Proceedings of the 2005 Systems and Information Engineering Design Symposium
Ellen J. Bass, ed.

ENGINEERING TRADE STUDY: EXTRACT, TRANSFORM, LOAD TOOLS FOR DATA MIGRATION

Scott Henry
Sherlynn Hoon
Meeky Hwang
Diane Lee
Michael D. DeVore

Department of Systems and Information Engineering
University of Virginia
Charlottesville, VA 22903

ABSTRACT

The widespread need within corporate Information Systems (IS) divisions to migrate large quantities of data between data stores has spawned a family of commercial products, which are commonly referred to as Extract-Transform-Load (ETL) tools. The focus of this paper is the development of engineering trade studies to be used for ETL tool evaluation. This approach: (1) Identifies selection criteria that are essential in the evaluation of an ETL tool, (2) Develops scenarios that examine each criterion, and (3) Develops quantitative measures useful for evaluating the various aspects of ETL usage. This approach generates replicable evaluation methods that can be used and modified by companies to address their own ETL product needs. With the results generated through such evaluations, companies will be able to make informed decisions and choose the best ETL tool for their purposes.

1 INTRODUCTION

The widespread need within corporations to migrate large quantities of data between data stores and to turn the data into strategic information has generated a set of commercial products, which are referred to as Extract-Transform-Load (ETL) tools. This paper documents an engineering trade study for evaluating ETL tools. The trade study consists of selection criteria, scenarios, quantitative measures, and evaluation procedures. This report outlines the structure, technology, and applications of the trade study.

1.1 Problem Definition: Selecting an ETL tool

In the 1990s, business executives became desperate for information to stay competitive and improve their business operations. Operational computer systems did provide information to run daily operations, but what executives needed were different types of information that could be used to make strategic decisions. This prevalent need within corporations spurred the development of ETL tools.

Since its origins in the mid-1990s, the ETL market has evolved rapidly. A number of products have reached a level of maturity and next generation products are becoming available. Many corporations have a need to integrate their data from relational databases and to transform data into information for more efficient decision-making.

The focus of this methodology is developing a standardized test suite consisting of selection criteria, scenarios, quantitative measures, and evaluation procedures. In addition, these procedures are applied to commercial ETL products to examine the effectiveness of the test suite in the evaluation of ETL tools. This methodology is replicable and adaptable, allowing any company to apply it to new generation ETL tools and other software products.

1.2 Context: Importance of Business Intelligence

Business intelligence (BI) has matured to be an increasingly well-defined subset of information technology that addresses the challenges of turning raw data into useful information. It encompasses issues such as reporting, analysis, data presentation, and data integration.

BI technologies attempt to help people understand data more quickly so that they can make better and faster decisions. The key drivers behind BI objectives are to increase organizational efficiency and effectiveness (Eckerson, 2003). The ETL process is the lifeline for any business intelligence solution. ETL design and development work consumes 60 to 80 percent of the resources of an entire BI project (Variar, 2004). ETL tools are used in designing and populating analytic databases such as data warehouses and data marts. Specifically, these tools provide for the periodic extraction, transformation, and integration of data from any number of heterogeneous data sources into time-variant databases used predominantly for business intelligence (Sweiger, 2002).

1.3 Role of ETL tools

ETL programs periodically extract data from source systems, transform the data into a common format, and then load the data into the target data store. ETL tools also move or transport data between sources and targets, document how data elements change as they move between source and target (i.e., metadata), exchange this metadata with other applications as needed, and administer all runtime processes and operations (e.g., scheduling, error management, audit logs, and statistics) (Eckerson, 2003).

ETL tools serve two very specific purposes. First, they provide a development environment that is easy to manage and embedded into the graphical interface. Second, ETL tools provide increased throughput, which increases the productivity of the user. The increased throughput is achieved by separating data management from data access. Data management is the coordination of data from source to target systems, while data access is the delivery of information from the database to the end user (Krudop, 2005).

1.4 ETL Market

The ETL market comprises vendors that offer tools used in designing and populating analytic databases. In the early days of ETL tools, the idea of being able to "move anything from anywhere to anywhere else" was nothing more than a pipe dream. Over the years, vendors came much closer to being able to provide real business value by accomplishing tasks such as recognizing metadata, examining complex source systems, moving very large datasets quickly, and providing a graphical development environment.

Market leaders have stable, mature products with the ability to connect to a broad array of data types, robust developer environments, and job parallelization capabilities (META Group, 2004). They also have strong financial stability that enables them to innovate and acquire ancillary capabilities, such as data quality, throughout a slow economy.

The ETL adoption rate will expand 10%-20% annually during the next three to five years (META Group, 2004). The estimated sales of ETL products doubled from 101 million in 1998 to 210 million in 2002 (Jarke et al., 2003). In 2004, the ETL market was expected to experience high growth rate due to an aggressive uptake in customer data integration efforts and consolidation of departmental data warehouse efforts (META Group, 2004). In 2005, Forrester Research (2004) forecasts that the ETL market will exceed $1 billion.

ETL tools have become a necessity in today's market, and investors are not giving the selection process as much attention as they should. Before purchasing an ETL product, investors need to first identify the needs of the corporation, followed by essential selection criteria.

1.5 Project Scope: Engineering Trade Study for the Evaluation of ETL tools

This study's final product is a standardized test suite consisting of selection criteria, scenarios, quantitative measures, and procedures. By designing and validating the engineering trade study methodology, four specific goals are accomplished: (1) Development of selection criteria that are essential in the evaluation of an ETL tool, (2) development of scenarios that examine each criterion and are effective in testing the various characteristics of an ETL tool, and (3) development of quantitative measures useful for evaluating the various aspects of ETL usage.

This work has the potential to add real value ETL tool decision makers. The methodology developed is effective in evaluating ETL tools and will help companies select the "best" ETL tool that meets the needs of their corporation.

1.6 Document Overview

The remainder of the document presents the development of an engineering trade study approach for the evaluation of ETL tools with the following: Methodology, validation, and conclusions and future work.

2 METHODOLOGY

The following sections present the development of the engineering trade study for the evaluation of ETL tools. The sections are 1) figures of merit, 2) criteria, 3) test scenarios, 4) quantitative measures, 5) evaluation matrix, and 6) procedure.

2.1 Figures of Merit

The first step in the developing the engineering trade study is to develop figures of merit, which are measures of quality, in this case, the performance of an ETL tool. The six figures of merit are: cost, ease of use, flexibility, robustness, scalability, and speed.

The cost of the ETL tool is important since most organizations will have a price limit. Consumers should take into account the actual product cost as well as hidden support costs. Hidden costs included the price of required training, implementation, maintenance, and future software costs for upgrades and support from the vendor.

Ease of use is another factor. The complexity and the size of the learning curve should be taken into consideration. The product would be worthless to the organization if an ETL tool with many features and functionalities were difficult to learn and complicated to use.

The flexibility of the ETL tool entails how easily the product can be customized to specific uses. It also measures how complete the functionalities of the tool are. Robustness measured the dependability of the product and whether the tool could continue operate well under the constantly changing environments. A robust product should be able to handle failures such as network outages, sever crashes, or a lack of disk space.

The scalability assesses the product's ability to handle different project sizes as well as different sizes of input data. Scalability could also measure how well the tool could process the complex projects well. The last figure of merit, speed, measures how quickly the ETL tool performed. The speed of the tool could vary by the size and complexities of the projects.

Since ETL tools had numerous features and functionalities to evaluate, it would not be feasible to directly apply these figures of merit to the products. Therefore, the second step involved the development of essential criteria in which the figures of merit could be applied to evaluate the various aspects of ETL usage.

2.2 Criteria

Selection criteria are standards by which one judges the achievement of required system operational effectiveness, suitability characteristics, or resolution of technical or operational issues. Since ETL tools are so complex, the criteria are grouped into eight different categories. Each category includes specific attributes that should be considered in the evaluation of ETL tools. The detailed descriptions of the eight main categories is included in the following sections.

Each main criteria category contains a detailed list of attributes. Each attribute falls under one or more of the relevant figures of merit. For instance, product architecture will be assessed according to the ease of use, flexibility, robustness, scalability, and speed. Listed below are the details of the eight groups.

2.2.1 Product Architecture

Product architecture specifies how the product is implemented. Information about the product such as the installation process or the platforms that the product supported fall under this category. Another important feature of product architecture is the ability to perform multiple jobs concurrently. Listed below are the criteria included in product architecture:

(i) *Installation process.* The ease of the installing the ETL tool.

(ii) *Platform support.* The support for source and target operating systems and databases.

(iii) *Recovery logic.* If there is a problem that results in a system crash during the ETL processing one should be able to recover to the point it last completed successfully. Likewise, the ETL product itself should not require manual work to correct problems after a crash due to its handling of metadata, schedules or job states.

(iv) *Restart logic.* Similar to recovery logic is the ability to restart ETL processing if a processing step fails to execute properly. One should have the ability to restart processing at the step where it failed as well as the ability to restart the entire ETL session.

(v) *Intermediate storage.* The product should allow for the use of intermediate storage between processing steps. Typically, this is in a relational database, but the product should allow staging and interim processing in both databases and files. A key item here is the support for simultaneous activities regardless of location. For example, one might have data coming from a relational source, to be combined with data from a flat file. You should not be forced to bring all data into a single format prior to working with it.

(vi) *Parallel support.* One component of parallel execution is that the product should be capable of executing multiple different jobs or modules concurrently. The second component is the ability of the product to support multi-threaded operation and parallel code execution internally, so that a given job can take advantage of the inherent parallelism of the platform it is running on. The third component is load balancing, if applicable. The product should be able to distribute load across multiple servers if it operates under a centralized control mode.

(vii) *Documentation.* The documentation of the ETL tool should be complete, containing information about all the essential technical aspects of the tool. The online help should be provided for both developer and administrator tools, and should be available in an online format that allows access from within the ETL environment as well as separately.

2.2.2 Data Support

Data Support focused on the different types of data that the ETL tool can support.

(i) *Data formats support.* The support for text, comma delimited files, excel, XML, HTML, etc.

(ii) *Data type support.* The support for integers, decimals, dates, times, string, user-defined types, etc.

(iii) *Real-time data.* The support for data that is time-dependent and information on current conditions.

2.2.3 Data Extraction

Data extraction is one of the three main functionalities of the ETL tools. A main consideration to assess is the product's ability to extract from a variety of various data sources.

2.2.4 Data Transformation

Data transformation is one of the more important functionalities of an ETL tool. This criterion includes being able to define the rules of transformation as well as the number of pre-built functions the product has. Examples of pre-built functions included changing the letter case or changing the time format from a 12 hour format to a 24 hour format. Data cleansing is another important factor that should be considered under this category due to the importance of high data quality.

(i) *Pre-built rules and transformations.* There are frequently encountered tasks that vendors can provide pre-built functionality to support. This can be a simple thing such as changing a letter case to more complicated tasks such as sorting a large file containing names and addresses.

(ii) *Rule-based transformations.* One should be able to specify that transformations be executed only when certain conditions are met, and be able to merge these rules together in many different ways. It is an advantage to have sample rule-based transformations in the environment and to be able to check how to implement them in the tool.

(iii) *Support for all basic mathematical and statistical functions.*

(iv) *Basic data cleansing functionality should be available.*

(v) *Recursive processing support.* One should be able to build new rules and transformations and save them for later use.

(vi) *Code/scripting support.* The support for code or scripts written in programming languages such as C++ and Java.

2.2.5 Data Loading

Loading, along with transformation and extraction, is one of the three main functionalities of the ETL tools. Data loading assesses the product's ability to load data to a variety of different targets and databases.

2.2.6 Matching

The ability to match and merge information between multiple data sets is an important capability for ETL tools to have and thus should be evaluated.

2.2.7 Metadata Management

Metadata is a component of data which describes the data. Essentially, it is the "data about data" which describes the content, quality, condition, and other characteristics of data. Metadata is crucial to building a data warehouse and therefore ETL tools should have functionalities to help the users manage metadata.

(i) *Extensibility.* The metadata repository should be easily extensible so that items can be added that the product did not take into consideration.

(ii) *Open storage format.* The metadata should be stored in an open, documented and easily accessible format. The obvious choice is in a relational database, preferably of the client's choice and not the vendor's.

(iii) *Metadata sharing.* Every ETL tool should have the ability to share metadata with third party applications. Typically, one should look for support of common metadata interchange standards. This sharing includes integration of business intelligence products.

(iv) *Content reporting.* Any ETL tool with a metadata repository should be able to report on the contents. It is important to see things like what systems data is coming from, where it is being used, what data is ignored, etc.

2.2.8 Development Environment

The development environment is where most of the users will spend their time developing the projects. Features to consider in the development environment include the interface of the product, the error reporting capabilities, and support for add-in tools and software upgrades.

(i) *Graphical User Interface (GUI) support.* The support for a graphical environment for both novice and expert users, particularly when dealing with scheduling and job dependencies.

(ii) *Command line support.* The support for invoking jobs or entire sessions is particularly useful. This allows for the creation of much more flexible ETL processes, such that every job can be executed without going back to the GUI.

(iii) *Integrated toolset.* With an integrated toolset, developers do not need to switch from one interface to another, as might happen when developing a transformation and then having to make changes to the metadata.

(iv) *Sequential processing.* In the context of information storage, sequential processing means that each item must be read one at a time, going through all the preceding records to get to the next record in sequential order. In an SQL environment, sequential processing is executed using a cursor. The support for cursors in relational databases is important for sequential processing.

(v) *Debugging support.* The ETL developer is able to see data both before and after transformations.

(vi) *ETL reporting.* Every product should provide reports on ETL sessions and jobs so one can see how the system is functioning.

(vii) *Centralized administration*. It is critical that there be a single place to view the entire ETL process, even if jobs are running across both the source and target platforms under different operating systems. Updating job schedules and managing jobs should be location-independent.

(viii) *Scheduling*. The scheduling of tasks within an ETL session is often not a trivial task. There are many dependencies between jobs, making it hard to determine what tasks can continue to run when an earlier job fails. It would be helpful to have built-in support for task scheduling.

2.3 Test Scenarios

After establishing a list of criteria, the next step in the development is to design test scenarios for each criterion. Different types of data must also be acquired to use in these scenarios. Ideally, the data used in these scenarios should be similar to the size and format of the data that is intended for final ETL processing. These data consist of text files, comma-delimited files, and XML.

When designing the test scenarios, we had two requirements: (1) They have to be effective in testing each criterion, and (2) they have to be scalable. Table 1 below shows an example of test scenarios designed for evaluating Data Extraction.

Table 1: Data Extraction Scenarios

Data Extraction	
Basic Processes	Using CD data, filter data by searching for information on any county where ELEM = SNOW. Continue with query by adding more restrictions such as querying for where ELEM = SNOW or PRCP, YEARMO <1980 but not 1975. Extract only this filtered data. Once data is extracted, sort the data by descending year. Save this query and run it on a different larger file of CD data. View the query and attempt to paste results into excel. Monitor display and edit extraction commands.

If a company were to use different data sets than those used in this study, the scenarios would need to be slightly edited to fit the different data. However, the basic processes performed in each scenarios should be comparable to those performed in this study.

2.4 Quantitative Measures

After designing the test scenarios, the next important step is to develop quantitative measures to assess the criteria. The following sections document the development of a scale and applying weighting factors.

2.4.1 Scale

In order to convert the qualitative figures of merit and criteria into quantitative values, a scale is created in which the products were scaled on a one to five scale (Table 2).

Table 2: Quantitative Scale

Numerical Value	Definition
1	Does not meet expectations
2	Slightly below expectations
3	Meets expectations
4	Slightly above expectations
5	Exceeds expectations

Since every consumer/organization has different needs and expectations of the tool, this scale would allow for a wide range of consumers to use this methodology to evaluate the products. The scale is applied to the six figures of merit and eight criteria in the evaluation matrix. For instance, the flexibility of the product architecture is ranked from one to five according to how well the expectations for flexibility were met.

2.4.2 Weights

Numerical weights are assigned to the selection criteria and figures of merit to quantify their relative importance. Numerical weights are given to reduce the effects of evaluator bias on the analysis; this allows the evaluator to obtain an objective assessment of the alternatives. Therefore, the weights should be assigned before the scenarios are performed. In addition, numerical weighting facilitates comparison among criteria that are not related. For ease in comparing diverse criteria, the sum of all these criteria weighting factors equals 1.00. After scoring, the weights are applied using the following equation:

$$\sum_i \left(b_i \cdot \sum_j a_j \cdot X_j \right) \qquad \text{Equation (1)}$$

Where,

a is weight of figure of merit
b is weight of criterion
X is figure of merit
i is the numbers of figures of merits
j is the numbers of criteria within each figure of merit

2.5 Evaluation Matrix

An evaluation matrix is created consisting of the figures of merit as column headings and the criteria as row headings. The evaluation matrix is used to quantitatively score the product according to how well it performed relative to the figures of merit. Table 3 below depicts how the criteria, figures of merit, and scores relate to one another in the evaluation matrix.

Table 3: Evaluation Matrix

	Ease of Use	*Flexibility*	*Robustness*	*Scalability*	*Speed*
PRODUCT ARCHITECTURE					
Installation process		▨	▨	▨	▨
Platform Support	▨		▨	▨	▨
Recovery logic		▨		▨	▨

2.6 Procedure

The procedure of this engineering trade study consists of five steps: (1) The developer will assign the weights. (2) The developer will perform the scenarios for a specific criterion. (3) The developer will assign scores for the relevant figures of merit. (4) The developer will repeat steps two and three for all the criteria. (5) The developer will apply the weights as shown in §2.4.2 and come up with a single value.

3 EXAMPLE

The evaluation methodology was validated against the ETL tools of some prominent ETL vendors. The following section describes an example of how a fictitious company called *Data Corp* could proceed with this evaluation.

3.1 Assigning Weights

Data Corp assigns weights to the figures of merit based on their experience with ETL tools and what is important to their needs. Table 4 shows these assigned weights.

Data Corp felt that they wanted an easy to use, flexible, robust tool that could easily be scaled up for large projects. They were not as concerned about the price or speed of the tool, therefore lower weights were assigned to those figures of merit.

Table 4: Weights for Figures of Merit

Figure of Merit	*Weight*
Cost	0.10
Ease of Use	0.20
Flexibility	0.20
Robustness	0.20
Scalability	0.20
Speed	0.10

Table 5: Weights for Criteria

	Weight
Ease of Use	
Product Architecture	0.10
Data Extraction	0.15
Data Transformation	0.15
Data Loading	0.15
Matching	0.05
Metadata Management	0.15
Development Environment	0.25
Flexibility	
Product Architecture	0.05
Data Support	0.15
Data Extraction	0.15
Data Transformation	0.15
Data Loading	0.15
Matching	0.05
Metadata Management	0.20
Development Environment	0.10
Speed	
Product Architecture	0.05
Data Extraction	0.30
Data Transformation	0.30
Data Loading	0.30
Matching	0.05
Robustness	
Product Architecture	0.20
Data Support	0.05
Data Extraction	0.20
Data Transformation	0.20
Data Loading	0.15
Matching	0.05
Metadata Management	0.10
Development Environment	0.05
Scalability	**Weight**
Product Architecture	0.10
Data Extraction	0.20
Data Transformation	0.20
Data Loading	0.20
Matching	0.05
Metadata Management	0.20
Development Environment	0.05

For the criteria, *Data Corp* assigned weights to 'each criterion category with respect to all of the figures of merit except cost. Again, these weights were assigned according to what *Data Corp* felt was the level of importance of the criteria in each of the areas. Table 5 shows the assigned weights for each criterion category that *Data Corp* decided upon.

The basis for assigning these weights should be an examination of each of the criteria in regards to each figure of merit. Questions like "Would our company rather have a robust user interface, an easy to use one, or both?" and "Would our company rather have flexible data loading or data support?" must be answered. Companies such as *Data* Corp must determine what areas of the criteria are more critical to their needs within each figure of merit. This means that under some figures of merit, certain criterion will me more important than others. A few examples of *Data Corp's* weighting assignment explanations are included below:

(i) *Data Corp* highly valued an easy to use development environment because the bulk of their work would reside in this area and they wanted to train multiple people on these tools quickly.

(ii) Matching, an aspect of ETL tools that *Data Corp* did not plan on utilizing often, received small weights in all five categories.

(iii) *Data Corp* wanted the ability to create and manipulate metadata extensively, therefore this area received a large weight under flexibility. Data support also received a high weighting under flexibility because *Data Corp* manipulates a large variety of data formats on a regular basis.

(iv) For the most part, *Data Corp* felt the extraction, transformation, and loading steps of their work were equally important across the board. Therefore, these three areas have the same high weightings in each of the categories. One exception to this however is under robustness, where *Data Corp* decided that since many of the loading jobs they would perform would be very straightforward and simple, loading's weight could be less than that of extract and transform.

(v) When looking at performing large jobs, *Data Corp* decided to give high weights to each part of the ETL process, as well as metadata management. In order for a job to be scaled effectively for their purposes, *Data Corp* needed each of these areas to perform equally well.

3.2 Test Scenarios

Data Corp performed the majority of the scenarios successfully with little difficulty. After completing each scenario, they wrote comments about how the ETL product performed. Table 6 shows an example of comments made after performing the scenario for rule-based transformations.

Table 6: Comments made on Rule-based Transformations

Criteria	Comments
Rule-based transformations	Completion of the scenario was successful. The process was relatively intuitive. Easy to define new columns and create different rules with expression builder.

Data Corp then scored each criteria based off their experience from performing the scenarios. Finally, *Data Corp* applied the weights using Equation 1 found in §2.4.2 to generate final scores on the tools they evaluated. These scores indicated the tools' performances according to *Data Corp's* specifications and allowed them to determine which tool was better for their purpose.

After testing the scenarios much like the *Data Corp* example above, the team revised the scenarios as needed. The figures of merits and criteria were also updated so that they could better serve the purpose of ETL evaluation. The final version of the evaluation matrix includes eight criteria (product architecture, data support, data extraction, data transformation, data loading, matching, metadata management, development environment), and six figures of merits (cost, ease of use, flexibility, robustness, scalability, speed). The final results were the ones included in section 2 of this paper.

4 CONCLUSIONS AND FUTURE WORK

In this paper, a useful engineering trade study for ETL tool selection was developed. In the end, all three initial objectives were achieved. Comprehensive ETL criteria were identified, testing procedures were developed, and this work was applied to commercial ETL tools. The study covered all major aspects of ETL usage and can be used to effectively compare and evaluate various ETL tools. This study will be an asset to any companies and organizations wishing to evaluate ETL tools in the future. This study produced replicable evaluation methods that can be used and modified by companies to address their own ETL tool needs. Using this study, companies will have the ability to make informed decisions about the best ETL tool for their purposes and will thereby reducing the risks of investing into an ETL venture.

This study could benefit from several improvements and future studies. For instance, the generated scenarios could be more comprehensive and complex. This study's scenarios could be used as building blocks to create more complex testing scenarios to better replicate ETL usage. Companies themselves could develop specific scenarios that might be more representative of their particular ETL needs. In the future, this study's testing procedures should be implemented on a company in need of an ETL product. Conducting a case study using the methodologies outlined in this paper would help to further validate this study. Finally, the list of criteria and scenarios generated in this study will undoubtedly need updating in the future as ETL processes continue to become more and more evolved as time progresses.

REFERENCES

Eckerson, Wayne (2003, May). The Evolution of ETL. What Works (Vol. 15). Retrieved March 3, 2005 from the TDWI online database: http://www.tdwi.org/research/display.aspx?ID=6716

Eckerson, Wayne (2003, November). Understanding Business Intelligence. What Works (Vol. 16). Retrieved March 3, 2005 from the TDWI online database: http://www.tdwi.org/research/display.aspx?ID=6838

Krudop, M. (2005, March). Maximizing Your ETL Tool Investment. Retrieved on March 15, 2005 from DM Review online magazine: http://www.dmreview.com/article_sub.cfm?articleId=1021507

MetaGroup Inc (2004). Metaspectrum Market Summary. Stamford, CT:Author.

Russom, P., Orlov, L, Teubner, C. (2004, June). The ETL tool market is back and growing. Forrester Research.

Sweiger, M., Madsen, M. R., Langston, J., & Lombard, H. (2002). Clickstream Data Warehousing. New York: John Wiley & Sons.

Variar, G (2004, August). Performing ETL with SQL. Retrieved on February 20, 2005 from the World Wide Web: http://www.bizintelligencepipeline.com/showArticle.jhtml?articleID=29105983

AUTHOR BIOGRAPHIES

SCOTT HENRY is a fourth year Systems and Information Engineering (SIE) student at the University of Virginia. He will be attending law school in the fall 2005. He can be reached at sch8x@virginia.edu.

SHERLYNN HOON is a fourth year SIE student at the University of Virginia. She will be working at Deloitte Consulting as a System Analyst starting in July 2005. She can be reached at sher@virginia.edu.

MEEKY HWANG is a fourth year Systems and Information Engineering (SIE) student at the University of Virginia. She will be working as an analyst at Accenture and can be reached at mkh7b@virginia.edu.

DIANE LEE is a fourth year Systems and Information Engineering (SIE) student at the University of Virginia. She will be working at Boeing Company in Seattle. She can be reached at dl9p@virginia.edu.

MICHAEL D. DEVORE is an Assistant Professor of Systems Engineering at the University of Virginia. He received his B.S., and M.S., degrees from the University of-Missouri-Columbia, and D.Sc. degree from Washington University in St. Louis. He has held a visiting assistant professor in the electrical engineering department at Washington University in St. Louis and prior to that managed system integrity and operations at Amdocs, Inc. He is a member of IEEE and SPIE. He can be reached at mdevore@virginia.edu.