ABSTRACT

Organizations are beginning to apply data mining and knowledge discovery techniques to their corporate data sets, thereby enabling the identification of trends and the discovery of inductive knowledge. Since traditional transaction databases are not optimized for analytical processing, they must be transformed. This article proposes the use of modular components to decrease the overall amount of human processing and intervention necessary for the transformation process. Our approach configures components to extract data-sets using a set of “extraction hints.” Our framework incorporates decentralized, generic components that are reusable across domains and databases. Finally, we detail an implementation of our component-based framework for an aviation data set.

Keywords: component-based software engineering; data warehousing; knowledge and data engineering; software reusability

INTRODUCTION

Over the past decade, government and industry organizations have enhanced their operations by utilizing emerging technologies in data management. Advances in database methodology and software (i.e., warehousing of transaction data) has increased the ability of organizations to extract useful knowledge from operational data and has helped build the foundation for the field of
knowledge discovery in databases (KDD) (Fayyad, Piatetsky-Shapiro, & Smyth, 1996; Sarawagi, Thomas, & Agrawal, 2000; Software Suites supporting Knowledge Discovery, 2005). KDD consists of such phases as selection, pre-processing, transformation, data mining, and interpretation/evaluation. Selection involves identifying the data that should be used for the data mining process. Typically, the data is obtained from multiple, heterogeneous data sources. The pre-processing phase includes steps for data cleansing and the development of strategies for handling missing data and various data anomalies. Data transformation involves converting data from the different sources into a single common format. This step also includes using data reduction techniques to reduce the complexity of the selected data, thereby simplifying future steps in the KDD process. Data mining tasks apply various algorithms to the transformed data to generate and identify “hidden knowledge.” Finally, the area of interpretation/evaluation focuses on creating an accurate and clear presentation of the data mining results to the user.

Excluding the data mining phase, where there are a plethora of automated algorithms and applications, the other phases are mostly human-driven. Data experts are required to complete the tasks related to the majority of steps in the KDD process as explained in the following.

- **Data formatting, loading, cleaning and Anomaly detection.** In the pre-processing phase, data experts must correct and update incorrect data values, populate missing data values, and fix data anomalies.
- **Adding important meta-data to the database.** In the data transformation phase, data must be integrated into a single model that supports analytical processing. This typically involves adding meta-data and converting data sets from text files and traditional relational schemas to star or multi-dimensional schemas.
- **User and tool-generated hints.** In the final phases (i.e., data mining and evaluation), general approaches are needed to assist users in preparing knowledge discovery routines and analyzing results. These general approaches must allow the user to manually specify potential correlation areas or “hints.” In the future, the suggestion of new hints may be automated by intelligent software mechanisms.

These human-driven tasks pose problems since the initial data set, which we will refer to as the raw data, is large, complex and heterogeneous. Our work attempts to reduce the amount of time required for human-driven tasks in the KDD setting. General reusable components may represent a feasible solution to assist in the execution of the time-consuming processing tasks underlying KDD. In this paper, specific tasks suitable for such components are identified and characterized. In addition, a component-based framework and corresponding process are described to address these tasks.

The paper proceeds in the following section with a discussion of related work with respect to component-based KDD. The paper then introduces the Component-Based Knowledge Discovery in Databases (C-KDD) framework. Subsequent sections provide specific low-level technical details of the C-KDD framework and, in the final
sections, the C-KDD is used in an aviation-based study.

RELATED WORK

In practice, the application of KDD to the aviation domain has been done in a limited number of studies. In fact, there are few approaches known by the authors that use data mining techniques in the aviation domain. Earlier aviation studies (Callahan, De Armon, Cooper, Goodfriend, Moch-Mooney, & Solomos, 2001; Nazeri & Jianping, 2002) use static, specialized techniques for aviation security and studies in weather. These earlier approaches do not leverage current data modeling approaches or follow a general purpose design. With respect to KDD-related research, there are many approaches that investigate the data mining phase (Agrawal & Shim, 1996; Netz, Chaudhuri, Fayyad, & Bernhardt, 2001), but few approaches that address the human efforts are particularly using component-based development. Bueno (1998) discusses the benefit of using components to assist with the KDD process. However, similar to other KDD-related research, Bueno focuses on the components for the data mining stage of the process. Bueno does not significantly detail the connection of their components to the underlying database. C-KDD focuses on all KDD stages with an emphasis on the tedious human-driven data collection phase.

Chattratichat, Darlington, Guo, Hedvall, Kohler, and Syed (1999) and Engels (1999) describe component architectures to assist human users in KDD. Chattratichat et al. (1999) address data mining across different organizational regions and Engels (1999) focuses on assisting in the selection of data mining algorithms. Neither project considers support for the human-driven steps required to initially capture the data.

Kim et al. (2002) describes Chamois, a component-based framework implemented by composing a number of third-party software packages. The goal of Chamois is similar to C-KDD; however, the focus is not on the integration of component capabilities at the specification level from the top-down, instead it focuses on building up a framework based on existing applications. The C-KDD approach focuses on the communication channels, particularly from human-to-component. The innovation of C-KDD is the formal data transfer specifications which have not been found in related frameworks supporting the full KDD process.

C-KDD FRAMEWORK

C-KDD is a component-based framework designed to support KDD phases. Five distinct KDD components were chosen to realize the various phases of the KDD process. These KDD components are the Data Extraction Component, Loader Component, Data Integration Component, Data Mining Component, and Hints Component. The Data Extraction Component and Loader Component combine to extract data from several heterogeneous data sources and populate a central relational data repository. A human, domain specialist identifies common fields between relational entities, and the Data Integration Component transforms these entities into the new component-accessible model. Finally, in the Data Mining, Interpretation, and Evaluation Phases, a human, domain specialist initiates a data mining scenario with the help of the Hints and Data Mining Components. The Hints Component presents the available data model to the domain special-
ists and accepts the human instructions to submit specialized studies to be executed by the Data Mining Component. The C-KDD framework is illustrated in Figure 1.

In the scope of this paper, we highlight the C-KDD specification techniques that allow human, domain experts to program components and automate portions of the KDD process. The following three sub-sections discuss how domain experts can specify data extraction/loading directives, data transformation instructions, and knowledge discovery instructions.

**Component Specification for Data Extraction and Loading**

In the selection and pre-processing phases of KDD, various data sources must be identified and the underlying data must be captured in machine-readable format. The approach taken in this work is to perform a component-mediated step that collects heterogeneous data and populates that data in several relational database tables using traditional approaches (i.e., existing data transformation techniques are embedded in component (Chawathe et al., 1994; Nodine, Fowler, Ksiezyk, Perry, Taylor, & Unruh, 1998)).

The C-KDD framework considers several types of data formats. It was not possible in the initial investigations to consider every possible format, but samples of data were taken from the initial deployment domain (i.e., aviation studies). There were several types of structured and semi-structured data considered as listed in the following:
Delimited Data in Row Format, Relational Database Format, Name/Value Format, and Mark-Up Language Format are standard structured formats for capturing data. Using general functions and minimal human intervention, the aforementioned formats are extracted with ease by the Data Extraction Component. Human intervention can be defined in two tasks. The first row in most input files contains the column headings. With respect to human intervention in this case, the human is required to confirm these headings or change the naming, if necessary. In other files, the column headings are not included; therefore, the human user must specify column headings. In mark-up files, XML element names are used as headings. In general, for these types of files, the overhead is relatively low, since the user is only required to enter the names or make confirmations with a couple of button presses. There are some exceptional cases, but it was discovered in the aviation domain that this degree of human intervention was sufficient in 17 out of 19 relevant cases.

Exploiting semi-structured data (i.e., Delimited Formats in Non-Fixed Row Format) requires a relatively higher degree of interaction between the human user and the Data Integration Component. The C-KDD framework extends an existing approach referred to as templating to describe the semi-structured data schema to the components. With respect to the C-KDD approach, the specification of a template allows the user to flatten hierarchical data into row format. The template consists of several aspects. We define a zone to be the area of a template identified by delimitation parameters (e.g., semi-colon, space, or tab-delimited). Data can be delimited by a string delimiter and also by using fixed column locations. There are specialized attributes that describe the start and end of a zone. Other attributes allow the user to specify a zone by number of columns. It is not in the scope of this paper to describe the templating technique in detail, but technical details can be found in related work (Blake, Singh, Williams, Norman, & Sliva, 2005).

Component-Mediated Processes for Data Transformation

The C-KDD framework includes a Data Transformation Component that generates a generalized, denormalized database. In the C-KDD framework, we introduce a process that generalizes the database schema into a model that C-KDD components can navigate. This is a reproducible process that is valid for any database with entities that share related columns. This process includes the creation of fact tables (Kimball, 1996; Kimball, Reeves, & Ross, 1998) that connect entities. With respect to the C-KDD framework, there is a CorrelationSpace entity that links multiple CorrelationAttribute entities. The CorrelationSpace table corresponds to the fact table in a star schema. The CorrelationAttribute entity corresponds to the dimension table in the star schema. Example CorrelationAttributes include time and location. Although the results will be discussed in a later section, Figure 2 contains an aviation-based model developed using this approach. There is a
CorrelationSpace table, and the AreaSpace and TimeSpace tables are both CorrelationAttributes.

The process for creating the denormalized database in Figure 2 is as follows:

1. Human domain specialist identifies related columns across multiple normalized tables.
2. The domain specialists identifies the specific range of data for the related columns/attributes
   - This specification may consist of a range of time, a specific area, or a range of keyed information.
3. Assuming columns are sorted, C-KDD components create generic CorrelationAttribute tables and preload records for data within that specified range.
   - For example, if the user specifies the time between May and June, the component would build a Correlation Attribute (i.e., TimeSpace) table that has a record for each time increment within that range.
4. Components query the initial data tables and record-by-record create correlation records in the CorrelationSpace entities based on matching the record to the preloaded correlation attributes.
   - Considering the earlier example in step 3, the component would extract a record from the original table and search for the time increment corre-
sponding to that record. The two individual records would be merged or “joined” into a single composite record in the CorrelationSpace table.

5. Additional detailed tables can be generated/connected to further describe the correlation attributes. (This is not a component-automated step.)

By using the generic Correlation Space and CorrelationAttribute table structures, component capabilities can be duplicated on any database containing these meta-information tables. The Data Mining Components are not limited by having to hard-code column names and table locations; instead, queries can be managed based on pre-defined conditions. Also, this process essentially pre-loads entity joins into the database model. During loading, the Data Integration Components do not create duplicate correlation identifiers (i.e., CSpaceid in Figure 2). Instead, existing correlation records are reused. In this way, for example, when a weather record has the same correlation conditions (the same time and same airport) as a terminal performance record, both records have the same CorrelationSpace identification. This modeling approach separates the domain-specific information from the database semantics, an approach akin to the separation of concerns in software design. This separation allows the Data Mining Components to be able to perform basic text comparisons without knowledge of domain-specific concerns when generating data sets.

Component-Mediated Data Mining Routines Using Extraction Hints

A major innovation in this work is the formal approach and corresponding user interface design that allows human domain specialists to collaborate with components on data mining scenarios. This approach uses a representation called extraction hints as instructions to the Hints Component. Domain specialists, at times, need to determine trends based on a composite list of constraints. In C-KDD framework, these constraints, based on qualifying events, are modeled as extraction hints. In the aviation domain, qualifying events can be defined as the combination of weather, processes, and performance conditions (i.e., “Give specified data when the temperature is greater than 70 degrees and the airport arrival rate is less than 15 per hour”).

There are two major aspects of the extraction hint that human users can provide. The first aspect, called the search criteria, allows the input of a basic constraint. For example, a user might direct the component to explore situations where the cloud ceiling was greater than 1,000 feet and temperature was greater than 90 degrees. This search criteria will specify records from the ASPMData table (Figure 2) where the values of the ceiling column is greater than 1,000 and the temperature column is greater than 90.

Once a search criteria is set, the user can also suggest information points. An information point is defined as other information related to the search criteria, as constrained by the correlation attributes (in this case, time and airport). For example, a user may specify an information point as the visibility at the same time and area as the search criteria constraint. A user can also specify an information point for a different location for a different time, perhaps at another airport and, for the time, three hours before the time captured when the search criteria is met. Both the search criteria and information points are composed
of the correlation attributes represented in the correlation records.

The Hints Component has a specialized user interface that supports the collaboration of extraction hints between the domain specialists and the Hints Component. At initiation, the domain specialist suggests an initial extraction hint using this interface. Once results have been generated, the Hints Component can suggest a variation of the original hint for further data mining. In future work, the intent is to convert the Hints Component into a software agent to manage new, fully automated data mining scenarios. The Hints Component user interface as customized for the aviation domain is shown in Figure 3.

The search criteria is specified on the left side of the user interface. The Table textbox dynamically pulls table names from the C-KDD database. Once the table is chosen, the database columns are dynamically populated in the Field textbox. The domain specialist can designate a value (Value textbox) using any of the most common relational operators, that are populated in the Relation field. The Preferred Name textbox allows the user to associate a personalized heading for the search criteria. The information point is specified on the right side of the user interface, but was named search information for consistency. The domain specialist again can specify a preferred name, table, and field. In addition, the domain specialist can specify if there will be one data return correlated with the search criteria or multiple data returns (Precision textbox). The CSpaceTimeType and CSpaceAreaType textboxes and corresponding textboxes al-
allows the domain specialist to vary the correlation between the search criteria and search information. This user interface has been customized for the aviation domain so, in the case of this illustration, the correlation is based on time and location. However, it should be noted that the correlation attributes are not fixed, but that this implementation is just one embodiment of the approach. The user interface in Figure 3 shows one tab, but multiple tabs can be added on-demand by the domain specialist. Each tab has a new search criteria and information point (search information). Finally, the Hints Component creates an XML message that directs the Data Mining Component on how to extract datasets from the relational database. In the interest of space, there is just an overview of the capabilities of this user interface presented here.

ASCEND: C-KDD FOR ANALYSIS IN THE AVIATION DOMAIN

In evaluating the effectiveness of the C-KDD framework and approaches, the software was customized and deployed in an operational setting and used for an analysis on real data sources. The C-KDD framework and initial prototype was used in a joint project of the Center for Advanced Aviation System Development (CAASD) of The MITRE Corporation and the Department of Computer Science of Georgetown University. This joint project called Agent-Based Software for the Correlation of EN-route Data (ASCEND) was a customization of the C-KDD framework in the aviation domain. The ASCEND software was built predominantly using the C-KDD prototype. An additional goal of the project was to extend the C-KDD components using agent-oriented concepts, but those efforts are out of the scope of this paper.

Initially, the C-KDD software consisted of about 23,000 lines of Java code. The learning components were implemented by integrating the WEKA data mining application and toolkit (WEKA, 2005). The WEKA software was chosen for this research project, since it was also written in Java code and provided run-time program interfaces. The denormalized database was implemented using Oracle9iLite. Oracle9iLite is a fully operational relational database management system that runs on a personal computer. This personal database was chosen to support the portability of the software and the database as one package. This ability made it possible to transfer the entire application and database from machine to machine on one CD. Both the C-KDD database and external databases (as data sources) were accessed using several JDBC interface software. The communication component was implemented using shared object space, specifically Sun Microsystems JavaSpaces implementation. The software for communication was an extension of the agent communication methods developed in earlier work (Blake, 2003b; Blake & Williams, 2003a).

Several software changes were required to customize the C-KDD framework for ASCEND. Several specialized objects were created to converge latitude and longitude values for airports, airspace locations, and weather locations into a common attribute. The user interfaces were customized for more specialized extraction hint generation. The sum of software changes was less than 2,000 lines of code which was favorable with regards to the usability of C-KDD.
The ASCEND Database

The ASCEND component-accessible database was populated with several data sources from multiple heterogeneous sources as listed in Table 1. Access to these data sources was provided by The MITRE Corporation. The final ASCEND data model is closely related to the schema used for the C-KDD approach. The final data model is illustrated in Figure 2.

The data sources listed in Table 1 are shown in the data model with direct association to the CorrelationSpace entity. The CorrelationAttribute entities were given more specific names expressed as database views. The names of the two CorrelationAttribute entities are AreaSpace and TimeSpace. The shaded area represents the meta-data tables added for component navigation by the Data Integration Component. The additional tables in the model are specialized tables added to further describe the CorrelationAttribute entities. The additional specialized Java code described in the previous section extended the C-KDD framework to create and access the additional entities (i.e., Vertice, Polygon, Time, Duration entities).

Verifying C-KDD by Regenerating Known Aviation Rules

C-KDD was evaluated by performing aviation-specific knowledge discovery routines using real operational data. The ASCEND database was populated with 6 months of data from May 2001 to October 2001 using the data sources described in the previous section. This six-month time frame is of great importance to traffic flow management in the United States because the convective weather (i.e., thunderstorms) that occurs during these months is the greatest cause of aviation delay (Convective Weather PDT, 2005). The final database was 600 megabytes in size.

The intention of the experimentation was to verify the correctness of the framework by performing studies that result in known rules. The intention of the first study, involving Los Angeles International Airport, was to verify the C-KDD data mining tools and the second study, involving Atlanta...

---

**Table 1. Data sources implemented in ASCEND**

<table>
<thead>
<tr>
<th>Data Source Name</th>
<th>Data Format Type</th>
<th>Source of Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Stop (Order 7210.3S, 2003)</td>
<td>Microsoft Access Database (structured)</td>
<td>Federal Aviation Administration (FAA)</td>
<td>Procedure used to stop operations at a specific airport or group of airports with time and duration</td>
</tr>
<tr>
<td>Ground Delay Program (GDP)</td>
<td>Microsoft Access Database (structured)</td>
<td>Federal Aviation Administration (FAA)</td>
<td>Procedure used to control the flow of arrivals into an airport or group of airports with a time and duration</td>
</tr>
<tr>
<td>Advisory Information</td>
<td>Delimited Text in Non-Fixed Row Format (unstructured)</td>
<td>Federal Aviation Administration (FAA)</td>
<td>Specific information about the status of the National Airspace System</td>
</tr>
<tr>
<td>ASPM Data (ASPM, 2005)</td>
<td>Delimited Text in Row Format</td>
<td>Federal Aviation Administration (FAA)</td>
<td>Airport performance metrics</td>
</tr>
<tr>
<td>National Convective Weather Forecast (NCWF) - STORM (National Weather Service, 2005)</td>
<td>Delimited Text in Non-Fixed Row Format (unstructured)</td>
<td>National Weather Service</td>
<td>Polygons representing areas of convective or inclement weather (prediction)</td>
</tr>
<tr>
<td>National Convective Weather Forecast (NCWF) - DETECTION</td>
<td>Delimited Text in Non-Fixed Row Format (unstructured)</td>
<td>National Weather Service</td>
<td>Polygons representing areas of convective or inclement weather (current forecast)</td>
</tr>
<tr>
<td>METAR</td>
<td>Delimited Text in Row Format (structured)</td>
<td>National Oceanic and Atmospheric Administration (NOAA)</td>
<td>Weather situation over a particular airport</td>
</tr>
</tbody>
</table>
Hartsfield International Airport focused on verifying the data transformation mechanisms. The rules resulting from both studies would indicate trends that support several well-known facts in aviation (i.e., when visibility is bad, aviators are required to use their instruments also certain weather conditions negatively affect the performance of airport departure and arrival operations).

The first study was helpful in evaluating the accuracy of the Data Mining Components by re-generating known rules in real operational data. The chosen experiment was to re-engineer flight rules. From an aviation perspective, when visibility is below a certain distance and ceiling (cloud-level) is below a certain altitude, the FAA institutes the instrument flight rules (IFR) restriction as opposed to the visual flight rules (VFR) designation during normal conditions. During IFR, any pilot flying an aircraft should be able to operate the plane using solely the instruments. In the M- 
TRE-maintained METAR data source (National Weather Service, 2005), the MetCon (Flight Rule) field is determined by the value of the Visib (visibility) and the Ceil (Cloud Ceiling) columns. If the Visib field is less than four miles and the Ceil field is less 700 feet, then the MetCon is set to IFR. Otherwise, the MetCon is set to VFR. There are other conditions, but it is not in the scope of this paper to discuss the rules in detail. It is more important to understand that the MetCon column is a known function of the Ceiling and Visibility. The study was executed on one data source (i.e., METAR) to concentrate on the data mining functions. In setting up this experiment, the search criteria was set to Los Angeles airport and the information points were set to most of the METAR attributes (i.e., ceiling, visibility, flight rules, temperature, barometric pressure, wind speed, wind direction, and dew point). The relevant columns were not specified; therefore, a successful experimental result could be shown in two outcomes. The first outcome would be that ceiling, visibility and flight rules should stand out as having the strongest correlation. A second successful outcome would be the determination of correct rules for flight rule designations. The experiment was successful from both aspects. Executing the experiment on all available METAR data for Los Angeles, the decision tree created using Data Mining Component (this component wraps the WEKA software (WEKA, 2005)) represented the known flight rule correlations. There were no significant deviations.

Using the same principle of discovering known rules, the second evaluation experiment was toward the verification of data integration processes and tools. In this experiment, the correlation between wind speed and airport performance was measured. Considering the fact that an aircraft lands against the wind and takes-off against the wind, excessively high wind speed tends to have a strong correlation to poor performance (low arrival/departure rates, high cancellation rates) in airports on a specific day. In this second experiment, we evaluated the affect of wind speed using the METAR data source with the airport performance using the ASPM data source (i.e., integration of separate data sources). As with the first experiment, the study contained many more columns than the columns with known correlation. The experiment was run on data for the Atlanta Hartsfield International Airport (ATL).

For a successful verification in this second evaluation, the expectation was that the high wind speed would result in low
departure and arrival counts at Atlanta airport. Again, there were sufficient results to verify the tools ability to integrate the two data sources. The METAR and ASPM data points were reported approximately twice a day during the period of time captured in this study. The Arrival and Departure counts were low in the majority of the cases when the wind speed was high. Several MITRE analysts were consulted to confirm that the results were consistent with the trends of weather-impacted operations at the facilities. The specific aviation-based results of the studies can be found in related literature (Blake et al., 2005).

DISCUSSION

In this paper, we describe a component-based framework that facilitates a human domain specialist in developing and executing KDD routines. This work represents a novel investigation of component capabilities for use in knowledge discovery. The C-KDD framework has been applied to the aviation domain and is currently in use for analysis purposes at The MITRE Corporation. We introduce the notion of a correlation space that enables us to generalize our components related to data preprocessing and data transformation. The innovation in the ASCEND project is the ability of the user to define the data mining scenario with the assistance of a general component-based framework. In addition, this is perhaps the first general-purpose framework that exploits KDD in the aviation domain. This paper presents an experience report of the C-KDD, but in-depth technical details can be found in related literature (Blake et al., 2005).

Although the current implementations have been deployed as a local workstation application, in future work, we plan to distribute the C-KDD engine as a network accessible web service. The underlying technologies (i.e., Java and Jini) would facilitate these future extensions. In other future work, we plan to evaluate the optimality of our correlation modeling approach by recapturing our data model in other more standard formats. With regards to aviation studies, we plan to extend our current studies using a full year of aviation data. Although the May through October period (as analyzed in this work) is most relevant for traffic flow experimentation with respect to delays (Convective Weather PDT, 2005), other times during the year may be more valid for other concerns such as ground operations.

ACKNOWLEDGMENTS

Our thanks to several undergraduate student developers, Lindsay Blohm, Todd Cornett, and Denitsa Apostolava. We thank David Hite, Urmila Hiremath, John Mack, and Joe Hoffman of the MITRE Corporation for their discussions in the evaluation of the framework and the interpretation of the data mining results. This work was produced for the U.S. Government under Contract DTFA01-01-C-00001 and is subject to Federal Aviation Administration Acquisition Management System Clause 3.5-13, Rights In Data-General, Alt. III and Alt. IV (Oct. 1996). The contents of this material reflect the views of the author and/or the Director of the Center for Advanced Aviation System Development. Neither the Federal Aviation Administration nor the Department of Transportation makes any warranty or guarantee, or promise, expressed or implied, concerning the content or accuracy of the views expressed herein.
REFERENCES


Kim, W., Park, S., Yong, H., Chae, K., Cho,


*M. Brian Blake received the BS in electrical engineering from the Georgia Institute of Technology, Atlanta, the MS in electrical engineering (minor in software engineering) from Mercer University, Atlanta, Georgia, and the PhD in information technology with a concentration in information and software engineering from George Mason University, Fairfax, Virginia. He is currently an associate professor in the Department of Computer Science, Georgetown University, Washington, DC. As a professor, he has published more than 50 journal and refereed conference papers in the domains of workflow and agent-based systems, service-oriented computing, distributed data management, and software engineering education.*

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
His investigations cover the spectrum of software engineering: design, specification, proof of correctness, implementation/experimentation, performance evaluation, and application.

Lisa O. Singh is currently an assistant professor in the Department of Computer Science at Georgetown University. Prior to joining the university in 2002, Dr. Singh was a database advisor/architect and data warehouse consultant for small- to mid-size firms in the Bay Area. She was also the Database Architect and Senior Application Programmer for the Internet startup, MyWorkTools.com. She received her PhD from Northwestern University in 1999. Her research is in database management systems, specifically addressing issues associated with knowledge discovery for streaming data sets, event prediction in social networks, and data mining of large, heterogeneous data sets.

Andrew B. Williams received the BS and PhD in electrical engineering from the University of Kansas, Lawrence. He received the MS in electrical and computer engineering from Marquette University, Milwaukee, Wisconsin. He is currently an assistant professor in the Department of Computer and Information Sciences, Spelman College, Atlanta, Georgia. He has published several journal and conference papers in the areas of autonomous agents and multi-agent systems, ontologies, bioinformatics, and computer engineering education. His current research interests include ontology merging and learning, ontology consensus for biomedical applications, teamwork in multi-agent systems, and robotics.

Wendell N. Norman received the BS in computer science from the Georgetown University, Washington, DC. He is currently a software engineer with The MITRE Corporation, McLean VA. While at Georgetown, he conducted several undergraduate research projects in the area of agent-based data management and java-based information systems. At The MITRE Corporation, he investigates the use of information technologies for next-generation aviation-based simulation systems.

Amy L. Sliva received the BS in computer science from the Georgetown University, Washington, DC. She is currently a PhD student in the Department of Computer Science at the University of Maryland-College Park. While at Georgetown, she spent three years investigating data management approaches for agent-based systems and for the service-oriented computing environment. In 2005, she was a finalist for the Computing Research Association (CRA) Undergraduate Research Award.