Inf Syst Front (2018) 20:275–288
DOI 10.1007/s10796-016-9722-2

Enabling self-service BI: A methodology and a case study for a model management warehouse

David Schuff[1] · **Karen Corral**[2] · **Robert D. St. Louis**[3] · **Greg Schymik**[4]

Published online: 22 November 2016

Abstract The promise of Self-Service Business Intelligence (BI) is its ability to give business users access to selection, analysis, and reporting tools without requiring intervention from IT. This is essential if BI is to maximize its contribution by radically transforming how people make decisions. However, while some progress has been made through tools such as SAS Enterprise Miner, IBM SPSS Modeler, and RapidMiner, analytical modeling remains firmly in the domain of IT departments and data scientists. The development of tools that mitigate the need for modeling expertise remains the "missing link" in self-service BI, but prior attempts at developing modeling languages for non-technical audiences have not been widely implemented. By introducing a structured methodology for model formulation specifically designed for practitioners, this paper fills the unmet need to bring model-building to a mainstream business audience. The paper also shows how to build a dimensional Model Management Warehouse that supports the proposed methodology, and demonstrates the viability of this approach by applying it to a problem faced by the Division of Fiscal and Actuarial Services of the US Department of Labor. The paper concludes by outlining several areas for future research.

Keywords Business intelligence · Model management · Analytics · Modeling · Self-service

✉ Karen Corral
karencorral@boisestate.edu

[1] Temple University, Philadelphia, PA, USA

[2] Boise State University, Boise, ID, USA

[3] Arizona State University, Tempe, AZ, USA

[4] Grand Valley State University, Allendale, MI, USA

1 Introduction

In 1987, Box and Draper wrote: "Essentially, all models are wrong, but some are useful" (p. 424). They went on to say: "Since all models are wrong the scientist cannot obtain a 'correct' one by excessive elaboration." Box and Draper's words are very relevant for today's business intelligence practitioners. The science and art of business intelligence has typically required a team with diverse skills ranging from data storage and retrieval, to model formulation and selection, to the presentation of actionable results to business managers.

Through products such as SAS Enterprise Miner, IBM SPSS Modeler, and RapidMiner, we are seeing the emergence of visual analytics model-building tools in the same way that we saw the emergence of visual programming tools 20 years ago. These tools seek to "democratize" analytics (see Henschen 2014; HBR Analytic Services 2012) through the realization of "self-service" BI, making advanced data analysis accessible to a wider audience. Self-service BI seeks to give business users access to selection, analysis, and reporting tools without requiring intervention from IT. Widespread use is a necessary condition for self-service BI to maximize its impact, and this resulting democratization of BI is a necessary condition for realizing its role in transforming individual and organizational decision-making. Unfortunately, just as visual programming tools don't make people better programmers, visual modeling tools don't make people better modelers. In fact, new tools can make things worse by misleading users into thinking they are doing "good" analytics simply because they are able to complete an analysis. In order to truly democratize analytics, we need tools that support decision-making around the model-building process and not simply mask the complexity of statistics and coding.

Information systems professionals have had a great deal of experience with managing, organizing, and presenting data in

both structured (e.g., spreadsheets and databases) and unstructured (e.g., textual documents) forms. However, model building historically has fallen within the domain of management science (Geoffrion 1987; Kottemann and Dolk 1992; Lin et al. 2000). That must change to enable the widespread adoption of business intelligence and analytics. For analytics to move beyond the purview of data scientists, business-facing practitioners must employ methodologies and tools that help them: 1) understand the difference between data, documents, and models, and the implications of those differences for model building and management; 2) identify relevant variables and their relationships; 3) assess the usefulness of models; and 4) know when to terminate the model building process.

This paper describes a structured methodology for model formulation specifically designed for practitioners, and the design for a dimensional model store that supports the methodology. We begin by reviewing the literature on data and document retrieval and extend this work to the retrieval of analytical models. We then review the work that was done by management scientists on model management and explain why that work was never sufficiently implemented in practice. Next, we present our approach and discuss why our methodology and underlying data store is uniquely poised to simultaneously democratize the use of analytics and encourage "good modeling behavior." We then present the data model for our Dimensional Model Warehouse and apply it to an original case study of a prediction problem faced by the United States Department of Labor. We conclude with future directions.

2 Data versus documents versus models

Blair (2002), through an analysis of the differences between data retrieval and document retrieval, proposed that the information search process changes based on the type of artifact being targeted (see the first two columns of Table 1). He argued that the task of finding information contained in documents is fundamentally different and more complex than the task of finding data. A data retrieval task is closed-ended and direct with an unambiguous answer – for example, "what grade did Chen receive for the Database Systems course in the fall semester of 2014?" Data retrieval success is characterized by a "correct" (and verifiable) answer. The time it takes to return the answer is dependent only on the speed of the software and hardware executing the query.

In document retrieval, the underlying questions are more open-ended and indirect, and there may not be a single correct answer – for example, "Which students are most likely to graduate?" The formal query often is phrased in several different ways to gather a set of documents that, together, are likely to provide a sufficient answer to the question. Queries might include "student success factors," "graduation rates," and "at-risk students." These searches are likely to return multiple results, as it frequently is the case that more than one document will contain relevant information. Document retrieval success is based on the utility of the documents returned for formulating an answer to the question being researched. The time it takes to formulate an answer is dependent on both how many documents are returned, and the speed with which the searcher can identify relevant documents, discard irrelevant ones, and conclude that a given set of documents sufficiently answers the question.

Model retrieval is even more complex than document retrieval. Adding to the complexity of model retrieval is the fact that the distributions of the variables and the correlations among the variables may differ from dataset to dataset, even if the datasets have similar metadata. This creates the need to specify the functional form of the relationships, and estimate the parameters of those functional forms for each data set. Because there is no single, correct model, this process has no finite end. Box's 'all models are wrong' aphorism has been discussed by countless scholars, including Cox (1995) and Burnham and Anderson (2002). Wit et al. (2012) summarize the results of a conference titled 'All models are wrong: model uncertainty and selection in complex models' that was held in March, 2011 in Groningen, the Netherlands to critically examine the field of statistical model selection methods over the past 40 years. Their summary differs little from the conclusion of Box and Draper (1987) that the modeler does not eventually arrive at the "correct" specification. Instead, the analyst can only achieve a "satisficing" model that balances the tradeoff

Table 1 Comparison of data, document and model retrieval (adapted from Blair 2002)

Data retrieval	Document retrieval	Model retrieval
Direct ("I want to know X")	Indirect ("I want to know about X")	Investigative ("I want to find a model that explains X")
Necessary relation between a formal query and the representation of a satisfactory answer	Probabilistic relation between a formal query and the representation of a satisfactory answer	Satisficing relation between a formal query and the representation of a useful model that recognizes tradeoffs between accuracy and complexity
Criterion of success = correctness	Criterion of success = utility	Criterion of success = improved ability to predict, manipulate, or understand X
Speed dependent on the time of physical access	Speed dependent on the number of logical decisions the searcher must make (include or discard)	Speed dependent on the number of modifications required to obtain a useful model

between accuracy and complexity. The analyst knows it is time to stop refining the model when the ability to predict, manipulate or understand the data cannot be further improved in a cost effective manner. Therefore, the speed of this process depends on the skill of the modeler, the strength of the relationships among the data items, and the support that can be provided by a modeling environment.

Clearly model retrieval includes aspects of both data and document retrieval. But it also requires a level of manual intervention that is fundamentally different from either of these. Because model retrieval is such a complex process, any information system designed to facilitate model retrieval must be part of a larger, structured methodology for model formulation. This need for manual intervention is why the model retrieval process cannot be considered complete until the intervention, i.e., the refinement of the retrieved model into a satisficing model, is complete.

3 Model management research

A great deal of work was done in the model management area during the 1980s and 90s. For example, Geoffrion (1987) identified two major problems confronting the management science/operations research (MS/OR) community. First, he noted that doing MS/OR tends to be a low productivity activity. Second, he noted that managers and policy makers are reluctant to ask for model-based assistance. Geoffrion, and others, tried to address these problems by developing modeling languages.

The modeling languages of the 1980s and 90s had four major design objectives. First, modeling languages should represent large and complex models using a few relatively simple statements (Geoffrion 1987; Brooke et al. 1988; Fourer et al. 1990). Second, modeling languages should support the entire modeling life-cycle (Fourer et al. 1990; Geoffrion 1987, 1989). Third, modeling languages should allow the accumulation, sharing, integration, and reuse of data, models, solvers, and derived knowledge (Brooke et al. 1988; Choobineh 1991). Fourth, modeling languages should improve the productivity and managerial acceptance of MS/OR activities (Geoffrion 1987). To that end, much of the work on model-driven decision support systems focused on taking models "as-is" or tuning parameters of an already established underlying model (i.e., see the review by Power and Sharda 2007).

Several modeling languages were developed. These included structured modeling language (SML) (Geoffrion 1987), generalized algorithm for mathematical systems (GAMS) (Brooke et al. 1988), a mathematical programming language (AMPL) (Fourer et al. 1990), linear, interactive and general optimizer (LINGO) (Cunningham and Schrage 2004), structured query language for mathematical programming (SQLMP) (Choobineh 1991), and the subscript-free modeling language (SFL). The developers of SFL (Lin et al. 2000) state that "In SFL, the steps the decision maker must go through to formulate a model are the same steps that the decision maker must go through to understand the problem. This makes SFL very user friendly" (p. 615). However, neither SFL nor any of the other modeling languages was widely adopted by non-technical managers, who continued to view MS/OR models as both confusing and expensive to build.

This is at odds with the notion of self-service BI. The *2014 State of Self-Service BI Report* (Logi Analytics 2014) notes that "Business users should be able to use all this information when they want, where they want, and do so without having IT in the way" (p. 3). Fifty-two percent of managers stated that it was important to have the capability to gain insight from data independent of their IT department (Logi Analytics 2014), but only 22 % of the respondents actually have access to those tools now. The study also reports misalignment in priorities between IT and business departments. IT considers the use of spreadsheets to be the most important modeling for business users, whereas business users said it's most important for them to not only consume preformatted reports, but also to analyze data and create reports on their own. Further, the report states that "the most important capabilities for business users were the ones they were the least satisfied with" (p. 3).

The model management work that was conducted in the 1980s and 90s failed to satisfy the desire of business managers for self-service BI tools. There are several reasons for this. First, the primary focus of prior model management research was how to build, store, and retrieve deterministic models. Second, the work assumed the modeler knew the relevant variables for the deterministic model, and was interested in finding the optimal solution to a structured problem. As pointed out by Davenport et al. (2001), business analytics deals with structured, semi-structured, and unstructured problems.

There are many parallels between modeling languages and CASE tools. CASE tools were developed with the intention to support, simplify and even automate portions of the very technical task of software development (McMurtrey et al. 2002). However, these tools failed to provide the "silver bullet" for application development (Guinan et al. 1997). A variety of reasons for nonadoption of CASE tools have been identified, e.g., they forced processes on developers (Lending and Chervany 1998; Senn and Wynekoop 1995), they were not perceived as contributing to an improvement in productivity, especially by experienced developers (Finlay and Mitchell 1994), they were considered to be too complex (Finlay and Mitchell 1994; Iivari 1996; Senn and Wynekoop 1995), and there was a large perceived gap between expectations for the tools and their capabilities (Lundell and Lings 2004). Fundamentally CASE tools failed because they never were able to overcome the need for the highly knowledgeable software developer, and yet they forced those developers to adhere to the tool's methodology.

A clear analogy exists between application development and the development of analytical models. CASE tools require that their users be software developers; that is, users have to be highly skilled and knowledgeable individuals in order to use the tools to build useful software. Similarly, modeling tools such as Enterprise Miner and SPSS Modeler require users to be savvy statisticians in order to build useful analytical models. That paradigm cannot enable the democratization of BI.

Several more recent studies have proposed automated systems for model building (Kridel and Dolk 2013; Deokar and El-Gayar 2011; El-Gayar and Deokar 2013). In these systems, the model-building process is implemented as a service that takes a data set as input and selects a model based on a preset template (i.e., the example of "retail acquisition" given by Kridel and Dolk 2013). These solutions are very promising and services such as the recently-released Watson Analytics by IBM offer one way to bring modeling to non-technical knowledge workers. However, one concern is that these tools are fundamentally data-driven – that is, they allow the model to be determined by the data. Beyond selection of the model template, there is a missed opportunity to leverage the analysts' domain knowledge.

Enabling business users to select, rather than build, models is an alternative solution that can enable self-service BI. Predictive Modeling Markup Language (PMML), developed by the vendor-led Data Mining Group (Guazzelli et al. 2009), is an XML specification designed to share models between software applications (Guazzelli et al. 2009; Pechter 2011) by encoding the model details in an application-neutral format. SPSS and SAS can store their models using PMML, disentangling the model itself from the software application, much like HTML disentangles web page content and formatting from the browser.

There also have been efforts by researchers to use PMML as an enhancement to knowledge management systems, where predictive models are executed as a result of rule firings by embedding the PMML within a rule base (Sottara et al. 2011). While PMML does facilitate some degree of knowledge sharing through the exchange of previously-constructed models, it still doesn't meet the requirements necessary for self-service BI. As a standard primarily for information interchange, PMML doesn't provide a mechanism for user-driven search for possible models or models that have already been built by others.

This is not to say that PMML does not have a role to play in a self-service BI architecture. Its software-neutral format would be useful in encoding a model selected by a user so that it could be exported to a variety of analytics software solutions. It also overcomes many of the problems that Madhusudan (2007) identifies as inhibiting the realization of distributed model management, such as "the lack of semantic and syntactic standards for model definition" (p. 9). In fact, PMML could be leveraged to enable a true model management solution, like the one we propose later in this paper, to be software-agnostic.

The ability to reuse BI models across different development platforms is enhanced by the use of PMML (Guazzelli et al. 2009), and the sharing of data warehouses is greatly aided by the use of the Common Warehouse Metamodel (Object Management Group 2003). However, many modern modeling tools, such as SAS Enterprise Miner and IBM SPSS Modeler, are largely based on the Sample, Explore, Modify, Model, and Assess (SEMMA) process for modeling that was developed by SAS (SAS Institute 1998; Rohanizadeh and Moghadam 2009). This five step process: 1) collects a sample of data for a set of variables; 2) graphically explores the univariate distributions of the variables and the bivariate relationships among the variables; 3) determines whether the variables need to be truncated, grouped, or transformed in order to eliminate outliers or adjust for nonlinearities; 4) models the multivariate relationship among the variables; and 5) assesses the accuracy of the model. An implicit assumption of this process is that the analyst knows the relevant variables to sample, and the potential relationships among those variables, before the modeling process begins. This step is critical – Davenport (2013, p. 77) states "The essence of analytical communication is describing the problem and the story behind it, the model, the data employed, and the relationships among the variables in the analysis." Davenport and Kim (2013, p. 186) cite Intel Fellow Karl Kempf's statement that "effective quantitative decisions 'are not about the math; they're about the relationships.'" Effective self-service BI modeling tools must help managers determine which variables are relevant and the nature of the relationships among them.

This is a key limitation of SEMMA – it doesn't formalize the incorporation of the domain knowledge of business and the data (Hampton 2011). CRISP-DM (Cross Industry Standard Process for Data Mining) is a more generalized methodology that offers some guidance regarding what's missing – specifically business understanding, data understanding, and actual deployment of the model. However, most software takes a SEMMA approach, essentially assuming the modeler is knowledgeable about analytics techniques. Tools that democratize BI will need to support those missing pieces that are included within the CRISP-DM methodology, as this will empower those very familiar with the business and the data, but who are lacking deep analytics skills, to build models.

In addition, managers need help determining when to stop modifying a model and how to assess its usefulness. If managers do not receive help with assessing the usefulness of models, they may do more harm than good with models that they build. This raises the question of whether non-technical managers ought to be building their own models. Pack (1987) recommended that an analyst have at least a master's degree in statistics, or the equivalent, in order to build and use

forecasting models successfully. Geoffrion (1987) and Murphy et al. (1992) further argue that most modeling work is understood only by a small group of professionals, not non-technical decision makers or managers. If that level of expertise is needed, then self-service BI will not be realized. However, Davenport (2013, p. 77) quotes Xiao-Li Meng, the chair of Harvard's Department of Statistics as saying:

> Intriguingly, the journey, guided by the philosophy that one can become a wine connoisseur without ever knowing how to make wine, apparently has led us to produce many more future winemakers than when we focused only on producing a vintage.

Apparently, as persons who did not know anything about wine making became involved in wine tasting, they also became more curious as to what creates the taste in wine. If managers are able to assess the usefulness of models, they also may become more interested in what allows their models to produce useful results; but this will occur only if managers are confident that they can assess the usefulness of models. If self-service BI is to be realized, a methodology is needed that helps managers and business analysts throughout the SEMMA process; locate the relevant variables, see how those variables relate to each other, know when to stop modifying a model, and assess its usefulness.

4 The model formulation process

As we've established, model formulation is a multi-step process based on the highly complex and open-ended task of identifying relationships. In fact, the model formulation process can be characterized as a "wicked problem," as it has "unstable requirements and constraints based on ill-defined environmental contexts" (Hevner et al. 2004, p. 81). It also requires some degree of human intervention to arrive at a solution (Hevner et al. 2004). Therefore, model formulation can be looked at as a non-deterministic, problem-solving process – some models may be more useful than others, but there never is a definitive, "correct" model.

To support this formulation process, we propose a structured methodology that both technical and non-technical analysts can use to formulate analytical models (see Fig. 1). While our approach does not reduce the complexity of the problem, it does provide a repeatable set of steps to approach model formulation. The steps are outlined below and demonstrated using the example of determining which incoming college students are least likely to graduate:

1) *Define the problem* by describing the decision to be made. In our example, the problem definition would be: "An inability to graduate on-time has added to the financial burden and accessibility of a college education. Which students are most likely to have difficulty graduating on time?"
2) *Determine the hypothesized relationships* that will inform the decision. This requires reducing the problem scope to a set of core concepts; such as retention, prior academic performance, and current working status.
3) *Define the data required* to test those relationships, specifically framed in terms of outcome (dependent) and input (independent) variables. In our example, the outcome variable is second-year retention and the input variables include family income, high-school GPA, first-semester college GPA, and hours worked per week.
4) *Assess available data* to determine what data the decision-maker already has and what data they are capable of getting. Data quality should also be considered, as data might be available but useless for analysis. For example, family income and GPA data could be part of a student's existing record, but whether a student is working would likely require manual collection.
5) *Retrieve a set of candidate models* that would test the hypothesized relationships. All candidate models would have to be appropriate given the characteristics of the data (e.g., type, distribution). In our example, we might find that some have used regressions to build a predictive model of student success, while others have used clustering techniques to create profiles of high risk and low risk students. We may also find that several regression models have been used in the past with different subsets of the independent variables.
6) *Evaluate and refine the candidate models* arriving at what the analyst believes to be the most useful final model to aid the decision making process. Models are chosen as a tradeoff between accuracy and complexity, weighted according to the decision-maker's preferences. The decision-maker may test all candidate models and further refine them based on the characteristics of the specific data set. For example, for non-traditional student populations, high school GPA may be irrelevant.

5 Dimensional document mart to support modeling: The model management warehouse

While the methodology outlined in the previous section is useful in providing structure for the inherently open-ended model formulation process, it requires access to a sophisticated body of knowledge that encompasses data, causal relationships, analytical modeling, and domain-

Fig. 1 A structured methodology for model formulation

specific organizational processes. More specifically, the modeler must have an understanding of:

- Organizational processes
- Data available within and outside the organization
- Hypothesized relationships among the variables to be explained/predicted (the *dependent* variables) and variables that influence the values of the dependent variables (the *independent* variables)
- Mathematical representations of the hypothesized relationships among the dependent and independent variables
- Measures of model effectiveness.

We propose that a Model Management Warehouse, implemented as a dimensional document mart, will facilitate model-building in a way that is consistent with our model formulation methodology. A dimensional data store is particularly well-suited for model management; specifically for facilitating model retrieval. There are two main reasons for this. First, each dimension maps to a major component of an analytical model: the modeling domain, possible dependent and independent variables, possible variable transformations, possible techniques to model the relationships among the variables, and possible measures of model effectiveness (see Table 2). This allows users to intuitively mix-and-match model components. Second, dimensional modeling facilitates "slicing" the data, thus enabling users to hold one or more aspects of the model constant while letting the others vary. For example, we can query the database to show all models that use a particular predictor and transformation. Table 2 describes each choice a modeler must make during the model building process, and how the dimensional model store helps with those choices, thereby enabling even a novice modeler to navigate the collection of potential models.

Table 2 illustrates the case where one wants to develop a model to forecast whether a customer is likely to drop your service. Note that very little knowledge of statistics is required on the part of the model builder. All the modeler needs to know is what he/she wants the model to do, and how to tell if the model is doing it. In this case the modeler would simply need to know that he/she wants to build a profiling model, and what a lift chart, ROC curve, and decile table are. They would not need to understand how to construct a lift chart, ROC curve, or decile table; but only what these represent. This is something that any non-technical analyst can understand.

Based on the dimensions identified in Table 2, we developed a star schema that stores the necessary data about the models (see Fig. 2). A star schema was chosen because dimensional models have been very successful at enabling self-service data retrieval. The structure of the star schema enables the data analyst to pick the precise piece of information for which he/she is searching out of the mountain of data that is stored in the data warehouse, and to do so without knowing anything about data structures or SQL. The structure of the star schema also will enable the data modeler to pick the precise model for which he/she is searching out of the mountain of models that are stored in the model warehouse, and to do so without knowing anything about model structures or statistics. As mentioned above, all the modeler has to have is enough domain specific knowledge to know what he/she wants the model to do, and how to tell if the model is doing it. This is very similar to the case with the user and the dimensional data mart.

This data warehouse also contains a set of documents that explain the models with the specified dependent and independent variables. Note that a Date Dimension is included in the star schema, but not in Table 2. The date dimension is included to enable modelers to screen on models that have been developed more recently. Note also that the effectiveness measures included in the fact table are specific to screening profiles. A more general star schema could include many additional effectiveness measures such as the root mean square error, the coefficient of determination, the coefficient of variation, etc.

6 Case study: The US department of labor

To test the feasibility of our approach, we applied it to an actual modeling problem that confronts the Division of Fiscal and Actuarial Services of the Office of Unemployment Insurance. The Unemployment Insurance (UI) system, as established by the Social Security Act of 1935, is a unique Federal-state[1] system grounded in Federal law, but executed

[1] There are 53 UI jurisdictions. They include the 50 states plus the District of Columbia, the Virgin Islands, and Puerto Rico. State is used here to refer to a UI jurisdiction.

Table 2 Dimensional document mart to support model formulation

Methodology step	Help provided by dimensional model store	Choice made by modeler
Step 1: Identify the business modeling domain	*Domain* dimension identifies domains for business modeling	Select the relevant modeling domain for the business problem being investigated – e.g., statistical profiling
Step 2: Identify variable to be explained/predicted	*Model* dimension shows keyword descriptors for prior models and the dependent variable in each model	Select the relevant dependent variable for the current problem – e.g., probability that an existing customer will drop your service
Step 3: Identify variables that have been used in prior studies to explain/predict the variable of interest for this study	*Independent Variables* dimension shows keyword descriptors for variables that were used in prior models/studies to explain/predict the dependent variable for this study	Select relevant independent variables for the current problem – e.g., age, income, education
Step 4: Identify possible analytic techniques for modeling the relationship between the dependent and independent variables	*Technique* dimension shows keyword descriptors for broad analytic techniques that were used in prior studies to explain/predict the dependent variable	Select relevant techniques to use with data that is available for the current problem – e.g., logistic regression, neural networks, decision trees
Step 5: Identify possible transformations to apply to the dependent and independent variables	*Transformation* dimension shows keyword descriptors for transformations that have been applied to the dependent and independent variables in prior studies to improve the usefulness of the models	Select appropriate transformations to apply to the dependent and independent variables – e.g., log, power, reciprocal, grouping, none
Step 6: Identify measures that can be used to assess the usefulness of models designed to explain/predict the dependent variable	*Fact Table* that links effectiveness measures to models and variables that have been used in prior studies	Select appropriate effectiveness measures for evaluating the usefulness of a model – lift chart, ROC curve, decile table
Step 7: Retrieve studies that provide useful information for building your model	*Document* dimension shows titles and associated URLs for documents that are relevant to your model building effort	Select relevant documents – e.g., documents that provide information for completing steps 1 through 6 above
Step 8: Select model to use	*Success Measures* show how well models have performed in the past	Select model that is likely to be the most effective given the desired performance measure – e.g., select model that most often did the best job of identifying customers that dropped the service

in its relationship to the employer and to the unemployed worker through state law. The Office of Unemployment Insurance (OUI) is the Federal partner in the system. Its role is one of overall oversight and coordination. The Division of Fiscal and Actuarial Services (DFAS) of the OUI provides technical assistance to the states to help them meet the Federal requirements.

One of the requirements of the UI program is that the state refers UI claimants likely to exhaust their benefits to reemployment services. This requirement assumes the state can correctly identify those UI claimants. The state uses statistical profiling models to identify those claimants, and each state is tasked to develop its own model. DFAS works with states to provide statistical training, tabulate and share state model building efforts, and provide a baseline model.

For the past 30 years, DFAS has offered week-long training sessions on profiling methods to state analysts. These sessions have primarily focused on teaching statistical procedures such as multiple linear regression, logistic regression, and neural networks to the state business analysts. Despite these efforts, state analysts have struggled to build effective profiling models. In 2016, DFAS is changing its approach for the seminars. Instead of emphasizing statistical procedures, they will emphasize the steps in the model-building process described in Table 1. Each analyst will bring data from her/his state to the seminar, and, supported by the dimensional model mart described in Fig. 2, is going to be asked to construct a profiling model for claimants in her/his state. DFAS believes this will greatly reduce the number of staff hours it spends consulting with state agencies, and enable analysts in the states to effectively build and maintain profiling models.

The analysts at the state agencies tasked with developing the profiling models generally do not have degrees in statistics, but do need to develop and update the models for their respective agencies. This has been a problem for DFAS. A recent study by the John J. Heldrich Center for Workforce Development (Powell 2015) found that state agencies frequently do not update their models even though it is very important to do so, and frequently do not progress beyond models they inherited from previous analysts or models that they developed jointly with DFAS. More specifically, they found that fewer than half of the UI jurisdictions had updated their profiling models since the "Great Recession" of 2007.

The staff at DFAS tasked with assisting states with their profiling efforts consists of four people: an Actuary and team leader for state modeling efforts; an Economist and DFAS contact for state modeling efforts; an Economist and state contact for state profiling models; and an Economist and Research Analyst. From conversations with these four

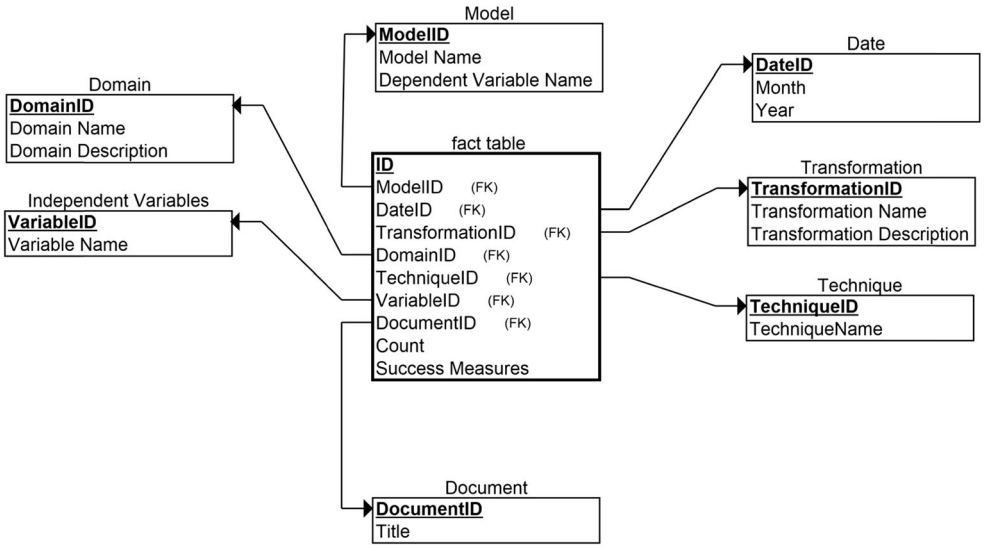


Fig. 2 Star schema for a dimensional model management warehouse

persons, we discovered several reasons why state analysts either did not re-estimate their models (keep the same variables but re-estimate the parameters using more current data), or update their models (include new variables or new functional forms for existing variables). First, there was uncertainty with respect to how to evaluate their models. Second, there was uncertainty with respect to what variables and functional forms to use. The Economist and DFAS contact person charged with assisting the states, commented that "analysts at the state level, if left unassisted, struggle with selecting variables, variable transformations, and performance measures." The Heldrich Center for Workforce Development (Powell 2015) also identified these two uncertainties as the major impediments to state efforts to update and refine existing models. The problem was not a lack of data, nor was it a lack of statistical software for analyzing the data.

Figure 3 depicts a dimensional model management warehouse, based on the star schema in Fig. 2, tailored for analysts at DFAS and State agencies. The DFAS star schema reflects how little customization is necessary to implement this type of model selection warehouse; it differs from the original star schema in only two minor ways. First, "State Name" was added to the "Model" dimension to reflect the state from which the model came. This is important if the analyst wants to select prior models for his/her own state, or examine models from states that are demographically or geographically similar. Second, specific success measures were added as measured facts to the fact table: count, lift chart, ROC curve, and decile table.

In this instance, the tailoring was done by an economist at DFAS. DFAS keeps track of the models by state, and has been evaluating state models for many years. Thus it literally took them only minutes to say they wanted to add "State Name" to the model dimension, and to specify lift chart, ROC curve, and decile table as the success measures in the fact table. In general, the tailoring would be done by the group that is responsible for building the model management warehouse. As long as this group is aware of the success measures for evaluating the models, it should be a simple task to tailor the general schema shown in Fig. 2 to their organization.

When shown the star schema in Fig. 3, the analysts at DFAS (who have master's degrees in statistics) saw many possibilities for using the schema's associated pivot table to assist states with their modeling efforts. For example, in February of 2015, the John J. Heldrich Center for Workforce Development produced a working paper for DFAS titled "Summary of State Models" (Powell 2015). For 34 different state profiling models, this report showed the dependent variables, the independent variables, and the statistical techniques used. DFAS felt that it would be very useful to incorporate this information into a dimensional model management warehouse using the star schema shown in Fig. 3. Figure 4 illustrates how an Excel pivot table based on that star schema makes it easy to discover what statistical techniques have been used by the various states to predict who will exhaust their benefits. In order to produce a list of the statistical techniques used in the past, an analyst simply needs to click on the domain name, dependent variable name, and technique name in the Pivot Table Fields list. The pivot table shows that three techniques have been used, and logit is the most commonly used technique by a very wide margin.

Finally, in any modeling environment, one has to assume that patterns that existed in the past will persist into the future. In unemployment insurance, the environment can change quickly. The unemployment rate can double in as little as six months, and industries that were growing quickly can just as quickly begin to lay off workers. Hence the analyst never knows whether the most useful model last year will continue to be the most useful model this year. However, for any given model, it is easy to observe whether the success measures for

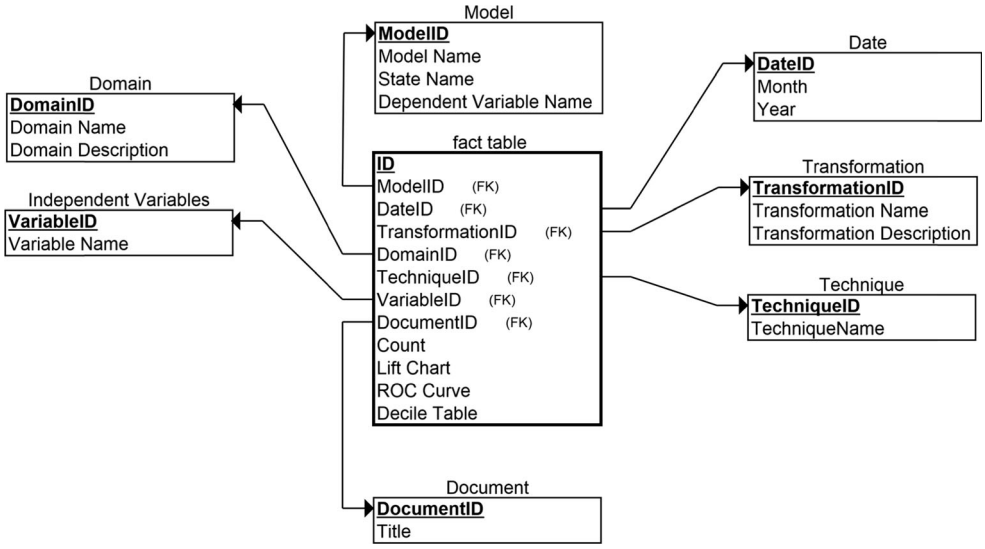


Fig. 3 Star schema for DFAS dimensional model management warehouse

the model have deteriorated over time. The DFAS encourages states to update the coefficients in their models yearly. If updating the coefficients does not restore the accuracy of the model, the time dimension enables analysts to see what models have worked best in environments similar to the current one. This provides very useful guidance for modifying the model.

Using this pivot table, the analysts at DFAS also saw that it would be very easy for state analysts to see what independent variables have been used by the various states, and what is the most widely used independent variable among the states. In order to produce this list, analysts simply need to click on the dependent variable name and variable name in the Pivot Table Fields list. Figure 5 shows the result of doing this. Although a large number of variables have been used by states to predict whether a claimant will exhaust her/his benefits, four variables stand out as the most commonly used predictors: education, industry, job tenure, and occupation. In total 64 different variables have been used as predictors, but only the top 15 are shown in Fig. 5.

Moreover, if additional information were gathered from the states, it would be equally easy for analysts to see the transformations used on the independent variables by the various states, and which state had the best performing model. This would remove some of the uncertainty that state analysts have with respect to how to evaluate their models, what variables should be in their models, and what functional forms are most appropriate for the variables that are in their models.

This simple example creates a compelling proof-of-concept for our Dimensional Model Warehouse. It is a powerful example of democratizing BI. Historically, when the only tools available were SAS Enterprise Miner or SPSS Modeler, non-technical business analysts at the states have had a difficult time building profiling models. In order to begin the SEMMA process, the analyst had to have already identified the relevant variables and collected data for those variables (SAS Institute 1998). In addition, the analyst had to have some idea of how the variables relate to each other, what transformations might be useful, what statistical techniques might be helpful, and

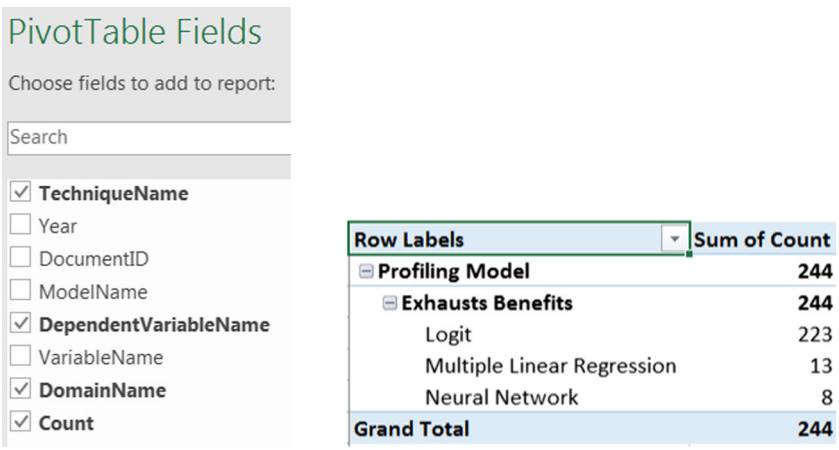
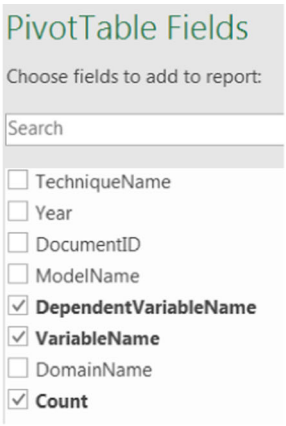


Fig. 4 Pivot table output showing techniques used



Row Labels	Sum of Count
Exhausts Benefits	**244**
Education	27
Industry Code	26
Job Tenure	23
Occupation Code	22
Delay in Filing	11
Replacement Ratio	10
Potential Duration	9
Local Unemployment Rate	9
Weekly Benefit Amount	8
Base Period Earnings	8
County of Residence	7
Claim Month	6
Number of Employers in Base Period	6
Prior Claims	4
Maximum Benefit Amount	4

Fig. 5 Pivot table output showing predictors used

how to assess the usefulness of the models. In general, the non-technical analysts would not know this information.

Of course, one can question whether a non-technical business analyst is capable of judging which model will be the best fit with her/his state's data. The answer to this question depends on whether one feels it is possible to understand what a lift chart, ROC curve, or decile table represents without understanding how they are constructed. There certainly are many examples where people are able to effectively evaluate performance without understanding the underlying mechanism: e.g., the check engine light on a car. Our experience from working with analysts at state agencies is that they can easily understand lift charts, ROC curves and decile tables without understanding how they are constructed. Consider decile tables for example. Analysts do not have to understand how logistic regression works, or even what it is, in order to understand what a predicted probability of exhaustion is, or to understand how a decile table indicates whether the predicted probabilities are accurate. Their lack of understanding of how logistic regression works does not interfere with their ability to use decile tables to correctly assess model accuracy and usefulness.

However, the business analyst *would* know that the modeling domain is "profiling," and the dependent variable is the probability that a UI claimant will exhaust his or her benefits during the claimant's current period of unemployment. With a dimensional model mart, an analyst could select "profiling models" as the Domain Name from the Domain dimension, and "exhausts benefits" as the Dependent Variable Name from the Model dimension. In our prototype model, "profiling models" is the only value that has been entered for Domain Name from the Domain dimension, and "exhausts benefits" is the only value that has been entered for Dependent Variable Name from the Model dimension. Hence in Fig. 4 we only needed to select Dependent Variable Name and Domain Name as the Pivot Table Fields. It was not necessary to slice the Pivot Table Fields any further since "profiling model" is all that was presented for the Domain Name attribute, and "exhaust benefits" is all that was presented for the Dependent Variable Name attribute. The US Department of Labor, several states, and various research organizations, such as Mathematica Policy Research and the Heldrich Center for Workforce Development, have published reports on the construction and use of profiling models to identify UI claimants that are likely to exhaust their benefits. Models for 32 different states are contained in our prototype dimensional model management warehouse. Constraining the dimensions using the problem-specific values of "profiling model" and "exhausts benefits" would enable the analyst to see:

- All of the independent variables (age, education, industry, income, etc.) that have been used to explain this dependent variable,
- All of the techniques (logistic regression, categorical models, neural nets, decision trees, etc.) that have been used to model this dependent variable,
- All of the transformations (logs, reciprocals, power functions, groupings, etc.) that have been performed on the dependent and independent variables to improve the fit of the models,
- All of the measures (lift, percent classified correctly, classification matrices, etc.) that have been used to assess the usefulness of the models, and
- The degree of fit that has been acceptable to other modelers.

Moreover, if the analyst has data for only a limited set of variables, he/she also can filter on that subset to see how well that model has performed for others. This information will enable business analysts to effectively use SAS Enterprise Miner or SPSS Modeler because it fills the gap between the problem definition, which is familiar to the analyst, and the modeling tool's graphical programming interface, which is becoming easier and easier for novices to use.

7 Discussion

As evidenced by the *2014 State of Self-Service BI Report* (Logi Analytics 2014), neither the model management work of the 1980s and 1990s, nor the current code-generating graphical interfaces of SAS and SPSS have enabled self-service BI. Model management research focused on developing modeling languages designed to enable operations research and management science (OR/MS) researchers – i.e., "experts" – to quickly build, store, retrieve, and reuse models. The two major shortcomings of this approach are: 1) it requires modelers to have a deep understanding of statistical modeling; and 2) it assumes the exact same instantiation of a model will be used multiple times. Given these shortcomings, it is not surprising that the systems did not achieve widespread use.

The code-generating interfaces of SAS Enterprise Miner and SPSS Modeler are promising – they greatly reduce the time required to learn the software and speed up the model-building process. However, like previous attempts at simplified modeling languages, they presume a level of mathematical, statistical, and modeling knowledge that is not present in most business analysts. As illustrated in our UI exhaustion example, a typical business analyst may not be able to identify the relevant predictor variables, and probably is even less certain about how to transform variables, select a technique for modeling the relationships among the variables, or assess the usefulness of the model. SAS's SEMMA process provides no assistance with selecting the dependent and independent variables, and provides very little guidance with respect to transformations, technique selection, or assessment.

Kimball (1997) argues that dimensional modeling is the only viable technique to support end-user queries in a data warehouse. We argue that dimensional model marts are the only viable technique to support self-service BI. The intuitive structure of a dimensional data mart enables users who know little about databases or query languages to get the information they need. The widespread availability of tools such as Excel provide a low-cost way of connecting users with these dimensional databases. This has contributed greatly to the ubiquitous use of dimensional data marts.

The same is true for dimensional marts to support model-building. Once the fact and dimension tables are constructed, Excel's pivot table functionality can query the database. The user merely has to click on a specific dimension (or "slicer") to see and assess what analysts in situations similar to their own have done.

In the case of dimensional data marts, non-technical business analysts clearly are able to understand the data and performance measures. Their domain-specific knowledge about the organization makes them more qualified than technical staff to interact with the data. The obstacle for non-technical analysts, prior to dimensional data marts, was simply access to the data.

In the case of model building, non-technical analysts clearly are able to understand the purpose of the model and whether it works. Their domain specific knowledge about the organization makes them more qualified than technical staff to apply the model to their problem. The current obstacle to using models is that non-technical analysts do not understand the models, and are dependent on technical staff to build the models.

Figure 6 shows how a dimensional model management warehouse can fill this gap. Our proposed warehouse can be considered part of a larger model-building decision support architecture, with additional components that facilitate the link between model selection and its implementation in an analytics software package. A Model Selection Layer, such as the Pivot Table described in our Department of Labor case study, would enable the user to select model components. An Analytics Connector Layer would be implemented as a software component that would take the selected model components and automatically build the application-specific code (i.e., SAS, SPSS, or R) that implements the model. This layer would be written by software developers, and would be opaque to the non-technical analysts. Finally, this model could be imported and executed by the target analytics software package to run the model and provide the results to the user. If a cross-application standard such as PMML were used, then the Analytics Connector Layer could create code readable by any analytics software that supports PMML.

Another major deficiency that has been associated with modeling efforts is the tendency to let the data determine the model. This can lead to a model with good fit but poor insight into the problem domain. The fact that humans tend to rely too much on familiar tools has been extensively studied. In 1966 Abraham Maslow noted, "I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail" (Maslow 1966, p. 15). SAS, SPSS, and RapidMiner do nothing to combat this bias on the part of modelers because they facilitate the mechanics of modeling without providing guidance. Clear evidence is the best way to combat biases. Our proposed methodology shows analysts what techniques,

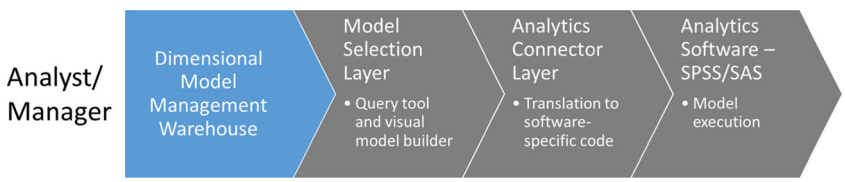


Fig. 6 Architectural role of a dimensional model management warehouse

data, and transformations have been used to model a particular problem domain, and how well they have worked. Allowing analysts to see all of the options that have been used to model a particular dependent variable in a specific context, and their relative performances, will give analysts a broader perspective for evaluating alternatives. Seeing all of the approaches and their relative performance should keep analysts from considering only the approach with which they are most familiar; and seeing the relative performance of the approaches should keep analysts from considering only the approach that has been used most frequently. The end result should be more appropriate model selection.

We show that our proposed methodology works well for a significant problem encountered by the Division of Fiscal and Actuarial Services. However, our methodology is not industry- or problem-specific. The main determinant of applicability to new domains is the type of problem to be solved. Our methodology is most applicable in scenarios where the problem is encountered by more than one person/team within the organization, and multiple attempts have been made to model the problem either across teams or through time. The model builder must also have working knowledge of the business domain in order to choose the correct variables for analysis. As this is a database of previously constructed models, some prior analysis must have been done to populate the database; completely new problems would make this approach less useful.

The problem confronting DFAS is a good example of a problem encountered by multiple teams with a set of previous models to populate the warehouse. While the DFAS case is somewhat unique – 53 geographically separated and distinct units confronting a nearly identical problem with no unit made worse off by sharing its models – the key characteristics of this case are not uncommon. For example, almost every organization needs to make demand forecasts, and frequently multiple units within the same organization need to make demand forecasts for their specific products. These forecasts have many elements in common, and all of the modelers benefit from knowledge sharing. Therefore, making the effort to encode that knowledge in a model management warehouse should be feasible and valuable for a very large number of organizations, ranging from banking to retail.

The scalability of our proposed methodology also can be demonstrated a-priori. In dimensional data marts, tens of millions of queries can be stored, and yet the user can find the one query that he or she is interested in by "slicing-and-dicing" (filtering) on the dimensions. The structure of the dimensional model enables the user to go directly to the proverbial "needle in the haystack." The same is true for a dimensional model mart. By filtering on the dimensions of domain, model, variable, technique, etc., the user will be able to filter out irrelevant information. Therefore, our proposed approach is quite robust and will continue to serve as an effective decision tool no matter how large the model base becomes.

Finally, our proposed methodology should help resolve the problem of when to stop iterating. Box and Draper (1987) point out that it is not possible to obtain a perfect model, and warn modelers against falling into the trap of excessive elaboration" (p. 424). This is a common trap, and the best way to avoid it is to know what it is possible to achieve with a given model. Our methodology enables analysts to see what others have been able to achieve, and therefore provides useful guidance for when it is time to stop the model building process.

In summary, this paper outlines a methodology and technology artifact that together provide modelers with the key pieces of information required to execute the SEMMA process. It is clear that providing this information is a necessary condition for self-service BI, but whether it is enough to give non-technical managers the ability to independently construct useful models must be tested. In this paper we have prototyped a solution for the United States Department of Labor, but future research can empirically test the benefits of our approach by examining metrics of success for model-building. These metrics should include time to create the model, effort expended, and the quality of the results produced by the selected models. This would be the true "missing link" in self-service BI, all but eliminating the need for the nontechnical analyst to be even moderately proficient with statistical packages.

References

Blair, D. C. (2002). The data-document distinction revisited. *The DATA BASE for Advances in Information Systems, 37*(1), 77–96.

Box, G. E. P., & Draper, N. R. (1987). *Empirical model building and response surfaces*. New York: Wiley.

Brooke, A., Kendrick, D., & Meeraus, A. (1988). *GAMS: a user's guide*. Redwood City: Scientific Press.

Burnham, K. P., & Anderson, D. R. (2002). *Model selection and multimodel inference: a practical information-theoretic approach (2nd ed.)*. New York: Springer.

Choobineh, J. (1991). SQLMP: a data sublanguage for representation and formulation of linear mathematical models. *ORSA Journal on Computing, 3*(4), 358–375.

Cox, D. R. (1995). Comment on "model uncertainty, data mining and statistical inference". *Journal of the Royal Statistical Society: Series A (Statistics in Society), 158*(3), 455–456.

Cunningham, K., & Schrage, L. (2004). The LINGO Algebraic Modeling Language. In J. Kallrath (Ed.), *Modeling languages in mathematical optimization* (pp. 159-171). Kluwer Academic Publishers.

Davenport, T. H. (2013). Telling a Story with Data. *Deloitte Review, 12*.

Davenport, T. H., & Kim, J. (2013). *Keeping up with the quants.* Harvard Business Review Press.

Davenport, T. H., Harris, J. G., De Long, D. W., & Jacobson, A. L. (2001). Data to knowledge to results: building an analytic capability. *California Management Review, 43*(2), 116–138.

Deokar, A., & El-Gayar, O. F. (2011). Decision-enabled dynamic process management for networked enterprises. *Information Systems Frontiers, 13*(5), 655–688.

El-Gayar, O. F., & Deokar, A. (2013). A semantic service-oriented architecture for distributed model management systems. *Decision Support Systems, 55*(1), 374–384.

Finlay, P. N., & Mitchell, A. C. (1994). Perceptions of the benefits from the introduction of CASE: an empirical study. *MIS Quarterly, 19*(4), 353–370.

Fourer, R., Gay, D. M., & Kernighan, B. W. (1990). A modeling language for mathematical programming. *Management Science, 36*(5), 519–554.

Geoffrion, A. M. (1987). An introduction to structured modeling. *Management Science, 33*(5), 547–588.

Geoffrion, A. M. (1989). The formal aspects of structured modeling. *Operations Research, 37*(1), 30–51.

Guazzelli, A., Zeller, M., Lin, W-C., & Williams, G. (2009). PMML: an open standard for sharing models. *The R Journal, 1*(1), 60. http://journal.r-project.org

Guinan, P. J., Cooprider, J. G., & Sawyer, S. (1997). The effective use of automated application development tools. *IBM Systems Journal, 36*(1), 124–139.

Hampton, J. (2011). SEMMA and CRISP-DM: data mining methodologies. JessHampton.com http://jesshampton.com/2011/02/16/semma-and-crisp-dm-data-mining-methodologies. Accessed 21 July 2016.

HBR Analytic Services (2012). The evolution of decision making: how leading organizations are adopting a data-driven culture. *Harvard Business Review.* https://hbr.org/resources/pdfs/tools/17568_HBR_SAS%20Report_webview.pdf. Accessed 18 September 2015.

Henschen, D. (2014). IBM Watson analytics goes public. *InformationWeek.* http://www.informationweek.com/big-data/big-data-analytics/ibm-watson-analytics-goes-public/d/d-id/1317887. Accessed 14 February 2015.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 77–105.

Iivari, J. (1996). Why Are CASE tools not used? *Communications of the ACM, 39*(10), 94–103.

Kimball, R. (1997). A dimensional modeling manifesto. *Database Magazine, 10*(9), 59–78.

Kottemann, J. E., & Dolk, D. R. (1992). Model integration and modeling languages. *Information Systems Research, 3*(1), 1–16.

Kridel, D., & Dolk, D. (2013). Automated self-service modeling: predictive analytics as a service. *Information Systems and E-Business Management, 11*(1), 119–140.

Lending, D., & Chervany, H. L. (1998). CASE tools: understanding the reasons for non-use. *Computer Personnel, 19*(2), 13–26.

Lin, E., Schuff, D., & St. Louis, R. (2000). Subscript-free modeling languages: a tool for facilitating the formulation and use of models. *European Journal of Operational Research, 123*(3), 614–627.

Logi Analytics (2014) *State of Self-Service BI Report.* http://images.learn.logixml.com/Web/LogiAnalyticsInc/%7B7c21cd62-221c-44af-9ecd-a35265bc8e34%7D_LogiAnalytics-2014StateOfSelfService-Artwork-1028.pdf. Accessed 8 February 2015.

Lundell, B., & Lings, B. (2004). Changing perceptions of CASE technology. *Journal of Systems and Software, 72*(2), 271–280.

Madhusudan, T. (2007). A web services framework for distributed model management. *Information Systems Frontiers, 9*(1), 9–27.

Maslow, A. H. (1966). *The psychology of science.* Chicago: J. Dewey Society.

McMurtrey, M. E., Grover, V., Teng, J. T. C., & Lightner, N. J. (2002). Job satisfaction of information technology workers: the impact of career orientation and task automation in a CASE environment. *Journal of Management Information Systems, 19*(2), 273–302.

Murphy, F. H., Stohr, E. A., & Asthana, A. (1992). Representation schemes for linear programming models. *Management Science, 38*(7), 964–991.

Object Management Group (2003). Common Warehouse Metamodel (CWM) Specification. http://www.omg.org/spec/CWM/1.1/PDF/. Accessed 24 April 2015.

Pack, D. J. (1987). A practical overview of ARIMA models for time series forecasting. In S. G. Makridakis & S. C. Wheelwright (Eds.), *The handbook of forecasting: a managers guide* (pp. 196–218). New York: Wiley.

Pechter, R. (2011). PMML conformance progress report – five years later. In *Proceedings of PMML'11* (pp. 6–15). New York: ACM Press.

Powell, S. R. (2015). *Summary of state models*. Unpublished working paper. New Brunswick, NJ: John J. Heldrich Center for Workforce Development.

Power, D., & Sharda, R. (2007). Model-driven decision support systems: concepts and research directions. *Decision Support Systems, 43*(3), 1044–1061.

Rohanizadeh, S., & Moghadam, M. (2009). A proposed data mining methodology and its application to industrial procedures. *Journal of Industrial Engineering, 4*, 37–50.

SAS Institute (1998). Data Mining and the Case for Sampling. *SAS Institute Best Practices Paper*, Carey, NC.

Senn, J. A., & Wynekoop, J. L. (1995). The other side of CASE implementation. *Information Systems Management, 12*(4), 7.

Sottara, D., Mello, P., Sartori, C., & Fry, E. (2011). Enhancing a production rule engine with predictive models using PMML. In *Proceedings of PMML'11* (pp. 39–47). New York: ACM Press.

Wit, E., van den Heuvel, E., & Romeijn, J.-W. (2012). 'All models are wrong…': an introduction to model uncertainty. *Statistica Neerlandica, 66*(3), 217–236.

David Schuff is Professor of Management Information Systems in the Fox School of Business and Management at Temple University. David's research interests include the application of information visualization to decision support systems, data warehousing, and the impact of user-generated content on organizations and society. His work has been published in MIS Quarterly, Decision Sciences, Decision Support Systems, Information & Management, Communications of the ACM, Computer, and Information Systems Journal. He holds a BA in Economics from the University of Pittsburgh, an MBA from Villanova University, an MS in Information Management from Arizona State University, and a Ph.D. in Business Administration from Arizona State University.

Karen Corral is an Associate Professor in the Department of Information Technology and Supply Chain Management at the College of Business and Economics at Boise State University. She holds a BA in English from the University of Michigan, an MS in Computer Information Systems from Arizona State University, and a Ph.D. in Business Administration from Arizona State University. Her research interests are in the area of data and knowledge management as related to decision support. Her work has been published in journals such as Communications of the ACM, Information Systems Frontiers, Decision Sciences, and Decision Support Systems.

Robert D. St. Louis is a Professor of Information Systems at Arizona State University. He received his AB degree from Rockhurst College, and his MS and Ph.D. degrees from Purdue University. He began teaching at ASU in 1969, but spent the period from 1976 through 1981 working full time as a researcher for state and federal agencies. Dr. St. Louis teaches classes on information management, and conducts research in the areas of document search, enterprise performance management systems, and evidence based decision making. He also conducts seminars for the US Department of Labor on data driven forecasts, and consults with state and federal agencies on predictive analytics.

Gregory Schymik is an Assistant Professor in the School of Computing and Information Systems at Grand Valley State University. Greg holds a BSE in Computer Engineering from the University of Michigan, an MSCIS from the University of Detroit-Mercy, and Ph.D. in Business

Administration from the W. P. Carey School of Business at Arizona State University. Before starting his doctoral studies, Greg spent 15 years in embedded systems development in the automotive industry. He is co-inventor on two patents. His research interests include enterprise search, information literacy and its impact on the use of information systems, information security and assurance, and work system theory. He has published in Decision Analysis, Decision Sciences, and the International Journal of Business Intelligence Research.