An approach to improving existing measurement frameworks

by M. G. Mendonça
V. R. Basili
I. S. Bhandari
J. Dawson

Software organizations are in need of methods for understanding, structuring, and improving the data they are collecting. This paper discusses an approach for use when a large number of diverse metrics are already being collected by a software organization. The approach combines two methods. One looks at an organization's measurement framework in a top-down fashion and the other looks at it in a bottom-up fashion. The top-down method, based on the goal-question-metric (GQM) paradigm, is used to identify the measurement goals of data users. These goals are then mapped to the metrics being used by the organization, allowing us to: (1) identify which metrics are and are not useful to the organization, and (2) determine whether the goals of data user groups can be satisfied by the data that are being collected by the organization. The bottom-up method is based on a data mining technique called attribute focusing (AF). Our method uses this technique to identify useful information in the data that the data users were not aware of. We describe our experience in analyzing data from a software customer satisfaction survey at IBM to illustrate how the AF technique can be combined with the GQM paradigm to improve measurement and data use inside software organizations.

There are many different groups involved in the processes of developing, maintaining, and managing software. Those groups need to use measurement to characterize, control, predict, and improve those processes. We define a *measurement framework* (MF) as a set of related metrics, data collection mechanisms, and data uses inside a software organization. It is not uncommon to find measurement frameworks that are: (1) collecting redundant data, (2) collecting data that nobody uses, or (3) collecting data that might be useful to people who do not even know the data exist inside their organization. For these reasons, improving ongoing measurement is an important problem for many software organizations. We believe the solution for this problem needs to address two key issues: (1) how to better understand and structure this ongoing measurement, and (2) how to better explore the data that the organization has already collected.

This paper describes an approach that addresses these two critical issues jointly. The approach combines a knowledge discovery technique, called attribute focusing (AF), with a measurement planning approach, called the goal-question-metric (GQM) paradigm. In this approach, a GQM-based method is used to understand and structure ongoing measurement, and an AF-based method is used to discover new interesting information in the legacy data. We have been using our approach to analyze the cus-

0018-8670/98/$5.00 © 1998 IBM

tomer satisfaction (CUSTSAT) survey data at the IBM Software Solutions Division Toronto Laboratory. We describe some of our experiences to illustrate how the approach works.

The paper is organized as follows. First, we introduce the basic concepts associated with the GQM paradigm and the AF technique. Next, we introduce our approach to improve existing measurement frameworks. Then we report our experiences in applying this approach to the CUSTSAT measurement framework at the IBM Toronto Laboratory. We close the paper by presenting the main conclusions drawn from those experiences.

Background

This section introduces the basic concepts that will be needed to discuss our approach: the terminology used throughout this paper (adapted from data mining[1] and software engineering[2,3] terminology), the goal-question-metric paradigm, and the attribute focusing technique.

Terminology. We define *application domain* as the real or abstract system a software organization wants to analyze using an MF. An *entity* (object, event, or unit) is a distinct member of an application domain. Similar entities can be grouped into classes such as persons, transactions, locations, events, products, and processes. Entities are characterized by *attributes* and *relations* to other entities. An attribute (field, variable, feature, property, or magnitude) is a single characteristic of all entities in a particular entity class, for instance "usability" of software products or "size" of source code. In the case of a measurement framework, an attribute defines what one wants to measure. A *relation* is a set of entity tuples that has a specific meaning, for instance "a is married to b" (for person entities "a" and "b"). We measure entity attributes to empirically define relations between entities; for instance we can determine the relation "a is heavier than b" by weighing entities "a" and "b."

Measurement is the process of assigning a value to an attribute. A *metric* is the mapping model used to assign values to a specific attribute of an entity class. A metric states how we measure something. It usually includes a measurement instrument, a value domain, and a scale. *Data* are sets of measured (collected, polled, surveyed, sensed, or observed) attribute values produced by specific metrics for certain *user groups*.

A *user group* is a formal group inside the organization that in some way utilizes (consumes, employs) the data produced by the MF. A *data use* is a description of the way a user group consumes the data. And, a *data user* is any member of a user group. A *data manager* is a person responsible for managing the collection and storage of, and access to, the data in a measurement framework. A person may play both roles—data manager and data user—in a given MF.

A measurement *goal* is an operational, tractable description of a user group objective in using the data. In this paper, a goal is always described using the template we introduce later. *Domain knowledge* is nontrivial and useful empirical information specific to the application domain believed to be true by the data users. *Background knowledge* is the domain knowledge that data users had before analyzing the data. And, new or *discovered knowledge* is the new domain knowledge that data users gain by analyzing the data.

The GQM paradigm. The goal-question-metric paradigm[4,5] is a mechanism for defining measurement in a purposeful way. It supports the definition of measurement goals tailored to the specific needs of an organization. Goals are defined in an operational, tractable way, as a set of quantifiable questions. Questions in turn imply a specific set of metrics and data for collection. This paradigm has been used successfully in several organizations, e.g., the National Aeronautics and Space Administration (NASA),[6] Motorola, Inc.,[7] and Hewlett-Packard Company.[8]

The GQM paradigm has been implemented in several different ways. In our work, we use our own version. Figure 1 shows an abstract example of what we call a GQM structure.

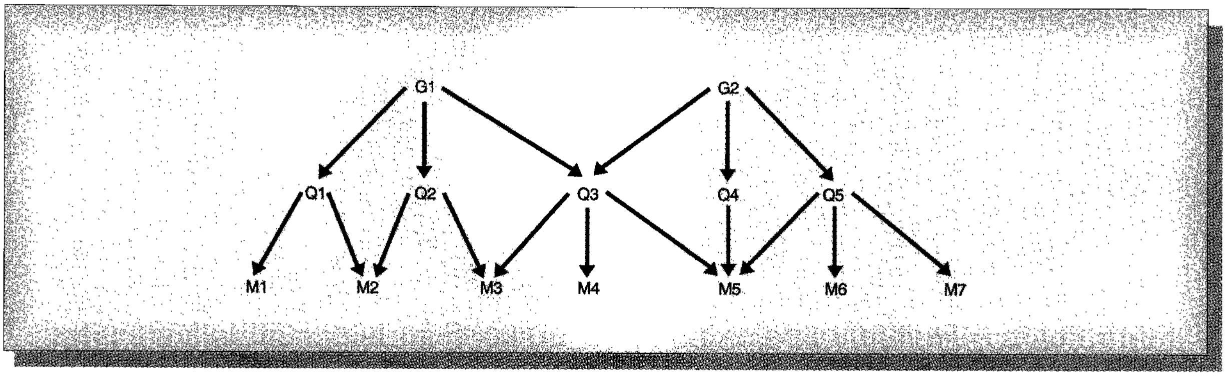
Typically, we use the following template to define measurement goals:

Analyze *"object of study"* in order to *"purpose"* with respect to *"focus"* from the point of view of *"point of view"* in the context of *"environment."*

Each of the italicized phrases represents a *facet* that must be considered in measurement planning. For example:

Analyze *service support for our product* in order to *evaluate it* with respect to *customer satisfaction* from the point of view of *service support personnel* in the context of the *small data processing companies.*

Figure 1 An abstract GQM structure



Each goal implies several questions based on its facets. For example, the purpose "evaluate" might generate questions of the type: "How does the service support of our product compare with its competitors?" or "How does the current service support satisfaction compare with previous years?" The questions will then be refined into the metrics needed. The goal facets are also used in this process. For example, the point of view determines the scale, granularity, and timing of the metrics used to answer a certain question.

In our approach, each GQM structure will specify the goals associated with a certain data user group (goals with the same "point of view"). Each structure will allow us to trace the goals of a certain user group to the measures that are intended to define the goals operationally. It will also provide a platform to interpret the data and better understand the data user needs.

The attribute focusing technique. Attribute focusing (AF) is a data mining technique that has been used in several different applications—including software process measurement,[9–11] customer satisfaction,[12] and sports[13] data analyses. This technique involves one or more experts in the knowledge discovery process.

The AF tool[14] searches an attribute-value (measurement) database for *interesting facts*. An interesting fact is characterized by the deviation of attribute values from some expected distribution or by an unexpected correlation between values of a set of attributes. The facts are presented in easily interpretable bar chart diagrams. The diagrams are sorted by *interestingness level*—a numeric value calculated to quantify how interesting each diagram might be to an expert.

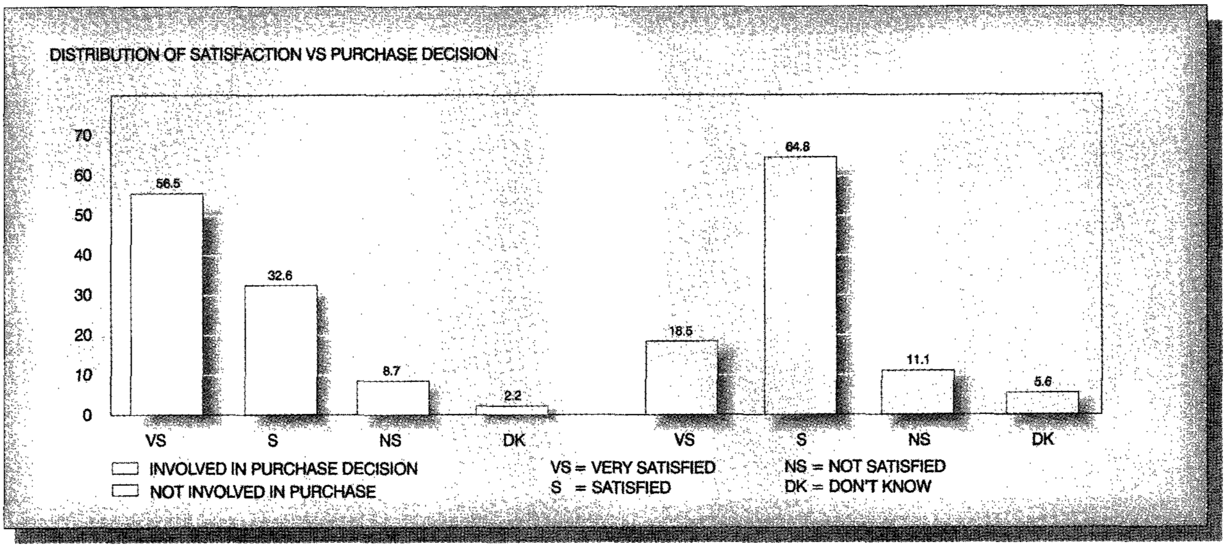
The ordered diagrams are presented to the experts. Knowledge discovery takes place when the experts address the questions raised by the diagrams. This technique can be called machine-assisted knowledge discovery because the computer program guides a person to discover interesting facts in the data.

Figure 2 shows an example of an attribute focusing diagram. It was obtained from a real data set pertaining to a particular class of software products.[12] Let us call it "Product Class X." This particular diagram has two attributes: overall satisfaction and customer involvement in the decision to purchase the product.

The satisfaction level by customer involvement in purchase is shown by bar patterns in the diagram. The possible values are: "involved in purchase decision," if the customer was involved in the decision to purchase the product he or she is evaluating, and "not involved in purchase," if not. The y-axis shows the percentage of occurrence of each "satisfaction" value per "purchase involvement" value. For example, the first vertical bar indicates that about 56.5 percent of those involved in the decision to buy the product were very satisfied with the product.

The diagram shows that if the customer was involved in purchasing a product of Product Class X, he or she is likely to evaluate the product more favorably than other customers who were not involved in the decision to buy this product (see the differences in

Figure 2 A two-way attribute focusing diagram



values between "very satisfied" and "satisfied" for "involved" and "not involved in purchase" decision). This diagram exemplifies how the AF tool helps knowledge discovery. It points out new facts to the experts. These facts may or may not lead to discovered knowledge. The experts are the ones who will look at the facts expressed in the diagrams, using their background knowledge, and conclude if the diagrams are saying something new and useful.

Suppose, for example, that the experts know that products of a certain class are expensive (background knowledge). This might lead to the discovery that purchasers of this class of products try to defend the product in order to justify their decision to invest in it.

"Interestingness" functions. The diagram in Figure 2 is said to be a two-way diagram. The function used to calculate the interestingness level of a two-way diagram—involving two attributes A_x and A_y in nominal or ordinal scale—is:

Interestingness $(A_x, A_y) = \max[\text{In}_2(A_x = v; A_y = u)]$, over all possible values v and u

$\text{In}_2(A_x = v; A_y = u) = |\text{Observed}(A_x = v) \times \text{Observed}(A_y = u) - \text{Observed}(A_x = v \wedge A_y = u)|$

The "In_2" function quantifies the correlation between A_x and A_y. It calculates the probability of co-occurrence of two particular values u and v as if the attributes were independent,[15] and subtracts from it the rate of occurrence of the combination observed in the data.[16]

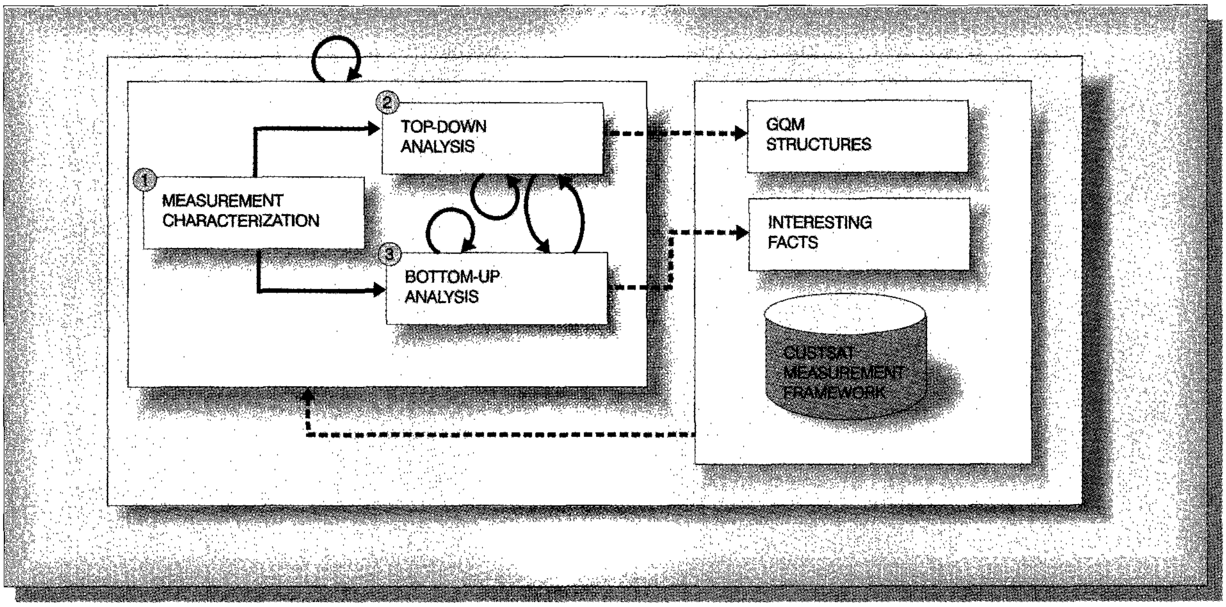
Other interestingness functions can be used with the AF technique. In our work, we have used functions that estimate the interestingness level for associations between an arbitrary number of attributes (n-way analysis). Colet and Bhandari[14] provide a good discussion on the concept of interestingness and the principles behind the AF technique.

The approach

As mentioned before, the purposes of our approach are to better: (1) understand the ongoing measurement, (2) structure the measurement, and (3) explore the MF legacy data. Our approach is divided into three phases: MF characterization, top-down analysis, and bottom-up analysis. The approach is depicted in Figure 3.

The first phase—characterization—is executed to identify the current and prospective data user groups and how they are (or could be) using the data. The second phase—top-down analysis—is based on the GQM paradigm. It is executed to capture the goals of the data users and to map these goals to the metrics and data in the MF. The third phase—bottom-up

Figure 3 The approach



analysis—is based on the AF technique. It is executed to extract knowledge (useful, interesting, and nontrivial information) from the already existing data. The approach combines these three phases to tackle the following practical questions:

1. How are the data being used inside the organization?
2. What are the goals of the data users in using the data?
3. Can these goals be satisfied by the data being collected?
4. What data should the organization consider not collecting anymore? What data should the organization consider beginning to collect?
5. What useful information is there in the data that data users are not aware of?

Figure 3 shows the information flow (dashed lines) and control flow (solid lines) of this process. The two main products of our approach are: (1) GQM structures, produced by the top-down analyses, and (2) interesting facts, produced by the bottom-up analyses.

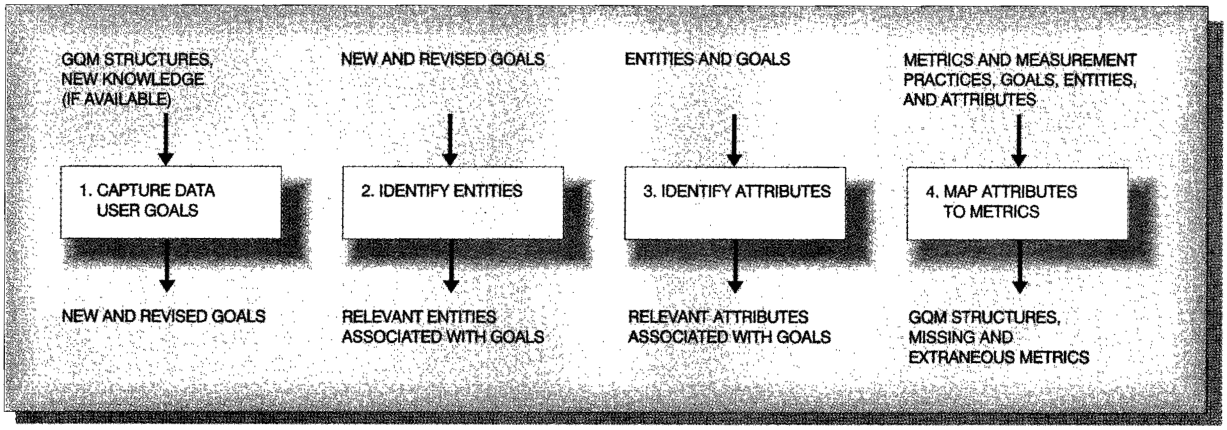
The control flow described by solid arrows in Figure 3 is determined by the interaction between the phases. The characterization results are used to execute the bottom-up and top-down analyses. Thus, the characterization can be seen as a prerequisite for the other two phases. The top-down and bottom-up phases can interact with each other. Interesting facts discovered during bottom-up analyses can lead to new measurement goals for the top-down analyses. Measurement goals can in turn be used to define new data sets for the bottom-up analyses.

The top-down and bottom-up analyses are designed to be applied incrementally. Our basic unit of analysis is a user group (also called a point of view). This makes it possible to use our approach to incrementally improve large MFs—one point of view at a time.

The measurement framework characterization. This first phase is executed to identify key components of a measurement framework (MF) and document how they relate to one another. These components are the metrics, attributes, data, user groups, and data uses.

We use a combination of structured interviews[17,18] and review of the available MF documents to capture and document those key components. The descriptions of metrics and available data can usually be obtained in the MF documents. The descriptions of user groups, attributes, and data uses are usually

Figure 4 The GQM-based method


GQM STRUCTURES, NEW KNOWLEDGE (IF AVAILABLE)
1. CAPTURE DATA USER GOALS
NEW AND REVISED GOALS
NEW AND REVISED GOALS
2. IDENTIFY ENTITIES
RELEVANT ENTITIES ASSOCIATED WITH GOALS
ENTITIES AND GOALS
3. IDENTIFY ATTRIBUTES
RELEVANT ATTRIBUTES ASSOCIATED WITH GOALS
METRICS AND MEASUREMENT PRACTICES, GOALS, ENTITIES, AND ATTRIBUTES
4. MAP ATTRIBUTES TO METRICS
GQM STRUCTURES, MISSING AND EXTRANEOUS METRICS

obtained by interviewing data managers and data users.

We use the following process to characterize a measurement framework:

Step 1—Identify metrics. The first components to be identified are the metrics used in an MF. Each of the metrics used to collect data must be listed, with an explanation of how it works—especially the measurement instrument, scale, and value domain. The metric scale will bound the type of operations that can be executed with the data. The measurement instrument and value domain will help in evaluating the precision of the data.

Step 2—Identify available data. The next components to be identified are the data available in the MF. This includes when and under what circumstances the metrics were used to collect data, where the resulting data were stored, and how to access the data. This last step may require an understanding of the format in which the data are stored and how to get authorization to use them. This may require a sizable amount of work if the data are stored in several different formats or locations.

Step 3—Identify data uses and user groups. Other components to be identified are data uses. Each type of data analysis and presentation that is generated with the data must be described. Each description should include the frequency and granularity with which the data are used. Together with the data uses, the users of the data must be identified. A user group description should include the objectives of the group as well as the importance of the data to the group.

Step 4—Identify attributes. The last components to be described are the attributes. This information is obtained by asking the user groups to describe their perception of what is being measured by the metrics. For example, when a person says a program size is 15 000 lines of code, it means that the metric "lines of code" is being used to measure the attribute "size of a program." It is important to make sure that user groups are correctly and consistently interpreting the meaning of the metrics used in an MF.

The top-down analysis. This phase is used to capture the data user goals and to map them to the data that are being collected. This helps to gain better understanding of the data user needs. It also helps to identify missing and extraneous metrics in the MF.

The top-down analysis uses a method based on the goal-question-metric paradigm. This method is applied to build (or revise) a structure that maps the data user goals to the metrics (and data) used in the organization. It is this structure that is used to identify missing or extraneous elements of an MF.

The GQM-based method is an example of how the principles proposed by the GQM paradigm can be applied when an MF is already established in an organization. Its objective is to build a GQM structure for each data user group. This structure is built by interviewing representatives of a data user group. Figure 4 shows the process for building a GQM structure.

Step 1—Capture data user goals. The method starts by capturing the goals of the user group, using the goal template described earlier. Using this template, each goal is described in terms of three facets: object of study, focus, and purpose (in the GQM-based model, the point of view is the user group itself).

The object of study is the entity that the user group wants to analyze (e.g., a particular product). The focus is the primary attribute that the user group wants to measure in order to analyze that entity (e.g., customer satisfaction). The purpose outlines what the user group wants to do with the object of study (e.g., evaluate it). Usually, the purpose has to be explained in detail by the user group representatives being interviewed.

The description of the data user goals is supported by the informal description of the data user objectives, obtained during the characterization process. Previous GQM structures and new knowledge (as defined earlier) are also used as inputs to this step when they are available.

Step 2—Identify relevant entities. The next step is to identify the entities for which attributes are to be measured—what we call "relevant entities." The relevant entities can be identified in two ways: (1) asking about them during the interview with the representatives of the data user group, or (2) looking for them in the documentation available about the object of study.

Usually, two entities can directly be derived from each goal: one is the object of study itself and the other is the entity with which the focus attribute is associated. We identify other relevant entities by finding out which entities are related to the object of study and which entities may affect the focus from the data user group point of view.

Consider the goal in our previous example. There are two relevant entities listed: the service support process (object of study) and the customer (related to the focus). The other relevant entities might be: the product, the support team, the problem, and the provided solution.

Step 3—Identify relevant attributes. The next step is to identify the attributes to be measured to achieve this goal—what we call "relevant attributes." For each relevant entity, we prepare an initial list of attributes that might be relevant for the stated goal. In order to produce a comprehensive list of attributes for each entity, we use a checklist based on the entity type (see sidebar).

The initial list of relevant attributes must be reviewed and expanded by the user group representatives during the interview. The end result of this step should be a list of attributes classified according to their relevance to the user group's goals.

Step 4—Map attributes to existing metrics. The last step is to map the relevant attributes to metrics that are being used in the organization. An attribute states what the user group wants to measure while the metrics define how the measurement is done. The mapping consists of checking to ensure that the metrics are measuring the desired attributes.

At this step, a partial GQM structure is assembled for the user group. This structure shows the mapping between the user goals, the relevant entities, the relevant attributes, and the metrics used in the MF, documenting the data user group's measurement needs.

At the end of this step we can derive a list of inconsistent, missing, and extraneous metrics from the user group point of view. We detect a missing metric when a relevant attribute has no metric to measure it. We detect an extraneous metric when a metric has no corresponding attribute in the GQM structure. We detect an inconsistent metric when a metric used to measure a relevant attribute is not consistent with the user group goals. Typical consistency problems occur when: (1) the metric's scale or range of values is not suitable for the user group needs, (2) the cost to apply a metric is unacceptable, or (3) a metric cannot be applied when or where it is needed by the user group.

The bottom-up analysis. The data already collected by an organization are the most important asset of any MF. It is important for an organization to have means to explore its legacy data. We believe that intelligent data exploration methods are an effective way to understand and learn more about the organization's business. We refer to them as bottom-up methods, because the raw data are the starting point for better use and understanding of the data themselves.

The top-down analyses are aimed at better planning for and execution of data collection. The bottom-up analyses are aimed at discovering new and useful information in the existing data, thus improving data awareness and data usage. The literature has many examples of the use of machine learning techniques to extract knowledge (new and useful information) from software engineering data sets.[19-24] Our bottom-up analyses use attribute focusing[25]—a data mining technique—to extract unexpected and useful information directly from the MF database.

The aim of the AF-based method is to establish procedures to effectively apply the AF technique—maximizing knowledge discovery and minimizing discovery cost.

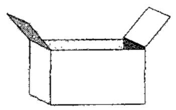
In the case of a measurement framework, the "experts" in the knowledge domain correspond to the MF data users. In this context, the bottom-up method allows the data users to gain knowledge about: (1) their application domain (learn about the things they are measuring), and (2) the components of the measurement process (learn about the way they are measuring things).

In order to effectively apply the AF technique, the method goes through the five steps shown in Figure 5. In the first two steps, the people in charge of applying the method to the legacy data (i.e., data analysts) interact with the data users to define the type of analysis that will be done. In the next two steps, the data analysts run the AF tool and organize the obtained results. In the last step, the results are reviewed by the data users. That is when knowledge discovery takes place.

Step 1—Establish relationship question. In the AF technique, it is very important to avoid the computation of uninteresting relations whenever possible. An uninteresting relation wastes machine time to compute, and yields uninteresting diagrams that will waste the user's time during the diagram reviews.

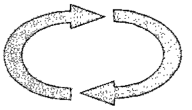
The AF tool avoids uninteresting metric combinations based on user-defined data (metric) groups. The metrics grouped together are not correlated during the analysis. For example, a typical two-way AF analysis will use two metric groups. The AF tool will pick one metric from each group and try to correlate the pair.

Our method uses generic *relationship questions* to select and group the data for the AF analyses. A relationship question states the relations between relevant attributes to be investigated empirically. A typical relationship question has the format:



If the entity is a *product*, consider:

1. Quality attributes (number of defects, changes, stability, reliability, etc.)
2. Logical attributes (functionality, capability, usability, maintainability, etc.)
3. Physical features (size, complexity, modularity, coupling, etc.)

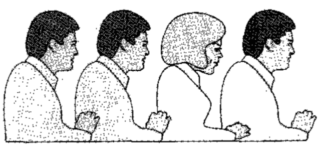


If the entity is a *process*, consider:

1. Physical attributes (size, complexity, etc.)
2. Managerial constraints (budget, schedule, quality targets, etc.)
3. Conformance in execution (how well the process was performed, unforeseen events, etc.)

If the entity is a *person* or a *team*, consider:

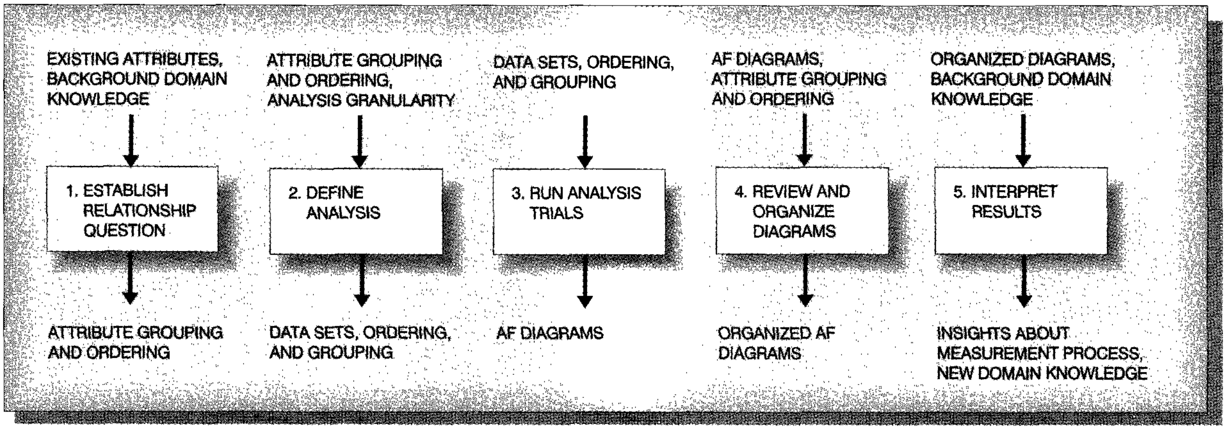
1. Career prospects (professional motivation, position in the organization, role in projects, etc.)
2. Familiarity with processes and products (experience and training with programming languages, tools, virtual machines, etc.)
3. Familiarity with the application domain (areas of expertise, experience and training in relevant knowledge areas, etc.)



If the entity is an *organization*, consider:

1. External environment (areas of activity, market niche, type of competitor, suppliers, customers, etc.)
2. Physical attributes (organizational structure, gross expenditure, gross sales, number of software installations, type of platforms, etc.)
3. Management features (rate of innovation, expenditure with software, views of software technology, etc.)

Figure 5 The AF-based method


EXISTING ATTRIBUTES, BACKGROUND DOMAIN KNOWLEDGE
ATTRIBUTE GROUPING AND ORDERING, ANALYSIS GRANULARITY
DATA SETS, ORDERING, AND GROUPING
AF DIAGRAMS, ATTRIBUTE GROUPING AND ORDERING
ORGANIZED DIAGRAMS, BACKGROUND DOMAIN KNOWLEDGE
1. ESTABLISH RELATIONSHIP QUESTION
2. DEFINE ANALYSIS
3. RUN ANALYSIS TRIALS
4. REVIEW AND ORGANIZE DIAGRAMS
5. INTERPRET RESULTS
ATTRIBUTE GROUPING AND ORDERING
DATA SETS, ORDERING, AND GROUPING
AF DIAGRAMS
ORGANIZED AF DIAGRAMS
INSIGHTS ABOUT MEASUREMENT PROCESS, NEW DOMAIN KNOWLEDGE

How does *attribute x* relate to *attribute y*?

In our AF-based method, we want to investigate several empirical relations in each analysis. We use a *generic relationship question* (GRQ) to state the set of relations to be investigated empirically. The following template[26] is used to define a GRQ:

How do *"attribute class x_1"* and . . . and *"attribute class x_{n-1}"* [relate to, affect, impact] *"attribute class y"*?

An *attribute class* is a set of attributes grouped according to certain criteria or features relevant to a user group. For example, attributes that represent logical features of the final products could be grouped in class *y*, while attributes representing managerial constraints over the project could be grouped in class x_1. In this example, the above template would result in the following question for AF analysis:

How do the *managerial constraints over the project* relate to the *logical features of the final products*?

The GRQs can be determined by: (1) interviewing user group representatives or (2) directly analyzing user group measurement goals. In the latter case, it is very useful to have a GQM structure defined for the user group.

Step 2—Define the analysis. After establishing a GRQ for an AF analysis, the analysis itself must be defined. First, attributes identified in the GRQ must be mapped to the metrics in the MF. This is straightforward, if the information collected during the characterization phase is used.

Next, the data granularity and scope of analysis must be determined. Consider, for example, our previous relationship question:

How do the *managerial constraints over the project* relate to the *logical features of the final products*?
Scope of the analysis: What products and projects should we consider?
Granularity: Should we analyze the data for the products individually or should we analyze classes of products?

The scope and granularity should be directly derived from the user group goal and data use descriptions. The key here is to understand the group's purpose in using the data.

The data sets are extracted after the scope and granularity of the analysis are defined. This task is usually simple, but it may take a sizable amount of effort if the data are not easily retrievable. The data sets may also need to be preprocessed and formatted to meet the data user and AF tool format requirements.

Step 3—Run the analysis. The next step is to run the tool itself. This step is almost completely automated. The inputs are: (1) the metric groupings, (2) the maximum number of diagrams (relations) to be produced, (3) the interestingness cutoff level, and (4) the analysis dimension. The groupings are directly derived from the GRQs as previously discussed. The maximum number of diagrams is based on the time that the data users can spend looking at the diagrams. The interestingness cutoff determines the minimum interestingness value for which the tool will produce a diagram for a given relation. The higher this cutoff is, the more interesting (and less numerous) the produced diagrams are. The analysis dimension determines the maximum number of metrics that can appear in a diagram (e.g., a Type-3 analysis results in up to three-way diagrams).

Step 4—Organize the diagrams. Although many uninteresting potential diagrams have already been pruned away in constructing the metric groupings, there may still be diagrams that are unsuitable for data user review. The next step is to manually review the diagrams. It may be necessary to rerun the analysis trials if: (1) too few diagrams were found for a given cutoff, or (2) missing or skewed data are driving the discoveries.

After a sizable number of useful diagrams have been compiled, we organize them to facilitate the data user's inspection. We group diagrams using the following algorithm:

1. Organize all the N diagrams obtained from the AF tool by order of interestingness.
2. Discard diagrams that are clearly uninteresting.
3. Following the order of interestingness given by the AF tool, select the first diagram.
4. Select all other diagrams that have the same set of explanatory metrics, and an explained metric in the same group as that of the diagram selected in Step 3.
5. Group all the diagrams selected in Steps 3 and 4 by order of interestingness in a unique "explanatory group" to be shown together to the data users.
6. Remove the diagrams gathered in Step 4 from the overall group of produced diagrams and return to Step 3, while diagrams remain.

This procedure produces several groups of diagrams to be shown to the data users. Diagrams in each group have the same explanatory metrics and related explained metrics. This allows the data users to concentrate on a unique reasoning thread while looking at each group.

Step 5—Interpret results. The last step of the AF-based method is analysis of the diagram groups by the data users. The diagram groups contain many types of information:

1. Unexpected correlations between metrics (direct analysis of an n-way diagram)
2. Unexpected value distributions (direct analysis of a one-way diagram)
3. Unexpected consistencies (or inconsistencies) in the relationships between explanatory metrics and related explained metrics (directly from analysis of a diagram group)

New knowledge is gained when the data users apply their background knowledge to interpret the information contained in the diagrams. There are two types of domain knowledge to be gained in this way: (1) insights into the application domain, and (2) insights about the components of the measurement process. The first is what is traditionally expected from the AF technique. The technique helps the experts to gain new insights into their activities. These insights may lead the data users to take adaptive, corrective, or preventive actions to improve the way they do business. The second happens when the AF diagrams lead the data users to realize that some previous assumption about the data or measurement process is incorrect. This may lead them to modify their measurement goals, metrics, predictive models, or data collection procedures.

Applying the approach

In this section, we report our experience in applying our approach to IBM Toronto's customer satisfaction CUSTSAT measurement framework. The CUSTSAT data are collected annually through surveys carried out by an independent party. Their purpose is to evaluate customer satisfaction with products of IBM's Software Solutions Division and competing products. The IBM Toronto Laboratory is only one of several IBM Software Solutions laboratories that use the CUSTSAT data. At IBM Toronto, the CUSTSAT data are used by several different groups (e.g., development, service, support, and senior management).

IBM surveys a large number of customers from several different countries. All the data are stored in one database. Currently, this database stores CUSTSAT data collected over several years. The large amount of data and the diversity of groups interested in the data made it desirable to apply our approach. Our two main objectives were: (1) better understanding of the user groups' needs with respect to the CUSTSAT measurement, and (2) better exploration of the data already stored in this database.

We effectively started this work in the summer of 1995. Most of the CUSTSAT MF characterization and some AF analyses were done that year. In the summer of 1996, we updated our MF characterization, ran several AF analyses, and executed the top-down analysis reported in this section. In 1997, we replicated some of the AF analyses we did in 1996, and executed top-down analyses for two other CUSTSAT data user groups.[27]

The characterization step. The first step was to document the metrics that composed the CUSTSAT MF, the user groups, and how those groups used the data. We focused on the data related to the Toronto Laboratory products. We did not work with any metrics or user groups associated with products developed in other IBM laboratories. The information collected in this step was gathered from existing documents in the MF (e.g., the survey questionnaire) or by interviewing the data manager responsible for the CUSTSAT data in the Toronto Laboratory.

The task of identifying what metrics composed the CUSTSAT framework was simple. Most of the metrics corresponded to questions in the survey questionnaire. The exception was the customer contact information stored separately in the CUSTSAT database. Next, we recorded the "meanings" of each metric. This corresponds to the attribute we believed the metric was measuring. This task was facilitated by the formulation of the questions on the questionnaire. The question usually explained what was to be measured. Terms like "capability," "performance," or "maintainability" were explained when they were used. Overall, we identified more than 100 attributes that were measured (mostly in ordinal scale) in the CUSTSAT framework.

Next, we described the data uses and user groups. This was done by interviewing the data manager. The interviews were somewhat structured. We had a checklist for the information we wanted to collect during the interviews, but we did not establish an ordering in which this information was to be collected. We started by: (1) listing the data analyses and data presentations that used the CUSTSAT data, and (2) identifying the persons who used those analyses or were present at those data presentations.

Each type of data analysis or presentation (DA/P) corresponded to a distinct data use. The data use descriptions included the frequency with which the DA/Ps were done, the list of metrics used in them, the granularity and scope of the DA/Ps, and the list of groups that took part in the DA/Ps. The list of user groups was compiled by mapping the list of persons who used the DA/Ps to the formal groups inside the laboratory. The user group descriptions included a statement of the data manager's perception of the group's objectives in using the data and a subjective ranking of the importance of the CUSTSAT data to the group.

Overall, we have identified 17 user groups that can be divided into four major areas: senior management, database development, compiler development, and support (e.g., market analysis, marketing, and sales). We identified about 16 different DA/Ps (data uses) associated with the CUSTSAT data. We also found that the amount and type of data usage varies even between user groups with similar goals. This allowed us to identify data usage experiences that can be transferred between groups.

The top-down analyses. We applied our GQM-based method to a limited number of data user groups in order to test the method feasibility and effectiveness. We built GQM structures for three user groups to propose improvements in the CUSTSAT questionnaire based on the obtained results. The three groups chosen were associated with the database product development at the laboratory:

1. The database customer service and support group
2. The database usability (user interface design) group
3. The database information development (documentation) group

In this section, we describe the building of the GQM structure for the database service support group to illustrate how we have applied the GQM-based method in the CUSTSAT MF. The service group gives vendor support to the client's database installations. Its responsibilities are to give fast resolution for client problems and provide permanent solutions to prevent these problems from recurring.

We used a structured interview[18] to build the GQM structure for the service support group. We interviewed a senior representative of the group. All the material for the interview was prepared beforehand. It included:

- A complete list and description of the metrics and DA/Ps associated with the service support group
- A tentative description of our perception of their goals
- A tentative list of entities and attributes that we believed were relevant for them

- A complete list of questions and topics to be discussed during the interview

During the interview we wanted to capture: (1) the database service support group goals in using the CUSTSAT data, (2) the relevant entities associated with their work, (3) the relevant attributes they want to measure through the CUSTSAT survey, and (4) the metrics (questions) that are effectively measuring them. We also wanted to use this interview to validate and rate the importance of the DA/Ps and metrics associated with the service support group. These last activities are part of the MF characterization step.

First we asked for their comments on the data analyses and presentations (DA/Ps) prepared for the group. We had two objectives: (1) to motivate and focus the rest of the interview around the CUSTSAT MF, and (2) to validate our understanding of the group's data usage (including assessing the importance of the data to them).

Next we attempted to capture their goals in using the CUSTSAT data. This was supported by the previous discussion of the group data usage. We asked the group representative what the group wanted to achieve in using the CUSTSAT data and expressed it in the form of GQM goals. We captured the following goals:

1. Analyze the service support process in order to characterize its key areas with respect to customer satisfaction and dissatisfaction.
2. Analyze the customer set in order to understand expectations with respect to support service.
3. Analyze the service support areas where customers were dissatisfied in order to improve them with respect to customer satisfaction.

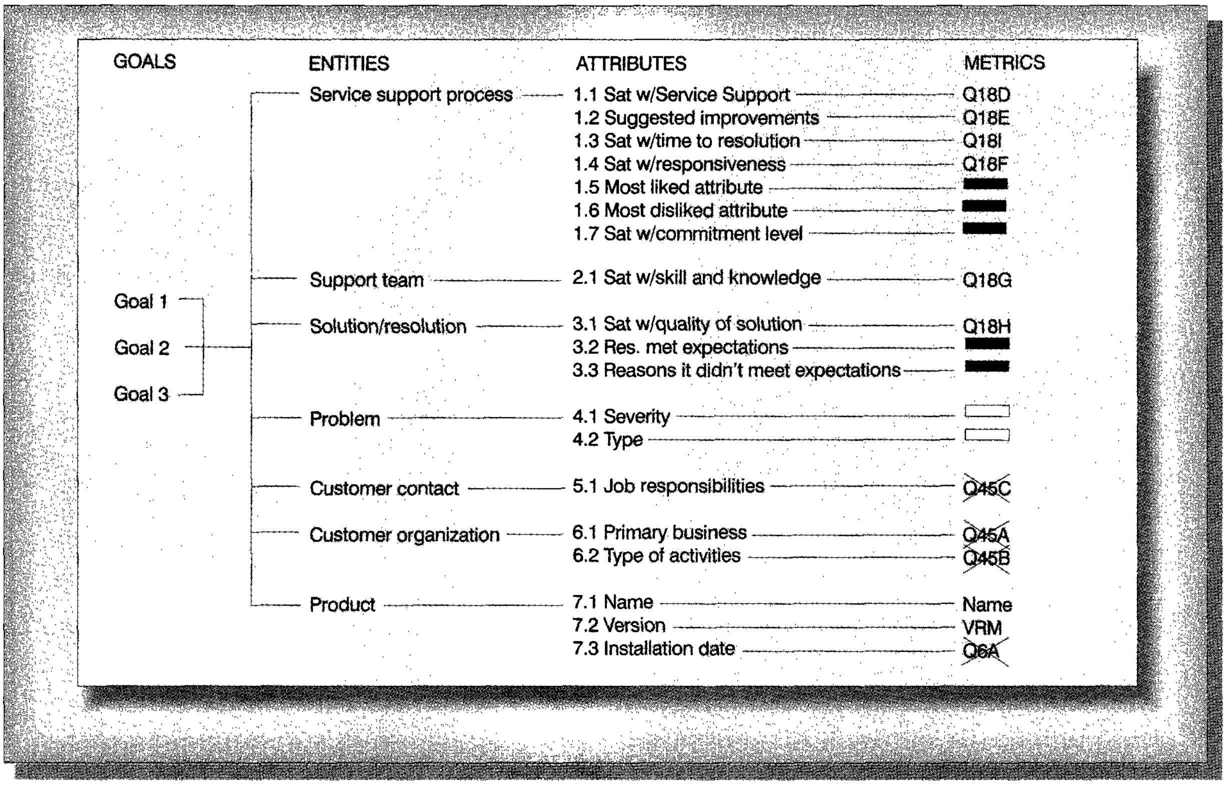
Table 1 Relevant entities and attributes

Entity 1: Service support (SS) process	
Attribute 1.1:	Overall customer satisfaction with SS
Attribute 1.2:	Improvements suggested to the SS by the customer
Attribute 1.3:	Customer satisfaction with time to resolution
Attribute 1.4:	Customer satisfaction with SS responsiveness
Attribute 1.5:	Aspects customer liked most about the SS
Attribute 1.6:	Aspects customer disliked most about the SS
Attribute 1.7:	Customer satisfaction with the SS commitment level
Entity 2: Support team	
Attribute 2.1:	Customer satisfaction with support team skill and knowledge
Entity 3: Solution/resolution provided	
Attribute 3.1:	Quality of the solution
Attribute 3.2:	Degree to which resolution met expectations
Attribute 3.3:	Reasons why resolution did not meet expectations
Entity 4: Reported problem	
Attribute 4.1:	Severity
Attribute 4.2:	Type
Entity 5: Customer contact (surveyed person)	
Attribute 5.1:	Role in organization (job responsibilities)
Entity 6: Customer organization	
Attribute 6.1:	Primary business
Attribute 6.2:	Type of activities
Entity 7: Product being supported	
Attribute 7.1:	Product name
Attribute 7.2:	Product version
Attribute 7.3:	Date the product was installed in the organization

The next step was to discuss the relevant entities, attributes, and metrics associated with those goals. We started by identifying the relevant entities. From the entities and goals, we discussed the relevance of the attributes shown in Table 1. This list includes the attributes associated with existing metrics as well as new attributes suggested by the interviewees. In the case of new attributes, it was important to make sure that we understood and recorded their meaning. Let us consider Attribute 1.7 as an example. According to the interviewee, this attribute refers to the degree to which the service support meets the commitment level contracted by the customer. Those levels are established in the support contract and correspond to well-defined preconditions on the time that IBM should take to provide satisfactory resolution of customer problems.

Figure 6 depicts the GQM structure for the service support group. It shows the mapping from the attributes to the metrics (questions in the survey questionnaire). In the structure, the metrics are referred to by the question number in the survey questionnaire. The rectangles indicate that the attribute was suggested by the interviewee's goals but was not being measured. From Figure 6, we concluded that there were seven missing metrics from the service support point of view (rectangles). These metrics are needed to measure Attributes 1.5, 1.6, 1.7, 3.2, 3.3, 4.1, and 4.2. However, Attributes 4.1 and 4.2 (open

Figure 6 GQM structure for the service support group



rectangles) were considered too difficult to measure.[28]

The crossed-out metrics—Q45C, Q45A, Q45B, and Q6A—indicate that their associated attributes were not relevant to the service support group. They were extraneous from the service support point of view. The structure depicted in Figure 6 also indicates that no information on the "customer contact" and "customer organization" entities is needed to achieve the goals currently established by the service support group.

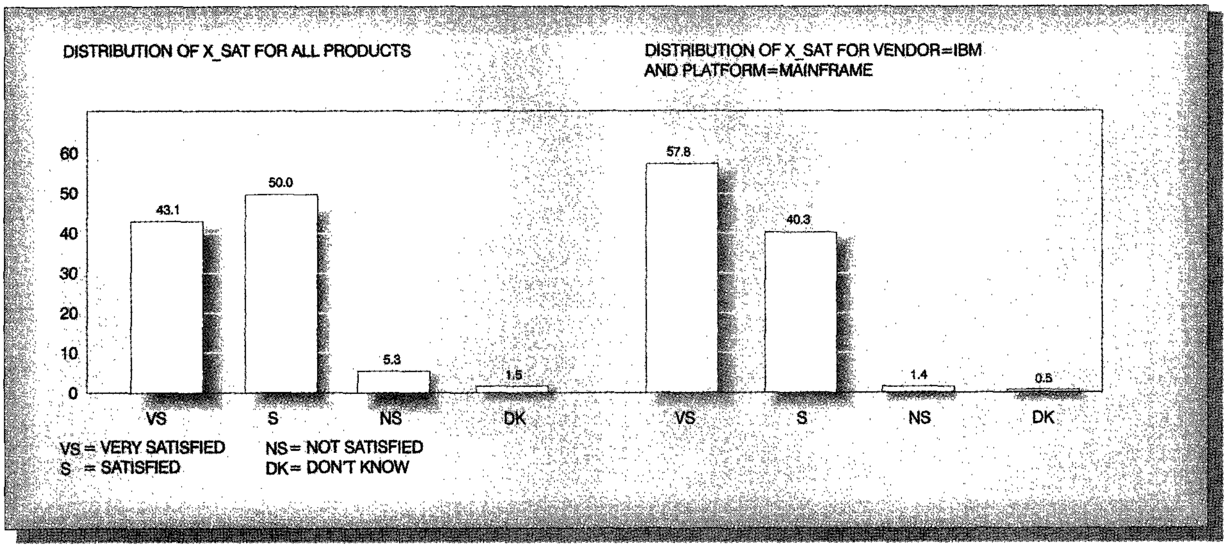
During the interview, we also asked if the interviewee had any comments on the structure of the metrics. In particular, we have asked for comments on the wording and the ranges of values of the questions. The interviewee comments and GQM structure for the service support group were recorded. They will be used as input to the annual questionnaire review and modification, and in future improvement cycles with the service support group.

The bottom-up analyses. We then applied our bottom-up method to the CUSTSAT database to extract new interesting information from it. Although several statistical analyses were already being done with the CUSTSAT data, we believed that, due to its size and diversity, there should be interesting information that remained hidden in the data. Here we focus on the analyses and results obtained, rather than on the procedures used to do the analyses, which were discussed earlier.

We performed analyses at several levels of data granularity. We will cover some examples of analyses done at a coarse level (many products analyzed together by platform, manufacturer, or type of application) to illustrate the method. Those analyses were aimed at the senior management user groups.

First, we want to give an example of how fast and inexpensive it is to explore the data with the AF technique. Consider the following relationship question:

Figure 7 Most "interesting" diagram produced from Analysis A



AF Analysis A: How do *product class* and *manufacturer* relate to *customer satisfaction by feature*?

This question defines an analysis that associates attributes from three attribute classes (as previously defined). The "product class" has attributes such as platform and product application type. The "manufacturer class" has only one attribute: whether the manufacturer of the product is IBM or another company. The "feature-wise satisfaction" (FSAT) class includes the customer satisfaction ratings of features of the products. Satisfaction with the product performance, user interface, documentation, and reliability are examples of FSAT attributes.

This analysis was designed to discover the FSATs in which IBM differs the most from its competition. The main result is shown in Figure 7. In general, IBM had a significantly better performance than the competition in a certain aspect of its products (let us call this aspect "X-feature satisfaction," or Xsat). The product class showed us that this advantage originated from the mainframe platform. In other words, IBM products were much better than the competition with respect to Xsat in the mainframe platform and had similar scores in other platforms.

This result is interesting because Xsat is a very important feature. However, the result was not surprising. The data managers already knew about it. In this case, there was no knowledge discovered. Nonetheless, this was an illustrative experience for us. In 1994, the data managers spent a lot of time in statistical analysis to discover this very same information. The bottom-up method was able to find this fact quickly and inexpensively.

The second example illustrates how new unexpected insights can arise from a bottom-up analysis. Consider the following GRQ involving three attribute classes:

AF Analysis B: How do *satisfaction with local support* and *type of local support provider* affect the *most important attributes*?

This analysis associates three attributes, one from each attribute class. The first class has attributes related to the satisfaction with the local support for the product (LSSATs). It includes satisfaction with training, local sales, and local technical support. The second class includes only the attribute "who is the local support provider (IBM, a competitor, or a third party)?" The "most important satisfaction attributes" (MIA) class contains the attributes that senior management consider most important for IBM. It includes attributes such as: overall satisfaction with product (Osat), whether the customer would recommend the product to someone else, whether the customer is planning to upgrade the product, etc.

Table 2 LSSAT and FSAT impact on the most important attributes

MIAs	LSSATS and FSATS			
	Positive Impact	New VS	Negative Impact	New NS
	Fsat1	67.2%	Rsat	42.6%
VS ⇒ 33.2%	Fsat2	65.6%	Fsat1	41.4%
	Fsat3	61.6%	Fsat2	29.3%
Attribute X	Fsat4	59.0%	Fsat3	29.3%
	Rsat	57.6%	LSsat1	20.8%
NS ⇒ 8.6%	LSsales	51.9%	Fsat4	20.2%
	LSsat1	51.8%	LSsales	15.4%
	LSsat2	50.3%	LSsat2	14.3%
	Fsat1	72.1%	Fsat1	44.9%
VS ⇒ 48.7%	LSsales	69.4%	Rsat	38.0%
	Fsat2	68.9%	Fsat3	37.7%
Attribute Y	Rsat	66.6%	Fsat2	35.0%
	Fsat3	66.4%	LSsat1	28.6%
NS ⇒ 13.3%	LSsat1	65.6%	Fsat4	27.8%
	LSsat2	64.5%	LSsales	21.8%
	Fsat4	61.1%	LSsat2	16.8%

This analysis tried to capture how the local support provider impacts the most important attributes (MIAs), and what the relation is between the LSSATs and the MIAs. It provided several interesting results. For example, although in general high "overall satisfaction" (Osat) with a product was associated with high satisfaction with the training for that product, it showed that this positive association is stronger for IBM training than for the competition or third-party training. This type of information, combined with the data users' background knowledge, has led to new business insights.

This analysis was particularly interesting because it provided a completely unexpected result to the data managers. They always considered the FSATs to be the most important drivers of the MIAs. However, this analysis revealed very high associations between the MIAs and the LSSATs. This result raised a new question: are the LSSATs as important as the FSATs with respect to the MIAs? This question was translated to the following GRQ:

AF Analysis C: How does the *local support satisfaction + customer satisfaction by feature* relate to the *most important attributes*?

This analysis compares the local support and feature-wise satisfaction attributes with respect to their impact on the most important attributes (MIAs). The tool produced diagrams for the strongest associations of the LSSATs and FSATs with the MIAs. But, instead of looking at particular diagrams, we grouped diagrams according to the positive and negative impacts of LSSATs and FSATs on the MIAs. The positive impact was determined by the percentage of "very satisfied" (VS) answers for an MIA, given that the customers were very satisfied with an FSAT attribute. The negative impact was determined by the percentage of "not satisfied" (NS) answers for an MIA, given that the customers were not satisfied with an FSAT attribute.

Table 2 shows the summary of positive and negative impacts of the LSSATs and FSATs in two particular MIAs—Attributes X and Y. For example, the first line of Table 2 shows that 67.2 percent of the customers who were very satisfied with respect to Fsat1 were also very satisfied with Attribute X. By the same token, 42.6 percent of the customers who were not satisfied with respect to Rsat were also not satisfied with Attribute X. Table 2 shows two attributes explicitly: Rsat (satisfaction with product reliability—an FSAT), and LSsales (satisfaction with local sales support—an LSSAT). The other FSATs (Fsat1, Fsat2, Fsat3, and Fsat4) and LSSATs (LSsat1 and LSsat2) are not made explicit to protect IBM proprietary information.

This analysis produced several interesting results:

- For some MIAs, LSSATs are sometimes as important as FSATs such as product performance or reliability satisfaction. For example, Table 2 shows that "local support sales" (an LSSAT) has a higher positive impact than "reliability" (an FSAT) with respect to "Attribute Y."
- The same FSATs and LSSATs had different types of impacts in different MIAs. For example, "local support sales" (an LSSAT) was one of the attributes with the highest positive association with Attribute Y (an MIA), while it was one of the attributes with the lowest positive associations with Attribute X (another MIA). This was a surprise because there had been an implicit assumption that the FSATs and LSSATs were associated in more or less the same way with different MIAs.
- The same attributes may have quite different positive and negative impacts in the same MIAs. For example, "reliability" has a very high negative impact and a surprisingly low positive impact in Attribute X.

These facts led to more than new business insights. They showed that some assumptions about the data were incorrect or incomplete. They implied that some of the data analyses and models needed to be revised or refined.

During the qualitative analysis of this case study (see Mendonça,[29] Chapter 5), we concluded that the insights obtained through the AF analyses were different in nature from the insights gained through the more traditional data analyses done using the CUSTSAT MF. The insights produced by traditional data analyses were rare but always very important. Traditional data analysis always monitors a small number of recognized key areas of the CUSTSAT data. The AF technique, on the other hand, explored several new and potentially interesting aspects of the CUSTSAT data. Its insights were numerous but not always important. Because of its exploratory nature, some of those insights pointed to further data analysis or to reviews of the way things are measured or interpreted in the CUSTSAT MF.

Concluding remarks

We believe that an important problem in software engineering is to understand and improve existing measurement frameworks. Our work tackles this problem on two key fronts: (1) how to understand and structure ongoing measurement, and (2) how to better explore the data that the organization has already collected. We use the characterization step and the GQM-based method to tackle the first problem. We use the AF-based method to tackle the second problem.

The characterization process tackles the problem of understanding how people are using the data in a measurement framework. In our approach, characterization is a prerequisite for applying the GQM- and AF-based methods. Although characterization is an important step for improving existing MFs, we do not know of other work that discusses this problem.

The GQM paradigm has been used by several software engineering organizations. However, it has been used to plan and implement measurement "from scratch." One of the main contributions of our work is to show how GQM can be applied when the measurement framework is already operational.

The AF-based method describes how the AF technique can be applied in a measurement framework. Our contributions here are: (1) in the area of data mining, we propose procedures to better explore data using the AF technique; and (2) in the area of software engineering, we show that this type of data exploration can produce important business insights and contribute to better understanding of the data, metrics, and measurement models used in software organizations.

The GQM- and AF-based approaches are complementary and can work in synergy. The GQM structures help us to choose and organize data for AF analyses.

Our approach was designed to be nonintrusive to the measurement framework management.

The new knowledge gained through the AF analyses can be fed back into our measurement goals and used to revise our GQM structures.

Our approach was designed to be nonintrusive to the MF management. Its main objective is *not* to implement modifications to an MF, but rather to point to where it can be improved. It is also important to point out that our approach is not a methodology for defining new metrics or measurement (predictive) models. It is rather a methodology for understanding the data and the metrics and how they are fulfilling the needs of data users in an MF.

Our approach of applying the GQM method incrementally—one point of view at a time—has a limitation in that it does not detect *overall* extraneous metrics (i.e., metrics that are extraneous to the MF as a whole). We can only detect metrics that are extraneous from a certain point of view. We have to interview all the user groups that are related to a certain metric before we can conclude that this metric is extraneous to the MF as a whole.

The AF-based method cannot substitute for statistical data analysis techniques; it complements them. This method gives us the ability to find interesting facts that might otherwise remain hidden in the data. It is geared toward discovering information. We should use statistics to further analyze the facts discovered using this method.

The bottom-up analyses may use data mining techniques other than AF. Our choice of AF was determined by its simplicity, ease of use, and availability. We intend to further explore the synergy between data mining in general and GQM. We want to couple data mining and GQM more tightly. Our idea is to better formalize the use of GQM to structure existing measurement frameworks, and to combine it with different types of data mining approaches, in order to define an integrated method for re-engineering measurement frameworks.

The reader interested in further information about the GQM- and AF-based methods may refer to Mendonça,[29] which discusses the approach presented in this paper in greater detail. It also describes the qualitative study that was used to validate these methods in the CUSTSAT MF.

Acknowledgments

This work is sponsored by the Centre for Advanced Studies at IBM Canada. We would like to acknowledge support from the staff of the Centre for Advanced Studies—Karen Bennet in particular—and the Market Revenue and Planning department at IBM Toronto. We also would like to thank the University of Maryland Experimental Software Engineering group—Carolyn Seaman in particular—for their input at various stages of this work.

Manoel Mendonça also recognizes the past support from CNPq (Brazil's Conselho Nacional de Desenvolvimento Científico e Tecnológico) for his research.

Cited references and notes

1. W. Klösgen and J. M. Żytkow, "Knowledge Discovery in Databases Terminology," *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Editors, AAAI Press/The MIT Press, Cambridge, MA (1996).
2. N. E. Fenton, *Software Metrics: A Rigorous Approach*, Chapman and Hall, New York (1991).
3. N. E. Fenton, "Software Measurement: A Necessary Scientific Basis," *IEEE Transactions on Software Engineering* **20,** No. 3, 199–206 (March 1994).
4. V. R. Basili and D. M. Weiss, "A Methodology for Collecting Valid Software Engineering Data," *IEEE Transactions on Software Engineering* **10,** No. 6, 728–738 (November 1984).
5. V. R. Basili and H. D. Rombach, "The TAME Project: Towards Improvement-Oriented Software Environments," *IEEE Transactions on Software Engineering* **14,** No. 6, 758–773 (June 1988).
6. V. R. Basili and S. Green, "Software Process Evolution at the SEL," *IEEE Software* **11,** No. 4, 58–66 (July 1994).
7. M. K. Daskalantonakis, "A Practical View of Software Measurement and Implementation Experiences within Motorola," *IEEE Transactions on Software Engineering* **18,** No. 11, 998–1010 (November 1992).
8. R. B. Grady, *Practical Software Metrics for Project Management and Process Improvement*, Chapter 3, Prentice-Hall, Inc., Englewood Cliffs, NJ (1992).
9. I. S. Bhandari, M. J. Halliday, E. Tarver, D. Brown, J. Chaar, and R. Chillarege, "A Case Study of Software Process Improvement during Development," *IEEE Transactions on Software Engineering* **19,** No. 12, 1157–1170 (December 1993).
10. I. S. Bhandari, M. J. Halliday, J. Chaar, R. Chillarege, K. Jones, J. S. Atkinson, C. Lepori-Costello, P. Y. Jasper, E. D. Tarver, C. C. Lewis, and M. Yonezawa, "In-Process Improvement through Defect Data Interpretation," *IBM Systems Journal* **33,** No. 1, 182–214 (1994).
11. I. S. Bhandari, B. Ray, M. Y. Wong, D. Choi, A. Watanabe, R. Chillarege, M. Halliday, A. Dooley, and J. Chaar, "An Inference Structure for Process Feedback: Technique and Implementation," *Software Quality Journal* **3,** No. 3, 167–189 (1994).
12. I. S. Bhandari, M. G. Mendonça, and J. Dawson, "On the Use of Machine-Assisted Knowledge Discovery to Analyze and Reengineer Measurement Frameworks," in *Proceedings of CASCON'95*, Toronto, Canada (November 1995), pp. 275–284.
13. I. S. Bhandari, E. Colet, J. Parker, Z. Pines, R. Pratap, and K. Ramanujam, "Advanced Scout: Data Mining and Knowledge Discovery in the NBA Data," *Data Mining and Knowledge Discovery* **1,** No. 1, 121–125 (January 1997).
14. E. Colet and I. S. Bhandari, "Statistical Issues in the Application of Data Mining to the NBA Using Attribute Focusing," in *Proceedings of the Section on Statistics in Sports of the 1997 Joint Statistical Meetings*, Anaheim, CA (August 1997), American Statistical Association, pp. 1–6.
15. Observed ($A_x = v$) × Observed($A_y = u$)
16. Observed ($A_x = v \wedge A_y = u$)
17. A structured interview is one in which the questions are in the hands of the interviewer and the response rests with the interviewee, as opposed to an unstructured interview in which the interviewer simply raises topics for discussion and the interviewee provides both the relevant questions and the answers.
18. Y. S. Lincoln and E. G. Guba, *Naturalistic Inquiry*, Sage, Beverly Hills, CA (1985).
19. H. Potier, J. Albin, R. Ferreol, and A. Bilodeau, "Experiments with Computer Complexity and Reliability," in *Proceedings of the 6th International Conference on Software Engineering*, Tokyo, Japan (September 1982), IEEE Computer Society Press, pp. 94–103.
20. R. Selby and A. H. Porter, "Learning from Examples: Generation and Evaluation of Decision Trees for Software Resource Analysis," *IEEE Transactions on Software Engineering* **14,** No. 12 (December 1988), pp. 1743–1757.
21. J. Tian and J. Palma, "Analyzing and Improving Reliability: A Tree-based Approach," *IEEE Software* **15,** No. 2, 97–104 (March/April 1998).
22. L. C. Briand, V. R. Basili, and W. M. Thomas, "A Pattern Recognition Approach for Software Engineering Data Analysis," *IEEE Transactions on Software Engineering* **18,** No. 11, 931–942 (November 1992).
23. L. C. Briand, V. R. Basili, and C. Hetmanski, "Developing Interpretable Models with Optimized Set Reduction for Identifying High-Risk Software Components," *IEEE Transactions*

on Software Engineering **19,** No. 11, 1028–1044 (November 1993).

24. K. Srinivasan and D. Fisher, "Machine Learning Approaches to Estimating Software Development Effort," *IEEE Transactions on Software Engineering* **21,** No. 2, 126–137 (February 1995).
25. I. S. Bhandari, "Attribute Focusing: Machine-Assisted Knowledge Discovery Applied to Software Production Process Control," *Knowledge Acquisition Journal* **6,** No. 3, 271–294 (September 1994).
26. Multiple attribute classes are used when we want to guarantee that different classes of metrics will appear in *n*-way diagrams.
27. During 1997, we also performed a qualitative evaluation of our improvement approach based on the experience gained by applying it to the CUSTSAT MF, as reported in Mendonça, Chapter 5.
28. The surveys are done up to six months after the problem occurred. It might be difficult for the customer to classify the severity and type of problems in those cases.
29. M. G. Mendonça, *An Approach to Improving Existing Measurement Frameworks in Software Development Organizations*, Ph.D. thesis, University of Maryland, College Park, MD (December 1997). Also available as CS-TR-3852 and UMIACS-TR-97-82 from the University of Maryland.

Accepted for publication June 4, 1998.

Manoel G. Mendonça *University of Maryland, Department of Computer Science, College Park, Maryland 20742 (electronic mail: manoel@cs.umd.edu).* Dr. Mendonça received the Ph.D. degree in computer science from the University of Maryland in 1997. He also holds an M.Sc. degree in computer engineering from the State University of Campinas, and a B.S. degree in electrical engineering from the Federal University of Bahia, both in Brazil. Dr. Mendonça is currently a research associate at the University of Maryland. His main research interests are data mining and the use of measurement for software engineering and management.

Victor R. Basili *University of Maryland, Department of Computer Science, College Park, Maryland 20742 (electronic mail: basili@cs.umd.edu).* Dr. Basili is a professor in the Institute for Advanced Studies and the Department of Computer Science at the University of Maryland. He received a B.Sc. degree in mathematics from Fordham College, Bronx, NY, an M.Sc. degree in mathematics from Syracuse University, NY, and a Ph.D. degree in computer science from the University of Texas at Austin. His main interest is in the development of quantitative approaches for software management, engineering, and quality assurance. He is one of the founders and principals of the Software Engineering Laboratory, a joint venture between NASA's Goddard Space Flight Center, the University of Maryland, and Computer Sciences Corporation. He has consulted with many agencies and organizations, including IBM and NASA, and has authored more than 130 fully refereed papers. Dr. Basili has been a member of the IEEE Computer Society Board of Governors, Editor-in-Chief of the *IEEE Transactions on Software Engineering*, and Program Chairman for several conferences, including the 6th and 15th International Conference on Software Engineering. He is an IEEE and an ACM Fellow. In 1997, Dr. Basili won the award for Outstanding Contributions to Mathematics and Computer Science from the Washington Academy of Sciences.

Inderpal S. Bhandari *Virtual Gold, Inc., Hartsdale, New York 10530 (electronic mail: info@virtualgold.com).* Dr. Bhandari received a Ph.D. degree in electrical and computer engineering from Carnegie Mellon University in 1990. His M.Sc. degree is from the University of Massachusetts at Amherst and his B. Tech. degree from the Birla Institute of Technology and Science, Pilani, India. His research interests are broad and he has published on topics relating to design automation, automated diagnostic methods, and distributed algorithms and systems. His current interests focus on data mining—specifically, on translating data mining technology into applications that offer real competitive advantage to organizations. Recently, he formed Virtual Gold, Inc., a data mining technology consulting company, to implement his technical vision. From 1990 to 1997, Dr. Bhandari was a member of the research staff at the IBM Thomas J. Watson Research Center, where he received several awards for his pioneering work in data mining and in software engineering. He was the creator and project director of IBM's Advanced Scout, a data mining program used by coaches of the National Basketball Association to devise new strategies based on the automatic identification of hidden patterns in game data and video.

Jack Dawson *IBM Software Solutions Division, IBM Toronto Laboratory, 895 Don Mills Road, North York, Ontario M3C 1W3, Canada (electronic mail: dawson@ca.ibm.com).* Mr. Dawson holds a B.Sc. degree in electrical engineering from the Technical University of Nova Scotia. He is currently a program manager in the market revenue and planning department of the IBM Toronto Laboratory. His research interests are in the areas of customer satisfaction data collection and analysis. At IBM, he is part of a team that measures customer satisfaction on a worldwide basis, and he has actively worked on the Software Solution Division's customer satisfaction surveys for product diagnosis. Before joining IBM, Mr. Dawson worked as a geophysicist for Amoco Canada Ltd., performing petroleum exploration studies in western Canada.

Reprint Order No. G321-5687.