) into a variety of architectural models. In many cases, these tools allow the glue logic to be created once and be able to support changes to the cognitive model.

The KEEL-FBD tool allows the staged development of complex models in a manner that automatically creates the glue logic. This is another feature that supports rapid development of complex systems.

Architecture Independence
KEEL engines are discrete information processing engines.

In many cases they will be part of a larger system.

Just like humans in a factory, in an army, or on a team, there is interaction between individuals and between devices and equipment. Computer protocols and network architectures have been developed in a manner that somewhat duplicates the development of human language.

People developed different languages to satisfy different needs: voice for people to communicate verbally; sign language for those that cannot hear; telephone and video conferencing for distance communication.

Computer based protocols have been developed to satisfy different needs: ASCII for simplicity; XML for structure; asynchronous / synchronous / isochronous for flexibility, speed, and time determinism.

KEEL engines can participate with any of these linkages, as long as the data can be normalized to meet the needs of the KEEL engines.

KEEL engines can sit at any point in any architecture. In this way KEEL engines can be integrated into almost any architecture. The simple API enables this valuable attribute.

The evolution of the KEEL engine over time
KEEL engines are often developed over time. New pieces of information are added to the design when it is apparent that they contribute to the interpretation of the data provided.

Compsim's KEEL Toolkit provides a number of services to enable enhancements to the engine over time without impacting the glue logic. This saves software development effort.

The KEEL FBD tools allow a complex system to be developed in stages and integrated and tuned as separate components.

The KEEL toolkit incorporates the idea of merge-able objects or decision-making modules.

There are other cases where the model doesn't need to expand, but only needs to be tuned when relationships between information change.

KEEL engines can be created as "classes" in some languages.

The "importance" of information
A key aspect of any KEEL based system is the dynamically changing importance of information.

In simple systems, there may be no changing importance of information. These types of systems might be built with hard coded solutions or discrete logic.

Cognitive solutions, however, usually deal with complex relationships where information is being interpreted in different ways; in different parts of the same problem.

Cognitive problems often deal with both strategic (future) and tactical (now) problems at the same time. Questions about what to do now and not destroy future opportunities are often addressed. This requires a balancing of information to obtain the best outcome.

Diagnostics and prognostics require that a system adapts its operation to performance variables. If you have difficulty breathing, then breathing becomes more important.

The value of "explainable actions"
Many systems can benefit from an engine that creates explainable decisions and actions.

This allows the systems to be audited for their performance in order to tune them over time.

It is possible to have KEEL engines monitor other KEEL engines and potentially provide a feedback mechanism to achieve optimal operation.

There are other cases where there are demands for the creation of explainable actions. This insures that a code of ethics is integrated into the design. Without the ability to decompose every action and every

decision, there is the ability to create a system that does not meet the needs of society.

So, KEEL based systems, because they are rule based, can explain why any action was taken or decision was made.

Even though the rules are defined graphically, by inserting a snapshot of the inputs back into the design, the reasoning can be displayed.

Who and how the KEEL actions are monitored
KEEL based systems offer several methods for monitoring their performance.

First, because all KEEL actions are visible and explainable in the development environment, they are available for analysis within the tool environment.

Second, because a KEEL engine is a rule based system, it will always respond the same way to an input. An XML schema exists that defines the format for an XML file produced by the real-world device or software applications. If the device or software application logs the input data in this format, it can be read back into the KEEL Toolkit where the reasoning can be reviewed and explained.

The system designer has the responsibility for determining when and how the data is logged and reviewed.

Certain applications may demand more auditing to insure that performance is satisfactory.

It is also likely that new data will enter the application space. This may trigger reviews of performance.

As in human systems, novice operators or players may require closer review than experienced operators or players. The same is true with a KEEL engine.

Consideration for how the KEEL engine(s) fits in the "chain of command"
In some cases KEEL engines will participate as components of a larger system.

They can perform administrative roles by interpreting information in a consistent manner and responding with consistent command decisions.


They can perform subservient roles by accepting direction from above and adapting the commands to modify their actions.

They can sit in the middle of a chain, by accepting commands from above and reviewing status from below. They can modify their own strategy according to the rules provided to them and the information they observe on their own.

They can make requests to humans and devices above them in the chain-of-command, and can deliver commands to humans and devices below. They can collaborate with their peers according to the rules that dictate responsibility.

Should it be appropriate, the KEEL engines can develop their own levels of trust in collaborative environments. The system designer will determine the communication protocols and the flexibility of the system.

The level of "trust" attached to input information
Many judgmental decisions are made by including a level of trust to validate the information. When the level of trust is diminished, then the information may be given a lower level of importance in the overall solution.

KEEL engines can include the level of "trust" as an input to the system. How this is interpreted, is the responsibility of the system designer.

The concept of risk associated with the decisions and actions associated with the system
Many cognitive decisions need to incorporate risk into the decision-making model.

Risk can be included as an input to a KEEL engine.

It is the responsibility of the system designer to determine how risk participates in the decision-making model.

How time and space relationships might contribute to the solution
Time and space often impact the importance of information when making cognitive decisions. KEEL supports these concepts with its "clipper" features.

This allows decisions and actions to be tuned for different times and locations.

In cases where an optimal solution is being targeted, such as the time to send a message or the time to shoot at a moving target, then tools to

support these kinds of decisions are required. They are built into the KEEL toolkit and they are available to the system designer.

Normalizing the data
KEEL engines normally operate on normalized data (0 to 100) values that can be either integer or floating point, as defined by the KEEL project. Any normalization scheme can be used.

While this suggests a linear range of normalized input data, the inputs can drive a curve which allows the data to be interpreted in almost any way.

In this manner a single normalized input value can be interpreted according to any number of independent curve relationships.

Much of the development work in architecting a KEEL solution is spent defining the relationships between information. Because this is all done graphically, there is no need to write "code" to see the results of the analysis.

The responsibility for the overall system remains with the solution architect
The system architect is responsible for the overall architecture of the system.

This will include segmentation of the system, determining when and how the KEEL engines will be scheduled.

The cognitive model for interpreting the input data and causing decisions and actions to be promoted is also the responsibility of the system designer.

System architecture
The system architecture is the definition of the relationships between all system components.

The cognitive segment is usually just part of the system. The system architecture defines the layout for performance, flexibility, extensibility, cost, resources, etc.

The system architecture is often the result of a balancing act: balancing time to market, resources, performance, and cost. The system architecture is driven with an objective where the features are defined. The objective is addressed by identifying potential solutions: selection of components and methods of tying them together.

KEEL technology can be integrated into the architecture from the beginning, or it can be an "add-on". Because individual KEEL engines are

architecture neutral, they can be integrated into an overall architecture at different times; even after a program is completed.

The potential for autonomous operation
Because KEEL engines can interpret information in a human-like manner, and direct relative actions to be taken based on that interpretation, KEEL based systems have the potential to operate without human intervention.

Alternatively, KEEL based systems can operate as either backups to human operators, giving advice or recommendations, or they can provide the primary decision-making engines that are backed up by humans.

Performance considerations
It would be possible to have every object in a game be driven by a KEEL engine. Every object would interact with its surroundings and have its own personality. All objects could constantly be "thinking".

This would create an excessive load on any system.

Different applications require different levels of performance for the KEEL engines.

Some real time control applications may require that information is constantly being evaluated and operated upon.

There will be other times when information changes relatively slowly and therefore does not merit constant evaluation.

There are a variety of techniques to balance performance and system complexity when creating KEEL based solutions.

Assign KEEL engines only to key objects.
In this way the dynamics of cognitive technology would be applied to the objects that are the most important.

Trigger the KEEL thinking process only on certain events.
This is similar to how humans react to change. They think and then they react. Reactions might be scripted. Which action to perform and how to perform it might be KEEL based.

Then as new events happen, small KEEL engines may determine whether to think on bigger problems or not.

Build levels of KEEL engine, thinking of small problems and then trigger more in-depth considerations if necessary.

This is similar to the way humans react. They spend little time considering relatively unimportant items and then spend much more time on significant issues.

Trigger KEEL engines based on a state machine
In this manner the KEEL engines would be triggered to turn on and off as the conditions of the complete system changed.

Distribute KEEL engines across multiple processors
Some applications may demand the full processing power of a microprocessor. It may be appropriate to distribute KEEL engines to multiple computing engines.

Use a multithreaded process with KEEL engines assigned to different priorities and different threads
By assigning KEEL engines to different processing threads (available in some systems), the KEEL engines demanding higher performance can be given a higher priority.

Utilize hardware implementations of the KEEL engines
Analog logic implementations of KEEL engines can likely provide the highest performance.

Complete digital implementations implemented as custom ASICs could also provide enhanced performance.

New digital circuitry techniques with extremely high bandwidths could also be used.

The sources of input data

Sensors
Sensors of all types can be used as inputs to a KEEL system. As long as the information can be transformed into one or more normalized input values, it can be interpreted by a KEEL engine.

Sensors exist to detect and measure almost all physical states. For example: time, temperature, pressure, torque, speed, acceleration, distance, density, color, edges, shapes, counts, volume, etc. There are probably sensors to measure anything for which a value can be assigned to it.

Collections of sensors can also detect and measure non-physical information: stress, truthfulness, pain. These values are determined with some algorithm that synthesizes the information.

Human operator

In systems where the human operator is part of the system there is the potential that the operator will be providing input data to the system.

For example, the operator could be supplying input data to the system as part of the job function. The operator could be reading values or recording physical observations about characteristics of the problem domain. This could be a doctor that records physical symptoms of the patient or of the environment that may contribute to the symptoms.

In other cases, the operator could be directed to take specific measurements. An example might be an automotive service technician that could be directed to take readings in an automotive electrical system to try and isolate the problem.

Inputs from human operators are commonly gathered through some kind of Human Interface Device that will transform the information from human terms to formats more conducive to digital processing. This might be via a keypad, a pushbutton, or in some cases it might be voice input. It could also be in some form of visual form where information is generated by physical movement. It is possible that any of the human senses could trigger inputs to a KEEL system.

Databases

Databases are used to store historic and synthesized data. This data can be manipulated by any number of mathematical processes to provide running averages, identify trends, detect shifts, etc.

The result of database queries can generate numeric information that can be used as inputs to KEEL engines.

Databases that are constantly updated have the ability to send evolving data to KEEL engines and thus tune the KEEL engines with new data.

External Data Sources

In addition to databases, external data sources can be any device or software application that generates information.

For example, machine tools may have counters embedded in them that count completed operations or completed orders. This information is gathered as the equipment operates and can provide input information to KEEL engines.

A clock or calendar is another example of an external device that can generate information for a KEEL engine. These devices generate time related information.

A communication network might generate traffic information.

Other KEEL Engines
KEEL engines may be components of a larger cognitive system. In these cases it is likely that one KEEL engine will feed other KEEL engines.

The KEEL FBD tool provides a mechanism for integrating multiple KEEL engines in a single application.

A more loosely coupled solution would be to connect KEEL engines across a network or some other connectivity approach.

Locally accumulated data
A KEEL engine will be embedded in a device or software application.

It is likely that the device or software application will be performing functions in addition to the cognitive process associated with the KEEL engine. In these cases other values generated by the application may be used as inputs to the KEEL engine. Certainly diagnostic and prognostic data generated by a device might be used to drive a KEEL engine.

Preprocessed data
Preprocessed data can exist anywhere in a system. This preprocessed data could have gone through a validation process or a transformation process. It could carry with it confidence data or some other biasing information that could be used in conjunction with the preprocessed data. It could be accumulated locally or it could exist anywhere in the system where it could be move to the KEEL engine for interpretation and processing.

Control Signals from Other Devices
Control Signals from pieces of equipment or software applications can be used as inputs to KEEL engines.

In some cases KEEL engines are part of autonomous devices. They react to their surroundings and decide what to do for themselves. In some of these cases, the KEEL engine could intercept control signals from another device that are directed to perform operations for that other device. This information could provide intelligence for the device containing the KEEL engine.

The outputs from the system

KEEL could generate control signals.

KEEL engines interpret information and provide outputs that represent a balancing of the inputs. These values can be used to generate complex commands in the form of control signals to other equipment.

KEEL could generate information for other KEEL engines.
A common practice is to segment a system into multiple KEEL engines. It is likely that one KEEL engine will provide data to the input(s) of other KEEL engines.

The KEEL FBD tool assists in connecting KEEL engines into a single solution.

KEEL engines could also be distributed across a network or in multiple tasks where messaging or data sharing could provide the mechanism for one KEEL engine to feed others.

KEEL could provide inputs to other systems (non-KEEL).
The outputs from a KEEL engine could be supplied to other non-KEEL subsystems for further processing.

KEEL outputs could be part of a local feedback loop.
KEEL engines can be part of a feedback loop, where the output of the system is connected back to the input through some other circuitry.

KEEL outputs could be part of a distributed feedback loop.
KEEL outputs could be fed to other external devices which, in turn, feed data back to the input of the KEEL engine. The other devices could be local or remote to the KEEL engine.

Warning messages could be triggered.
The outputs from the KEEL engine could be used to trigger warning messages. The warning messages could use other outputs to describe the warning in relative terms.

Information messages could be triggered to indicate status. Analog values could be included to explain subjective interpretation.
KEEL engines can used to supply variable information in the form of informational messages. Complex messages could be structured from multiple variable output signals.

Commands to operators could be generated.
KEEL engines interpret information and provide outputs that represent a balancing of the inputs. These values can be used to generate complex commands to an operator.

Logging of information could be triggered. KEEL could log its own decisions or it could log other inputs and outputs.
The outputs from KEEL engines could be logged for historical records or for audits.

The inputs to the system could also be logged. If the log format is in XML compliant with the KEEL Input Schema, then the data could be used in the development environment to recreate the decision-making model for exact interpretation.

KEEL could cause state changes of the system to take place based on subjective evaluations.
KEEL engines interpret input information and drive outputs. These outputs could drive an external state machine that could cause the equipment holding the KEEL engine (or any other system with or without the KEEL engine) to change state. In this case, the KEEL engine is supplying inputs to the state machine.

KEEL could generate diagnostic interpretations.
Beyond just generating processed / interpreted information, KEEL engines can interpret diagnostic information and explain the interpretation in detail.

In addition to explaining the interpretation, it can provide the information to explain why other diagnostic interpretations are not considered.

Connectivity

Directly wired to source
In its simplest form a sensor can be directly wired to an input pin on a microprocessor where the signal is transformed to a normalized data format used by the KEEL engine.

The same is true for the output. In its simplest form the normalized data output from the KEEL engine is transformed before sending it out a pin on the microprocessor where a wire carries the signal to an actuator (control point).

A direct wire is the simplest form of network.

Network connected - any topology
Connectivity to and from KEEL engines can be accomplished with any type of network with any topology.

Some wired networks might be termed point to point, multi-drop, token passing, star, web, loop, trunk, etc.

Messaging techniques might include: Broadcast, Store and Forward, Directly Addressable, Group Addressable, All-call.

Any message packaging technique might be used: structured or unstructured, packed, block mode, etc.

Any character encoding can be used: ASCII, Async, Bisync, Isochronous, or any other.

The data can be encrypted or non-encrypted.

The choice of network connectivity is left to the system designer.

Infrared connection
The KEEL engine is not restricted to any specific connectivity to its inputs and outputs. Infrared links can be used.

Supplied by the same processor running the KEEL engine
Because some data sources and data sinks for KEEL engines will be within the same microprocessor, input and output data can be generated and consumed locally.

Radio Frequency
KEEL engines are independent of the connectivity between inputs and outputs and the KEEL engine. Radio frequency connectivity is appropriate for some applications where wired and infrared connections are not appropriate.

Description:

Determine where KEEL engines might be located in the system and what information will be exchanged

Evaluate KEEL segmentation

Define the expectations (outputs) from the system

KEEL outputs could be part of a local feedback loop.

KEEL outputs could be part of a distributed feedback loop.

Warning messages could be triggered.

Information messages could be triggered to indicate status. Analog values could be included to explain subjective interpretation.

Commands to operators could be generated.

Logging of information could be triggered. KEEL could log its own decisions or it could log other inputs and outputs.

KEEL could cause state changes of the system to take place based on subjective evaluations.

KEEL could generate diagnostic interpretations.

KEEL could generate control signals.

KEEL could generate information for other KEEL engines.

KEEL could provide inputs to other systems (non-KEEL).

Identify the sources of information

Human operator

Sensor

Database

Preprocessed data

External Data Sources

Locally accumulated data

Other KEEL Engines

In addition to normal inputs and outputs, model of how soft drivers like emotion will be integrated into the solution
The designer will be able to model how the animatron will react to different stimuli and this reaction will impact other activities.

Plan for staged introduction

KEEL General:

Judgmental decisions by trained operators are potentially "tricked" into overlooking critical attributes
There are some applications where there is an attempt to trick humans.

Magic shows attempt to redirect attention while the trick is performed.

Some markets like airport security target people that may try to trick the system. A terrorist is not going to advertise he is a terrorist.

When people are trained for airport security, they are trained to look for specific attributes. These are the rules.

KEEL engines are rule based systems. The rules identify the trigger points.

While the perfect airport security agent will never miss a critical attribute, human nature makes them susceptible to structured trickery.

A KEEL engine will perform according to its rules.

Human experts take too long to make judgmental decisions
Human experts are trained to perform a task. This works well when the task operates in normal ways.

However, when humans are "surprised", it takes them some amount of time to react.

Adrenaline is used to supercharge the human in special circumstances. This accelerates their response mechanism and makes them more aware of their surroundings.

Even with adrenaline, however, reaction time is based on the complexity of the task.

Computers running structured code react to emergencies (events) in a structured way.

Computers, however, cannot watch out for everything. The code to develop this type of system would be too complex.

The adaptable nature of a KEEL engine may provide a middle ground: faster than a human and not so complex that one would have to write too much code to respond.

The dynamic nature of a KEEL engine enables it to handle complex situations in adaptable ways.

Applications where the judgmental decisions must be explained
While not all decisions or actions need to be explained, there are a number of reasons for having an explainable system.

First, it is difficult to enhance a system when you don't know why it is doing what it is doing. With a KEEL system, if one takes a snapshot of the

critical inputs, those inputs can be loaded into a KEEL engine where the reasoning can be investigated.

Some systems need an explainable solution for legal reasons.

A system that creates explainable actions may be more marketable and may be more likely to receive market acceptance.

Complex situations where it is not economical to develop and maintain straight line code (IF, THEN, ELSE)
It is very expensive to develop conventional software. Complex software is prone to logic and typing errors.

Large complex systems are brittle and break with unexpected, untested situations.

KEEL systems allow the development of complex systems without writing code. The decision-making model is developed graphically and the code is created automatically. The model can be tested from within the toolkit, before it is translated to source code.

The same cognitive model can be translated into multiple languages at the click of a button.

Situations where there is an advantage to be able to create one design and execute it on multiple platforms: device, software simulation, web
In some circumstances there is a large value for being able to demonstrate the same model in a device, in a simulator, and on a web page.

Using the KEEL Toolkit, the same design can be implemented in C, C#, VB, VB.NET, Flash, Java, PLC Structured Text.

When it is important to demonstrate the cognitive technology in multiple environments, KEEL can provide a significant advantage.

Situations where the environment is dynamic and the importance of information changes; the system must react to change
A key attribute of KEEL systems is that the dynamic importance of information is built into the KEEL technology.

KEEL was designed to accept and adapt to the changing importance of information.

Information changes in importance when it is being evaluated from different time and space aspects. (For example, the temperature is

important because you are close to the source. It is not important if you are far away from the source.) What is NOW is more important than what is FAR IN THE FUTURE, if you are thinking about a problem now. It is less important now if you are addressing a strategic problem.

Where architectural issues may prohibit other solutions (KEEL technology is architecture independent: localized, distributed, web based, multiprocessor, etc.)
KEEL engines are implemented as two or three subroutines in any of a variety of languages.

One subroutine is called to initialize the arrays with data.

The second subroutine is scheduled when appropriate. Before this routine is called, the KEEL arrays are loaded with input data. Then the second routine is called repetitively until a flag is set indicating that a stable system has been determined. After that, the outputs from the system are distributed to the appropriate places.

This simple API and small subroutine set allows KEEL engines to be integrated into a variety of system architectures.

KEEL engines don't care where the input data come from or where the outputs are distributed.

This makes the KEEL engines appropriate for almost any system configuration.

The architecture selection is left to the system architect.

Human experts are required to interpret information to make the best decisions or take the most appropriate actions
KEEL engines interpret information in a human-like manner. In this light, KEEL technology is an expert system that makes decisions or takes actions using rules established by the designer.

Like human decision-making, KEEL engines utilize a linkage of information to solve problems. An input may impact a number of problem domains; each in a different way. A single problem domain may impact a number of other problem domains in a variety of ways.

The linkages defined for the KEEL engine can be compared to the various ways that a human might link pieces of information.

The KEEL toolkit provides a unique way to document the human decision-making model for specific applications.

Devices must operate autonomously and make judgmental decisions on their own
Each KEEL engine has its set of inputs and outputs.

When the KEEL engine is scheduled to evaluate the inputs, it iterates until a stable answer to all of its problems is achieved.

Each KEEL engine is a stand-alone function. If a complete system is incorporated into a single engine, then a complete stand-alone solution is developed.

A system can be built with one engine or it can be built with several engines that are integrated into a single autonomous solution by using the KEEL FBD tool. Services provided within the KEEL Toolkit support the software engineers' need to compile multiple engines with a single compile.

KEEL technology has been developed to specifically address the needs of adding cognitive technology to devices in order to provide autonomous operation.

KEEL technology supports addressing the judgmental decisions required when there is a need to make subjective decisions in a human-like manner.

Devices can make control decisions when human operators are not present
Because KEEL engines can operate autonomously, they are able to provide backup to human operators when the human operators are unavailable.

KEEL engines can make the same judgmental decisions as their human operators. These judgmental decisions are appropriate for making control decisions.

These control decisions can be binary (on/off) or relative (analog). Just like a human operator, a KEEL system can take observed data (inputs) and apply cognitive judgmental rules that are defined graphically and make the control decisions.

When the small memory footprint of a KEEL engine is an advantage
KEEL technology is implemented in two or three small subroutines. The size of the subroutines does NOT expand if the system increases in size or complexity.

The remainder of the system is made up of tables. These take much less space than "code".

This allows KEEL engines to run in 8 bit microprocessors and above.

This is important in many "device" applications.

Repetitive judgmental decisions are prone to error
Some human activities or occupations require that those humans are trained to observe and react to situations. This could be an airport security guard or it could be a medical diagnostician.

The airport security guard is asked to look for things out of the ordinary. This triggers increased scrutiny of passengers and of luggage.

The medical diagnostician is trained to look at diagnostic tests.

Both of these activities are repetitive. There are potentially large amounts of data (characteristics) to observe.

The decisions or observations by the reviewer are prone to error, just by the number of repetitive judgments that are made.

The airport guard that hasn't identified a problem passenger in five years may get passive as time goes by.

The medical diagnostician, that never finds a disease, may miss the one critical test that is pertinent in a particular diagnosis.

Summary:

Animatrons that can think and react on their own without scripted functionality can cause a paradigm shift in the electronic gaming industry.

Animatrons that can think and react on their own without the biases of the game authors, can create situations not conceived by their human creators.

The commercial bridge between the gaming industry and the simulation industry can open the doors for new ventures.

By having KEEL technology in the toolkit for electronic game developers, the opportunity for increased revenues can be stimulated.

Disclaimer

This application note suggests the potential for KEEL technology to respond to certain market needs. The end users are totally responsible for assuring that the technology performs as expected.

The application note may also assume that certain external technology exists to support the KEEL engine in an effective manner. This may or may not be accurate in all cases.