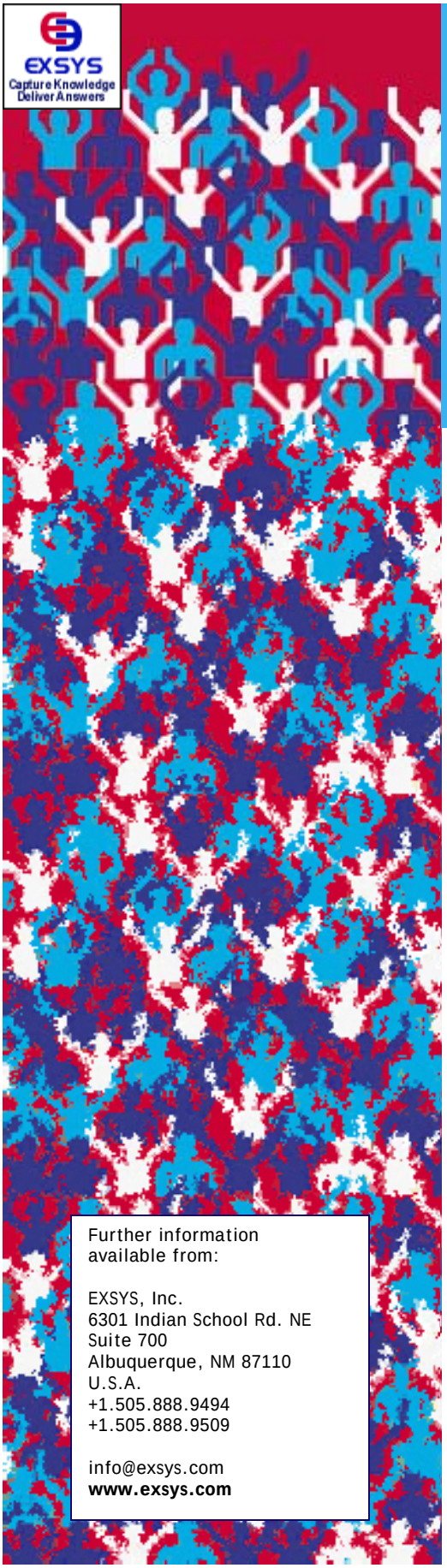


Further information
available from:

EXSYS, Inc.
6301 Indian School Rd. NE
Suite 700
Albuquerque, NM 87110
U.S.A.
+1.505.888.9494
+1.505.888.9509

info@exsys.com
www.exsys.com

Effective Collaboration

Capturing Corporate Knowledge for the Dissemination of Expertise

WHITE PAPER

" Your most precious possession is not your financial assets. Your most precious possession is the people you have working there, and what they carry around in their heads, and their ability to work together."

Robert Reich

Most projects have an effect on many people in an organization, and should be based on input from the people most knowledgeable in each aspect of the project. This applies to everything from arranging new office space to designing a new piece of software. Collaboration is the process of obtaining and coordinating the input from the appropriate people in a systematic way so that nothing important gets left out or overlooked. There are a variety of "Collaboration Tools" that assist in this process by allowing various documents and information to be shared online. This can be as simple as a shared blackboard in a Web-meeting to specialized tools for project planning that provide detailed diagrams of steps and milestones.

Brainstorm or Brain Capture

In most cases, systematic collaboration has been used most effectively for "creative" projects. These types of projects are usually about creating something new with many aspects to choose among, e.g. arrangement of new office space, package design for a new product, or user interface on a new piece of software. While there may be constraints imposed by regulations, budget, or the laws of physics, many options are available. The collaboration process is used to allow the appropriate and involved people to decide which options are most desirable and beneficial. The somewhat freeform nature of many collaboration tools is particularly well suited to this type of brainstorming approach to finding creative new solutions.

Corporate knowledge on a particular subject also tends to be spread across various experts in a company. The reasons and basis for taking particular actions or making particular decisions often involves input from multiple sources.

The current popular term for this is “Business Rules”, but that often limits it to specific procedural steps. Corporate knowledge actually takes in all aspects of how a decision is achieved, including various subjective factors and assessments made by experts, which are often not documented in specific terms. There is a tremendous need to document and preserve this decision-making expertise, and ideally do it in a way that makes it deliverable and usable by others in the company.

Collaboration tools would seem to be an ideal approach to this. They should allow all the appropriate people to contribute their input to describe the overall decision-making process. However, in practice the freeform nature of most collaboration approaches actually gets in the way of effectively describing and capturing useful knowledge. When collaborating on a "Creative" project, having various ad hoc bits of input can be useful. They can lead to new ideas and approaches. The fact that they are not structured generally does not matter. However, when trying to capture an existing decision-making process, ad hoc factors are not really that useful unless they are fully stated in terms of the context of when they apply, when they don’t apply and any special situations that must be considered.

Representing Business Processes

For example, when designing a new office space, the input "It would be good for everyone to have at least 6 power outlets" is useful on its own. It provides a full statement of a need that should be considered in the design. However, when capturing a corporate decision-making process, the statement "We often need to include a Form XYZ" is not particularly useful. It may be true, but does not say when or why the form is included or when it is not needed. To be useful, all the aspects of when the form is needed must be incorporated in some overall systematic description of the decision-making process. Most business processes fall into 3 types of categories (though there is overlap). The representation and documentation of the decision-making process, and related knowledge, is quite different for each category.

1. **Purely procedural**. A series of steps that are performed in a specific way, simply because that is the prescribed way to do them. This is generally done to make sure that all steps are done in a way that can be easily documented and taught. No step should be omitted or changed. While there may have been logical reasons for setting up the procedural steps, they are now executed without any logical reasoning or options.

2. **Simple Logic**. Steps that are still largely procedural, but there is some level of logic used to allow steps to be modified or skipped. The logic is typically "simple", based on If/Then rules with a relatively simple IF condition and a straightforward action in the Then part. For example, when shipping a package, a step that says, "If the shipment is has a value over $100, then insure it for the value."

3. **Knowledge**. These are steps that require considering many factors simultaneously. Often the factors are competing, or must be prioritized and weighted, or ranked probabilistically. This is the sort of knowledge that comes from experience and developed expertise. For the shipping example, when selecting the best carrier for shipping a package, many factors have to be considered - destination, cost, delivery date, performance, size, contents, hazardous materials, etc. All of these have to be combined together to arrive at the best recommendation.

Each category has different issues of representation when using collaboration.

Reaching Consistent Consensus

The key to using collaboration to produce a consensus and agreement on any process is the way in which the decision-making steps and factors are described. Fundamentally, this is the “knowledge representation” scheme used. To make collaboration possible, it is essential that the technique of knowledge representation be both:

1. Appropriate to the problem – capable of fully describing the process
2. Understandable by all collaborators

Purely procedural steps are the easiest to document and collaborate upon. The steps can be fully documented in a simple list of operations to follow. The list contains the full description of the process and is easy to understand.

It can be circulated to the appropriate people who can provide input on the individual items. Collaboration tools can be used to simplify this process. Items can be added, deleted or moved, but it remains a single list of steps. While this does not include any underlying logic and knowledge, it does include all the information needed to perform the operation.

Simple Logic is somewhat more difficult to document, and Knowledge is the most difficult. The degree of difficulty is largely a matter of how complex the decision-making logic actually is. "Simple Logic" can become more and more complex until if evolves into "Knowledge" - with no clear dividing point.

Generally "Simple Logic" operations can be easily diagrammed as a single flow chart or tree logic diagram that does not have large sections of the tree that are repetitive. Such diagrams are widely used and easy to understand. They can be handled with collaboration tools, allowing the tree to be modified or expanded.

Handling the Logic

As long as the logic can be fully described in a single diagram, it can be read and understood by the people collaborating. However, once multiple competing factors, or probabilistic rankings start to be involved, a single tree cannot easily diagram it. It may require multiple tree diagrams, each of which contributes to a final decision. Or it may involve some trees that are used to derive facts used by other trees. Most real-world decisions that have any complexity fall in this category.

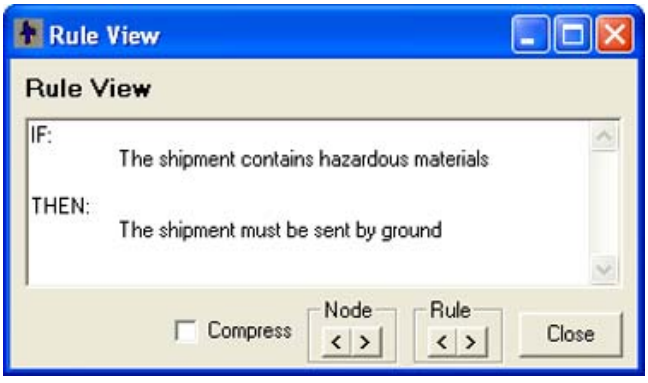
Once a process reaches the point of real knowledge, and there are multiple trees or flow charts that each contribute to a decision, it is quite difficult to use a simple "paper based" representation scheme. Each collaborator may look at the various aspects of the decision differently, and have a different interpretation of the same documents. This is typically where collaboration on capturing high-level corporate knowledge breaks down. Unfortunately it is also the area where there is the greatest need and benefit to capturing the corporate knowledge.

To get over this obstacle, a different approach to knowledge representation is needed. What is required is a process to easily and unambiguously describe the steps in the decision-making process, which collaborators can understand. It should also have a systematic and consistent way of applying and combining the steps to reach a conclusion. This is an "Inference Engine". It is a program that uses the individual decision-making steps as data, and processes them with situation-specific information, to reach a recommendation or advice for each particular situation. The collaborators can use the inference engine to test various scenarios. If the advice is not what they expected, the underlying steps can be examined to see if any are incorrect, or do not completely cover all situations. The steps can be expanded or modified as needed. The changes or enhancements are done on individual details of the process – that allows the collaborators to focus on a specific issue that can be more easily resolved. While changes will have overall effects on recommendations, specific aspects can be tested in a framework that systematically and consistently applies the changes.

Representing the Knowledge

There are 3 main ways to represent the knowledge and steps in a complex decision-making process – rules, tree diagrams and other diagrammatic techniques specific to certain problem types.

Rules are the most flexible way to describe steps. Inference Engines that process "Heuristic Rules" have been in use for many years. A Heuristic Rule is typically an If/Then statement about some aspect of a decision. In the shipping example, "If shipping a hazardous material, Then the shipment must be sent by ground transportation".



Heuristic Rule about an aspect of a problem

Most capable inference engines support "Backward Chaining", which allows one heuristic rule to use facts

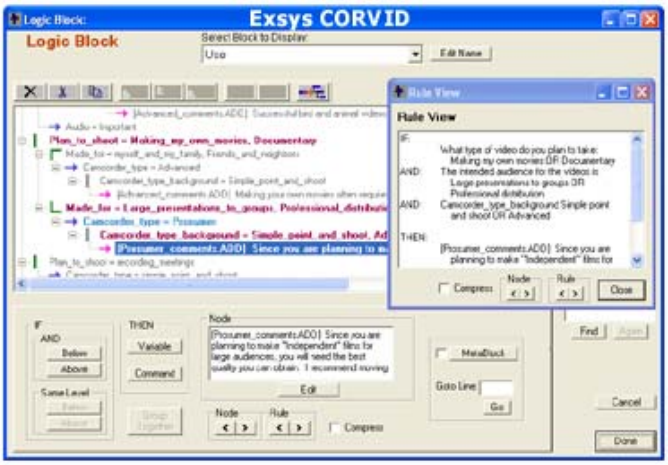
derived from another rule. For example, a rule "**If** the item to ship is Part #123, **Then** the shipment involves a hazardous material". The Inference Engine in effect combines these rules to recognize that "If the item to ship is Part #123, Then the shipment must be by ground transportation", even though this rule is not explicitly stated.

Rules can also be used to make probabilistic statements. For example, "**If** cost savings is a main factor, **Then** carrier X is unlikely to be a good choice". There are various ways to formalize what "unlikely" means, from simple "certainty" systems to complex systems that use statistical probability factors. The exact meaning and implementation of probability is another aspect that needs to be understood by the system developers.

A backward chaining inference engine can combine any number of rules together to make decisions. However, each rule is just a specific element that can be examined by the experts. If a body of rules is created that covers all relevant factors in a decision, it will effectively describe and represent all of the knowledge related to that decision. The inference engine will be able to use those rules to make specific recommendations based on input for individual cases. In addition to capturing and preserving the decision-making knowledge, the inference engine provides an interactive mechanism for the knowledge to be disseminated to others. This can be done in many ways, but the most effective way is to use a Web-enabled Inference Engine that allows the expert decision-making knowledge to be delivered in an interactive system in a Web page. The system will ask focused relevant questions and, based on the input and underlying heuristic rules, provide advice and recommendations.

The big benefit is that the experts can focus on reaching agreement on specific rules. Some may be unanimous views and some controversial, but the rule structure allows a very precise and focused debate on specific issues, rather than more general overviews of the problem that cannot be directly applied or tested. Once all the rules are agreed to, the inference engine takes care of how they are combined.

Tree diagrams are a popular way to organize and structure a section of a decision-making process. Actually, rules are the underlying way to represent knowledge in all of the approaches, and tree diagrams are really just a way to organize related rules. If a group of rules have related IF parts, diagramming them in a tree may make it easier to see the structure and fill in possible gaps. However, each branch in the tree is still an individual rule. Tree-based development environments are much faster and easier to work with. The intrinsic structure of the tree helps collaborators to fill in any gaps in the logic and helps to prompt them to consider all possible cases.



Exsys CORVID organizes and structures decision-making processes in rules and tree diagrams

As with rules, collaborators can share the tree diagrams and make changes based on their views. This makes it even easier to "see" the logic, and many users find it easier to work with tree diagrams than individual rules – especially when there are many related options that need to be considered. Since the tree branches are actually just rules, the Inference Engine can systematically and consistently apply the tree logic, and also deliver the system to end users via the Web.

A relatively new approach to knowledge representation is through tools aimed at specific types of problems. Most rule or tree-based tools are rather generic and allow a wide range of decision-making problems to be described. Other tools are aimed at particular types of problems, such as product selection, smart questionnaires, or diagnostics. Each of these types of systems has a standard rule structure. Rather than having the user build this structure from scratch, the tool starts with the structure and user just fills in the appropriate blanks. This results in a "constrained" development environment where the knowledge representation is expressed in the context of the specific problem area. This allows approaches that

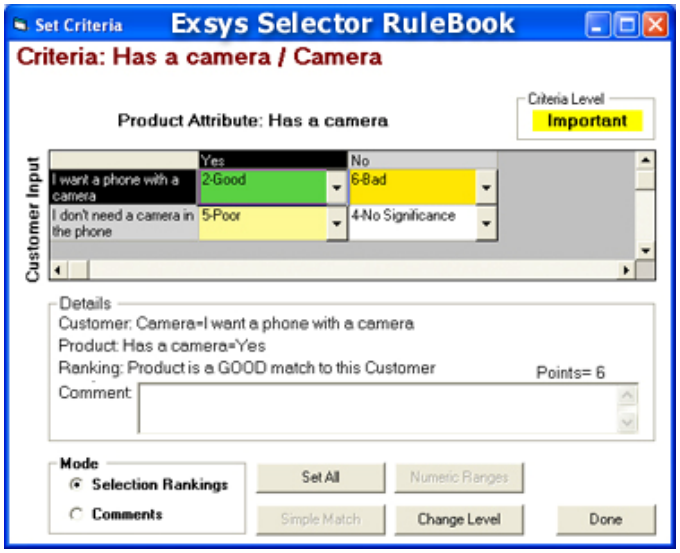
that are not rule or tree based, and which can be very easily learned and shared. For example, a product recommendation system recommends the best product(s) for a particular customer, selecting among a group of related items, based on the customers unique needs and requirements. All that needs to be entered by the developer is:

1. The possible products to select among
2. What characteristics of the products will be used in the selection process
3. When an end user has a particular requirement, how well does each possible value for a characteristic meet that requirement

The first 2 steps are rather easy to do since they are really just lists. There are many ways to use collaboration tools to develop the lists. It is the last step that is more complicated. A spreadsheet interface can be used, where each cell represents a ranking. This ranking is set in the context of the problem, and compares a customer's requirements with specific product features.

For example in a simple case, a cell phone recommendation system might have a characteristic of the phone having a built-in camera. A spreadsheet of product data lists each of the phones in the system and has a column to indicate if the phone has a camera. The customer is asked to select if they want a camera in the phone. A simple matrix can then be used to rank this individual factor in the decision. The cells in the matrix represent if the combination is "good" (based on this criteria, the phone is a good one to recommend), or "bad" (based on this criteria the phone is not a good one to recommend).

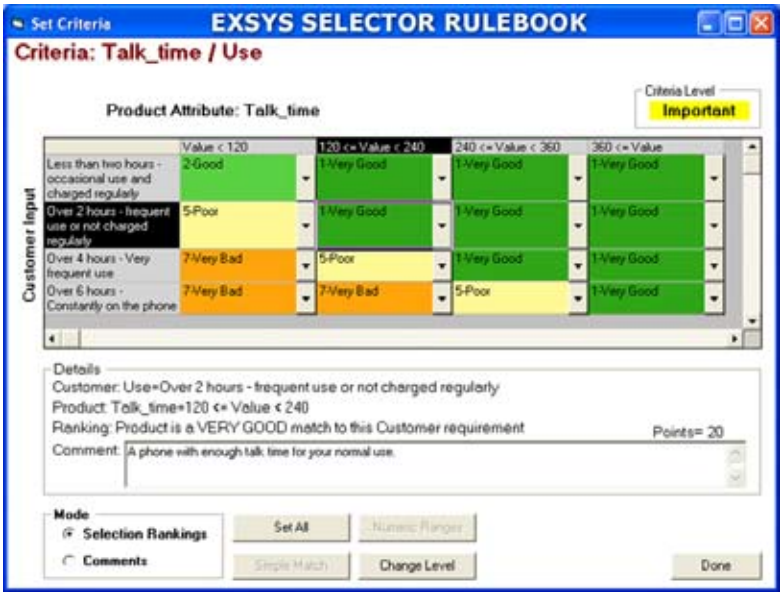
The top row is easy. If the customer wants a camera and if the phone has one -that is "good" and if it does not, it is "bad". The second line is more complicated. If the customer does not want a camera, is having one "bad"? If there is none, is that "good" or neutral? When there are many more possible values than "Yes" and "No", and multiple levels from "very good" to "very bad", the situation becomes one where collaboration among various experts and points of view can produce the best system.



Simple matrix representation of custom requirements and product characteristics

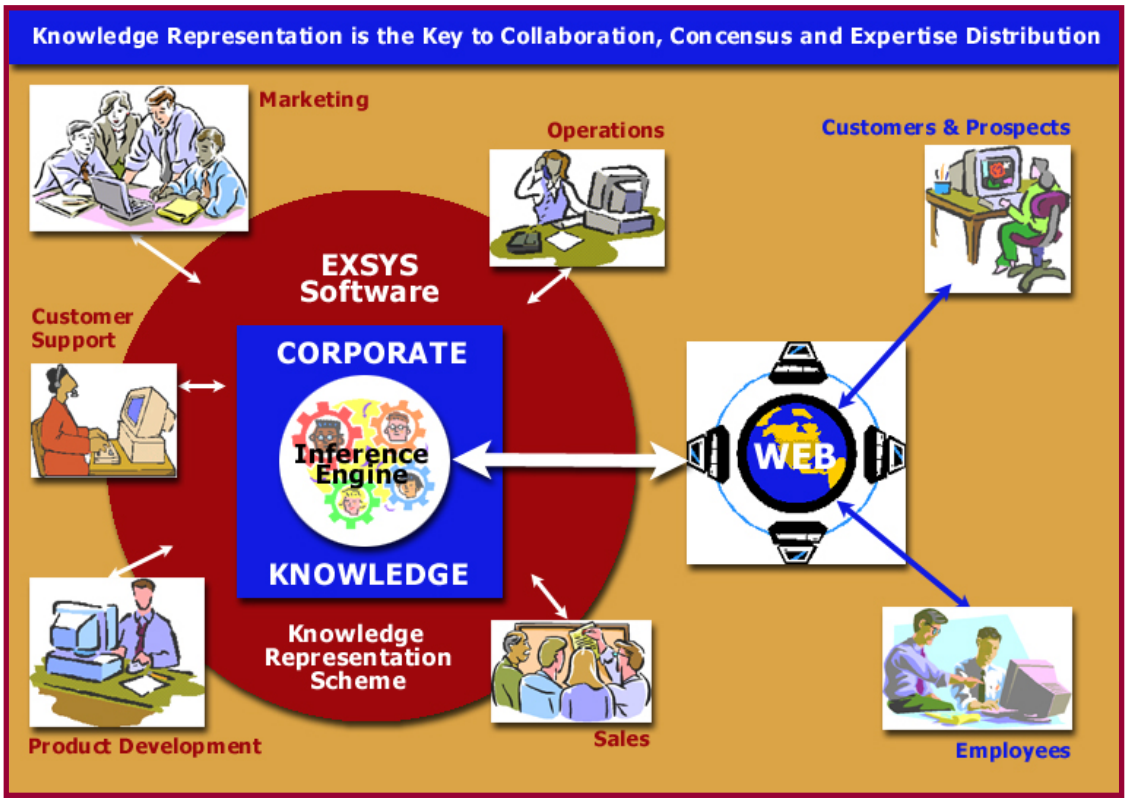
However, the key is to be able to use a knowledge representation scheme and tool to that makes it easy for the collaborators to enter their opinion on the issue.

By having the knowledge representation in the context of the problem, it is much easier for collaborators to contribute effectively. The underlying Inference Engine is still used to combine all the various criteria in a final decision. An actual selection decision will involve many factors that need to be weighted and combined.



Tools such as Exsys® Selector RuleBook®, designed specifically for Product Selection, enables collaborators to contribute effectively and individually without having to learn "rule" syntax.

A change in one will have ripple effects on the overall selection. The inference engine assures that all participants have their changes and input processed in exactly the same way to reach the overall conclusion. With such an approach, changes can be made, tested and sent to others in a consistent way.



Specialized knowledge representation tools only work well for the specific problems types that they were designed to handle - but for those problems, they are very effective at capturing the knowledge and produce a system that can be delivered via the Web to give advice based on expert knowledge. For problems that are beyond the scope of problem-specific tools, generic rule-based representation tools are a very effective way to handle more complex logic in a way that still allows it to be understood, commented on and processed in a consistent and logical way.

Consensus Makes Collaboration Work

The goal of collaboration is to reach consensus – otherwise it is just a structured way to argue and disagree. By reducing the topic being discussed to a single rule, decision tree branch or matrix cell, the discussion among the experts can be much more focused and precise. It is much easier and faster to reach a consensus on a specific fact than on a much larger decision-making process. The use of the proper knowledge representation tools allows the larger process to be systematically broken down into small pieces that will be consistently applied by the inference engine. This can allow consensus to be reached even for complex and controversial issues. The inference engine both provides a logical and consistent way to apply the pieces, but also allows the overall knowledge and decision-making expertise to be delivered to others via the Web.

Collaboration tools are a means to share "documents", however the key to successful capture of knowledge and consensus of expertise, is to share a representation of the knowledge that is meaningful and consistently interpreted. Whether helping customers find the most appropriate product, automating forms, meeting compliance or diagnosing processes, using knowledge automation in collaboration is helping businesses, organizations and government agencies move beyond concepts and brainstorming to delivering top-level knowledge assets of problem-solving expertise online to customers and employees whenever/wherever it's needed.

EXSYS
Capture Knowledge. Deliver Answers

Sales and product Information
sales@exsys.com

Partnerships
info@exsys.com

Technical Support for Existing Customers
support@exsys.com

EXSYS Inc.
6301 Indian School Rd. NE, Suite 700
Albuquerque, NM 87110 U.S.A.

t. +1.505.888.9494 ▌ f. +1.505.888.9509 ▌ www.exsys.com