Executive Information Systems, Inc.

Knowledge Management Metrics Development:

A Technical Approach

By

Joseph M. Firestone, Ph.D.

White Paper No. Ten

June 25, 1998

Introduction

This is one of a series of studies developing a Use Case approach to Knowledge Management (KM). This paper is primarily focused on the problem of developing measurement models for KM metrics in the context of a projected information systems application fulfilling the ***"Perform Measurement Modeling Task"*** within the ***"Perform Knowledge Discovery in Databases"*** use case.

To "drill down" to this focus I need to properly set a context involving a number of elements. These are: the nature of measurement; its relation to KM Metrics; the connection between KM metrics development and a business process use case view of KM; the connection between the high level business use case: "***Discover New Knowledge in Knowledge and information Bases,"*** and its ***"Perform Measurement Modeling"*** Task; and finally, the connection of these to the Distributed Knowledge Management Systems (DKMS) concept and the specific, projected information systems application. A good bit of this paper will set this context.

Following these preliminaries, the remainder of the paper will specify the Perform Knowledge Discovery in Databases use case, and the Perform Measurement Modeling Task in some detail. In the course of this specification, a number of techniques for developing ratio-scaled measurement models applicable to KM metrics development will be described. These will include techniques for developing rules that map: (1) categorical variable (e.g., event or type) values onto a ratio scaled abstract metric; (2) frequencies of an event occurrence onto a ratio scaled abstract metric; (3) multiple indicators into a ratio-scaled composite.

Measurement, Knowledge Management Measurement and Metrics

"Measurement is the assignment of numerals to things according to any determinative, non-degenerate, rule." [1] Determinative means the constant assignment of numerals given constant conditions. Non-degenerate means allowing for the possibility of assignment of different numerals under varying conditions.

Given this fairly broad definition it is common to distinguish classification, linear rank ordering, and metrical measurement [2]. Metrical measurement is quantitative. It involves *assigning a real number* to any selected item in the domain of a concept. Classical examples of metrical concepts are temperature in degrees Celsius, and length in centimeters. The metrics in these concepts are "degrees Celsius," and "length in centimeters," respectively. To establish these metrics, the abstractions "temperature," and length," are related to observational events through rules. The rules determine the Celsius and centimeter metrical measurement scales. A quantitative concept, the rules associated with it, and the observational events, taken together, constitute a measurement model for a metrical scale concept [3]. It is the measurement model, as much as the quantitative concept and the associated observations, that establishes the metric.

In knowledge management measurement, we are trying to select and/or formulate those concepts useful in measuring and influencing knowledge management performance. Some concepts will prove useful because they directly relate to core notions about the goals of knowledge management, and in that sense, have *normative significance* as performance criteria. For example, providing for the growth of knowledge is one of the goals of knowledge management. The abstraction "growth of knowledge," is therefore a normative concept we may seek to metricize, and establish as a performance criterion for knowledge management.

Other concepts may at first not seem directly related to the goals of knowledge management. But, insofar as they represent causes of the core concepts, or possible side effects of the knowledge management process, we will still need to measure and perhaps to metricize them, in order to explain, predict, influence, or properly assess progress on the performance criteria. These other concepts provide *descriptive criteria* for knowledge management.

The two types of criteria: normative and descriptive suggest two types of metrics for knowledge management: normative and descriptive metrics. Though at first blush it seems that we should be less interested in descriptive than in normative metrics, this is not the case. Some descriptive metrics, in fact, are likely to make the KMS "go round," and to be determinative of many of the normative metrics. These descriptive metrics then, provide a second set of knowledge performance metrics, a set whose members derive significance from their role in determining the course of the KMS, not from their direct normative significance.

KM Metrics Development and Knowledge Discovery Use Cases

Metrics development requires measurement modeling, a process of specifying the rules relating quantitative abstract concepts to observational concepts, thereby creating a metric. Put another way, a quantitative concept, the rules associated with it, and the observational events, taken together, constitute a measurement model establishing a metrical concept. It is the measurement model, as much as the quantitative concept and the associated observations that establishes the metric.

So, an approach to metrics development is an approach to measurement modeling. Measurement Modeling fits within the Organizational Knowledge Management Process (OKMP) [4] as a task within a specific OKMP business system use case [5] called: ***Discover New Knowledge in Knowledge and information Bases***. This use case is part of a set of twenty-two KM use cases I have under development and I will be call it Use Case Eleven for convenience. Here is an outline of the use case and some of its output descriptors to provide a context for the measurement modeling task.

Use Case Eleven: Discover New Knowledge in Knowledge and Information Bases.

- **Actors: Executive, Knowledge Management Engineer, Knowledge Management Consultant, Knowledge Engineer, IT Consultant**
- **Structure -- The tasks comprising the use case**

a. **Retrieve strategic goals and objectives, tactical goals and objectives and plans for knowledge discovery from outputs of earlier use cases.**
b. **Sample data.**
c. **Explore Data and Clean for Analytical Modeling.**
d. **Recode and transform data.**
e. **Reduce data**
f. **Select variables for modeling.**
g. **Transform variables for modeling.**
h. **Perform measurement modeling.**
i. **Select modeling techniques for causal, predictive, and dynamic modeling.**
j. **Estimate Models using data.**
k. **Validate Models.**

- **Output**

This use case further enhances the explicit organizational knowledge base by adding new models developed in interaction with the existing knowledge base. The enhanced knowledge base is resident in, or at least measured by, organizational cultural artifacts -- media of various kinds. Descriptors of the enhanced organizational knowledge base effected by this use case, as well as descriptors relating to growth and change are:

- Knowledge Base Descriptors
  - Knowledge Domains of knowledge components;
  - Subsystem locus of knowledge;
  - Media locus of knowledge;
  - Type of knowledge (knowledge, meta-knowledge, planning knowledge, descriptive knowledge, knowledge about impact, predictive knowledge, assessment knowledge);
  - Distributed/centralized architecture of knowledge base;
  - Degree of integration/coherence of the knowledge base within

    or between knowledge types or domains;
    - Scope of the knowledge base within and across knowledge types or domains;
    - Level of measurement of attributes in knowledge base within and across domains;
    - Extent of quantification of attributes in the knowledge base;
    - Extent of logical consistency of the knowledge base;
    - Types of models used in the knowledge base (conceptual, analytic, data models, object models, structural models);
    - Types of formal languages used in the knowledge base (set theory, mathematics, fuzzy logic, etc.)
    - Types of semi-formal languages used in the knowledge base (object modeling language, knowledge modeling language, etc.);
    - Types of methods (features, benefits, specifications);
    - Types of methodologies (features, benefits, specifications);
    - Software applications (features, benefits, specifications, performance, interface);
    - Type of validation of various components of the knowledge base (logical consistency, empirical fit; simplicity, projectibility, commensurability, continuity, coherent measurement modeling, systematic fruitfulness, heuristic quality, comparison set completeness, etc.);
    - Extent of validation within each type;
    - Composite extent of validation of various components;
    - Priority of knowledge components.
    - Performance metric on quality of organizational knowledge base
- Growth and Change Descriptors
    - Growth/decline of various types of knowledge,
    - Changes in knowledge architecture centralization,
    - Growth/decline in integration/coherence of knowledge,
    - Increase/decrease in scope of the knowledge base,
    - Changes in levels of measurement of attributes in knowledge base,
    - Increase/decrease in quantification of attributes,
    - Increase/decrease in logical consistency,
    - Change in types of models used in knowledge base,
    - Development in formal languages,
    - Development in semi-formal languages,
    - Changes in types of methods (reduction in costs, increase/decrease in capabilities);
    - Change in types of methodologies (reduction in costs, increase in scope, increase/decrease in capabilities);

- Increase/decrease in IT-assisted support for decision making provided by software applications;
- Increase/decrease in type of validation of various components of the knowledge base (logical consistency, empirical fit; simplicity, projectibility, commensurability, continuity, coherent measurement modeling, systematic fruitfulness, heuristic quality, completeness of the comparison set, etc.);
- Increase/decrease in extent of validation within each type;
- Increase/decrease in composite extent of validation of various components.
- Performance metric on discovery of new knowledge.

In the first instance, an approach to metrics is to specify further what is entailed in the measurement modeling task. And since the measurement modeling task is integrated within the broader context of Use Case Eleven, we also need to further clarify this context on which measurement modeling is dependent. At this point we could proceed in either of two directions.

We could specify OKMP Business Process Use Case Eleven in more detail and analyze measurement modeling as a task within this use case. Or we could move to a more concrete level of analysis and consider measurement modeling as a task in knowledge management software applications.

We choose to continue this conceptual development at the more concrete level of the software application for three reasons. (a) We need to go there anyway, eventually. (b) Measurement modeling is closely related to analytical modeling and data mining, two subjects closely identified with automated analysis and software applications. And (c) moving to the software application level will allow specification of more of the information systems application underlying metrics development than if we continued the analysis at the OKMP level.

In developing the idea of measurement modeling or metrics development as part of a software application, we need the concept of an Information Systems Use Case. This concept is defined by Jacobson [6] as "A behaviourally-related sequence of transactions performed by an actor in a dialogue with the system to provide some measurable value to the actor." A behaviorally related set of information system use cases, in turn, constitutes an information systems application supporting a business process through its use cases. Figure One shows the relationships of business and information system use cases in an application.

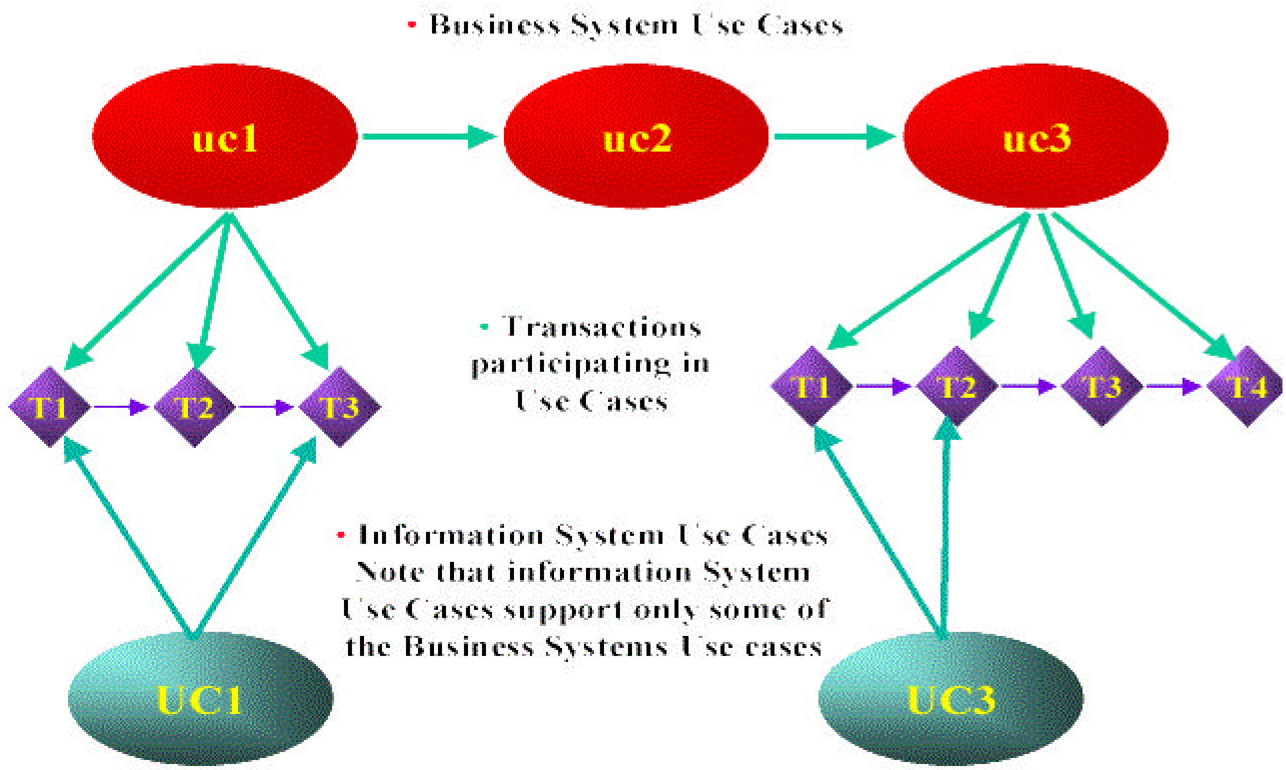

Figure One -- Relationships of Business System Use Cases to Information System Use cases

Measurement modeling is not only a task in the Knowledge Discovery use case of the OKMP; it is also a task in the use case of the information systems application supporting the OKMP. This generic information systems application is called the Distributed Knowledge Management System (DKMS). [7] Its Use Case Eleven is called "Perform Knowledge Discovery in Databases (KDD)." [8]

Distributed Knowledge Management Systems (DKMS) and Automated Knowledge Bases

An OKMP may be supported by a software application system that manages the integration of distributed objects into a functioning whole producing, maintaining, and enhancing an automated knowledge base. This software application is the DKMS. The functionality of a DKMS is specified by a set of DKMS use cases partially automating the knowledge management use cases of the OKMP. These DKMS use cases then, are conceptually distinct from OKMP use cases.

An automated knowledge base is the set of data, validated models, metamodels, and software used for manipulating these, pertaining to an organization, produced either by using a DKMS, or imported from other sources upon creation of a DKMS. A DKMS, in this view, requires a knowledge base to begin operation. But it enhances its own knowledge base with the passage of time because it is a self-correcting system, subject to testing against experience.

The DKMS knowledge base is a subsystem of the broader organizational knowledge base. But it is a subsystem that will grow in size and importance in pursuing the goals of improved knowledge management and growth in organizational knowledge. The DKMS must not only manage data, but all of the objects, object models, process models, use case models, object interaction models, and dynamic models, used to process data and to interpret it to produce an organization's automated knowledge base. It is because of its role in managing and processing data, objects, and models to produce a knowledge base that the term Distributed Knowledge Management System is so appropriate for this software application.

The Perform KDD Use Case

Here is a detailed exposition of the Perform KDD use case. Its importance for metrics development is to make clear the context of measurement modeling and its dependencies on the tasks that come both before and after measurement modeling in the use case sequence.

- **Actors: Executive, Knowledge Management Engineer, Knowledge Management Consultant, Knowledge Engineer, IT Consultant**

The Knowledge Management Engineer, and Knowledge Management Consultant will perform "what-if" simulations of policy impact, develop forecasts, and perform model adjustment, adaptation, and refinement of previously formulated models. These tasks may require that these actors execute a highly automated form of KDD/data mining workflow, where the actor is led through the data mining process according to a fixed set of procedures and dialogs in order to adjust, adapt, or refine previously formulated models.

The Knowledge Engineer performs the same tasks as the Knowledge Management Engineer, or Knowledge Management Consultant, except this role adds a more flexible KDD/data mining work flow beginning with data input, moving through various data processing and KDD stages including model estimation and model validation. The Knowledge Engineer will have complete flexibility in performing the data mining workflow and exercising professional judgment in arriving at estimated and validated models.

- **Structure**

a. **Retrieve and display strategic goals and objectives, tactical goals and objectives, and plans for knowledge discovery from outputs of Use Cases One-Four.**

   The actor interfaces with dialogues displaying the plans and priorities for new knowledge discovery. Based on this examination the actor selects the business domain area of the discovery effort.

b. **Select entity objects representing business domains to be mined for new knowledge.**

   Once the area of investigation is selected, the actor searches for domain entity objects that may be relevant to the investigation, and selects from among them the targets of the KDD process.

c. **sample data**

The actor will work through a dialogue to select samples of the data to be mined within the entity objects selected. A full range of one-of-n and random sampling options should be available through the GUI interface.

d. explore data and clean for modeling

This task refers to basic exploratory data analysis. This is a task that requires considerable expertise and judgment to perform well, and it also requires access to a diversity of analytical techniques that would be used in a relatively free-wheeling manner. Therefore, the Knowledge Management Engineer, and Knowledge Management Consultant will not perform this data mining task. It is restricted instead to the Knowledge Engineer Role.

The Knowledge Engineer needs access to, and appropriate dialogues for: general purpose descriptive statistics, specialized descriptive statistics and diagnostics, graphs for exploratory data analysis, tests for fitting probability distributions to data, descriptive statistics and graphics coupled with segmentations by grouping variables, categorization of continuous variables, spreadsheet-like browsing of data, point-and-click facilities to switch from data to graphics, comprehensive sets of options in computing correlations between and among variables, including the ability to handle missing data, graphics for visualizing correlations and correlation matrices, descriptive statistics and graphics for blocks of data selected while browsing the database, an interactive probability calculator to allow an actor to explore distributions of segments of data, t- and other tests of group differences in means, frequency tables, cross-tabulation tables, stub-and-banner tables, multiway tables, graphics, and statistical tests correlated to the tables, a comprehensive set of non-parametric statistical tests and associated graphics, and distribution fitting with statistical tests and associated graphics. Once subtasks related to dialogues representing these techniques are performed, responses from the software must include attractively formatted and highly customizable reports for the Knowledge Engineer. The reports must be integratable with other documents, and also highly editable for presentations and publications.

Cleaning data for modeling, is very different from cleaning data in the course of creating a data warehouse. Assuming that data warehouse-related cleaning has been done, specialized data cleaning for data mining is cleaning for the purpose of adapting to the data mining software or the analytical purpose of the data mining task. Thus, the actor may need to remove missing data codes that are inconsistent with the data mining software, even though they fit the data warehousing software. It may also be necessary to scrub the data of certain specialized fields that are inconsistent with data mining. Finally it may be necessary to remove duplicate records that have value for data warehouse reporting, but are in conflict with the specific purposes of an analysis, and would distort the results of model estimation. The software must provide support for the actor to perform this type of specialized cleansing for data mining.

d. recode and transform data

The Knowledge Engineer follows exploratory data analysis and data cleansing with recoding and transformation of data variables. The model manager will often recode continuous variables to ordinal or categorical ones, or re-code categorical variables to consolidate categories. The model manager will create "dummy" variables out of both categorical and continuous variables. The capacity for recoding must be one that fully supports all common logical operations and conditional statements of unlimited complexity.

Data transformations will be performed to create distributions that better fit statistical norms, to modify outliers, and to handle missing data. The data transformation capability required here must support all standard mathematical functions, a large variety of statistical functions, conditional operations, variable names, comments, and missing data. The Knowledge Engineers will both transform existing variables and define new ones using the facility. The Knowledge Engineers will also write their own data transformation algorithms, and/or interface the software with external software containing other data transformation algorithms. The data mining software must support these activities.

Both the recoding, and data transformation activities will be highly interactive. Results of recoding or data transformation activities must be immediately observable in tabular or graphic form, since Knowledge Engineers will move from action to inspection of results and back, in a continuous and rapid workflow.

e. Reduce data

While performing recoding and data transformation, the Knowledge Engineer also makes decisions about which variables are relevant to the specific data mining problem providing the context of the use case. The Knowledge Engineer might base the relevance decisions on the results of previous activities supplemented by intuition, or alternatively, a formal evaluation procedure rating variables for relevance to the data mining problem might be employed.

If the latter scenario is selected by the actor, the Analytic Hierarchy Process (AHP) [9] would be used to derive the relevance ratings, since this is a prioritization problem. The software capability to handle this process would have been implemented for earlier use cases of the DKMS. For this use case, exactly the same software might be used, or if this alternative is not specifically focused enough on the problem of relevance ratings for variables, a template would be derived specifically for this use case.

Once either intuition, or AHP-based decisions are made to reduce the data variables involved in the data mining process, the remaining data variables are subject to empirically-based methods of data reduction. Knowledge Engineers use a variety of techniques in performing this next task including: further descriptive statistical analysis, further exploratory data analysis, contingency table analysis, correlation analysis, cluster analysis, principal components analysis with rotation, and multiple and stepwise linear regression.

The objectives in performing these tasks are to know intimately the shape of the distribution of each independent and dependent variable, and to learn about their degree of redundancy. The Knowledge Engineer also tries to select variables that measure different things in different ways, or at least the same thing in different ways, and also determine whether the variables selected will meet the distributional assumptions of the modeling techniques being considered.

f. select variables for modeling

This step is one of further work in variable selection. Here attention focuses on techniques such as multiple and stepwise regression in preliminary attempts to estimate a final model, while also determining whether additional variables can be pruned from the model.

g. transform variables

By the end of task (f) the Knowledge Engineer has reduced the data set by a great deal, often by as much as ninety percent (90%). A much more exacting effort is then made to model the non-linear relationships among variables. Transformations will be performed to derive nonlinear forms or combinations of the independent variables, to be used in the remaining stages of modeling. These non-linear transformations are often dictated by theory, but sometimes transformations are employed based on hunches or desires to experiment with new functional forms.

h. perform measurement modeling

The Knowledge Engineer now moves to a stage of explicit measurement modeling, where attempts are made to model the relations between and among data variables and abstract attributes they are supposed to measure. The more specific objective is to formulate models that define tight clusterings of data variables around derived abstractions. For measurement relations that are linear in their functional forms, the well-known techniques of structural equation and path modeling are often used. If nonlinear measurement modeling is involved, a diversity of techniques including fuzzy measurement modeling, bayesian belief networks and neural networks are appropriate.

If specific measurement models prove unsatisfactory, the Knowledge Engineer may iterate within the data mining use case, by returning to earlier tasks and re-transforming variables. Another favorite alternative, is to subset the data into more homogeneous groups of cases that may produce more satisfactory measurement models. Group Clustering techniques are used for this purpose. Clustering techniques must be selected by the Knowledge Engineer, and then applied to derive more homogeneous clusters. Measurement modeling may then be attempted once again, within each homogeneous group of cases.

i. **select modeling techniques**

When measurement models are formulated, techniques must be selected for causal or predictive modeling. The Knowledge Engineer must select from a range of techniques now provided by statistical packages including: various forms of multiple regression, analysis of variance, classification and regression trees, log-linear analysis, general non-linear estimation (including probit and logit analysis), canonical correlation and regression analysis, discriminant and canonical discriminant analysis, survival/failure-time analysis, time series analysis, and forecasting methods, structural equation modeling and path analysis. In addition, the Knowledge Engineer needs to be able to select from a number of techniques based on more recent fields of analytical research such as: neural networks, fuzzy engineering, genetic algorithms, chaos and fractal theory, graphical belief models, and case-based reasoning.

j. **estimate models**

Once the modeling technique is selected, the Knowledge Engineer uses the GUI interface to point the technique(s) to the data set to be mined, runs the data mining software applying the model estimation technique, and receives the results in tabular, graphic and other visual formats.

k. **validate models**

Results of model estimation are likely to conflict both within and across modeling techniques. Since a good analysis will provide many different points of view on the data, such conflicts are to be expected and welcomed as part of the ordinary procedure of model estimation. The validation task is one of conflict resolution, where the Knowledge Engineer decides on acceptance of one or more models for future application. This means the Knowledge Engineer must walk through a multi-criterion decision process to rate or at least rank candidate models, and to specify a cut-off point for excluding models from future application.

In this decision process the Knowledge Engineer will first develop, revise or reengineer an attribute hierarchy specifying the validity concept [10]. The attribute hierarchy will have validity at its first level and then will specify a second level concept cluster containing the components of validity from the perspective of the Knowledge Engineer. In turn, each component of this second level cluster will be specified by a third cluster, and so on until each component of the validity concept is specified to the point where quantitative data or logically consistent judgment-based ratings may be specified.

In specifying the attribute hierarchy, the actor will again apply the AHP. But it is here applied to validity assessment, rather than to developing a hierarchy of goals and objectives. Once the hierarchy, including its priority weights, derived from pair comparison ratings is developed, it will be applied to assess validity. Actors will perform such assessments by applying global weights to values of the bottom level components of the hierarchy. These values will be test statistics of goodness of fit of models, or tests of statistical significance, or other data on test criteria provided by the various statistical and analytical algorithms used with the workbench.

In relation to components where such "operational" measures of validity do not exist, actors will use the AHP pair comparison rating and ratio scaling facilities to produce such measurement. When all measurements are specified, existing quantitative measurements will be normalized to the same scale as ratio scale ratings. Finally, measurements of validity for each competing model in a model comparison set will be derived by using criterion values and global weights to aggregate a global measurement of validity for each model. All criterion values, weights, hierarchy relationships and hierarchy metadata will be saved to a commercial database.

Actors will review and report on validity assessments by using a set of standard reports displaying various aspects of validity hierarchies. Actors will also use ad hoc reporting to construct new views of the validity data and metadata. Graphics, and charts as well as tables will be provided for this reporting subtask. Following examination of results of the validity assessment, actors will decide on cutoff levels for valid models to be retained for future applications and for use by Knowledge Management Engineers, and Knowledge Management Consultants. A GUI dialog will assist actors in making this choice by summarizing the results of the validity assessment and by providing access to the full set of reporting capabilities on validity data and metadata. The dialogue will also allow the actor to save the cutoff decision to the database and to apply any cutoff criterion formulated by the actor to future validity decision-making.

To facilitate the actor’s attempts to construct validity hierarchies, the software will offer a baseline validity hierarchy developed for the application. This template will specify all of the clusters and their components, but will provide equal weighting for each component in the hierarchy as a set of default values. It will then be up to the individual Knowledge Engineers to customize their own hierarchy by deleting or adding components and providing weights. The various subtasks of the validate models task are listed below.

1. Enter, edit, or review the highest level attributes for specifying validity in an analytic hierarchy interface.
2. Rate the highest-level validity attributes relative to each other with respect to validity.
3. Enter, edit, or review the highest level attributes for further specifying each highest-level validity attribute.
4. Rate the attributes specified in (3) above, relative to each other with respect to each highest-level validity attribute to which they contribute.
5. Enter, edit, or review the next lower-level objectives contributing to each attribute specified in (3).
6. Rate these next lower-level attributes (those specified in (5)) relative to each other with respect to each attribute specified in (4) above.
7. Repeat (5) and (6) until the lowest level attributes for specifying validity are selected and rated.
8. Compute and save the analytic hierarchy of attributes.
9. Report on results of (1) to (8) above, the various levels of validity-related attributes, their relative importance, and relations to each other.
10. Retrieve and display lowest level validity attribute values, where these are provided by algorithmic software.
11. Provide missing lowest-level attribute level validity values by performing ratings directly pair-comparing models in the comparison set, against one another with respect to the lowest level validity criterion whose measurement values are being estimated. Use the AHP rating methods to derive these validity values, and to provide consistency measures of the ratings.
12. Derive global values for the lowest-level validity attributes by applying the global importance weights to the results of the ratings in (11), and to the results provided by algorithmic software.
13. Compare models according to their global validity scores. Determine cutoff points or range for designating models as valid. Designate models as valid. Save results to the metadata repository.

l. Repeat a specific data mining process on the same or new data

This task is the basis for the Knowledge Management Engineer's, and Knowledge Management Consultant’s involvement in the data mining process. They need to be able to adjust, refine, and adapt models previously formulated by the Knowledge Engineer. In this task, they will browse models in a model repository, or metadata about these models. Based on such metadata, the Knowledge Management Engineer, and Knowledge Management Consultant will select both models and associated data sets to work with. They will execute selections through the measurement modeling task in an automated fashion. That is, when the Knowledge Management Engineer, or Knowledge Management Consultant executes a model, all of the earlier decisions made by the Knowledge Engineer who initially formulated a particular model will be repeated on both old and new data records. The Knowledge Management Engineer, and Knowledge Management Consultant will not be allowed to add any new data variables to the analysis.

When the model selection task is reached, the Knowledge Management Engineer, and Knowledge Management Consultant will use a wizard to decide whether a modeling technique not previously used will be included in the new analysis. If a new technique is selected through use of the wizard, it will be used along with the old techniques to re-estimate models.

In the validation task finally, the Knowledge Management Engineer, and Knowledge Management Consultant will be guided in final model selection by a wizard incorporating the previous weightings and criteria used by the Knowledge Engineer. They can use the wizard to include models for future application, if these were selected during the validation task by using the evaluation criteria incorporated into the wizard by the Knowledge Engineer. But they will not be able to change the evaluation criteria, or priority weights specified by the Knowledge Engineer, or either add or delete models without using the Wizard.

The "Perform Measurement Modeling" Task

The Knowledge Engineer now moves to a stage of explicit measurement modeling, where attempts are made to model the relations between and among data variables and abstract attributes they are supposed to measure. The more specific objective is to formulate models that define tight clusterings of data variables around derived abstractions, and that metricize these abstractions to establish ratio scales [11].

a. **Open the concept mapping, cognitive mapping, knowledge mapping, graphical belief modeling or semantic networking dialog, and create, label, and define a new node [12].**

   Not that the above techniques are the same, but their function in graphically representing a network of concepts related by propositions is similar. Use of one of the techniques to specify concepts is both more effective and more economical for concept specification. The measurement scale level intended for the new node is specified as part of this step.

b. **Develop a network of abstract concepts encompassing the quantitative concept to be modeled, by establishing new nodes for related concepts and linking them to the original concept.**

A technique should be used that supports fuzzy cognitive mapping [13], as well as crisp cognitive mapping, and that allows distinctions among different forms of entailment such as logical entailment or causal entailment.

c. **Select data variables as candidate measures of the original concept**

The task should support database browsing for information about candidate variables, and an interface that allows drag-and-drop selection of the candidates. Results of the previous concept specification tasks should support selection of tables or classes relevant to the underlying concepts. The existing object or data models, if properly done, will tend to associate candidate data variables in the same classes or tables.

d. **Open dialog guiding actor in specifying rules relating abstract and data variables in such a way that values of the data variables can be used along with the rules to compute quantitative scores for the concepts.**

This is an involved dialog providing for a number of options. Frequently, rule specification for metrics is very direct, even simple. For example, "extent of accessibility of the hierarchy" produced by use case one to knowledge management consumers can be measured by the percentage of organization actors who have documents detailing the hierarchy, or who can access it through their workstations either locally, or over a network. The rule of correspondence in this case associates extent of accessibility with percentage of individuals having access. Each additional individual increases the percentage and the extent of accessibility by corresponding amounts.

More generally, though, metrics development may require: (1) a rule that will map categorical variable (e.g., event or type) values onto a ratio scaled abstract metric; (2) a rule that maps frequencies of an event occurrence onto a ratio scaled abstract metric; or perhaps (3) a combination of multiple indicators into a composite, mapping data variable values to values of an abstract metric. Here are procedures for developing ratio scaled metrics for these three situations specified in terms appropriate for the DKMS application context.

(1) The actor opens a dialog for establishing a ratio scale metric for an abstract concept from specified individual categorical data attribute values. The actor activates the rating task and selects the categorical attribute values as the object of judgmental ratings. Upon selection, these are presented to the actor in a series of pair-wise comparisons with other data attribute values on a priority scale relative to the abstraction.

The comparative judgments will split 100 points between each pair member (the constant sum comparison method), to rate the relative priority of one pair member against another.

Alternatively, the actor will assign a value to the right-hand member of a pair by making a proportional comparison to a fixed value of 100 given to the left-hand pair member. Alternatively, the actor will rate relative priority by adjusting the height of bars on a GUI control representing the relative importance of each pair member. Whichever, method is selected by the actor. The results of ratings will be displayed by the software using all three methods for the actor’s review of ratings.

The actor enters the pair-wise comparisons using one of the rating methods, until all ratings are completed. The actor then saves the judgments, and after doing so, causes the software to assemble these into a positive reciprocal matrix. From this matrix it computes ratio-scaled priority ratings for each of the categorical attribute values, along with a measure of the consistency of the matrix of judgments on which the scale is based [14].

The result of the above procedure is a set of rules of correspondence of the form "if E then S" where: E is the event, event sequence, or event co-occurrence represented by the value of the categorical observational variable; and S is the value of the abstract quantity on a ratio scale [15]. In fact, the set of these rules of correspondence establishes the metrical standard of the underlying concept the actor is constructing.

(2) What if a categorical variable can be aggregated to produce event, event co-occurrence, or event sequence frequencies? Then the actor proceeds to open a dialog supporting formulating a "principle of correlation," [16] between a data variable and an abstraction. If E Then S, is a rule of correspondence; then $S = a + bf(E)$ (where S is the abstract scaled variable, f is some unspecified function, linear or non-linear as appropriate, and E is an event frequency variable), is the principle of correlation relating event frequencies to values on the abstract scale.

This general form encompasses all individual rules of correspondence of the if-then form relating any possible event frequency to an S-value. An actor has unlimited freedom in specifying a principle of correlation, which is just another hypothesis among many in a measurement model. The application should therefore support selection of any of a diverse set of commonly available functional forms, plus the ability for actors to specify a function of their choosing as a principle of correlation.

To help arrive at a principle of correlation, the actor should again apply pair comparisons, but now to event frequencies rather than single events. This will produce individual rules associating ratio scaled values with particular event frequencies, but no principle of correlation.

To get to a principle of correlation, the actor uses the application to graph event frequencies against scale values, and to non-linearly regress scale values against frequencies. The regression result is the principle of correlation. Once the principle of correlation is specified for each type of event establishing the original ratio relationships, the measurement model relating events and event frequencies to the underlying linear order is complete, though the model is purely hypothetical and requires external validation.

(3) For the composite case, here are two alternative techniques.

(i) *Assuming the proposed component attributes of a composite are not statistically correlated*, one procedure begins with the actor performing pair comparisons of the relative ability of each of the attributes of the composite to represent the abstract quantity. The procedure is no different from the one for categorical attributes up to this point. Once logically consistent judgments are forthcoming, it produces a set of relative ability ratio scaled values of weights to be applied to the attributes in computing the composite.

The actor next computes a ratio scale from an algorithm for computing the composite. The algorithm normalizes and translates each of the attributes so that their values prior to the computation of the final scores are *calibrated to one of the attributes already defined as a ratio scaled metric*.

The calibration is done through simple linear regression against the criterion attribute variable, and is part of the algorithm. The algorithm then proceeds to compute the composite by weighting the transformed data variables, or transformed functions of these variables (if theoretical considerations dictate using something other than a simple linear composite), and then summing the weighted transformed scores. The result is a ratio scale since both the relative ability weights and all the component attributes in the composite have been defined on such a scale.

An alternative to using regression against one of the component attributes in order to normalize all attributes to the same input ratio scale, is to use a ratio-scaled criterion variable for regression that is external to the composite. The zero point for such a criterion may be established non-arbitrarily, if there are enough objects available having the ratio scaled abstract attribute to support another round of pair comparisons.

Specifically the actor can rate objects comparatively in relation to the attribute being measured. Following consistency tests and computation of ratio scale values, an attribute directly scaling the objects relative to the underlying attribute is produced. At this point the actor completes the procedure by regressing the composite predictor of the abstract attribute against the directly scaled attribute, or by regressing the attributes entering the composite directly against the criterion attribute. Once the composite is calibrated in this way, it can be used without the criterion variable to produce ratio scaled values.

(ii) The second alternative technique for producing ratio scaled composites is based on fuzzy measurement modeling.

Assuming that quantitative component attributes have already been selected for the proposed composite (for example, a multi-attribute performance measure), the actor's first step is to map these quantitative attributes into fuzzy linguistic variables, composed of fuzzy term subsets. This mapping is called fuzzification.

A fuzzy linguistic variable is an abstraction that maps a quantitative variable into a set of overlapping, categorical, subdivisions. The overlapping categories are the values of the linguistic variable. A fuzzy term subset is one of these linguistic categories. Each fuzzy term subset is specified by a surface, called a membership function, which maps the values of the underlying quantitative variable into the interval [0,1].

The significance of the mapping is that it measures the extent to which a value of the underlying quantity is a member of the fuzzy term subset whose surface determines the mapping. An illustration of such a mapping and its associated quantitative and linguistic variables, and term subsets is in Figure Two.

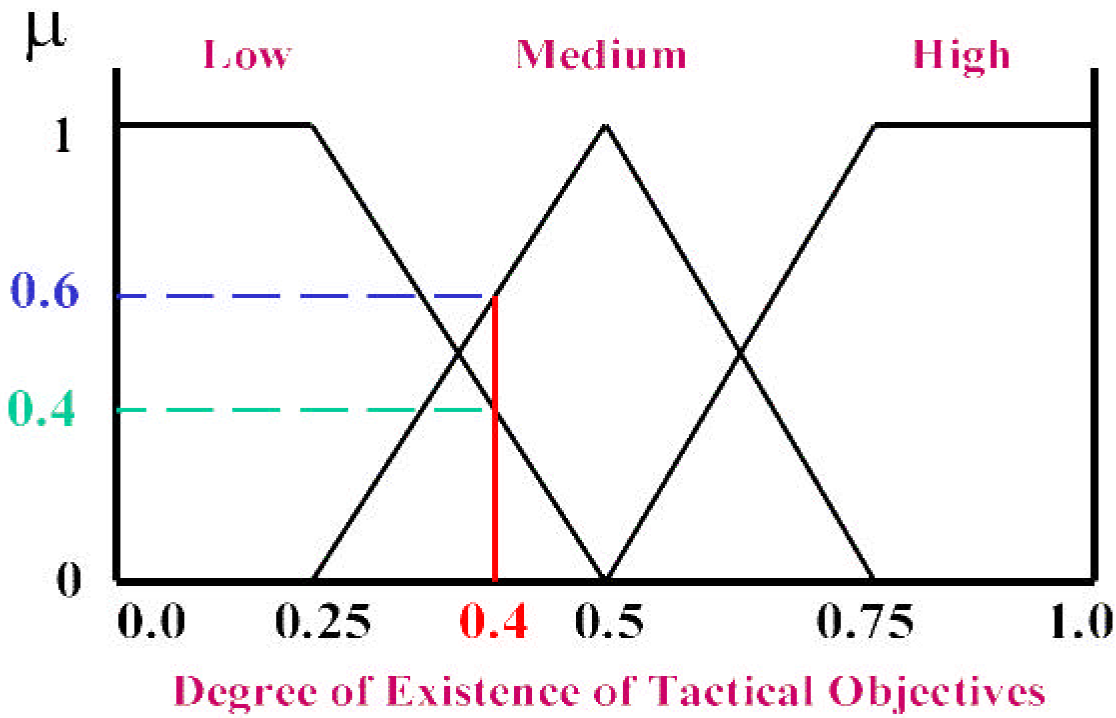

Figure Two -- Mapping A Quantitative Variable To A Linguistic Variable With a Membership Function

Figure Two shows a fuzzy linguistic variable with three fuzzy term subsets accompanied by three overlapping membership functions. The values of the quantitative variable are on the horizontal axis. The vertical axis provides the values of the membership functions corresponding to various points on the surfaces of these functions. The surfaces map values on the horizontal to both the term subsets and degrees of membership on the vertical axis.

For example, the value 0.4 units for degree of existence of tactical objectives maps to both the low and medium term sets. Its degree of membership in the low term set is 0.4. Its degree of membership in the medium term set is 0.6. Every other value on the horizontal axis is also mapped by one of the overlapping membership functions. The figure represents a complete measurement model for the quantitative variable relative to the linguistic variable and its fuzzy term subsets.

Once the mapping of quantitative to fuzzy linguistic variables and term sets is complete for all components of the composite, the actor is guided by the system in formulating the output variable. This variable may be a performance index, such as the metric on the quality of the knowledge base mentioned in Use Case Eleven. The actor selects the fuzzy term sets for the performance index, the shape of the membership functions, and the appropriate metric scale of the output quantitative variable. Degree of performance of any use case has a theoretical zero point, and full performance has a theoretical value of one, so the actor can specify the interval between zero and one as the range of values for the metric.

Next, the actor uses the system to formulate fuzzy rules connecting the input linguistic variables to the output. In the composite situation each of these rules have the form: If LVI(1) is A(1), and LVI(2) is A(2), and . . . LVI (n) is A(n) then LVO is B(1), where the LVI(1) . . . LVI(n) are linguistic variables input, A(1) . . . are fuzzy subsets (term sets), LVO is the linguistic performance output variable, and B(1) is a fuzzy output subset. The rules are linguistic expressions. An abbreviated example of such a rule is:

> **If degree of existence of the hierarchical network is high, and depth of the hierarchy is moderate, and dissemination of the hierarchy is medium, than performance in constructing and disseminating the hierarchy is moderate.**

In a composite with ten attributes, with seven term subsets per variable, and one output variable also with seven term sets, the number of possible rules is more than 282 million, a prohibitive number to model. Fortunately, Kosko [17] has shown that all multi-antecedent rules in a Fuzzy Associative Memory (FAM), can be composed from single antecedent rules, and therefore add no new information. In the ten attribute example, there are 490 such rules, a much more manageable number. The system will automatically generate the rules in a manner transparent to the actor.

Once the rules are generated, the actor needs to specify the degree of support for each rule. Degree of support is used in fuzzy inference to specify the actor's hypothesis about the validity of each rule. Degree of support can therefore be used to weight each rule in the process of inference from input to output fuzzy term subsets.

To get degree of support, the actor performs pair comparisons of the relative ability of each of the attributes of the composite to represent the abstract quantity as in section (i), above. The procedure produces a set of relative ability ratio scaled values of weights. These are the degrees of support to be applied in fuzzy inference. Degree of support is constant for all rules of a given linguistic variable, but varies across linguistic variables. In the case of the ten attribute composite, there would only be ten weights, each applying to 49 rules. The system would assign weights to rules for the actor.

When fuzzy inference is used in this type of measurement model, the scale values of the original attributes entering the composite are transformed into ratio scaled membership function values (varying between zero and one) by the membership functions specifying the term sets (see Figure Two). A non-zero membership function value of a member of a term set activates a fuzzy rule connecting a linguistic antecedent with a consequent *to the degree represented by the membership function value*. This degree of membership value is passed from the antecedent to the consequent in the inference process. So when inference is carried out, both a term set value (e.g., "performance is moderate") and a degree of membership value (e.g., 0.8) in the consequent term set, are deduced when using a fuzzy rule.

The values generated from a single rule are one element in a fuzzy surface generated by the full set of rules as they are applied to data describing an object. This fuzzy surface is the full information outcome of the fuzzy inference process.

To get from this outcome to a single ratio scale composite value, the actor needs to perform de-fuzzification. In de-fuzzification, the output surface generated by the fuzzy inference process is transformed into a single value most representative of the surface. Depending on the specific situation, different concepts of "most representative" can lead to different de-fuzzification calculations.

Here the centroid method of arriving at a single-valued output of the measurement process will be used. This method is essentially an average of the degree of membership values passed from the antecedent to the consequent terms during the fuzzy inference process [18]. Since the method operates on ratio scale values produced by the inference process, and computes a result based on the membership function values, the result is itself a ratio-scaled metric. In fact, in the performance index case mentioned above, the performance outcome values inferred by the fuzzy measurement model will vary over the interval from zero to one.

e. Perform internal validation of knowledge management metric.

The consistency measure is reported to the actor, along with the option to revise ratings by recycling through the ratings process, and advice on the need to revise ratings if the consistency measure exceeds a threshold level. When the actor is satisfied with the consistency of a set of ratings, the actor indicates acceptance.

Consistency across judges is also an aspect of internal validation. A dialog should support processing of comparisons of judgment matrices across judges, and provide measures of agreement/disagreement.

f. Perform external validation.

Strictly speaking, external validation is not a part of this task. It is part of the general KDD use case. There is no successful measurement modeling without KDD. Metrics are part of the model network that gets tested in an attempt to establish new knowledge. Metrics are finally established as valid only when the causal and dynamic models they are associated with survive testing.

Conclusion

This White Paper:

- characterized the nature of measurement and metrics development for knowledge management;
- showed how the OKMP and its use cases are related to the software application level of Distributed Knowledge Management Systems;
- presented the Perform Knowledge Discovery in Databases (KDD) use case which would provide the context for KM Metrics development at the software application level; and
- presented a general characterization of the Perform Measurement Modeling task within the Perform KDD use case focusing on development of ratio scaled metrics of quantitative abstractions.

Here are some possibilities for future developments.

First, KM Metric Templates based on the measurement modeling approach and methods would be a very desirable direction for new work. It is obvious that specific KM metrics will be tied to specific organizational contexts. Domain details affecting KM metrics will differ across organizations. Also, in the area of performance metrics, value interpretations entering composite performance metrics will differ across individual organizations. In spite of the diversity of specific KM metrics across organizations, it should, shortly, be possible to define general types of metrics: for example, use case performance metrics that are structurally similar across organizations. If these are formally defined as software patterns and are instantiated as user-oriented template components, they would then be ready for customization in organizational contexts.

Second, this paper has not covered very much in the area of KM Metrics conceptualization. It hasn't covered either KM domain metrics or business domain-based knowledge management metrics. Such metrics are necessary to provide utility to the abstractions I've provided here. The first step in developing these is conceptual specification of metrics (as distinct from data) in various business domains including KM. A future White Paper in this series will provide conceptualization in KM itself. In addition, the Knowledge Management Consortium (KMC) KM Metrics Task Force [19] is currently working vigorously in this area. But lots more effort is needed both in KM and in other domains.

Third, in this development, I haven't given attention to the environment of the OKMP, or to any other organizational processes and their relation to the OKMP. Clearly, a "balanced scorecard" viewpoint for organizations is needed to place KM Metrics in the broader context. Kaplan and Norton [20] distinguish Financial, Internal, Customer, and Learning and Growth perspectives on organizational processes essential to an overall strategy. Knowledge Management clearly fits within, if it does not define, the Learning and Growth aspect of their framework. If this is true, Knowledge Management outputs, the products of the use cases of the OKMP, will impact on other processes. I have not treated these impacts in this study of KM Metrics; but, because of its significance in measuring knowledge management benefits or costs to other processes in organizations, this is an important area for extending the present work.

Finally, if KM takes hold as a field, one of its main thrusts will be KM metrics development and implementation. In developing the above conception of the measurement modeling task, I've done some of the specification necessary for a software application in the area of KM Metrics development. It is doubtful that such an application should stand alone, but it is applicable as a part of an application fulfilling the "Perform KDD" use case. As yet no data mining products offer an application to facilitate measurement modeling and KM Metrics development.

References

[1] Brian Ellis, Basic Concepts of Measurement (Cambridge, England: Cambridge University Press, 1966), P. 41

[2] Carl G. Hempel, Fundamentals of Concept Formation in Empirical Science (Chicago: University of Chicago Press, 1952), 54-55.

[3] Joseph M. Firestone and Richard W. Chadwick, "A New Procedure for Constructing Measurement Models of Ratio Scale Concepts," International journal of General Systems, 2 (1975), 35-53.

[4] The OKMP, along with a number of other key concepts of knowledge management, is defined in my "Basic Concepts of Knowledge Management," White Paper at http://www.dkms.com/KMBASIC.html.

[5] A Business System Use Case is defined by Jacobson as "A sequence of transactions in a system whose task is to yield a result of measurable value to an individual actor of the business system." See Ivar Jacobson, Maria Ericsson and Agneta Jacobson., The Object Advantage: Business Process Reengineering with Object Technology (Reading, MA: Addison-Wesley, 1995), P. 343

[6] Ibid.

[7] I introduced the DKMS concept in two White Papers "Object-Oriented Data Warehouse," and "Distributed Knowledge Management Systems: The Next Wave in DSS." Both are available at http://www.dkms.com/White_Papers.htm.

[8] See Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasmy Uthurusamy (eds.), Advances in Knowledge Discovery and Data Mining (Cambridge, MA: M.I.T. Press, 1996).

[9] The hierarchy is an analytical hierarchy as defined by Thomas L. Saaty, The Analytic Hierarchy Process: Planning, Priority Setting, resource Allocation, 2nd Edition (Pittsburgh, PA: RWS Publications, 1990).

[10] The proposal is that the AHP be used to define the meta-knowledge validity concept, in much the same way as it is used to define other value interpretative concepts in terms of an analytic hierarchy.

[11] See S. S. Stevens "On the Theory of Scales and Measurement," Science, 103, (1946), 667-680.

[12] R. White and R. Gunstone, Probing Understanding (New York: Falmer Press, 1992), Bart Kosko. Neural Networks and Fuzzy Systems (Englewood Cliffs, NJ: Prentice-Hall, 1992), Robert Axelrod, (Ed.) The Structure of Decision (Princeton, NJ: Princeton University Press, 1976), Ted Kesik, Knowledge Mapping" @http://www.acs.ryerson.ca/~bsc/kmapmain.htm, Russell G. Almond, Graphical Belief Modeling (London, UK: Chapman & Hall, 1995), Francis Heylighen, "Structuring Knowledge in a Network of Concepts, @http://pespmc1.vub.ac.be/Papers/StructuringKnowledge.htm.

[13] Stephen T. Mohr, "Software Design For a Fuzzy Cognitive Map Modeling Tool," Rensselaer Polytechnic Institute, Fall 1997, Master's project.

[14] Saaty, 1990, Warren S. Torgerson. Theory and Methods of Scaling (New York: John Wiley, 1958), Firestone and Chadwick, 1975.

[15] Firestone and Chadwick. Ibid. Pp. 45-47.

[16] Ellis, 1966, P. 96.

[17] Kosko, 1992, Pp. 322-326.

[18] Ibid. Pp. 315-316.

[19] Work with the KMC KM Metrics Task Force led to this paper. See KMC's web site at http://www.km.org.

[20] Robert S. Kaplan and David P. Norton. The Balanced Scorecard (Boston, MA: Harvard Business School Press, 1996).

Biography

Joseph M. Firestone is an independent Information Technology consultant working in the areas of Decision Support (especially Data Marts and Data Mining), Business Process Reengineering and Database Marketing. He formulated and is developing the idea of Market Systems Reengineering (MSR). In addition, he is developing an integrated data mining approach incorporating a fair comparison methodology for evaluating data mining results. Finally, he is formulating the concept of Distributed Knowledge Management Systems (DKMS) as an organizing framework for the next business "killer app." You can e-mail Joe at eisai@home.com.

[Up] [KMBenefitEstimation.PDF] [MethodologyKIv1n2.pdf] [EKPwtawtdKI11.pdf]
[KMFAMrev1.PDF] [EKPebussol1.PDF] [The EKP Revisited]
[Information on "Approaching Enterprise Information Portals"]
[Benefits of Enterprise Information Portals and Corporate Goals]
[Defining the Enterprise Information Portal]
[Enterprise Integration, Data Federation And The DKMS: A Commentary]
[Enterprise Information Portals and Enterprise Knowledge Portals]
[The Metaprise, The AKMS, and The EKP] [The KBMS and Knowledge Warehouse]
[The AKM Standard]
[Business Process Engines in Distributed Knowledge Management Systems]
[Software Agents in Distributed Knowledge Management Systems]
[Prophecy: META Group and the Future of Knowledge Management]
[Accelerating Innovation and KM Impact]
[Enterprise Knowledge Management Modeling and the DKMS]
[Knowledge Management Metrics Development]
[Basic Concepts of Knowledge Management]
[Distributed Knowledge Management Systems (DKMS): The Next Wave in DSS]