

Modeling an Expert The Missing Piece of Knowledge Management

by

**Charles W. Faulkner & Cathryn C. Modrall
Influential Communications, Inc.**

Executive Summary

The field of Knowledge Management has emerged out of the growing recognition that many of a company's most valuable resources reside in the knowledge of its customers, business processes and employees. A business that can capture these resources, the 'knowledge' of its:

- *Consumer's desires
- *Best business practices and
- *Exceptional employees

and distribute them company-wide will experience a strategic competitive advantage. Companies that don't do this risk the same levels of strategic loss to downsizing, reorganizations and employee raids.

In light of this, Knowledge Management efforts have focused on three areas:

- | | | |
|---------------------------|-------|--------------------|
| * Data Mining | (for) | Customer Capital |
| * Enterprise Architecture | (for) | Structural Capital |
| * Communities of Practice | (for) | Human Capital |

These three areas address the customers' knowledge, the business infrastructure, and the knowledge that is created in a company's groups and teams, but they leave untapped the expert processes held within each of their exceptional, and often key, individuals.

Expert Modeling is a method for capturing and codifying this high-value-added, high-leverage knowledge – the essential "know-how" inside these key individuals in a way that allows for database codification and effective individual transfer. Expert Modeling is based on recent advances in cognitive science and linguistics. It makes use of the natural way individuals organize their experience, and provides a bridge to information technologies (IT).

When corporations can gain access to, replicate, and distribute, the essential expert knowledge found in their key individuals, teams, they can cost effectively leverage their investment in Knowledge Management and raise the baseline of performance and productivity company-wide.

Modeling an Expert

The Missing Piece of Knowledge Management

by
Charles W. Faulkner & Cathryn C. Modrall
Influential Communications, Inc.

Charles Faulkner and Cathryn Modrall are consultants who use a cognitive-behavioral systems approach for the elicitation, codification and transfer of experts' expertise.

The following is an abridged account of an actual expert modeling conducted in the Silicon Valley in the summer of 1997. The parties involved have requested confidentiality regarding themselves and the exact details of the resulting models.

In Search of Genius

The identified software genius enters the conference room. Everyone has come to expect the lack of a suit coat or tie. Their absence have become the 90's sign of genius. In their place are the jeans, high tech running shoes and developer t-shirt. Large black binders are passed out to everyone assembled. They are stuffed thick with pages. The software genius begins to speak. His first slide is dense with multi-syllabic acronyms and inter-connecting lines. A mix of industry experts, project managers, product developers and software testing specialists watch and listen. They have a large investment in understanding him. His methods are the cornerstone of a company-wide effort to raise the quality and speed of software development. The subject matter is as complex as preliminary reports had led them to believe. Several take another gulp of coffee before they physically hunker down in their seats. It's going to be a long day.

What is going to happen over the next hour, this day and the next? No one questions the intelligence of the identified expert. He is a leader in his field. No one questions the intelligence of those there to learn from him. After all, they have identified him as an important, perhaps unique, source of what they seek. They already know where they want to apply his expertise. The questions are: Can he deliver this to them? and Will they be able to 'get it?' Is this an effective form of knowledge transfer?

It is commonly assumed that the expert is the best person to transfer his expertise to others. It has a grand tradition. In the royal courts throughout history, it has been a pleasure of the powerful to be taught by the famous and wise. Aristotle was a teacher of Alexander. Beethoven, Descartes and many less luminous intellectuals were forced to take royal pupils. The idea is always the same. Only the best! Like having Michael Jordan teach you basketball or Eric Clapton the guitar. Who wouldn't want the chance to rub shoulders with greatness? "Perhaps some of it will rub off." "Maybe he'll let some secret slip." "If I watch carefully, maybe I can pick up something." "At least I'll be able to say I met him." There is currently a growth industry in instructional books, tapes and CDs authored by acclaimed experts. So far, this has not produced a rising number of people with an equal level of skill. This would argue that there must be something missing from our appreciation and understanding of expertise.

The desire for instruction by an expert often leads to less than expert results.

Understanding an Expert

Along the back of the room, the members our 'Expert Modeling' team look at each other and open their notebooks. We have been invited to observe this software genius in action. From Corporate Engineering to the Software Testing Division, there is an acknowledged need for what this exceptional individual knows. He has presented this program a half-a-dozen times at this organization. Each time, the participating employees have endeavored to apply his work. Each time they have had uneven results. "I understand it when he's talking, but when I get back to my team, it's gone." "You've got to keep so many things in mind." "He says he's drawing on at least seven separate disciplines. How can I do that? How does he do it?"

The first hurdle in understanding this software genius becomes apparent almost immediately. Turning to the group, he asks, "What is optimization?" Without waiting for an answer, he continues speaking. "Well, it's not maximization and it's not minimization." Several chairs squeak involuntarily; people tilt their heads as if to say "What?" Meanwhile, he briskly moves on. As Expert Modelers, we have been hired to understand and decode how this software expert organizes his thinking to do the work he does so expertly. We turn to a page in our modeling notebooks entitled 'Private Language' and write down his exact words. We anticipate our expert will be using many other such phrases in the days to come. Not another minute passes before he emphatically states that 'Systems Thinking' is fundamental to his approach....

Every discipline has its own jargon, its own terms-of-art that allow specialists to draw on large amounts of knowledge and experience in just a few words. Experts extend this into private labels - terms for their new ideas. Their words mean certain things in specific contexts and this acts as a kind of internal shorthand. Difficulties arise when the experts use these words around non-experts who don't know the shorthand. As Expert Modelers, we know that these keywords refer to facts, patterns and inferences within the mental maps of the expert. They are not held in isolation, but rather in patterns ranging from simple to complex that can be 'called up', each with its own private label.

While rich and complex, these private labels are not entirely private. Recent advances in the separate fields of neuroscience and cognitive linguistics are reaching the same conclusion: we are all using the same "building blocks" for language and understanding. What makes our thinking unique or conventional is how we "build up" our understanding. The difficulty with trying to automate these "building blocks" processes of understanding is that they are largely metaphoric and intuitive. "Metaphoric and intuitive" are not words immediately associated with a business context. But that is exactly the case when an executive says, "He has a gut feeling" or when he creates the solution to a business problem in terms of a football play, or when computer scientists discover how software design and architecture share the same fundamental rules. Computerized search engines may be able to determine a frequency count of the expert's private labels and associated terms, but they can not address the unique meaning an expert puts to these terms. It will necessarily leave out what is meant and what is inferred by the expert.

On the other hand, the examples - 'business as football' and 'software as architecture' - show how easily these patterns are available to human understanding and codification. While these metaphoric patterns number in the hundreds, all of them are as immediately understandable to another human being. The best knowledge codification for effective transfer may well be human as it gets to the heart of an expert's understanding of his or her expertise.

To gain an expert's expertise, first understand his understanding.

Not Every Language is in the Words

The software genius is now gesturing and moving about the room as he speaks. He wants his audience to understand that his concepts are hierarchical. He talks in detail about how he selected a particular photograph of the Great Pyramid in Egypt for the cover of his book. Some participants' eyes begin to wander. However, as Expert Modelers, we know that there are many ways that someone can emphasize the importance of something. These include: words ("This is important!"), the volume, an emphatic vocal tonality, or the total amount of time spent on the topic. Clearly, in terms of the time spent on a topic, hierarchy is a key concept. We also observe the ways in which his hands move and where his eyes alight. Some members of our team go so far as to try out these movements and gestures at the session breaks. Very often, his hands and eyes pause in the same places as he emphasizes several words in his 'private language'. What are we to make of this?

The software expert continues to refer to the same places in space over the next hour, the afternoon and throughout the following days. Meanwhile, the participants are increasingly losing the gist of his presentation. Questions are arising more frequently. On sessions breaks, they are now directly asking, 'How does he remember all these details?' 'How does he keep it straight?' Meanwhile, we have noticed that he has been non-verbally and systematically stacking one set of his principles on top of another. We realize he is 'seeing' the relationships among all his private labels, and so with his gestures he has been putting them together. At the same time, those observing him were listening for the different concepts and trying to separate them in order to understand them as distinct ideas. Instead of seeing him build the relationships between the different concepts, they see him mushing them together into the same 'mental' space. Our team makes a note to replay the videotapes of these words and gestures for them later.

Most everyone has heard or read something about the importance of body language and non-verbal communication. Books, tapes and communication classes have made people aware of the idea that gestures and voice tones can enhance or detract from their presentations. Recently, advances in cognitive science and linguistics have shown that body movements have much more to do with a person's states of the mind than was originally imagined. High speed video recordings have documented that gestures often begin in advance of the words, and that they tend to reflect the situation in one's thoughts much more than the situation in which they are expressed. Everyday examples abound: A busy account executive excitedly describes a recent negotiation. As he does so, what he refers and responds to is not in the room, but in his memory, and this is easily understood by everyone listening to his recollection. This same principle applies to our software genius's ideas. They are as real and present to him as the account executive's recollections. In linguistics, gestures and other non-verbal expressions constitute another language that is parallel to our verbal one. Everyone speaks these two languages, verbal and non-verbal, with their own degree of fluency. They are usually unaware of which one predominates in any given situation, or how one can substitute for the other. In most of life, it doesn't matter that much. In expert knowledge elicitation it matters a lot. In this case, our software expert is often substituting a 'Private Gesture' for what he realizes would be a long and convoluted verbal explanation of his multiple and connected concepts. Instead, he gestures, and his gestures equal a long verbal explanation for him - and so they are entirely clear to him. Using spatial relations and gestures to mark the spaces in front of him in order to organize reams of data and at least five separate disciplines, he creates the 'magic' of his expertise as only he can 'see' it. However his genius is not mysterious, nor tacit. It can be seen, and understood, by others who know what to look for.

The knowledge said in words may not be the important knowledge to know.

Look for Patterns, not Things

The software expert continues to speak; rapidly and disjointedly on his subject. He talks technically, suddenly veers off into an anecdote, and then mentions a fundamental principle before returning to the topic at hand. He makes frequent critical asides and jabs at various management practices. As the hours pass, his language becomes more technical. On a session break we question him about this. Yes, the increasing use of technical language and topic complexity is conscious and deliberate on his part. He thinks that his anecdotes make the material more memorable. We then ask him about his fault finding with current software practices and various management approaches. He tells us that he started out in software testing and he was paid to find faults. When we ask him about his disjointed speaking style, he smiles and reminds us that, "Everything is a system." We inquire further about this public and private language term. "What does it mean that everything is a system?" "It's parallel, iterative, and recursive." "And what does that mean?" "That it's parallel, iterative, and recursive." "Oh."

As Expert Modelers, we know that when an expert becomes unable to explain himself further, to provide additional details, or uses the same phrases of 'private language' over and over again, it is because he is nearing the edge of what he knows explicitly. We are going into an area of his knowledge that was unconsciously learned, or is so deeply held and tacit, that the assumptions and connections have never been consciously examined by him. At this point, we know to pay careful attention because it is in this realm that we will find out what truly makes this expert different and his expertise uniquely valuable.

Some of the seminar participants around us have been listening in on our conversation. A few of them nod knowingly at the software expert's remarks. They are drawing on their own private language meanings for these terms. Perhaps there is a match with what our software expert means. Perhaps not, but they won't know because they haven't compared their definitions. Other participants, having not understood these terms and others like them, begin to feel left out and unable to grasp the topic.

* REFERENCE The Knowledge Creating Company

When knowledge is explicit it held in common by a community. This community can be managers, or technical professionals or any group that has specialized knowledge shared in common. These shared understandings can be used to clarify the new ideas of professionals in the community their 'private' language' understandings. In these circumstances, a Best Practices approach yields high value returns. But knowledge is just as often held privately and tacitly and this is a quite different situation.

Imagine this software genius in a typical Best Practices scenario. The software team has been encouraged by management to find and implement the best software testing practices. They decide to look into the work of this expert to see what it has to offer. There are several possible outcomes. In one, the software team simply dismisses his work. After all, the number of implementations is few, owing to the difficulty understanding his way of thinking. In another, the software team recommends further study or simply waits for more practical applications to emerge. In these cases, a decisive technological advantage could be lost. Or, the software team has discerned value in this expert's work, and then, given their mandate, they seek it in the form of a specific practice. Rather than seeing this expert's integrated system, they look for the best elements. While they are certain to gain value, they will have missed much of what he has to offer. Most importantly, in all of these scenarios, the software team's Best Practice selections are restricted to the explicit knowledge this expert is able to offer them. Essentially, his expertise is

evaluated by to his ability as a communicator.

Faced with a variety of situations, the knowledge manager has to decide which of the many knowledge management 'tools' available to him will deliver the most value. Data Mining, Enterprise Architecture, Best Practices, Knowledge Mapping, Communities of Practice, and Expert Modeling each make certain strategic gains possible. However, one does not substitute for another. In common with all tools, each one is designed for some tasks and not for others. What is impossible with wrench may easily be accomplished by a screwdriver. What can not be found with one knowledge management 'tool' may easily be achieved with another.

What makes expert knowledge may be unknown, but it is not unknowable.

Looking at the Whole (Expert), as well as the Parts (of the Expertise)

In the software company where this software genius was engaged, the professionals who hired him all agreed there was something worthwhile in his work. They took their confusion as a sign of having reached the limits of their own expertise. They decided to bring in Expert Modelers.

As the Expert Modelers, we now had numerous examples of our software genius's 'private language', his predictable, if elliptical sentence structure, and his special, exemplative 'private gestures'. If he truly believed that everything was a system; "that it's parallel, iterative, and recursive," then this assumption was also 'organizing' his presentation style. While this created a confusing presentation for some, it was the key to what made his methodologies work.

Research into experts and expertise, in fields ranging from computer science to developmental psychology, have come to the conclusion that expertise is not additive. That is, knowing more is not what makes an expert. Experts do have a command of an extraordinary number of facts or distinctions, often numbering as many as 50,000. The difference that makes someone an expert is the organization of these facts, or rather their reorganization. In numerous studies referenced by _____, _____, _____ in "Expertise in Context", the expert emerged as the one who had, at some point, completely reorganized the information he has learned, and in that process, turned those many facts into his own expert knowledge system.

In the case of our software expert, he had reorganized himself in accordance with his understanding of systems thinking. Consequently, he looked at all the material he had to present as examples of the same things (systems concepts) but of different 'sizes' and on different levels. He simply started with the biggest 'chunk' of each one, and then moved from one to another by his own inner logic (remember his gestures) to smaller and smaller chunks. His anecdotes could fit anywhere since his presentation was "fractal" (a repeating pattern that can be scaled up or down). Since he was internally organized by the relative size (or scale) of his concepts, he couldn't present his material in a standard, linear or thematic fashion. However in doing this, the mystified participants, while knowing he was onto something important, couldn't grasp it because "he was doing it as he talked" NOT "talking about doing it."

The expert's expressions are often expressions of the expertise.

Human Expertise is an Expert System

All of the attempts to create Expert Systems in Artificial Intelligence (AI) and Knowledge Engineering (KE) share the assumption that the expertise is something separate and distinct from the expert. The living expert is thought of as merely a "carbon-based instantiation" of the expertise. This approach is a natural result of those wanting to transfer the expert knowledge from humans to computers. In some of these AI/KE approaches, time consuming one-on-one interviews are conducted between the expert and the knowledge engineer. Most notably, while these efforts seldom result in a working computer model, the knowledge engineer often emerges as an expert in the expert's expertise. This clearly shows that there is something that can be termed 'expert knowledge,' and that it is transferable. However, organizations are forced to reject this as a method of knowledge transfer because it is too time consuming for their experts and too expensive for the organization.

In the case of our software genius, years of effort, including assisting in the publication of a book and the development of a seminar, had not brought this high technology firm closer to extracting or implementing this expert's expertise. His value was recognized, but his skills remained elusive. The expert is considered the best person to transfer his expertise, his expertise can be understood piecemeal, his expertise is in the content of his work, and all the important information will be found in his words. With these assumptions, it is not surprising that expert knowledge has gained a reputation for being tacit and unknowable.

In order for us, as Expert Knowledge Modelers, to understand our expert's expertise, we had understanding him, that is, how he makes sense of the world. We had to understand our expert's field in the way he does. We needed to see how our expert fits things together, and we needed a way to fit them together - models, metaphors, diagrams, charts and actions. We gained these through analyzing his figurative language, his grammar, his narrative style, and his gestures in order to discover how he organizes his expertise. It was through these knowledge organizing structures that we were able 'see' his expertise the way he does.

The organization of expert knowledge very often is the knowledge.

Expertise is not the End

Something has concerned the members of our Expert Modeling team since the first morning we met our software genius. In developing his expertise for software testing, our expert had also developed his ability to find faults. The organization is anticipating the transfer of his models to other corporate employees. We have grave concerns about releasing a totally scalable fault finding strategy into their corporate culture. We huddle with our corporate liaison and review the original goals for the project. They said they want to understand and implement how their expert thinks about software testing to all the software quality assurance teams. We point out that a scalable fault finding strategy may increase software quality, but could be detrimental to productivity and morale. We propose that the scalable fault finding portions of the expert's strategy be replaced with a local, non-scalable quality standards strategy. They agree.

As we continue our modeling project, we find that nit-picky detailed scrutiny is common among many software testers, while reasoning and organizing in terms of systems is not. Thinking of everything as a hierarchical system with lots of little hierarchical systems inside it, each with its own special space that identifies its form and function, makes it much easier to keep track of a highly complex computer system. It also makes it possible to envision a comprehensive software reuse and testing strategy. However, this is not how software programmers and testers are typically or culturally trained. We were especially interested in specifying how this would effect of implementing his full software testing system. There would be a large amount of change management and retraining. It would require change of the existing company personnel and policies. Intrinsic to the implementation of a working expert knowledge model is to check the "fit" of the expert's way of doing things to the rest of the corporate system - checking for impacts, implications and consequences of the expert's approach.

Knowledge Management is knowing what knowledge fits with your organization.

Expert Knowledge as an Affordable Resource

At a review board team meeting, we present the completed model. Present are our corporate liaisons, a group of software testing specialists, and our software genius. We have reduced the content delivery time of the expert's program from four days to three hours, saving the firm tens of thousands of dollars of employee and consultant time. Our Expert Modeling team has written a lexicon of his private language terms with standard software development 'translations' at each level of his model. We have also made multiple layered color graphics and 3-D models to show everyone the interconnections the expert only alluded to with his gestures. It's now clear to everyone how he does it. We have also provided a metaphoric framework that is readily understood within the firm as well as containing the shift from fault finding strategies to quality assurance. The software genius is delighted that he is, finally, understood. He plans to update and revise his work as well as his presentations based on the insights he gained by having his expertise reflected back to him. The organization's team sets to work on on-site implementation of the material.

In this case study, using "cognitive-behavioral" knowledge elicitation techniques, what the organization had not gained in two years time working with this software genius became available to them in about one month, using only six days of the expert's time. Of these six days, four were concurrent with his normal work activities, with only two additional days spent with him testing our findings.

From this case study, it is clear that Expert Modeling adds value to the aggregate of corporate Best Practices by working with key personnel in a new way. In this new approach, people are seen as the source of particular genius, a specific, repeatable resource that can be distributed through a company for its competitive advantage.

Expert knowledge can be easily lost or it can captured and multiplied.

Summary

Knowledge Management is only a few years old, and it has already brought together a formidable number of disciplines and technologies. Data Mining has established its importance for documenting individual customer buying habits and discerning new consumer trends. Enterprise Architecture has become an established practice for the streamlining and automating of certain types of work. 'Communities of Practice' are solving long standing problems held in common by widely dispersed professionals. And as successful as all these endeavors have been and will be, none of them addresses the expertise held within an organization's best people – a resource that can 'walk out the door' tomorrow. This makes Expert Modeling the missing piece of Knowledge Management. It is missing because it has been thought impossible. In the words of the leading Knowledge Management consultants Thomas Davenport and Laurence Prusak, "If it were possible to extract knowledge from the knower in this way, it would radically change our compensation and educational policies." (*Working Knowledge*, p. 71.) The reason this has not been done is that the problems posed by knowledge elicitation have been approached from the point of view of the computer rather than the human being. However, one does not understand a human expert's expertise separate from the expert. Initially, they are one and same and need to be understood that way. Further, this understanding will only be partially expressed in words. This does not mean that it is unknowable or unqualifiable. Instead, it means the instruments of inquiry will be human ones - metaphors, analogies, models, drawings, and actions. In this, the Expert Modeler is seeking to find the expert's 'patterns of the whole'. Knowledge, especially expert knowledge, is as much in the relationships of the 'facts' as it is in the 'facts' themselves. Once this expert organization of knowledge is perceived, the expertise can be separated from its expert. Then it can be revised and enriched or even rewritten to meet the needs of other individuals and the organization. Expert Modeling is not the end of Knowledge Management, but rather the beginning.

Expert Modeling has solutions to business problems not addressed by other Knowledge Management approaches. Expert Modeling makes possible a kind of intellectual "key man insurance" by capturing the critical knowledge in irreplaceable individuals. It provides a means of multiplying scarce resources by modeling and codifying employees mental and behavioral skill sets. It can be used to increase the quality of existing resources by providing the means to streamline and enhance their processes. It makes possible accelerated knowledge transfer. It opens the possibility of new groupware and database search categories. Others will find an opportunity to discover how our best minds think and create. Expert Modeling offers corporations the reality of capturing their intellectual capital and retaining what their individual employees do so well. Like Knowledge Management, it is not a question whether Expert Modeling will become an accepted practice, but rather, who will secure that strategic advantage first.

©1998 Charles W. Faulkner & Cathryn C. Modrall, Influential Communications, Inc. All rights reserved.