



## **Application Note: Resource Management (9/30/2003)**

This application note addresses a variety of topics under the umbrella of "Resource Management".

More specifically it addresses people-based systems, network systems, and computer based systems. All of these systems deal with the management of resources, whether they are people resources, network resources, or processing resources.

### **Objective:**

**The management of resources, whether they are people resources, network resources, or computing resources, can be administered by supervisory personnel.**

In human systems, supervisors or managers apply people to tasks to satisfy the needs of the organizations. They apply those people based on a variety of reasons: the skill level of the people, the type of tasks, the importance of the tasks compared to the tasks the people were previously performing, other options for responding to the business need, etc.

In network systems, the network administrator may open and close ports, add or remove network connections, add / remove and authorize users, tighten or loosen security measures, etc. The overall goal would be to maximize throughput while limiting cost and security loopholes.

In computer systems the system manager will allocate computing resources to respond to the needs of the customers.

In semiconductor systems, the resource manager may be designed into the chip set. Like its human counterpart, it will be responsible for allocating functionality according to priority or need.

**The general function of a Resource Manager will be to optimize performance using all of the resources under "his/her" control.**

This is commonly a balancing act where there are likely to be more demands for service than the ability to provide the service.

This balancing act will require judgmental decisions to be made. The value versus risk balance will be triggered at different times. In human systems, the reallocation of resources will happen at a relatively slow rate (or the people will become frustrated and nothing will ever get done). In network management and in semiconductor systems, the reallocation of resources may happen quite often.



In either case, the systems may benefit from human-like reasoning to insure that the best solution is provided on a continuous basis.

Even in human-based systems, where resources are directed primarily by humans (managers and supervisors), there may be a benefit from having the administration monitored for best practices.

**KEEL technology has the potential to incorporate human-like reasoning in devices and software applications.**

Because KEEL technology has the potential to make the same judgmental decisions as humans, it could be applied to resource management tasks, whether they are human-based systems, network systems, computer systems, or semiconductor systems.

**Software implementations of KEEL technology are likely to be fast enough for resource management of human systems.**

**The management of network, computer, and semiconductor resources may require the higher performance of hardware implementations of KEEL technology.**

**Because resource management is likely to be a continuously evolving task, the dynamic rules that define a KEEL based solution will allow the systems to adapt to changing requirements.**

**The need to provide an audit trail of the management function can also be addressed with KEEL technology as all decisions and actions can be directly explained using the KEEL toolkit.**

**The Resource Management function can benefit from the ability to use KEEL technology to interpret system needs and provide an explainable relative response.**

**Resource Management functions can also be distributed across loosely coupled systems when there is a hierarchy of control.**

In people systems there are often layers of management, each with a different level of authority. Because KEEL systems can operate autonomously, while still accepting some control from external sources, KEEL engines fit well within the structure of a distributed resource management system. For example: any one KEEL engine can administer the resources under its control. A higher level control function could add or remove resources from the lower level resource manager. A KEEL based system could react to the changing resources under its control and continue to perform.

**Market:**



**Resource Management functions exist in all levels of business and government as well as within "systems" and even within devices.**

**The ability to include human like reasoning in Resource Management functions may allow systems administered by humans to be replaced or supplemented with a computing solution.**

**The ability to add human-like reasoning in Resource Management functions that are currently implemented by hardware and software devices, can allow these systems to more effectively adapt to change.**

**The ultimate goal is to support optimal performance and this cannot always be provided by hard-coded solutions.**

**Companies that implement KEEL based human-like reasoning in their products and systems can potentially demonstrate improved performance, lower downtime and more consistent operation.**

**This could enable them to have increased market share and / or increased margins.**

#### **Business Drivers**

**Reduced Engineering Design Cost & Time**

**Reduced Start-up/Commissioning Cost & Time**

**Optimization of Product Performance/Life to the Task**

**Increased Machine Performance/Speed - Production Rates**

**Increased Differentiation of Their Products**

**Reduced Maintenance Cost & Time**

**Reduced Support Cost & Improved Global Support Responsiveness**

**Increase Machine/Process Uptime (OEE)**

**Reduced Project Risk**

**Increased Market Coverage**

**Improved Internal Efficiency - Reduced Fixed Expense**

**Improved Access to Information**

#### **Buying Factors**



**Ease of Use**

**Time to Market**

**Flexibility**

**Extensibility / Scalability**

**Desire for Special Features**

**Performance**

**Options**

**Single Source for All Products**

**Complex Systems**

**Enterprise Connectivity**

**Trends**

**Product Availability**

**Availability of Support**

**Quality**

**Lowest Product Cost**

**Lowest Installed System Cost**

**Lowest Lifecycle System Cost**

**By-Products**

**Expectations of the Future**

**Indirect (External Influences)**

**Efficiency**

**Reduce or Eliminate Other Costs**

**Form Factor**

**Explainable / Auditable**

**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.

### **Performance of the 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.

#### **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.

#### **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.

**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 "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 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 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.

### **The outputs from the system**

The outputs from a KEEL engine are normalized values between 0 and 100. This information may have to be transformed into other forms for use by the external controls or monitoring equipment.

#### **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 be 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.

**The sources of input data**



KEEL engines can accept inputs from almost any type of data source, as long as the data can be transformed into the normalized data format required by the KEEL engine.

When the input is textual or verbal, it will have to be transformed into the normalized format.

### **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.

### **Connectivity**

The ability to install KEEL engines at any point in an architecture makes this a valuable attribute where components of the system are likely to be distributed across different pieces of equipment in different locations.

KEEL engines can be connected by any media and in any format as long as the data is converted to the normalized format before triggering the KEEL engine to process it.

#### **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.

### **Development Process:**

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

This is where the number of KEEL engines will be identified to satisfy the system need.

This is also where the information model will be defined: What input information is needed at each KEEL engine and how that information will flow through the system.

#### **Define the information hierarchy which includes how and when information is accumulated**

Some systems will fit in a "chain-of-command". The concept of authority needs to be considered in these types of systems.

This will dictate who is a supplier of information and who is a receiver of information.



This is where the concept of trust needs to be included in the model.

The concept of risk may also be considered.

#### **Plan for staged introduction**

In systems that might require multiple KEEL engines, consideration for a staged introduction might be appropriate.

Since KEEL engines operate autonomously, this is primarily an issue that is driven by other parts of the system.

If cognitive technology is to be included in the overall system, then it should be scheduled for introduction when it can be tested with the availability of its inputs and outputs.

#### **Evaluate KEEL segmentation**

KEEL engine segmentation is an issue that is required with distributed systems. It should also be considered with complex cognitive situations.

The KEEL FBD tools provide support for the integration of multiple KEEL engines into a single compile for a localized solution.

There is also the potential for integrating multiple independent segments into a single KEEL engine. In situations where one is dealing with simple cognitive processes, it might be appropriate to include several of them in the same engine.

#### **For each KEEL engine, define the expectations (outputs) from the system**

The outputs of the system can drive external controls, provide information or logging data, drive external logic, provide warnings or triggers, or drive other processing engines (like other KEEL engines) or hard coded logic.

The outputs of the system define what the KEEL engine is trying to accomplish.

#### **For each KEEL engine, identify the sources of information**

The sources of information are the inputs to the system as viewed from the outside of the KEEL engine. They can come from sensors, databases, human operators, other KEEL engines, other control systems, or internal logic (like self diagnostics and prognostics or internal calculations).

#### **KEEL General:**

This section identifies the general benefits that could be derived from a KEEL based solution.



**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:**

Because Resource Management functionality can be applied to a wide variety of situations, any mechanism to improve its performance and reduce costs will be appreciated.

Resource Management can benefit from human-like reasoning as it is incorporated into KEEL technology.



Any company offering KEEL technology in Resource Management tasks can differentiate their offerings from the competition and demand higher prices or consume a larger market share.

### **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.