Proceedings of the International Computer Science and Technology Conference 2008

Edited by
John Bugado, Mohammad Amin, Pradip Peter Dey,
Chuck Brown, and Arun Datta
Preface

Welcome to the International Computer Science and Technology Conference 2008 (ICSTC). This conference is intended to bring together researchers and practitioners from industry, academia and government to advance the state of the art in computer science, software engineering and technology and to encourage wider collaboration between academics and industry. The Conference is held for the computing community and hosted by National University. The large number of submitted papers is a clear indication of enthusiastic cooperation and response from the community. Out of the 183 submissions, 55 papers were initially accepted based on recommendations from our reviewers. However, many papers were subsequently withdrawn by mostly international authors due to some serious reasons including, visa and international situations. The program committee made their best efforts to accommodate all submissions with academic merit and scholarship.

National and internationally recognized speakers will present their research contributions in the field. In addition, a number of panelists will attend for the following six panel sessions: 1) Multi-model Multi-strategy Teaching/Learning in Science, Engineering and Technology, 2) Database Systems for Health Informatics, 3) Regulatory Compliance, 4) Strategic Management, 5) Green Computing, and 6) Physical Data Protection Panel. There will be two special presentations: 1) Virtualization by Mr. Darel Ison, Director of Technical Services at GTC Systems and 2) Business Process Management by Mr. Neal Fischer is the Founder of Hershey Technologies.

Finally, we would like to acknowledge the support and cooperation of all the authors and reviewers of ICSTC-08. We are thankful to our distinguished keynotes speakers: Dr. Dana Gibson, Dr. Peter Arzberger, Chair of the Pacific Rim Application and Grid Middleware Assembly, Dr. Subhas C. Misra, Director of Web World Network, Mr. Norbert J. Kubilus, COO of DataLEAD Communications and the Network Services Division of National Data Corporation, and Todd Walter, CTO, Teradata Research & Development. Our special thanks to Dr. J. Lee, Chancellor of National University System, Dr. Tom Green, VPAA of National University and Dr. Howard Evans, Dean of School of Engineering and Technology for their valuable suggestions and supports.

Conference Chair: John Bugado
Program Chairs: Chuck Brown, Pradip Peter Dey, Mohammad Amin, and Arun Datta
Event Management Chair: Albert Cruz

International Computer Science and Technology Conference (WWW.ICSTC.ORG)
San Diego, California, USA
April 1, 2008
<table>
<thead>
<tr>
<th></th>
<th>Table of Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Structuring Meta-search Research by Design Patterns</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Jürgen Dorn and Tabbasum Naz</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>APRIORI_PROCREATION- TOOL for Optimizing WEB PERFORMANCE</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Aruna Jain</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Formal Privacy Ontology, a Definition</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Waël A. Hassan</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>A Method for Optimizing the Size of Wireless Mobile Messages</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>P P Abdul Haleem and M P Sebastian</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Test Case Generation for White-Box Unit Testing</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>Samira Hussain and Salam N. Salloum</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Integrating Generative and Aspect-Oriented Technologies for Framework</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>Instantiation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R. P. Mahapatra and Dr. Jayaram Pradhan</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Semi-Automatic Generation of Transformation Rules in Model Driven Engineering:</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>The Challenge and First Steps</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Slimane Hammoudi and Samyr Vale</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Use of Simulation in Systems Engineering</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>Hassan Badkoobehi</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Content Delivery Network: A Dominant Technology for Internet</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>Tulika Srivastava and Mudita Srivastava</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>Secure Aroma Transfer</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>Vibhore Kumar Jain, Subha Joshi and Chaman Kumar Bansal</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>RESOURCE DESCRIPTION FRAMEWORK : A SMALL CASE STUY ON EDUCATIONAL DOMAIN</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>V. Kiran Kumar</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>Cooperative Intrusion Detection for Detecting Novel Attacks Using Real Time Data</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>Mining Approach</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R. Pushparaj</td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>Data Privacy and Technical Solutions, Iqbal Khan,</td>
<td>120</td>
</tr>
<tr>
<td>14.</td>
<td>Application of Cognitive Styles and Inference to Project Management,</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>David J. Sudbeck</td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>Early Muslim Scholars and Binary Mathematics</td>
<td>128</td>
</tr>
</tbody>
</table>
16. Acceptance Test Driven Story Card Development for XP
   (Agile Software Development)  
   Chetankumar Patel and Muthu Ramachandran  

17. Analysis of Near-infrared Phase Effect on Biometric Iris Data Demonstrates Viability of Iris-scan Biometric Device for Identity Authentication in Public Venues  
   Brady R. Stevenson and Gordon W. Romney  

18. Challenges in Vendor Collaboration in A Mix Mode Offshore Centre - Coexistence of Captive And Vendor Offshore  
   Satyen Vats and Rathi Dasgupta  

19. COMPARATIVE STUDY AND ANALYSIS OF FLANK WEAR OF SINGLE POINT COATED AND UNCOATED TURNING TOOL INSERTS USING ACOUSTIC EMISSION SIGNAL PROCESSING AND NEURAL NETWORK TECHNIQUES  
   N.PRASANNA, R. SRINIDHI, M.SUKUMAR, and C.S.VENKATESHA  

20. Multi-model Multi-strategy Teaching/Learning in Science, Engineering and Technology  
   Pradip Peter Dey, Mohammad Amin, Michelle Bright, Arun Datta, Shakil Akhtar, Albert Cruz, Mudasser F. Wyne, Patrick Kennedy, Patrick Olson, Rell Snyder, Alireza Farahani, Samuel Afuwape, Hassan Badkoobehi and Amber Lo  

21. Exploration Strategy Based on Terrain Coverage for Multiple Robots Using Genetic Algorithm  
   K.S. Senthilkumar and K. K. Bharadwaj  

22. Capitalising Software-Engineering Tacit Knowledge  
   Abdulmajid Hissen Mohamed  

23. On Simplification of Grid Scheduling and Resource Discovery Problem - A Survey and Proposal  
   Waqas Ahmed, Shaleeza Sohail and Aasia Khanum  

24. Joint Modeling and Optimization of Business Processes and Information Technologies  
   Igor Safonov and Vadim Safonov  

25. Prediction Based Dynamic Load Balancing Techniques in Heterogeneous Clusters  
   Pushpendra Kumar Chandra and Bibhudatta Sahoo  

26. QUERY PROCESSING IN MAIN MEMORY OODB SYSTEMS  

27. Rapid Prototyping Data Integration
   Barbara Lauridsen 204

28. Relational Database Loader for Implementing Advanced Queries in
    Protein Informatics
   Sharen A. Eskelsen, Scott S. Kelman, and Carl S. Ewig 210

29. Categorization of Functional Dependencies for a Minimal Cover
    Ajeet A. Chikkamanur and Shivanand M. Handigund 214

30. CHOMS: A Portal for Obesity Management
    Radha Nandkumar, Arun Datta, Jacqueline Caesar, Weimo Zhu,
    Peter Bajcsy, Jeff Sunderlin, Victoria Jackson, Jill Sproat, and Heidi Krahling 219

31. Why is C4I Software Hard to Develop?
    Lee Whitt 220

32. Waveform Distortion Estimation during Multiple Power Quality Events
    Amritpal Singh and Manmohan Singh 238

33. How to secure e-mail and instant messages
    Bari Siddique, Rozina Siddique, and Mohammad Amin 245

34. Database Panel Discussions: Database Systems for Health Informatics
    Moderator: Jacqueline Caesar 249

35. Integrating Multiple Data Sources Into One Analytical Environment
    Imad Birouty 250

36. Adaptive Neuro-Fuzzy Inference System based Inverse Kinematics Solution
    of 3DOF Planar Robot
    Srinivasan Alavandar and M.J.Nigam 251

37. Intelligent Cash Management System for an ATM network
    Rimvydas Simutis, Darius Dilijonas, and Lidija Bastina 256

38. RIN Evaluation with dispersion compensation for Optimum Performance
    of Optical Transmission System
    Kamaljit Singh Bhatia and Dr.Manoj Kumar 266

39. Simulation Study and Implementation of Single Piece Lean Line Design
    for Cycle Time Reduction - A Practical Approach
    D.Arunkumar, V.Ramesh, K.V.Sreenivasa Prasad, and T.R.Srinivas 271
40. Performance Analysis of AODV and DSR in MANET-A Simulative Study
   Gaurav Sethi and Manoj Kumar

41. Development of a Trusted Third Party Time-Date Stamp Server in an
   IT Security Course
   Gordon Romney

42. Simulative Analysis of operating conditions for Flat amplitude Multiwavelength
   Brillouin-Raman comb fiber laser
   Kamaljit Singh Bhatia and Manoj Kumar

43. GridARTCA: A collaborative technology environment utilizing
   Cyberinfrastructure - A preliminary report
   Kevin Franklin, Santiago Nunez-Corrales, Jose Castro, Peter Bajcsy,
   Alvaro de la Ossa, Thomas MacCalla, and Arun Datta

44. Serialized Tokens as a Means of Information Access Control
   Miles Romney IV, Todd H. Romney and Gordon W. Romney

45. There Exists a Critical Need for Role-based IDs in a University IT
   Infrastructure in Order to Meet Security Best Practices
   Gordon W. Romney

46. The Globally Distributed Ethernet Switch
   Barry Brueske

47. Modern methods to solve Industry Problems A Lean Management
   Case Study carried out in a Processing Industry
   N.Prasanna, V.Ramesh and D.Arunkumar

48. Using Enterprise Architecture to Gain Strategic Alignment for Project
   Management in Information Technology
   Jack Elson, Jeanne Stacy, Brian Turner, and Russell F. Zimmermann
Inaugural Speaker
Dr. Dana Gibson, President of National University
(Tuesday, April 01, 8:30-10:00 am)

In July 2007, Dana Gibson became National University’s third president. Dr. Gibson joined NU from Southern Methodist University, where she had served as Vice President for Business and Finance. Dr. Gibson also served as Vice Chancellor for Administration and Finance at the University of Colorado at Denver and University of Colorado at Denver Health Sciences Center. Her experience includes service as Vice President of Finance and Administration and Chief Financial Officer of the YMCA of Denver and as Vice President for Academic and Information Services at Texas Women’s University. As a tenured Associate Professor of Accounting and Information System at TWU, Dr. Gibson also served as Speaker Pro Tem of the Faculty Senate.

Dr. Gibson earned both a Bachelor of Science degree in Business-Accounting and an MBA from TWU. She received a Ph.D. from the University of Texas at Arlington, with a major field of business accounting and minor fields of information systems and research methods. She has been a Certified Public Accountant since November 1984.

Conference Keynote Speaker
Mr. Mark S. Johnson, VP Applications Product Marketing, Oracle
"Sustaining Profitable Growth: The Phases of ERP in a Growing Company"
(Tuesday, April 1, 8:45 – 9:25am)

Mr. Johnson will discuss the challenges facing growing businesses - the (often) urgent technology needs that arise when a company experiences exponential growth. The situation demands a top tier technology platform while faced with budgetary constraints. Fast growing companies need to establish consistent, automated business processes across global operations quickly - in implementations that last for weeks not months. This requires, first, to get executive buy-in for the need to invest in technology. Whatever solution is deployed must be easy to maintain, adapt, and scale with low Total Cost of Ownership. Mr. Johnson will illustrate how Oracle technology meets the challenges of fast growing companies.
Peter Arzberger is Chair of the Pacific Rim Application and Grid Middleware Assembly (PRAGMA; www.pragma-grid.net), an open, institution-based organization of 30 institutions. PRAGMA, founded in 2002, has a mission to build sustained collaborations among researchers around the Pacific Rim by building applications on top of emerging Grid hardware and software. Connected with PRAGMA is PRIME, the Pacific Rim Undergraduate Experiences (prime.ucsd.edu) program, which provides international research and cultural internship experiences to undergraduate students. PRIME, founded in 2004, has admitted 36 students and sent students to four PRAGMA sites. Arzberger is a founding member of the Steering Committee another international activity, GLEON (http://www.gleon.org), the Global Lake Ecological Observatory Network. GLEON is a grassroots network of people, institutions, programs, and data linked by cyberinfrastructure and united by the mission to understand and predict the response of lake ecosystems to natural processes and human activities at regional, continental, and global scales. In addition, Arzberger is Director of the National Biomedical Computation Resources (http://nbcr.net), an NIH National Center for Research Resource award. NBCR’s mission is to develop computing and information technologies (e.g., end-to-end tools in cyberinfrastructure) to catalyze and facilitate biomedical research across a broad range of biological scales. He is also Chair of the National Advisory Board to the U.S. Long Term Ecological Research (LTER) network. Arzberger is the former Executive Director of the National Partnership for Advanced Computational Infrastructure (NPACI) and a former Program Officer at the National Science Foundation in Computational Biology.
Dr. Subhas C. Misra is currently a visiting scientist at State University of New York, Buffalo, USA, and is also a Director of Web World Network. He received his Ph.D. degree from Carleton University, in Ottawa, Canada, and M.S. and M.Tech. degrees respectively from the University of New Brunswick, in Fredericton, Canada, and the Indian Institute of Technology (IIT), at Kharagpur, India. Dr. Misra has several years of experience working in the academia, and the public and private sectors in research, teaching, consulting, project management, architecture, software design and product engineering roles. His current research interests include the areas of software management, software quality management, and information systems security, which are multidisciplinary, combining the fields of software engineering, information systems, organizational behavior, and technology and operations management. Dr. Misra has authored over 50 scholarly research papers and published (yet to appear) 4 books. He has won Best Research Paper Award in an international conference held in the United States. He was also the recipient of more than 15 academic awards and fellowships such as the Achievement Award at the 2007 World Congress held in Las Vegas in the United States for “contribution and dedication” to his field, and the Canadian Government’s NSERC Post Doctoral Fellowship. A mention about him and his work has also appeared in the June 8, 2007 issue of the Carleton Now newspaper. His biography has also been selected to appear in the Cambridge Blue Book, Cambridge, England, 2008. Dr. Misra is the Managing Editor of two international journals - the International Journal of Information and Coding Theory (IJICoT), U.K and the International Journal of Communication Networks and Distributed Systems (IJCNDS), U.K. He is a Guest Editor of a special issue of Computer Communications Journal (Elsevier Science). Dr. Misra is an Associate Editor of the Security and Communication Networks Journal (Wiley) and ICIC Express Letters (an international journal motivated to Innovative Computing, Information, and Control; published from Japan). He is an Editor of the International Journal of Systemics, Cybernatics, and Informatics.
A Computerworld Premier 100 Technology Leader, Norbert Kubilus has over 30 years of information systems, technology and operations experience in companies ranging from start-up to Global 1000, as well as with non-profits and government agencies. This includes having been CIO or CTO for Diamond Resorts International®, Stellcom Inc., The Leading Hotels of the World, BCM Inc., and Educational Testing Service, as well as COO of DataLEAD Communications and the Network Services Division of National Data Corporation.

He was also a Technology Leadership Partner at Tatum LLC, providing technology management leadership to SMB companies primarily in San Diego, and a Faculty Fellow & Professor of Computer Science at The College of New Jersey. Five years ago, Nicholas Carr penned his Harvard Business Review article "IT Doesn't Matter" in which he argued that information technology no longer gives businesses a competitive edge. Carr called IT managers impatient, wasteful, and passive, and he criticized the hype about the so-called strategic value of IT. A year later, he expanded his thesis into a book titled Does IT Matter? Carr’s primary argument is that IT has become a commodity, like electricity, that confers no competitive advantage upon its business users. Among business leaders who have adopted Carr’s hypothesis, the result has been a de-emphasis of the role of IT in their companies. Such is the case in companies where the IT function has moved in the corporate organization to Sales or Marketing … or back to Finance and Accounting. Gartner and IDC studies tend to show that IT does matter. Companies that invest wisely in IT continue to increase revenues much faster than those that invest unwisely, too little, or not at all. A company that invests poorly in IT doesn’t increase revenues as quickly as its competitors. One that invests unwisely soon goes out of business or is acquired. This presentation addresses what technology leaders need to do to help their organizations make these wise IT investments.
Database Track Keynote Speaker

Mr. Todd Walter, CTO, Teradata R&D
"Data Warehouses"
(Wednesday, April 2, 9:00-10:00am)

Mr. Walter joined Teradata (a division of former NCR) in 1987. He designed and implemented features of Teradata, managed engineering teams and researched advanced database topics. Todd has worked directly with many customers at the leading edge of adopting data warehouse technology, and guiding the technology to solve real business problems.

Conference Closing Speaker

Dr. Shakil Akhtar
Distance Learning Techniques to Teach Networking and Databases
(Thursday, April 3, 10:30-11:30am)

While modern distance learning techniques such as video/audio conferencing, electronic communication, course online systems and course management systems work well for most classes offered in distance learning format, special techniques are used to teach IT courses that involve higher level of interactivity. The main philosophy in distance learning is the flexibility of self-paced learning that is anytime, anywhere but challenging at the same time to keep the participants actively involved in the learning process. Many students taking such classes are working professionals who seek opportunities for career advancement through professional development course and degree programs. They do not have much time to explore the technologies that support the learning system. Therefore one of the main requirements of any distance learning technique is to provide a quality education with an outstanding learner support system. The challenge becomes higher when effort is made to teach IT classes in distance learning format. The distance learning technologies commonly used nowadays are video/audio conferencing, email, and electronic conferencing. In addition, content management systems such as webct and blackboard are used as a centralized resource for class contents, students’ assessments, grades and class messages. It is possible to have some level of interactivity using these current technologies, which is very much needed for IT classes. However, a better interactivity is possible by using animated simulation, flash animation and screen capture of an activity as applied to teaching courses in programming, databases, networking, security and web design within the field of IT. The proposed presentation would cover such technologies with examples.
Abstract

Design patterns shall support the reuse of a software architecture in different application domains as well as the flexible reuse of components. In this paper, we propose design patterns for meta-search engines. We also introduce design patterns for common components of meta-search engines e.g. query interface generator, information extraction, result merger and result ranker. Presented design patterns for meta-search engines and their components are reusable, extendable and flexible. These design patterns accelerate the development process in meta-search domain and other related domains. Moreover, it promises higher quality of developed solutions. These design patterns also provide developers with a shared vocabulary for easy communication.

Keywords: meta-search, specialized search engine, design patterns

1. Introduction

Internet is flooded with information and contains hundreds of Web sites with thousands of topics in which searchers get lost while searching a topic. Second most popular activity in the Internet after e-mail is “search” [1]. One Microsoft researcher says: “Estimates are that information workers spend as much as 30 percent of their time searching for information, at a cost of $18,000 each year per employee in lost productivity” [2]. Primary search tools i.e. search engines, subject directories, social network search engines are available for the searchers but these are not sufficient to meet the requirements of users and are unable to provide the desired results. They have limited coverage, cannot locate high quality information from the invisible Web stored in accessible databases because of technical limitations or because of exclusion from the indices and the returned search results consist of long documents. Searchers have to navigate through long documents to find the relevant information from these long documents. One of Microsoft reports says: “people search an average of 11 minutes before they find what they are looking for” [3]. Subject directories are organized on the Internet sites by subject and users choose a subject of interest from the list of subjects. Subject directories depend on human editors for listings. If the description of a site is not specific enough then search may be unsuccessful. Social network services allow people with shared interests, hobbies, or causes to come together online. Social network search engines (del.icio.us, digg, reddit) are a class of search engines that use social networks to organize, prioritize, or filter search results. Social bookmark sites, allow Internet users to share content they like best with others for searching and viewing. Users can assign tags to their favourite Web pages. But the problem with such type of tag-based systems is that there is no set of controlled vocabulary for tags and no standard for tag structures. There also
exist spelling errors and multiple meanings of tags. Some users are also misusing tags to make their Web sites more visible [4].

To overcome all these problems with traditional search tools, “meta-search engine” are proposed as an choice for specific topic search. Meta-search engines (MSE) also known as multi-threaded engines, do not necessarily maintain their own listings/databases, but send the user’s query simultaneously to other search engines, Web directories or to deep Web, collect the results, remove the duplicate links, merge and rank them according to their own algorithm in a single list and display it to the user. Meta-search engines provide fast and easy access to the desired search, because they can search from multiple search engines simultaneously and save the precious time of the searcher. Meta-search engines provide a broader overview of a topic as compared to traditional search engines and increase the coverage of Web by combining the coverage of multiple search engines. Querying multiple search engines is more scalable then the centralized general purpose search engine. Meta-search engines have the ability to search the invisible Web too thus increasing the precision, recall, and quality of results. They make the user task much easier by searching and ranking the results from multiple search engines [5].

A lot of research is in progress in the field of developing configurable meta-search engines in different domains i.e. jobs, hotels, flights, news, research papers and real estate etc. It has been observed that developing a configurable meta-search engine in any domain is a tedious and time consuming task. Every time developers have to start the development process from scratch. It is desirable to have a reusable and flexible design for meta-search engines. After a detailed study of meta-search engine development research, we identified different processes and components for meta-search engines that meet specific requirements. In this paper, we propose reusable and flexible design patterns for meta-search engines and their components so that they can be reused in several times after some modifications. The design patterns of meta-search engine i.e. result ranker can also be reused in other application domains.

The rest of the paper is organized as follows. Section 2 describes the related work in the development of meta-search engines, design patterns and frameworks for domain specific search engines from a design pattern perspective. The discussion of different solutions motivate also the rationale for the design pattern. Section 3 provides an overall architecture for meta-search engines. Section 4 contains design patterns for meta-search construction and usage as well as important common components of meta-search engines. Finally, sections 5 concludes our work.

2. Related Work

In object-oriented systems there exist recurring patterns of classes and communicating objects. These patterns provide simple and elegant solutions to specific design problems and make object-oriented designs more flexible. In [6], Gof (Gang of four) describe 23 of the most common patterns in detail. These design patterns can help developers to structure their own specific applications and give them a common vocabulary to describe design concepts, rather than particular implementations [6][7].
To our knowledge, [8] is the only research that describes the framework for domain specific search engines from design patterns perspective. [8] also present design patterns for some components of domain specific search engines.

The Web Database Metasearch Engine project is developing technologies for providing integrated access to Web databases. Important phases for interface integration are automatic schema matching, schema integration and data integration. [9] use meta-information from Web search interfaces and present a two-step clustering based approach i.e. positive match based clustering and predictive match based clustering for schema matching.

MetaQuerier project (http://metaquerier.cs.uiuc.edu/) explores and integrates the query databases that are not visible to the traditional crawlers. Main components of MetaQuerier are MetaExplorer and MetaIntegrator. MetaExplorer discover sources on the deep Web and builds a search engine of Web databases. Moreover, MetaExplorer also creates models to represent discovered databases and develop wrappers to automatically extract schema details. MetaQuerier applies a holistic schema matching approach for schema matching. Holistic schema matching approach is used to identify simple 1:1 matching and complex 1:n or m:n matchings [10] [11].

In our previous research [12][13], we integrate job portals by meta-search and use a domain ontology for schema matching, schema integration, and data integration. The domain ontology is also used for information extraction from the result pages returned by various search engines. In [14] we have applied meta-search in the domain of accommodation search. Here the extraction from Web sources is based on Web service interfaces, requiring not all components that are required in the job domain. In a further project we address the search for orders in the logistics domain.

[15] introduce some techniques to automatically extract search result records (SRR) from dynamically generated HTML result pages. They present a tool ViNTs (Visual Information aNd Tag Structure) that utilizes visual content features and HTML tag structure of HTML result pages for the automatic wrapper generation of any given search engine. [16] presents techniques for supervised wrapper generation and automated Web information extraction. They implemented these techniques in a system called Lixto. Lixto provides a visual, interactive and convenient user interface for the creation of semi-automatic wrapper programs. The Lixto wrapper generator consists of modules i.e. navigator, extractor and visual developer. Lixto wrapper generator translates required piece of information from HTML pages into XML. [17] proposes an algorithm MDR (Mining Data Records in Web Pages) to mine data records in a Web page automatically. Authors claim that their data mining technique is able to mine both contiguous and non-contiguous data records.

[18] investigate result merging algorithms for meta-search engines, to merge results from different search engines into a single ranked list and state that merging based on titles and snippets of retrieved results can outperform other approaches based on full document analysis. They present five algorithms i.e. TopD, TopSRR, SRRSim, SRRRank and SRRSimMF for merging results into a single ranked list. Search result records (SRR) contain URL, title, and summary (snippet) of the extracted document. TopD algorithm uses the top document while TopSRR algorithm use top SRRs to compute the search engine score. The SRRSim algorithm
computes similarities between SRRs and query. SRRRank algorithm rank SRRs using more features like location of the occurred query term or total number of occurrences of the query term in the title and snippet etc. SRRSimMF algorithm computes similarities between SRRs and query using more features.

3. Meta-Search Architecture

This section describes the overall design of a meta-search engine. We identified that there are two processes in meta-search engines i) the meta-search engine creation process and ii) the meta-search engine usage process. Figure 1 and 2 shows the main components involved in the meta-search engine creation and usage process.

![Fig. 1: Meta-Search Creation Process Components](image)

Meta-search engine creation process work as follows. First of all, a developer specifies preferences like for which type, country or geographical area a meta-search engine is required by a preferences collector component. After getting preferences the search engine selector component will be activated and search engines meeting the preferences of the developer will be selected. Next, the Interface extractor component derives and analyse attributes from Web search interfaces. Then, an XML Schema generator component creates XML schemes for every search interface. Finally, a query interface generator component matches and integrates different XML-Schemes to have a single query interface for the meta-search engine.

When a user/seeker sends a query from the “meta-search query interface” to the meta-search engine, the components involved in meta-search engine usage process (Figure 2) will be activated and works as follows. Queries from the query interface are dispatched by a query dispatcher component and result pages with lists of results are collected. The information extractor component is responsible for extraction of records and the results from the result pages. Next all the identified results are merged together by a result merger component and stored in a
database or in XML format. Duplicate results are removed by a duplicate result eliminator component and finally results are ranked by a result ranker component.

**Fig. 2: Meta-Search Usage Process Components**

4. **Design Patterns for Meta-Search Engines**

This section contains the designed patterns for both processes and the main components of meta-search engines. Figure 3 and 4 describes our meta-search processes by using an abstract factory design pattern. “Abstract factory pattern provides an interface for creating families of related or dependent objects without specifying their concrete classes. It defines the interface that all concrete factories must implement, which consists of a set of methods for producing products” [6][7]. The meta-search abstract factory can produce any type of meta-search engine, as long as it gets proper set of directions, called factories. A factory design pattern will build a certain type of meta-search engine, depending upon the type of factory. It is clear from the abstract factory pattern in Figure 3 and 4, that families of job meta-search objects or hotel meta-search objects can be created from a developer or a user perspective. Each type of meta-search engine has the same overall structure that all meta-search engines share in common i.e. interface extractor, XML-schema generator, query interface generator, information extractor, result merger, duplicate result eliminator and result ranker etc. New factories for flights or real estates search etc. can be added easily. Using the meta-search abstract factory pattern, we do not need to worry about what kind of meta-search we are building. An abstract class may contain a default method i.e. a simple ranking algorithm, that can be used by every concrete meta-search engine, but will be refined if more specific ranking is required. Job-MSC-Factory in Figure 3 represents a Job-Meta-Search-Creation-Factory and Job-MSU-Factory in Figure 4 represents Job-Meta-Search-Usage-Factory. Few parts in Figure 3 and 4 are not drawn for the sake of saving space.
Fig. 3: Abstract Factory Pattern for Meta-Search Engines Creation Process

Fig. 4: Abstract Factory Pattern for Meta-Search Engines Usage Process
Below are some design patterns for common and important components of meta-search engines.

### 4.1. Query Interface-Generator

The query interface generator component of a meta-search engine is responsible for schema integration, data integration and then production of query interface for meta-search engine. Integration of interface schemes is divided into two parts i.e. schema matching and schema merging. During schema matching, semantic correspondence between interface attributes is identified and each schema is translated into a single schema for the query interface. During data integration, the values of different attributes for the user interface are determined. It is required that values are semantically unique and compatible with the local values. Different methods have been proposed for schema and data integration by using i) domain ontology [12][13], ii) clustering approach [9] or iii) holistic schema matching approach [10][11]. Different approaches for schema matching meet specific requirements according to the specific context. So developers should have a facility to choose one of the above mentioned approaches according to the specific requirement. It is required that new algorithms for schema and data integration comply with the same interface. These algorithms can easily be introduced with less effort and without changing the other code.

![Fig. 5: Strategy Pattern for Meta-Search Query Interface Generator](image)

To meet above requirements, the strategy design pattern as shown in Figure 5 is used for the query interface generator. “Strategy design pattern defines a family of algorithms, encapsulate each one, and makes them interchangeable. Strategy lets the algorithm vary independently from clients that use it” [7]. The design pattern in Figure 5 implements that the Meta-Search-Query-Interface-Generator is a class that is responsible for schema and data integration of different search engines and Schema-And-Data-Integrator is an interface. Integrating strategies is not implemented by the class Meta-Search-Query-Interface-Generator. Instead, they are implemented separately by sub-classes of abstract Schema-And-Data-Integrator class. Sub-classes of abstract Schema-And-Data-Integrator class implement different integrate strategies i.e. Integrate-By-Ontology, Integrate-By-Clustering and Integrate-By-Holistic-Schema-Matching. To switch schema and data integrator strategies, each meta-search engine calls the integrate method that it prefers. If a developer wants to add a new schema and data integration algorithm into the system, this can be done easily by implementing a new class using the Schema-And-Data-Integrator interface.
4.2. Information Extractor

The *information extractor* component is responsible for extraction of results from the result pages. It consists of a *Record collector* component and a *Result Collector* component. The *record collector* component is responsible for identification of the record section from the result page i.e. list of jobs, table with flights etc and *Result collector* component is responsible to extract the exact fields i.e. job salary, hotel price etc, from the identified record section.

Information extraction research shows that information extraction from different websites is often performed by using wrappers. Wrappers can be constructed manually, semi-automatically and automatically for record section identification. For identification of record section, different approaches i.e. i) automatic wrapper generation as ViNTs [15], ii) supervised wrapper generation as Lixto [16] or iii) data mining approaches [17] can be utilized. After identification of record section, a meta-search *result identifier* component is utilized for the extraction of exact results by using i) a domain ontology [12] [13] or ii) Lixto information extraction tool again [16].

For schema matching and merging it is required to normalize the terms like “Posted Date” into “Post Date” or “Job Types” to “Job Type”. Stemming algorithms i.e., Porter’s stemming algorithm can be utilized for term normalization process. A stemming algorithm is a method to convert word to their related form i.e. root, stem, or base. The stemming process is useful to find similar terms by only considering the word stem in search engines, natural language processing, and text processing [19].

The strategy design pattern is utilized for the information extraction component of the meta-search engine. The design pattern in Figure 6 shows that the Meta-Search-Information-Extraction class is responsible for information extraction from different result pages. Result-Collector and Record-Collector are interfaces. Sub-classes of Record-Collector abstract class implement different record identification strategies i.e. Record-Identifier-By-Automatic-Wrapper, Record-Identifier-By-Lixto, Record-Identifier-By-Data-Mining, Record-Identifier-By-Ontology.
Identifier-By-Lixto and Record-Identifier-By-Data-Mining. Sub-classes of Result-Collector abstract class implements two result identification strategies i.e. Result-Identifier-By-Ontology and Result-Identifier-By-Lixto.

4.3. Result Merger

Result extracted from different search engines need to be merged and then stored for future use. Results can be stored in a database or in XML format. We re-used result merger design pattern for result merger component from [8] with small changes.

![Abstract Factory Pattern for Meta-Search Result Merger](image)

The abstract factory pattern is used for the result-merger component of the meta-search engine (see Figure 7). “Meta-Search-Result-Merger” abstract factory defines the interface that all concrete factories i.e. Database-Factory and XML-Factory must implement, which consists of a set of methods for merging results. The concrete factories i.e. Database-Factory and XML-Factory implement the different product families i.e. Create, Query etc. To merge results, the client uses one of the factories and each factory knows how to create the right object for the right merging process. Few parts in Figure 7 are not drawn for the sake of saving space.

4.4. Result Ranker

A result ranker component ranks the results according to user preferences. Rank preferences can vary according to personal choices or meta-search engine type (i.e. job, hotel). For example, a seeker may want to rank the flight results according to the price, hotel according to the nearest location or job according to the query relevance. For merging results into a single ranked list according to the query relevance alone of the algorithm from TopD, TopSRR, SRRSim, SRRRank, SRRSimMF [18] can be used.
A strategy design pattern is used for the result-ranker component of the meta-search engine (see Figure 8). The Meta-Search-Result-Ranker class is responsible for ranking of results. Sub-classes of the Rank-Algorithm abstract class implement different ranking strategies i.e. Rank-By-Price, Rank-By-Location, and Rank-By-Relevance.

5. Conclusion

In this paper, we presented design patterns that can structure complex meta-search engines construction processes and provides us with flexible design strategy. We introduce design patterns for meta-search engine construction and usage as well as its components.

The difference between our and existing work is that we also introduced design patterns for i) a query interface generator component of meta-search engines that is required for user convenience and ii) information extraction component for information extraction from search engine’s result pages.

The reusable design patterns for meta-search components can be reused several times not only in meta-search domain but also in some different application domain after some modifications. Design patterns for meta-search engines are flexible enough and have facility to add or remove features for different components with minimum effort in future. These design patterns can speed up the development process for new developers as they do not need to rediscover the design problems and can save time. These design patterns for meta-search also enhance communication between project team members by providing them with shared vocabulary and provide a way to alter or extend some part of the system independently of all other parts.

The extend of reuse of software and software designs is difficult to evaluate since a reuse typically occurs years later after the design process. Moreover, the critical aspect is the reuse of software by developers not involved in the original design.
We have designed the design patterns with three applications in mind (job search, accommodation search, and search for transport orders). Moreover we have analyzed different algorithms described in the literature that can be used in meta-search.

References


[6] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995

[7] Eric Freeman, and Elisabeth Freeman, Head First Design Patterns, O’Reilly, 2004


APRIOI_PROCREATION- TOOL for Optimizing WEB PERFORMANCE

Aruna Jain
Department of Computer Science & Engineering
Birla Institute of Technology
Mesra, Ranchi

Abstract: Mining Association Rule is one of the common forms of Data Mining, in which Frequent Item sets are retrieved efficiently. In this paper, we review the APRIORI class of Data Mining algorithm for solving the Frequent Item set counting problem and develop a new algorithm, APRIORI_PROCREATION that reduces the redundancy of Candidate Item set generated directly from Frequent Item set by narrowing the Breadth First Search Space and hence reducing its execution time leading to enhancement in mining efficiency of classical APRIORI. To validate our claim, we test our algorithm against transaction database containing Web Logs of www.ddppl.in, which shows that our approach improves the performance of Web Server considerably in terms of Cache Hit Ratio and Bandwidth Ratio by reducing the Execution Time of the web request.

Introduction:

The expansion of the World-Wide-Web (WWW) has created an unlimited opportunity for users to disseminate and gather information online. As more and more data are becoming available on the servers, there is need to make searching comprehending, and retrieval of more sophisticated and dynamic semi-structured information’s(data) stored on the Web efficiently[1]

This necessitates study and analysis of the Web-user behavior to serve the users better and increase the value of an enterprise. One important aspect of this study is the Web-Log data which embodies much of web users’ browsing behavior. From these Web Logs, we can discover patterns that predict the users’ future requests and store them in anticipation of their future requirement thereby reducing response time and improving the server performance [2]. This technique of data mining is employed to discover new information in terms of patterns or rules from vast amounts of data by building predictive and descriptive models. A predictive model is used to explicitly predict values whereas
descriptive model describes patterns in existing data, which provide information such as identifying different customer segments. Web predictive model developed by [3] shows that data mining based caching enhances hit rate and weighted hit rate significantly.

An important class of data mining problem is mining Association Rules from Web Log data. Association rule is a data mining technique which discovers strong correlation among data to understand user access patterns and improve the website design (provide efficient access between highly correlated objects, better authoring design for pages, etc.) and enable better marketing decisions (better customer classification and behavior analysis, putting advertisements in proper places, etc.)[4,5,6]. Web Log analysis described in [7] is dedicated for improving Web Server Performance. The implemented algorithm is based on psychological human memory retrieval research which collects past access patterns and predicts further user action. It is observed that the Association Rule based Pre-fetching model has better predictive value than Site Structure Pre-fetching model and provides a good cache hit ratio without much addition to traffic load.

Association Rule Mining finds the set of all subsets of items/attributes that frequently occur in database records/transactions and extracts rules on how a subset of items influences the presence of another subset. Association Rule can be expressed as:

\[ A \implies B \ [S, C] \] where A and B are sets of items; S is the support of the rules, defined as the rate of the transactions containing all items in A and all items in B i.e. Support \( (A \implies B) = P (A \cup B) \) and C is the confidence of the rule, defined as the ratio of S with the rate of transactions containing A i.e. \( P (B / A) \).

Support and confidence are measures of the interestingness of the rule. A high level of support indicates that the rule is frequent enough for a business to be interested in it. A high level of confidence shows that the rule is true often to justify a decision based on it. Minimum support/confidence required for a rule to be reported is its threshold value.

One of the main attributes needed in an Association Rule-mining algorithm is Scalability, the ability to handle massive data stores. As a result fast and efficient Association rules are required to handle increasing number of transactions in real world databases. These rules are able to discover related items occurring together in the same
transaction. Since these transaction databases contain extremely large amount of data, current Association Rule discovery techniques try to prune the search space according to the support for the items under consideration.

In the application domain, items correspond to web resources, while transactions correspond to user sessions. Thereby, a rule such as \( \text{res}_1 \Rightarrow \text{res}_2 \); mean that if \( \text{res}_1 \) appears in a user session, \( \text{res}_2 \) is expected to appear in the same session, though possibly in reverse order and not consecutively. The basic algorithm called Apriori Algorithm for finding the association rules was proposed in 1993 [8] and later modified by [9,10] uses Breadth First Search, Bottom Up Approach and performs well when the Frequent Items are short and thus it is easy to implement when the support required is high as it leads to a smaller number of frequent items. But for larger number of frequent items it generates huge set of candidate items leading to high memory requirement and more searching time. Our work therefore focuses on developing an efficient algorithm of mining frequent item sets for association rules with Procreation count. We modify Apriori by using Procreation Count of Frequent Item sets at a level, which is related to Support Count of Candidate Item sets at a next level. Our modification leads to reduction of total number of Candidate Item sets by reducing number of rows in a transaction database that reduces the Cardinality of candidate item sets to improve efficiency of finding frequent item sets.

The organization of the paper is as follows –

In Section 2, we give analysis of APRIORI Algorithm and define terms, which are needed to understand our algorithm. In Section 3, we describe our new algorithm, APRIORI_PROCREATION. Section 4 gives the result of implementation of this algorithm on the data set of www.ddppl.in and finally Conclusions are drawn in Section 5 with a comment on future research opportunities in this area.

2 Analysis of APRIORI Algorithm:

A set of items containing k-items is referred as a k-item set. Support count or frequency support count or frequency of an item set is the number of transactions that contain the item set. An item set satisfies minimum support if the occurrence frequency of the item set is greater than or equal to the product of minimum support and the total number of transactions in \( D \). The number of transactions required for the item set to
satisfy minimum support is referred to as minimum support count. If an item set satisfies minimum support, then it is a frequent item set.

Discovering Association Rules is divided into two steps. In the first step, all sets of items with Support ≥ Minsup called the Frequent Item sets are found. In the second step the Frequent Item sets are used to discover the Association Rules. Mining Frequent Item set is a fundamental and essential operation in Data Mining application including discovery of Association Rule and Sequential Rules and is difficult to resolve. We try to solve and optimize this search for Frequent Item set in the proposed algorithm.

2.1 Finding Frequent Item sets with procreation Count:

In Apriori algorithm, we generate candidate item sets \( C_{k+1} \) from frequent item sets \( L_k \) in the \( k^{th} \) mining loop. A large number of redundant candidate item sets are generated in the process because the support of some \( k+1 \) item sets which are supersets of frequent item sets in \( L_k \) is smaller than Minimum Threshold value. We want to make cardinal number of candidate item sets \( C_{k+1} \) near to cardinal number of \( L_{k+1} \). So, when we count the support of candidates in \( C_k \), we also calculate procreation of those candidates and if procreation of a candidate in \( C_{k+1} \) is less than minimum threshold then we do not include that in \( L_{k+1} \). Thus we are reducing cardinal number of item sets in each level and ultimately reducing the search space to a significant extent in the final level. Hence, volume of the database will be reduced and mining efficiency will be improved.

2.2 Procreation count of frequent item sets:

We define procreation count of frequent item sets and then use this procreation count along with already existing support count to modify Apriori algorithm to increase efficiency of Apriori by reducing Breadth First Search space.

Let \( I = \{i_1, i_2, \ldots, i_n\} \) be a set of items. Let \( D \), the transaction database, be a set of transactions, where each transaction \( t \) is a set of items. Thus, \( t \) is a subset of \( I \). Let \( x \) be a frequent item set and \( x \cup \{i\} \) is next level procreated element, \( x \cup \{i\} \) is generated through candidate generation procedure used in Apriori algorithm. If \( D \) contains total \( n \) transactions and if total \( m \) transactions contain \( x \cup \{i\} \)

then

\[
\frac{(m / n) \times 100}{P}
\]
P is called Procreation of \( x \).

### 2.3 Relationship between Procreation and support of an item

If procreation count of frequent item set \( x \) in ordinal item sets tree is less than minimum threshold value, then support count of all children \( x \) denoted by \( xU \{i\} \) is also less than minimum threshold value.

**Vegetal item sets:**

If procreation count of frequent item sets \( x \) is greater than or equal to minimum threshold value, then \( x \) is called vegetal item sets.

### 3 Algorithm APRIORI_PROCREATION for finding frequent item sets:

Let \( U_k \) and \( U_2 \) are two infrequent item sets, which are used to decrease volume of database. Counter 1 is used to count the procreation in every loop. Algorithm generates frequent 1 item sets, frequent 2 items sets and finally frequent \( k \) item sets.

Transactional database and Minimum Threshold values are the two inputs of the algorithm. With a given minimum confidence threshold and a minimum support threshold, the problem of mining association rules is to find all the association rules whose confidence and support are larger than the respective thresholds. The higher the threshold value, the more precise the prediction and the less aggressive prefetching. In our earlier work [7], we have defined 25% minimum support and 60% minimum confidence as good breaking point.

Begin;

1: \( F_1 k = \{ \} \) \( F_1 k \) is a set of frequent item sets.

\( F_2 k = \{ \} \) \( F_2 k \) is a set of vegetal item sets.

\( U_k = \{ \} \) \( U_k \) is a set of k-infirequent item sets.

Result = \{ \} Result is a set of final outputs.

Initially there are empty sets */

2: Initialize \( k = 1; \) /* we start from first level */

3: \( A_k = \{ \text{frequent 1 – itemsets} \} \) /*\( A_k \) is a set of item sets which contains all 1- item sets */

4: If (\( A_k ! = \text{NULL} \)) /* if \( A_k \) is empty for some \( k \) then algorithm ends*/
   repeat Step 5 to 17

5: Initialize \( j = 1; \) /* We will consider each transaction \( T_j \) one by one
repeat Steps 7 to 12 until j > m
m is total number of transactions in our database*/

6: If
   cardinality of Tj < k
   Then
       Remove Tj from database

7: If (1k belongs to Uk^ cardinality of Tj=k+2^ Tj-1k belongs to Uk-(k-2) /*
   Then
   k-itemst.Tj
       remove Tj and all it’s subsets

8: If
   Tj belongs to Uk-1
   Then
       Remove Tj from database

9: for (all i, where i belongs to Tj) /*
   support_counter++; i is an item in transaction

10: for(all i, where i belongs to Tj^ cardinality of Tj>k)
    procreation_counter++;;

11: j++;

12: for all I do
   if
       support_counter of I >= minimum threshold value
   Then
       F1k=F1k U {I}

13: for all I do
   If
       procreation_counter>=minimum threshold value
   Then
       F2k=F2k U {I}

14: Uk=Ak-F1k;

15: Result=U F1k

16: k++

17: Ak=gen_candidate_itemsets with the F2(k-1)

End:
In step 6 of the algorithm, we consider transactions which have minimum cardinality1 during first iteration. In the next iteration we consider transactions having minimum cardinality 2 and so on. When we are in kth loop, all item sets of cardinality k-1 are already included in the set of frequent F-1k or to the set of infrequent itemsets Uk. So now we can remove Tj.

In step 7 we have an itemset which is known to be infrequent and whose length is 2 less than the length of transaction Tj in database. If after removing the infrequent itemset from the transaction, remaining set of items is still infrequent then remove Tj and all its possible subsets.

i is an item in transaction. If we find a transaction Tj which has already been included in infrequent itemsets then delete Tj because we cannot procreate any new frequent itemset from Tj.

In step 9 we increment support count of each itemset belonging to Tj because the support for an itemset is probability of inclusion of the itemset in a transition Tj.

In step 10 if the cardinality of item set is greater than present k value, the procreation counter is incremented by 1. This procreation counter is used to determine whether itemsets are vegetal or not.

Itemset having support count greater than minimum threshold values are included to two set of frequent item sets F1k in step 12. Similarly itemsets having procreation count greater than minimum threshold value are included to set of vegetal itemsets in step 13.

We remove all frequent itemsets from candidate itemset to get a set of infrequent itemsets in step 14 and take union of all frequent itemsets generated so far to get final result in step 15. Before going to next iteration, k is incremented.

Since candidate itemset Ak for current iteration should include all frequent and vegetal itemsets of previous iteration, so we modify Ak in step 17.

4 Implementation Methodology And Results: The proposed implementation methodology approach is primarily based on knowledge discovery of server access patterns through data mining. The data mart is populated starting from raw web/proxy server log files of www.ddppl.in, a leading travel industry publishing house which has offices in Delhi, Mumbai, Sri Lanka and Dubai. This site is not big in terms of contents
but then it has several directories and html files. The data mart population consists of a number of preprocessing and coding steps that perform data selection, cleaning and transformation.

The data mart has been implemented as a relational database, using Microsoft SQL Server 2000 Beta2. The process of data preparation and data mart population have been designed using SQL Server 2000 Data Transformation Services (DTS), a tool that allows to specify import / export / transformations processes of data through text files, databases or applications. Important tasks include field extraction, URL normalization and hash coding.

The quality of a web caching scheme is evaluated on the basis of Cache Hit Ratio, Bandwidth Ratio and Latency depending on the resource that is being focused on. We measured the Cache Hit Ratio which is defined as the ratio of number of objects served from cache to the total number of objects requested. A high hit rate reduces server load by reducing the number of requests a server must process. We also measure Bandwidth Ratio, which is defined as the ratio of total number of files fetched from the server to the total number of requests.

We extracted session IDs and the list of URLs requested in each session from the server log and assigned a unique number to each URL.

4.1 Results:

Some of the key findings are:

1. To justify our statement that APRIORI_PROCREATION reduces the searching space for Breadth First Search by lowering the scale of candidate items as compared with classical APRIORI, we plotted a graph of Execution Time Vs. Minimum Support and observed that the new algorithm is more effective in decreasing data size leading to reduction in execution Time as shown in Figure1. This fall in execution time is
prominent when the minimum support level changes from 20 to 30 in which case Apriori Procreation decrease.

2. To study what effect this reduction in Execution Time makes on Quality of a Web Caching Scheme, which is evaluated in terms of Cache Hit Ratio and Bandwidth Ratio, we plotted these parameters against Minimum Support. Cache Hit Ratio is the ratio of number of pages found in Cache to the total number of pages requested. From Figure 2, which shows a plot of Cache Hit Ratio against Minimum Support we found...
that Cache Hit Ratio is increased from 0.35 to 0.50 at a support level of 10. However for higher range of Support (between 20 to 30), Cache Hit Ratio increases at a lesser rate implying that aggressive prefetching beyond a certain point does not yield proportionate benefit.

Figure 3 shows a plot of Bandwidth Ratio against Minimum Support. Bandwidth Ratio is defined as the total number of files fetched from the server divided by the total number of requests. We note that as far as Bandwidth is concerned though the number of files prefetched increases with decrease in Support the Bandwidth Ratio decreases marginally because low support generates large number of rules thus causing more files to be prefetched.

These figures would vary for different weblog datas from different sites because it depends on how strongly various pages are related to each other and how good the association rules are that we find for those sites.

5 Conclusion: We have studied how the knowledge discovered through Web Log Mining can be used to optimize Web Server Performance thus improving user perceived latency on the World Wide Web. Implementation of our new algorithm,
APRIORI_PROCREATION on the Web Logs of www.ddppl.in shows that execution time for serving the user request is reduced as compared to classical APRIORI. This reduction in execution time brings proportionate increase in quality of Web Caching Scheme evaluated in terms of performance measure such as cache Hit Ratio and Bandwidth Ratio.

From our experience in this study, we believe that there are many open research issues. One direction would be to obtain a predictive model based on Sequence Rules and compare its performance with that of model based on association rule with respect to both Classical APRIORI and APRIORI_PROCREATION.

REFERENCES:

5. J. Han, J. Pei, “Mining Frequent Patterns by Pattern-Growth Methodology and Implications”, ACM SIGKDD Explorations (Special issue on Scalable Data Mining Algorithms), December 2000.
Formal Privacy Ontology, a Definition
Position Paper

Waël A. Hassan
University of Ottawa, Ottawa, Ontario, 800 King Edward, K1N 6N5
whassan@site.uottawa.ca

Abstract. Legislative compliance in the area of information management is of growing concern to institutions and enterprises. The Privacy legislation often adds interdependence and possibly ambiguities that need to be resolved prior to enforcement, audit, or implementation. Privacy must communicate requirements in a precise language understood by security information technologists and architects. Organizations must address privacy requirements to meet (a) their legal obligations, (b) their contractual obligations and (c) the expectations of the data subjects and client groups. A privacy ontology provides a basis for collecting engineering requirements specifying privacy governance rules, and in this sense, the represents the privacy meta rules for an organization.

Keywords: Privacy, Formal, Ontology, Policy, Governance.

1 Introduction

Computing systems are taking an increasing role in validating privacy compliance [4] [8] [14] . Validating legal compliance to privacy aims at ensuring that an enterprise has the necessary processes, roles, technical measures, and rules to govern and protect personal information. Canadian privacy laws stem from the Canadian Standards Association’s “Privacy Code” otherwise known as the ten principles. These principles are accountability, identifying purposes, consent, limiting collection, limiting use, disclosure, and retention, accuracy, safeguards, openness, individual access, and challenging compliance. These principles govern collection, use, retention, and disclosure of personal information. Canada being a federal state has several layers and areas of jurisdiction (federal, provincial / territorial, and municipal). Organizations operating in Ontario, for example, may be subject to the Personal Information Protection of Electronic Documents Act (PIPEDA)[19] , the Privacy Act, Freedom of Information and Protection of Privacy Act (FIPPA) [22] , the Municipal Freedom of Information and Protection of Privacy Act (MFIPPA) [23] , and / or the Personal Health Information Protection Act (PHIPA) [24] . In this paper we propose the use of legal ontologies for the purposes of capturing consistent privacy requirements.
The privacy domain faces the challenge of requirements capturing, refinement of specifications, in addition to interaction-detection. Privacy requirements are presented at different levels of specification and using various representation styles. A legal requirement, for example, may be represented as a declarative rule, whereas enterprise policies are represented in a functional ‘if-then-else’ style. Combining rules of operational style with declarative style is a challenge. Translating requirements to specifications and implementation is a generic problem that is the major concern of Computer Science (‘CS’) research. Jackson and Zave [11] [16] argue that “requirements exist only in the environment” and domains should be described explicitly independent of the system to be built. Privacy requirements introduce two further complexities; the first is enforcement, the second is context dependency. Enforcement is a challenge since people’s trust and use of data is a predicate in the underlying system. Context capturing, on the other hand, is challenge since the applicable set of policies changes based on context of execution. These issues and others are captured in a roadmap presented by [4]. It is certain that the trend of requirement translation presented in [7] and further studied in [2] still holds:

Table 1 Layers in Software Engineering

<table>
<thead>
<tr>
<th>Requirements Engineering</th>
<th>Requirements Specification</th>
<th>System specification and design</th>
<th>Programming Languages</th>
<th>Assembly Code</th>
<th>Machine Code</th>
</tr>
</thead>
</table>

Requirements engineering [9] has traditionally been concerned with investigating the goals, functions, and constraints of (software) systems. The process of capturing requirements can be broken down into four tasks: elicitation of information related to the problem domain; modeling of the problem; analysis of costs, completeness, and consistency; and validation with the customer [1]. These tasks pave the way to the generation of complete, consistent, and unambiguous specifications of system behaviour that are well suited for design and implementation activities [6].

There are several examples [4] of requirement engineering resulting in privacy breaches, violations, and conflicts. Therefore we classify the issues facing privacy requirements specification, translation, and conflict detection into Privacy generic problems and domain specific challenge.

Privacy domain [4] specific issues include the:
- Need for an abstract representation of privacy domain;
- Lack of precise specification models in standards;
- Need for a technique that assists in translating privacy semantics to operational constructs; and
- Need for a method that defines the integration of privacy policies into a multi-model security environment.
Generic specification issues include the inability to combine operational and declarative specifications, and the need for policy interaction detection. We suggest that ontologies can assist in the abstraction, increasing of preciseness and provide meta rules for creation of system rules from requirements or specifications.

Ontologies consist of sets of statements that describe definitions or characteristics that must be satisfied by (the ontology designer’s idea of) the “reasonable” state of the world. Formally, such statements correspond to defining terms used in logical sentences, and ontology corresponds to a logical theory. Privacy ontology will be able to offer the core abstraction to deliver such a promise. We propose the use of a decoupled ontology capturing:

- Legal Entity Ontology
- Data Entity Ontology

The paper proceeds to describe related work in Section 2. Section 3 contains the related work laying out existing ontologies and frameworks.

2. Related Work

Creation of a privacy problem as specified in the Privacy Incorporate Software Agent (‘PISA’) [15] requirements documents is rooted in the abstraction and implementation of privacy legal requirements approached in three distinct methods:

1. Existing privacy ontologies
2. Existing privacy frameworks.

2.1 Existing Privacy Ontologies

There are a limited number of privacy ontology examples, including DAML and PISA.

**DAML Privacy Ontology**

DAML developed a generic, simple and easy-to-use ontology for expressing privacy policies as well as a protocol to support matching of privacy policies. DAML distinguishes between three types of rules: Authorization, Capability, and Obligation.

**PISA Ontology**

The PISA ontology includes some meta ontology concepts such as data protection authority, privacy policy, privacy preference, and privacy principle. In addition it includes a classification of personal data into three data types. Finally, it breaks down the privacy principle into the concepts of transparency, finality, data subject rights, legal processing, and transfer.
2.2 Existing Privacy Frameworks

The current International Standards Organization (‘ISO’) and Organization for the Advancement of Structured Information Standards (‘OASIS’) documents lack a formal meta-model that captures privacy requirements. For example, eXtensible Access Control Markup Language (‘XACML’), a product of OASIS framework has a language model is used to represent security policies.

These standards provide recommendations on presentation and management of policies through languages such as XACML. We suggest and others [3] agree, XACML is missing the constructs needed to capture requirements and operational semantics required to implement privacy. XACML policies, for example, relate targets (subject, action, resource) to an obligation, which is a generic solution to a security problem. Context (purpose), which would speak to privacy compliance, is not included in this specification.

Canadian Standards Committee on Privacy (‘CAC-P’) handles recommendations to the ISO. Their work is limited to business recommendations and framework without any privacy model. One reason behind the lack of initiative to create a specification model is that standards bodies view privacy as front-end to security[30]. The private sector industry, on the other hand, has followed a pragmatic approach by adopting managed-policy governance systems[13] [17]. These systems combine process and technology to implement varying degrees of privacy. The military’s security work, for example, since the 1970s has focused on access models and algorithms related to program analysis.

3. Privacy Ontology

In this section we explain the goals, provide a formal definition for legal and data ontologies, in addition to suggesting potential applications of an ontology. Advanced normative systems use extensive sets of definitions to structure the domain on which they act. Companies have organizational structure that is taken into consideration in company policies. For example, employees can be characterized by roles. These definitions form ontologies, which are hierarchical data structures containing attributes for the entities in a certain domain, together with their relationships. Some literature refers to ontologies with the name of world knowledge, which they contrast with normative knowledge.[14] [31] The study of ontologies for legal systems is a research area in which Jurisprudence seem to be taking the greatest inspiration from Information Technology.[14] [33] [34]

3.1 Goals

We believe that future work in the area of defining legal ontologies can help the following three areas: Requirements Engineering, Requirements specification, and system specification and design.
**Formal Privacy Ontology, a Definition**

**Requirements Engineering:**
- Establish a common vocabulary: the ability to implement precise semantics is highly dependent on a consistent and precise vocabulary. A privacy ontology is able to provide the redefinitions of key terms, in addition to connectivity between equivalent concepts.
- Establish structure and legal dependency: In a multi-level, dual jurisdiction system, such as the case of Canadian provinces, the ability to understand legal applicability is important to applying the appropriate provision.

**Requirements Specification**
- Assist in identifying entities: Given that laws are entity based, a precise definition of entities removes ambiguities, it also assists in finding applicable laws.

**System specification and design**
- Defines policy format: A legal and data ontology should specify enterprise and institutional Meta Policy requirements, and hence they should affect policy format.
- Apply rules of priority and precedence: A legal hierarchy defines precedence and applicability. The ability for ontologies to define legal precedence is also an interesting area of research.

### 3.2 Semi-Formal Definition of a Privacy Ontology

In this subsection we present a formal definition of atomic and composite elements represented in the UML model. The we provide a predicate logic definition. The first definition formally presents a Legal Ontology. The second refers to a data ontology. We realize the importance of the data and policy components of the ontology. Both are instrumental in completing requirements for privacy policy definition and abstraction. As we mentioned in the motivation section, this will be future work.

**Legal Representation**

![Legal Ontology MetaModel](image)

**Fig. 1. Legal Ontology MetaModel**

Figure 1 shows the Meta Model of the legal ontology using a UML class diagram. The ['Legal Entity'] class represents a super class. It is composed as UML generalization. The idea here is that Legal Entities are generalization of one another. ['Law'] Meta class is attached to some Legal Entities.
**Data Representation**

![UML class diagram](image)

**Fig. 2. Data Ontology**

Figure 2 shows the Meta Model of a Data ontology using a UML class diagram. The ['Data'] class represents a composition relationship to other data classes. The idea here is that Data sets are compositions of one another. ['Sensitivity'] classes has a one-one relationship with datasets. The sensitivity class suggests a sensitivity level for various data compositions.

### 3.3 Formal Representation

We define formally a decoupled legal and data ontology. A Legal Ontology as defined in the UML MetaModel defines relationships between laws and applicable entities, whereas a data ontology separates between various classifications of data sensitivity.

**Definition 1. Legal Ontology**

A privacy ontology denoted by \( \mathcal{PO} = (\lambda, \varepsilon, \alpha) \). \( \lambda \) is a finite set of laws, \( \varepsilon \) is a set of legal entities, and \( \alpha \) is an ordered relationship between members of \( \lambda \) and \( \varepsilon \).

\[
P = \left\{ \begin{array}{l}
\lambda \text{ is a set of Laws} \\
\varepsilon \text{ is a set of Legal Entities} \\
\alpha \text{ is a set of Legal Applicability}
\end{array} \right. \quad \alpha \subseteq \left\{ \{u,v\} | \forall u \in \lambda, \forall v \in \varepsilon \right\}
\]

The super set of entities denoted by \( \mathcal{E} = (\varepsilon, \tau) \). Where \( \varepsilon \) is a finite set of entities. And \( \tau \) is a set of composition relationship. For example, \( \{u,v\} \in \tau \Rightarrow u \) is a type of entity \( v \).

\[
\mathcal{E} = \begin{cases}
\varepsilon \text{ is a set of Legal Entities} \\
\tau \text{ is a type of relationship}
\end{cases} \quad \tau \subseteq \left\{ \{u,v\} | \forall u, v \in \varepsilon \right\}
\]


**Definition 2. Data Ontology**

A data ontology denoted by \(DO=(\delta,\sigma,\theta)\). \(\rho\) is a finite set of processes and \(\sigma\) is a set of ordered pairs of members of \(\rho\). We call members of \(\rho\) and \(\sigma\) processes and contained processes, respectively. The process set \(\rho\) of a process ontology \(P\) is denoted as \(\rho(P)\). Similarly, the containment relationship is denoted as \(\sigma(P)\). Finally, the set of purposes is denoted by \(\theta(P)\). A containment relationship

\[
DO = \begin{cases} 
\{ \delta \text{ is a set of data elements} \} \times \{ \sigma \text{ is a sensitivity Level} \} \subseteq \{ \{ u, v \} | u, v \in \delta \} 
\end{cases}
\]

The super set of data elements denoted by \(\Delta=(\delta,\kappa)\). Where \(\delta\) is a finite set of Data Elements. And \(\kappa\) is a set of composition relationships. For example, \(\{(u,v)\} \in \kappa \Rightarrow v\) is included in the composition of \(u\).

\[
\Delta = \begin{cases} 
\{ \delta \text{ is a set of data elements} \} \times \{ \kappa \text{ is a composition Property} \} \subseteq \{ \{ u, v \} | u, v \in \delta \} 
\end{cases}
\]

### 3.4 Example Application

Figure 3, illustrates a legal applicability map for entities in Ontario, including individuals, corporations, government, non-profits, non-governmental organizations, and registered charities.¹

![Diagram](http://ontarioprivacyontology.googlepages.com/home)

**Fig. 3.** Privacy Laws Applicable to Organizations Operating in Ontario

¹ [http://ontarioprivacyontology.googlepages.com/home](http://ontarioprivacyontology.googlepages.com/home)
3.5 Using a Privacy Ontology

We adopt the methods of constructing a privacy policies being proposed by Hassan and Logrippo to appear. At the abstraction stage, their method suggests the following process.
1. collect the list of purposes for which the entity consumes data while identifying the data without a specific purpose;
2. map the purposes to existing entity processes, note that the mapping from purpose to process is a many-many relationship;
3. identify applicable laws and specify system policies enforcing legal requirements;
4. define contractual obligations;
5. define policies using privacy constructs of (consent, collection, distribution, retention, secondary use).

Privacy design principles, expressed by the meta-rules in the proposed ontology, provide the cornerstone for organizations operating in Ontario to be privacy compliant. Yet, the creation of a support organizational culture of privacy goes beyond the requirements. Creation of policy and management processes are of great importance to the fields of information management and privacy. Other areas for research include issues relating to the responsibility for educating individuals on the exercise of their privacy rights under the legislation, analysis on the extension of privacy impact assessment requirements beyond the realm of health care, and specific procedures questions regarding the use of data sharing agreements between multiple levels of government in a pan-Canadian jurisdictional environment for the purpose of protecting informational privacy. These are questions of the broad sort, questions of education, expertise, assessments, scope and cooperation.

4 Conclusion

We are proposing a generic definition of an ontology that can be used at the requirements engineering level in the software engineering process. The definition provides a precise specification both semi-formally and formally. The specification’s applicability needs to be validated. Moreover, we intend to develop a formal analysis tool to validate its features. We believe that legal and data ontologies can set the basic definitions, describe dependencies, meta rules and rules of precedence for privacy policy construction in federal jurisdiction. It is our intent to prove and validate that the ontology is implementation independent, decouples rules from structure, in addition it is designed for change.
Acknowledgment. I am indebted to my supervisor Luigi Logrippo at the UQO Université du Québec en Outaouais for introducing me to this area of research and for sharing with me his ideas on Ontologies and formal methods in general. As well, I am thankful to Tracy Kosa for her numerous detailed comments on earlier work related to this paper, and for introducing me to Privacy Impact Assessment Models, laws in Ontario and providing me with applicable examples.

References


http://www.site.uottawa.ca/~luigi/papers/07_logrippo.pdf


Formal Privacy Ontology, a Definition


A Method for Optimizing the Size of Wireless Mobile Messages

P P Abdul Haleem and M P Sebastian

Abstract—Optimization of the message size is a matter of concern in the resource limited, mobile wireless devices. As on today, no uniform standard is available for optimized wireless mobile messaging. The heterogeneous nature of mobile wireless devices makes the formulation of a uniform messaging standard a challenging task. The eXtensible Markup Language (XML), a universally accepted method for representing data, is a potential choice for this purpose. However, limitations like verbosity and the need for strict structuring make the XML format less attractive for wireless mobile applications. This paper proposes SA-YAML, a schema attached messaging format based on YAML Ain Markup Language (YAML), as an alternative for optimizing the size of mobile messages. SA-YAML makes use of the inherent properties of YAML in squeezing the message size, in addition to two phases of optimization. The evaluation results indicate the usefulness of SA-YAML over XML as a promising message format in wireless mobile devices.

Index Terms—Mobile Computing, Wireless Mobile Environment, XML, Knowledge Representation, Squeezing of Mobile Messages, Streaming of messages.

1 INTRODUCTION

Wireless mobile devices have many limitations such as power, memory, etc. These limitations in turn prevent the use of processors with high computing power in mobile devices. Limited bandwidth, increased latency and increased retransmission due to packet losses are still concerns in the wireless networking world. Formulating an efficient method for the transfer of messages in wireless mobile devices is a need of the hour. It is desirable to have a cross platform technology that can be used to achieve seamless access to the various ranges of devices and systems. The search for such an acceptable cross platform standard normally converges to XML. In Simple Object Access Protocol (SOAP), XML is already in use as a message syntax format [2]. The main factor in favor of XML is its continued use as the de-facto standard to represent data over net connected heterogeneous systems. However, XML is crippled with many limitations [1] such as high verbosity (results in larger message size and vulnerability to retransmissions), strict structuring (verification of this structure makes XML parsing a heavyweight process), and high textual content (requires string parsing for further processing). Many web services make heavy use of messages for exchanging data.

This paper is aimed at proposing a standard message format, particularly for the wireless mobile environment. The standard is expected to retain the merits of the XML format, with improvements on its limitations with respect to the wireless mobile environment. This paper is organized as follows: Section 2 presents the related work. Section 3 discusses the design principles. Section 4 presents the system architecture. Section 5 gives the performance evaluation and results. Section 6 concludes the paper.

2 RELATED WORK

Several solutions are proposed in the literature to overcome the limitations of XML in wireless mobile environment. Some of these approaches are successful in tackling the limitations for narrow application. A closer look at these methods reveals certain problems when applied to the wireless mobile environment. For instance, standard compression techniques such as Millau [6], Gzip [7] and XMill [7] give good performance only with larger messages with a high redundancy rate. But it may not yield good results with short messages with less redundancy, typically used in the wireless
mobile networks. The compression/decompression layer is considered as an additional burden for the resource critical mobile devices [3]. Several alternative serialization methods ([10], [11], [12] and [13] for instance) are also proposed. As these formats remain neutral to the natural language, problems like readability issues and loss of description could be possible when these techniques are employed.

Our proposed work retains the agility of the legacy XML technology - at the same time, some of its inherent deficiencies specific to wireless mobile environment are addressed.

3 THE PROPOSED METHOD

We have chosen YAML as the vehicle to achieve our goal. YAML is a light weight and computationally powerful data serialization language. The primary reason for considering YAML for our work is its low verbosity, at the same time being expressive and extensible. It is also easily readable and has a consistent information model. YAML supports stream-based processing too. YAML has a human-readable data serialization format that takes inputs from the languages such as XML, C, Python, Perl, and also the format of electronic mail as specified by RFC 2822. It is optimized for data serialization, configuration settings, logs files, Internet messaging and filtering [4].

Apart from just suggesting YAML as an alternative, we have proposed important value additions to ensure better throughput. The proposed format, christened as SA-YAML (Schema Attached YAML), is obtained in five stages. In the first stage, the message is encoded in the natural YAML format. This natural YAML format is optimized in the remaining stages. The second stage bifurcates the structure and content to create a schema definition of the message content, christened as YASchema, and the third stage verifies and validates the definitions made in the second stage. The fourth and fifth stages squeeze the message as obtained from stage II by making use of the properties of YAML and the YASchema, respectively.

4 DESIGN

4.1 Stage I: Message encoding in YAML format

YAML streams are encoded using the set of print-able Unicode characters, either in UTF-8 or UTF-16. Creation of messages is easy and can be done with any text processor. A detailed account of YAML specifications is given in [8]. The output of this stage is the message encoded in YAML natural format.

4.2 Stage II: YASchema Creation

The YAML data serialization includes both data description and content as an interwoven combination. Removing the clutter from the message can be achieved by bifurcating the structure and content. There exists a schema validator for YAML files called kwalify [5], developed by Kuwata-lab. YASchema attempts to improve upon the Kwalify schema in the following ways: (i) the YASchema is constructed as per the flow-style method of creating YAML messages. This helps to squeeze down the overall verbosity of the message, (ii) without affecting the readability, elements of YASchema and their properties are stored in the schema definition file in such a way that they require only less number of bytes, (iii) facility to define ID codes for elements of YAML are given - this arrangement significantly reduces the size of the original message, and (iv) the process of reusing the same schema across multiple sessions is envisaged - special directive is included in the YAML file to specify the schema definition to be referred.

There are three kinds of nodes in YAML - scalar, sequence, and mapping [8]. Apart from this, there can be mappings of sequences and sequences of mappings also. We evaluate the message structure to decide about the possibility of folding node contents. This is necessary as a blind conversion of the message to flow-style may not produce a convincing result. Nodes of the categories belong to sequences, mappings, sequences of mappings and mappings of sequences are marked as squeezable. In addition to this, we initiate the schema creation in this stage. Evaluation of the messages structure is used to decide the type of the schema to be constructed. Type can be of any of the three kinds - scalar, sequence and mapping. Primitive details like name of the field, its data type and a ID value are added to the schema for every member in a node. The output of this stage is YASchema together with message formatted in YAML natural format.

4.3 Stage III: Validation and Verification

This stage is used for the validation and verification of the schema constructed in stage II. This stage is used to correct anomalies, and to add more details like whether mandatory or not, expected pattern of data, default values etc. The output of this stage
is validated YASchema together with message formatted in YAML natural format.

4.4 Stage IV: Phase I Optimization

Normally, information representation in YAML is identical to that of XML. This enhances the readability of the content, but results in more number of bytes for preparing the messages. Interestingly, the so-called Flow style method specified in YAML helps one to squeeze the size considerably. Reorganization of the contents in flow style yields better performance. For e.g., a complex message consisting of customer invoices having many product details each can be presented to this stage, for better conservation of message size. Stage IV scans the message contents with the help of YASchema and reorganizes the information that are identified as fold-able (in stage I) to flow style. The output of this stage is semi optimized message formatted in YAML together with YASchema.

4.5 Stage V: Phase II Optimization

This stage concentrates on further optimizing the messages, making use of the enhancement to YAML, namely YASchema. All primitive details regarding every element in the message are included in the schema at the schema creation stage itself. Special ID codes are provided for each element. Equipped with these information, a scan over the message encodes the elements with the ID codes. The ID codes can be obtained from the accompanying schema (Figure 1). If a sender likes to reuse a schema that was sent earlier, the schema to be referred can be specified in the message itself. This makes the decoding easier. Such re-usability of schema over multiple streams is not possible for YAML in its natural format. The output of this stage is the optimized message formatted in YAML together with YASchema.

5 Testing and Performance Evaluation

The testing is done for five categories of messages including short (simple messaging format with only text), small (a single record consisting of String, Float and DateTime types of data), medium (details of 25 customer records), large and composite (these two categories can be invoices with many records). These messages are tested for five types of formatting which include YAML natural format, semi optimized format (after applying Phase I of squeezing), optimized format (after applying Phase II of squeezing), XML, and SOAP. XML and SOAP formats are used as benchmarks. Three measures that are indicative of the resource usage and viability of the proposed message format are identified for performance evaluation. These measures are Schema Size, Message Size and Transmission Speed.

5.1 Schema Size

For each category of messages the schema definitions are created as per the specifications of Kwalify and YASchema. The gain in number of bytes is tabulated in Table 1. YASchema offers considerable optimization for all types of messages. The Short category yields the maximum gain. The Large and Composite categories (with gains 53.23% and 54.78%, respectively) are profitable for repeated occurrences of records. In the Medium category the gain is over 50%. This is also impressive taking into consideration of the re-usability of the schema over multiple streams.

5.2 Message Size

The message sizes obtained in each of these categories are listed in Table 2. The sizes of the YASchema (developed in stage II) in each category are also included. The performance advantage in squeezing down the message size can be derived from the table. For example, the gain in message size (in bytes) using YAML natural format, over messages formed in XML for Short, Small, Medium, Large, and Composite categories are 46, 153, 2816, 5024, and 19277 bytes, respectively.

The results for semi optimized message format are as expected. The gain in message size for optimized format is tabulated in Table 3. The performance gain for the five categories of messages compared to XML are 74, 250, 7217, 12951 and 44927 bytes, respectively. This measurement excludes the size needed for schema (as shown in Figure 2). For small message types, especially for Short and Simple, squeezing beyond YAML natural format is not profitable (if these messages need to send only once), as seen from Figure 3. When the overhead needed to maintain the schema for XML is also taken into account, the optimized format outperforms XML in all categories. Even though there is an apparent marginal gain (e.g., in Short category the gain of 28 bytes - 38.7% and in Simple category the gain is 97 bytes - 54%) in message size,
introduction of YASchema adds more bytes to the total message size. However, YASchema is always advantageous when there is a need to send multiple messages with the same structure (as transmission of the schema will be only a one-time payload).

5.3 Transmission Speed

Three classes of record sets with varying levels of complexity are formed for this analysis, which include Simple, Medium, and Complex. The bytes required for the representation of record sets are considered to represent for the transmission speed. The number of bytes needed to represent each of these record sets is tabulated in Table 4. No packet losses and maximum link speeds (64kbps/256kbps) are assumed for the analysis. It can be seen from Table 4 that all data sizes, especially the Complex type, fit well in the limits of a single TCP/IP maximum segment size when it is in optimized message format.

The time required for transmitting the record sets over the 64kbps link are tabulated in Table 5. The advantages of YAML, especially for the optimized message format, are evident from Table 5. In the case of Medium category, there is an advantage of 15.26 micro secs over XML and 20.09 micro secs over SOAP. The gap is widened for semi optimized YAML SA-YAML (output after stage IV) is getting the maximum advantage (Table 6) by a factor of 2.18 for Small type, 1.71 for Medium type, and 5.14 for Complex type. Similar results can be derived for SOAP record sets also. Thus, the performance of the optimized message format is better than XML and SOAP in all categories. The optimized message format has notable gains over YAML natural format also (6.99, 7.32, and 125.36 micro secs for Small, Medium, and Complex categories, respectively, as can be seen from Table 5).

6 Conclusions

This paper proposes an optimized messaging format for wireless mobile devices based on YAML, a user friendly and lightweight messaging format, supported by YASchema. This formatting requires five stages of processing, starting with YAML. In terms of schema size, message size and transmission speed, this format shows better performance compared to that of XML format. The optimized message format performs better than YAML natural format also. SOAP is also used as a benchmark format, in addition to XML. It is worth mentioning that the proposed method retains the readability of the original message. The proposed optimized message format could be a candidate for standardizing the formats for wireless mobile messaging as it is simple, human readable, easily editable and, most importantly, with less bandwidth requirements.

The future work is to develop this Schema Attached YAML into a messaging standard. This requires many improvements including (i) facility for better encoding in to Abstract Syntax Notation One (ASN.1), (ii) security standards at par with the specifications of XML encryption and related specifications, and (iii) introduction of a policy based security mechanism. The possibility of weaving semantics to the messaging standard is also worth investigating.

References

[10] WAP Binary XML Content Format, Available at http://www.w3.org/TR/wbxml/.
### TABLE 1
Schema Size Analysis

| Message Categories | Schema Size (in bytes) | Kvalify (a) | YASchema (b) | Gain % 
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Short</td>
<td>366</td>
<td>121</td>
<td>66.94</td>
<td></td>
</tr>
<tr>
<td>Small</td>
<td>987</td>
<td>448</td>
<td>54.61</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>1177</td>
<td>576</td>
<td>51.06</td>
<td></td>
</tr>
<tr>
<td>Large</td>
<td>1922</td>
<td>899</td>
<td>53.23</td>
<td></td>
</tr>
<tr>
<td>Composite</td>
<td>1988</td>
<td>899</td>
<td>54.78</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 2
Comparison - Message Size

<table>
<thead>
<tr>
<th>Type</th>
<th>YAML</th>
<th>XML</th>
<th>SOAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short</td>
<td>145</td>
<td>120</td>
<td>221</td>
</tr>
<tr>
<td>Small</td>
<td>312</td>
<td>245</td>
<td>250</td>
</tr>
<tr>
<td>Medium</td>
<td>8128</td>
<td>4423</td>
<td>576</td>
</tr>
<tr>
<td>Large</td>
<td>13799</td>
<td>5901</td>
<td>899</td>
</tr>
<tr>
<td>Composite</td>
<td>54292</td>
<td>29592</td>
<td>73569</td>
</tr>
</tbody>
</table>
Fig. 1. System Architecture - Stage V
### TABLE 3
Comparison - Gain Achieved in Message Size (in bytes) with SA-YAML

<table>
<thead>
<tr>
<th>Message Category</th>
<th>YAML Phase II</th>
<th>XML Phase II</th>
<th>SOAP Phase II</th>
<th>Gain Over XML (Excluding Schema)</th>
<th>Gain Over SOAP (Including Schema)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
<td>(c)</td>
<td>(d)</td>
<td>(c - a)</td>
</tr>
<tr>
<td>Short</td>
<td>117</td>
<td>221</td>
<td>191</td>
<td>312</td>
<td>74</td>
</tr>
<tr>
<td>Small</td>
<td>215</td>
<td>250</td>
<td>465</td>
<td>604</td>
<td>250</td>
</tr>
<tr>
<td>Medium</td>
<td>3727</td>
<td>576</td>
<td>10944</td>
<td>13317</td>
<td>7217</td>
</tr>
<tr>
<td>Large</td>
<td>5872</td>
<td>899</td>
<td>18823</td>
<td>19913</td>
<td>12951</td>
</tr>
<tr>
<td>Composite</td>
<td>28642</td>
<td>899</td>
<td>73569</td>
<td>44927</td>
<td>46302</td>
</tr>
</tbody>
</table>

### TABLE 4
Record Sets - Categories

<table>
<thead>
<tr>
<th>Type</th>
<th>YAML Phase I</th>
<th>XML Phase I</th>
<th>SOAP Phase I</th>
<th>Original Phase I</th>
<th>Phase I</th>
<th>Phase II</th>
<th>Phase II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>145</td>
<td>120</td>
<td>117</td>
<td>145</td>
<td>120</td>
<td>117</td>
<td>312</td>
</tr>
<tr>
<td>Medium</td>
<td>320</td>
<td>281</td>
<td>260</td>
<td>320</td>
<td>281</td>
<td>260</td>
<td>487</td>
</tr>
<tr>
<td>Complex</td>
<td>2171</td>
<td>1182</td>
<td>1144</td>
<td>2171</td>
<td>1182</td>
<td>1144</td>
<td>6250</td>
</tr>
</tbody>
</table>

### TABLE 5
Transmission Speed (in micro secs) using 64kbps link

<table>
<thead>
<tr>
<th>Type</th>
<th>YAML Phase I</th>
<th>XML Phase I</th>
<th>SOAP Phase I</th>
<th>Original Phase I</th>
<th>Phase I</th>
<th>Phase II</th>
<th>Phase II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>17.7</td>
<td>11</td>
<td>10.71</td>
<td>17.7</td>
<td>11</td>
<td>10.71</td>
<td>38.09</td>
</tr>
<tr>
<td>Medium</td>
<td>39.06</td>
<td>34.3</td>
<td>31.74</td>
<td>39.06</td>
<td>34.3</td>
<td>31.74</td>
<td>59.45</td>
</tr>
<tr>
<td>Complex</td>
<td>265.01</td>
<td>144.23</td>
<td>139.65</td>
<td>265.01</td>
<td>144.23</td>
<td>139.65</td>
<td>762.9</td>
</tr>
</tbody>
</table>

### TABLE 6
Comparison (SA-YAML Vs XML & SOAP) - Transmission Speed (in micro secs) using 64kbps link

<table>
<thead>
<tr>
<th>Record Set</th>
<th>YAML Phase II</th>
<th>XML Phase II</th>
<th>SOAP Phase II</th>
<th>Gain Over XML (b/a)</th>
<th>SOAP (c/a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>10.71</td>
<td>23.32</td>
<td>38.09</td>
<td>2.18</td>
<td>3.56</td>
</tr>
<tr>
<td>Medium</td>
<td>31.74</td>
<td>54.32</td>
<td>59.45</td>
<td>1.71</td>
<td>1.87</td>
</tr>
<tr>
<td>Complex</td>
<td>139.65</td>
<td>717.9</td>
<td>762.9</td>
<td>5.14</td>
<td>5.46</td>
</tr>
</tbody>
</table>
Fig. 2. Comparison of Message Size (Excluding Schema Size) - SA-YAML Vs XML & SOAP

Fig. 3. Comparison of Message Size (Including Schema Size) - SA-YAML Vs XML & SOAP
Test Case Generation for White-Box Unit Testing

Samira Hussain and Salam N. Salloum
Department of Computer Science
California State Polytechnic University Pomona
ssalloum@csupomona.edu

Abstract: Black-box and white-box testing are the two major techniques for unit testing. In black-box testing, no information about the internal structure of the program under testing is available. However, in white-box testing, a complete source code or the internal structure is available. Basis path testing is a white-box testing technique that uses a control flow graph (CFG) of a given program to generate a basis set of independent paths for the CFG. Different techniques have been proposed to generate test data that cover all the paths of a basis set. In this paper, we present an interactive tool that performs three tasks:
- It constructs a control flow graph of a given program based on the pseudocode and information provided by the user;
- It computes a basis set of independent paths of the control flow graph;
- It generates test data using genetic algorithm to exercise all basis paths.

We evaluated the performance of different mutation operators for the genetic algorithm based on the percentage of basis paths covered by the generated test data. Experiments show that the use of two known mutation operators, input value and one-point crossover, provide the best path coverage for the programs tested.

Keywords: unit testing, test case generation, control flow testing, genetic algorithms

1 Introduction

Software testing has two main aspects: generating test data and applying test data adequacy criterion. A test data generation technique is an algorithm that generates test cases, whereas a test data adequacy criterion is a predicate that determines whether the testing process is finished. Several test data adequacy criteria have been proposed, such as control flow-based and data flow-based criteria.

The use of genetic algorithms (GAs) in test data generation has become the focus of several recent research studies. This paper presents an interactive tool for automatic test data generation for basis path testing using a genetic algorithm. Our tool consists of modules for constructing the control flow graph (CFG) of a given program based on the pseudocode provided by the user, computing the basis set of independent test paths in the CFG and lastly generating test data using a GA to exercise all the basis test paths. The genetic algorithm conducts its search by constructing a new test data from previously generated test data that was evaluated as effective test data. In the parent selection process, the GA uses the random selection method. In the mutation process, the GA uses five different mutation operators: mutation of input value, one-point crossover, boundary value, uniform value and non-uniform value. We evaluate the performance of the mutation operators in the genetic algorithm on the basis path coverage criterion.
This paper is organized as follows: Section 2 describes the basis path testing technique. Section 3 describes the principles of GAs. Section 4 describes the proposed GA for test data generation and gives the results of applying different mutation operators in the GA to an example program. Section 5 presents the results of the experiments that are conducted to evaluate the effectiveness of the mutation operators in the GA.

2 The Basis Path Technique

Basis path testing is a white-box testing technique that is widely used during unit testing to test the control flow in the program. Basis path testing uses a control flow graph to depict the logical control flow of program under test. A basis set of independent paths is identified from the CFG of a given program (each path starts with a start node and ends with the stop node) and tested to guarantee that all statements and branches in the program have been executed at least once.

The control flow of a program can be represented by a directed graph with a set of nodes (procedure and predicate nodes) and a set of edges. Each node represents a group of consecutive statements. The edges of the graph are the possible transfers of control flow between the nodes. A path is a finite sequence of nodes connected by edges.

Our tool constructs the CFG directly from the pseudocode of a program entered by the user using standard pseudocode notations [Pressman 2003]. The use of pseudocode ensures that there are no syntax errors in the input program and hence in the construction of the CFG.

The CFG of a given program is built from the basic/prime control flow graph notations, using only two operations: sequencing and nesting [Salloum and Salloum 2006]. All programming constructs can be uniquely represented by the prime control flow graph notations. Our tool implements the following programming constructs: sequence (statements), if-then-else loop, while-do loop, do-while loop and for loop. The Boolean constructs (and, or and not) are also dealt with.

Our tool uses the linked list data structure to store both the predicate and procedure nodes created. The predicate and procedure nodes stored in the linked list aid in the derivation of test cases. As the CFG of a program is not a fully connected graph, so it justifies the use of a linked list. Linked list is updated dynamically as user enters the pseudocode. Also, it allows efficient traversal of data nodes when converting the CFG in the shape of a tree for basis path computation.

It is assumed that the parameters and variables in a method call use the built-in data types. A lookup table stores the definition of variables in the given pseudocode. We follow the strongly typed convention of Java language i.e. all variables must first be declared before they can be used. Lookup table holds the following information: variable type, variable name and variable value (if any). Variable type can be any of the built-in data types in Java: integer, long, float, double, short, byte, boolean and character and string. Variable value, if entered, can be either positive or negative. Our program stores the possible range of values for the built-in data types. This range of values, together with the entered variable value is used to generate the initial test data.
The next step is to calculate the cyclomatic complexity of the resultant CFG. Cyclomatic complexity is software metric that provides a quantitative measure of the logical complexity of the program. The value computed for cyclomatic complexity provides the number of independent paths in the basis set of a program’s CFG. It also provides us with an answer to the number of paths to test in a given program. We calculate cyclomatic complexity by adding 1 to the number of predicate nodes in the CFG.

Our tool implements an efficient and formal algorithm [Salloum and Salloum 2006] to compute a basis set of independent paths from the CFG. The set of basis paths are linearly independent paths that are sufficient to express any other paths of the CFG. For a given CFG, the algorithm constructs a tree structure by visiting every node and edge of the CFG at least once. The paths of the resulting tree from the root to the leaves represent a basis set of paths for the CFG.

3 The Principles of Genetic Algorithms

Genetic algorithm is a heuristic that mimics the evolution of natural species (survival of the fittest) to search for optimal solution to a problem. GA generates a sequence of populations by using a selection mechanism, and use crossover and mutation as search mechanisms.

Each individual in the population receives a measure of its fitness in the environment. Reproduction selects individuals with high fitness values in the population. Crossover and mutation of fitter individuals derives a new population in which individuals may be even better fitted to their environment. Crossover involves swapping some genes in two individuals. Mutation introduces slight changes to one or several genes in an individual. The structure of simple GA is given below.

Simple Genetic Algorithm ()
{
    initialize population;
    evaluate fitness of population;
    while termination criterion is not reached
    {
        select solutions for next population;
        perform crossover and mutation;
        evaluate fitness of population;
    }
}

The algorithm uses a single crossover and mutation operator throughout the entire genetic process. Also, the algorithm will iterate until the population has evolved to form a solution to the problem or until a maximum number of iterations have taken place (suggesting that a solution is not going to be found given the resources available).
4 A Genetic Algorithm for Test-Data Generation

This section describes a GA that we have implemented for automatic test data generation, which is guided by the control flow dependencies in the program. The algorithm, proposed by Tonella [2004] searches for test cases that achieve maximum path coverage from the basis set of test paths. In other words, the GA in our tool generates test cases for a method under test until a satisfactory level of path coverage (i.e. 100% of method’s paths from the basis set of paths) is attained.

The GA accepts as input the control flow graph of the program to be tested, the set of basis paths to be covered, the number of input variables and the domain and precision of each input variable. Also, it accepts some other GA parameters: population size, maximum number of generations, and the probability of mutation. The algorithm produces a set of test cases, the set of basis paths covered by each test case and the list of uncovered basis paths, if any.

The algorithm uses an integer vector, called the path coverage vector, to record the traversed basis paths. In this vector, each element (initially zero) corresponds to a basis path. Whenever a basis path is covered, the number of the test case that caused this coverage is stored in the corresponding element of the path coverage vector. Our tool keeps track of all the generated test cases that cover new paths in the basis set. We use a counter to count them. These test cases are stored for later use.

Firstly, the overall algorithm is presented, and then the major components of this GA are discussed in turn.

Input:
- Control flow graph of the program P to be tested;
- Basis set of paths to be covered;
- Cyclomatic complexity of the program P;
- Number of program input variables;
- Domain and precision of input data;
- Population size;
- Maximum number of generations;
- Probability of mutation;

Output:
- Set of test cases for program P and set of basis paths covered by each test case;
- List of uncovered basis paths, if any;
**TestCaseGeneration (Input, Output)**

Begin

**Step1: Initialization**
- Initialize the path coverage vector to zeros;
- Create initial_population randomly based on number of variables in P;
- Current_population = initial_population;
- Target_paths_to_cover = cyclomatic_complexity of P;
- Set of test cases for P = 0;
- Coverage_percent = 0;
- No_of_generations = 0;
- Counter = 0;

**Step 2: Generate test cases**

For each member of Current_population do
Begin
    Execute test cases with Current_population as input;
    Evaluate the fitness of test case in Current_population;
    If (some basis set of paths are covered) then
        Counter = Counter + 1;
        Add effective test case to set of test cases for P;
        Update the path coverage vector;
        Update Coverage_percent;
        Target_paths_to_cover--;
    Endif
Endfor

While (Target_paths_to_cover != 0 and Coverage_percent != 100 and No_of_generations <= Max_gen) do
Begin
    Select set of parents of new population from effective members of Current_population according to fitness of test cases using random selection method;
    Set of parents of new population = Current_population;
    Create New_population using crossover and mutation operators;
    Current_population = New_population;
    For each member of Current_population do
        Begin
            Execute test cases with Current_population as input;
            Evaluate the fitness of test case in Current_population;
        End
    Endfor
Endwhile
If (some basis set of paths are covered) then
    Counter = Counter + 1;
    Add effective test cases to set of test cases for P;
    Update the path coverage vector;
    Update Coverage_percent;
    Target_paths_to_cover--;
    Break from for loop;
Endif
Endfor
Increment No_of_generations;
Endwhile

Step 3: Produce output
    Return set of test cases for P and set of basis paths covered by each test case;
    Report the uncovered basis paths, if any;
End.

4.1 Initial population

We randomly generate pop_size to represent the initial population, where pop_size is the population size. The appropriate value of pop_size is experimentally determined. Initial population of test cases are generated randomly using the random number generator of Java language.

4.2 Evaluation function

The fitness of a test case is obtained from the control flow graph edges that are traversed during its execution. The algorithm evaluates each test case by traversing the CFG, and checking whether the path covered after reaching the stop node is in the set of basis paths. If above is true, then the fitness value for the particular test case is closer to one. Otherwise, the fitness value is close to zero. The fitness value is the only feedback from the problem for the GA. A test case is considered effective if its fitness value is one. Each time a target basis path is covered, the test case covering is added to the result set as one of those necessary test cases to achieve the final level of coverage.

4.3 Selection

After computing the fitness of each test case in the current population, the algorithm selects test cases from the effective members of the current population that will be
parents of the new population. If none of the members of the current population was effective, all the members of current population are considered the parents of the new population. The selection process of the GA uses the random selection method described below.

In the random selection method, the selection of parents is made randomly, so that every effective member of the current population has an equal chance of being selected for recombination.

Assume that \( n \) members of the current population were effective, where \( n \leq pop\_size \). The parents are selected as follows:

Isolate the effective members and number them from 1 to \( n \);
For \( i = 1 \) to \( pop\_size \) do
Begin
    Generate a random integer number \( j \) from the range \([0…n]\);
    Select test case \( j \) from the effective members;
Endfor

### 4.4 Recombination

In the recombination phase, we use the mutation operator, which is critical to the success of GAs, since it diversifies the search directions and avoids convergence to local optima. The mutation operator creates new individuals from the selected parents to form a new population.

Mutation is used to maintain genetic diversity from one generation of population to the next. The performance of GA is influenced by the choice of mutation operator. Determining which mutation operator to use is quite difficult and is usually learned through experience or by trial-and-error. That is, experiments must be done using all candidate mutation operators to find the best operator for a specific problem, which consumes considerable time and computation resources.

We have used the following known mutation operators in the above genetic algorithm to introduce a small and random change in the next population:

- **Mutation of input value**: involves replacing a value with another random value of the same type.
- **One point crossover**: involves taking the midpoint of current and new population.
- **Boundary value**: involves replacing the chosen value with either the upper or lower bound (chosen randomly).
- **Uniform value**: involves replacing the chosen value with a uniform random value selected between upper and lower bound.
- **Non-uniform value**: involves replacing the chosen value with a non-uniform random value selected between upper and lower bound.
In the traditional GAs, the population evolves until one individual, which represents the solution, is found. In our case, this would correspond to many groups of test cases and several runs of the program to achieve maximum path coverage of the given program (i.e. traversing all the basis paths of a given program). We record which basis paths of the program each test case has covered and halt the evolution when a set of test cases has traversed the entire basis paths of the program. The solution is this set.

5 Experimental Results

The algorithm presented above gives acceptable results for constructing the control flow graph from the pseudocode of a given program. Some of the programs used to test CFG construction include bubble sort, insertion sort, selection sort, quick sort’s partition algorithm and sum of squares from Standish [1998]. It has been observed that CFG for large programs (with four or more predicate nodes) is generated in a reasonable time. The CFG updates dynamically and shows all the pertinent control flow information. The algorithm implemented in the tool for generating a basis set of paths for a given CFG by using Salloum and Salloum’s [2006] approach yields results in linear time.

We evaluated the performance of each of the five mutation operators individually in the genetic algorithm on the maximum basis path coverage criterion. Sometimes, a single mutation operator does not achieve the best mutation results; instead, several mutation operators may have to be applied to achieve the best performance. This observation led us to test the performance of any two mutation operators (out of the five operators) applied simultaneously on the genetic algorithm’s maximum path coverage criterion. The set of mutation operators that provides the best results for path coverage criterion are mutation of input value and one point crossover when applied simultaneously.

Figure 1 shows a table comparing the performance of the five mutation operators applied individually and in combinations of two. The table gives a list of test programs along with the number of paths in the basis set of each program. These test programs were run on our tool. The mutation operators are numbered 1 to 5 in the table where 1 is mutation of input value, 2 is one point crossover, 3 is boundary value, 4 is uniform value and 5 is non-uniform value.
Results from the experiments show that maximum (100%) level of path coverage is achieved when mutation of input value and one-point crossover mutation operators are applied simultaneously. The genetic algorithm implemented in our tool for generating test data achieves optimal coverage of a method’s branches within a reasonable computation time. The resulting test suites are generally compact.
6 Conclusions

Unit testing makes heavy use of white-box testing techniques, specifically basis path testing. Basis path testing uses the control flow graph of a program to generate a set of independent control flow paths.

Our work focuses on generating test cases for basis path testing using genetic algorithms. The GA technique presented in this paper is guided by control flow dependencies in the given program to search for test data that fulfills the maximum path coverage criterion.

The basic steps followed in our work are:

a) Control flow graph construction
b) Basis set of paths selection
c) Test case generation using genetic algorithms

Experiments on our tool show that the algorithms implemented for control flow graph construction, basis set of paths selection, and test case generation for a given program yield results in a reasonable time. The CFG updates dynamically and shows all the pertinent control flow information. Furthermore, the use of genetic algorithms for basis path testing proves extremely powerful. The simultaneous use of two known mutation operators, mutation of input value and one point crossover, achieves 100% basis path coverage.

The use of genetic algorithm with basis path testing has shown satisfying results in terms of path coverage, time taken to generate test cases and compactness of the resulting test suites. A software tool has been implemented and the power of the approach demonstrated.
References


Integrating Generative and Aspect-Oriented Technologies For Framework Instantiation

R. P. Mahapatra  
Asst. Professor,  
S.R.M. University, Department of Computer Science and Engineering)Chennai.  
email : mahapatra.rp@gmail.com

Dr. Jayaram Pradhan  
Professor  
Berhampur University,  
Department of Computer Science, Orissa  
email : jayarampradhan@hotmail.com

Abstract

Application frameworks have been successfully used as valuable tools to improve software quality while reducing development efforts. Nevertheless, frameworks still face important challenges in order to be widely adopted. In particular, framework instantiation is still a painful task requiring application developers to understand the intricate details surrounding the framework design. Some approaches to alleviate this problem have already been proposed in the literature but they are usually either just a textual cross-referenced document of the instantiation activities or too tied to technology or specific application domains. In this paper, we present the results of latest investigations to improving our approach to framework instantiation. In particular, we discuss a process language that have been developed to guide framework instantiation explicitly, and the most recent updates that have made to improve the language expressive-ness. Furthermore, we present a case study used to evaluate our approach and to identify current and future extensions.

Keywords: Frameworks, instantiation, software process, software, design, transformation, crosscutting concerns, aspect oriented, generative approach.

1. Introduction

Over the last years, generative programming and aspect-oriented software develop-ment have been proposed aiming at increasing maintainability and reusability of software systems. While several research works have focused on the investigation of the individual use of each of these software engineering approaches, less attention has been paid to the integration of these two techniques. Aspect-Oriented Software Development (AOSD) is an evolving approach to modularize crosscutting concern that existing paradigms (e.g.:
object-oriented) are not able to capture explicitly. Crosscutting concerns are concerns that often crosscut several modules in a software system. AOSD encourages modular descriptions of complex software by providing support for cleanly separating the basic system functionality from its crosscutting concerns. Aspect is the abstraction used to modularize the crosscutting concerns. The use of aspect-oriented techniques in the definition of a generative approach can bring additional benefits for the development of system families, such as: (i) clear separation of orthogonal and crosscutting features in the problem and solution space; and (ii) direct mapping of crosscutting features in aspects. Despite these advantages, we believe that the integration of GP and AOSD techniques is not a trivial task. Interesting questions arise and need to be considered when developing an aspect-oriented generative approach, including How to model crosscutting features in the problem space? How to design aspect-oriented architectures that address the crosscutting and non-crosscutting features modeled? Which technologies (domain-specific languages, frameworks) are appropriate to implement these aspect-oriented generative approaches for framework instantiation?

Framework concepts have been successfully employed as important tools to achieve software re-use. At the same time they reduce developments efforts and increase the overall quality of produced software systems. ET++, MacApp, Hotdraw, MFC, just to name a few, are important examples of early frameworks that helped demonstrate the feasibility of a framework-centered development approach. They were able to capture common features successfully and represent the variability of a family of applications within a specific domain. There are now a large number of frameworks that have been developed for a variety of different purposes including CORBA (middleware for distributed systems), JADE (agent systems), Strut (web applications), JBoss-AS (enterprise applications), and JUnit (application testing). However, as frameworks become popular their weaknesses as well as their strengths are becoming apparent. In particular, framework instantiation is still a painful task because application developers must understand the intricate details surrounding the framework design. Thus, instantiation
of a specific application can often be a slow and costly process. For some frameworks such as the MFC it may take up to 12 months for an application developer to be highly productive. Thus, the instantiation process is a time-consuming activity which is counter to one of the most valuable properties of reuse, i.e., significant shortening in development time. Some approaches to alleviate the framework instantiation problems we mentioned have been proposed in the literature. However, they are normally either just a textual cross-referenced documentation of the instantiation activities or too closely to technology or specific application domains.

In this paper we initially give an overview of an approach to framework instantiation. We present RDL (Reuse Definition Language) a process language created to represent framework instantiation activities explicitly. RDL along with xFIT, our supporting instantiation tool, operates on UML models through transformations in order to produce valid application instances. Finally, we discuss a case study we conducted to assess how our framework improved framework instantiation.

The sections of this paper are organized as follows. Section 2 presents our approach to framework instantiation and its latest enhancements. Section 3 depicts the case study we carried out in order to assess our approach properly. In section 4 we discuss how our approach improved framework instantiation in the light of our case study evidences. Section 5 includes some related work, and finally Section 6 presents our conclusions and future work.

### 2. Understanding Framework Instantiation Problems

However, developers find there is still a steep learning curve when extracting the design rationale and understanding the framework documentation during framework instantiation. Thus, instantiation is a costly process in terms of time, people, and other resources. The framework specifications are usually unstructured or loosely structured and use natural language to describe the artifacts and processes. Even when frameworks have reasonably clear and detailed documentation and modelling, abstractions (e.g., UML) are used in the design, maintenance today is still performed in reality using source code. Instantiation tasks are not explicit, but remain as tacit
assumptions in the developers’ minds. In this context, there is a need for extended design abstractions that can allow framework instantiation to be explicitly represented. As a result, framework instantiation is currently a time-consuming activity, which negates one of the most valuable properties of reuse, that is a significant reduction in application development time. In fact, achieving high productivity with framework reuse can take months of effort. Another problem relates to the consistency of the final application design. Some instantiation processes introduce unexpected states that can violate some of the framework design constraints. Therefore, it is important that the framework documentation provides a set of properties (structural and behavioural) that must be preserved after each instantiation process is performed. In summary, design modeling notations need to be extended and the instantiation tasks explicitly represented at the design level and, in addition, there is a need for methods that allow validation of these tasks.

3. An Aspect-Oriented Generative Approach

The aspect-oriented (AO) generative approach aims at exploring the horizontal domain of multi-agent systems (MASs) to improve their quality and productivity.

The purpose of the generative approach is threefold: (i) to uniformly support cross-cutting and orthogonal (noncrosscutting) features of software agents starting at early development stages; (ii) to abstract the common and variable features; and (iii) to enable the code generation of AO agent architectures.

Figure 1 depicts our generative approach that is composed of:

(i) a domain-specific language (DSL), called Agent-DSL, used to collect and model orthogonal and crosscutting features of software agents;
(ii) an AO architecture modeling a family of software agents. It is centered on the definition of aspectual components to modularize the crosscutting agent features;
(iii) a code generator that maps abstractions of the Agent-DSL to
specific compositions of objects and aspects in agent architectures.
The definition of our generative approach encompassed a typical
domain engineering process. The steps followed in the development of the
generative approach were:
1. Domain Analysis
   a. Definition of the domain
   b. Identification and modeling of common and variable features of the
domain
   c. Identification and modeling of the crosscuting features of the domain
2. Domain Design
   a. Specification of the generic AO architecture
   b. Identification and specification of the DSLs
   c. Specification of the configuration knowledge
3. Domain Implementation
   a. Implementation of the DSLs
   b. Implementation of the AO architecture and additional components
   c. Implementation of the code generator

4. Our Approach to Framework Instantiation
A typical framework adaptation has two phases: i) understanding the
overall rationale behind the framework design; ii) extending the framework
flexible points according to specific requirements in order to produce
application specific increments (ASI).
As we have mentioned, the first phase has been supported by some
framework documentation approaches. Basically, they describe the purpose of
the framework, its major design elements, their relationships, how the
flexible points can be adapted to produce applications, and provide
some examples. For example, in the cookbook approach, recipes are used to
explain how a certain extension point can be adapted. Recipes can reference
each other, thus helping application developers to understand better how
the hotspots (and the design elements they represent) are interrelated.
Our approach complements framework documentation techniques, in
particular cookbooks. It closes the gap left by purely text-based approaches by
providing means to represent instantiation activities explicitly. Our
approach consists of a process language, RDL, that allow framework
developers to represent adaptation steps, and a supporting tool, xFIT, that
operates on UML models by transforming a framework’s class
diagrams into application class diagrams based on application
developer’s inputs. Figure 1 below depicts our approach.

![Diagram](image)

Figure 1: Overview of Our Approach

The steps required to instantiate a framework using our approach consists of:

- The **framework developer** provides a framework class diagram conforming with the XMI format (an XML file representing the model);
- The **framework developer** provides an RDL script containing the framework instantiations steps;
- The **application developer** runs xFIT providing it with the RDL script and the framework UML class diagram. Likewise, the application developer provides feedback according to specific application requirements.
- At the end of the generation process xFIT will run validation tasks and report the errors encountered (if some exist). Otherwise, a UML class diagram is produced including the framework and the specific application instance classes.
- The **application developer** can then use a Case tool to open the application model and generate stubs for the classes produced. By filling out the stubs with appropriate code the process is ended.

### 4.1 The Process Language (RDL)

RDL is a process language that aims at providing mechanisms for framework developers to represent instantiation tasks explicitly. RDL is programming-language and framework-domain independent and manipulates design elements expressed in UML. RDL abstractions have been proposed based on the cookbook approach and exploit the use of design patterns. In the next sections we describe the main construct of the RDL and the latest enhancements we have made to increase the language expressiveness. Since it is not the purpose of this paper...
to be an RDL reference manual we suggest reading for a more detailed description of the language.

**RDL Main Structure** - RDL higher level constructs are represented by cookbooks, recipes and patterns. An RDL cookbook contains a set of RDL recipes. RDL recipes embody instantiation tasks related to particular variable aspect of a framework architecture. RDL Patterns describe recurring instantiation steps encountered during a framework adaptation (e.g. design patterns).

RDL can be used to produce two types of artefacts: RDL scripts and Pattern Libraries. A general structure of an RDL script is shown in Table 1.

```
COOKBOOK myCookBook
    RECIPE main
    ...
    CALL_RECIPE( R1, (...) );
    ...
    END_RECIPE
    RECIPE R1(…)
    ...
    END_RECIPE
    ...
    END_COOKBOOK
```

Table 1: General structure-RDL Script

In an RDL script, at least one recipe must be named `main` representing the cookbook start point. Recipes can call each other, receive parameters and return values in a way similar to functions in procedural languages.

RDL Pattern Libraries describe instantiation patterns, i.e., recurring instantiation tasks. Since some design patterns exhibit an abstract and a concrete part (e.g. Template Method, Abstract Factory, and Strategy) they can be properly used to expose framework hotspots. Therefore, design pattern instances can be represented as RDL Patterns. Pattern Libraries represent an enhancement we have made to our approach. The general structure of an RDL Pattern Library can be found in Table 2.

```
PATTERN_LIBRARY
    myPatternLibraryName
    PATTERN Pattern1(…)
    ...
    END_PATTERN
    PATTERN Pattern2(…)
    ...
    END_PATTERN
    ...
    END_PATTERN_LIBRARY
```

Table 2: General structure-RDL Pattern Library

Pattern libraries are normally stored in files with the `.rdp` extension (reuse definition pattern). UML class models are expected to conform to the XMI
format and are stored in .xmi files. In the current version of RDL only one RDL script (.rdl) is allowed to specify the instantiation steps of a framework. There is still no way to import and combine RDL scripts. This has been left as a possible enhancement for future versions of our language.

**RDL Types**- In order to keep the syntax of the language simple, the previous versions of RDL did not considered data types explicitly. However, as we used the language in practical situations the need for a strong typed-language became apparent. Thus, the types now encountered in RDL are basically those found in UML class diagrams plus some additional ones to represent strings, numbers, booleans and list of types (Table 3). Each type has a set of associated operations and attributes that allow framework developers (RDL script users) to make proper references to model elements.

<table>
<thead>
<tr>
<th>RDL Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRING</td>
<td>Represent Strings in RDL</td>
</tr>
</tbody>
</table>

**Table 3: RDL Types**

RDL commands fall into three categories: Basic, Instantiation, and Pattern Commands. Following we discuss each one of the categories.

**RDL Basic Commands**- The basic commands provide low-level facilities to manipulate the framework design elements. For instance, new classes, methods or attributes can be created and added to UML class diagram models. Table 4 below illustrates some of the RDL basic commands.

<table>
<thead>
<tr>
<th>Description</th>
<th>Basic Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class creation</td>
<td>NEW_CLASS(…)</td>
</tr>
<tr>
<td>Method creation</td>
<td>NEW_METHOD(… )</td>
</tr>
<tr>
<td>Attribute creation</td>
<td>NEW_ATTRIBUTE( …)</td>
</tr>
<tr>
<td>Inheritance</td>
<td>NEW_INHERITANCE(…)</td>
</tr>
</tbody>
</table>
Selection | IF (e) ... [ELSE ...] 
---|---
END_IF

Repetition | LOOP (e) ... 
---|---
END_LOOP

Assignment | Var = expression

Table 4: Main RDL Basic Commands

**Instantiation Commands** - RDL Instantiation Commands increase the level of abstraction by combining basic commands into single tasks. Basically, RDL Instantiation Commands represent object-oriented reuse activities such as extending a class, overriding a method, and assigning a value to a class attribute. Table 5 depicts the main Instantiation Commands.

<table>
<thead>
<tr>
<th>Description</th>
<th>Instantiation Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Extension</td>
<td>CLASS_EXTENSION(…)</td>
</tr>
<tr>
<td>Method Extension</td>
<td>METHOD_EXTENSION(…)</td>
</tr>
<tr>
<td>Value Assignment</td>
<td>VALUE_ASSIGNMENT(…)</td>
</tr>
<tr>
<td>Value Selection</td>
<td>VALUE_SELECTION(…)</td>
</tr>
</tbody>
</table>

Table 5: Main RDL Instantiation Commands

**Pattern Commands** - The highest level statements in RDL are represented by Pattern Commands. Pattern Commands allow framework developers to reuse a set of recurring instantiation activities previously specified. In previous versions of RDL, Patterns Commands were represented by the *Pattern Class Extension* and *Pattern Method Extension* commands. These commands required specific types to be passed as input parameters in order to be used properly. We decided to simplify the language support for patterns by defining a single command for an RDL Pattern call. No parameters are required and it is up to the framework developer to define how patterns will be properly described. Table 6 describes the RDL command to call an RDL Pattern.

<table>
<thead>
<tr>
<th>Description</th>
<th>Pattern Command</th>
</tr>
</thead>
</table>
| Pattern call command | CALL_PATTERN(…)

Table 6: RDL Pattern Call Command

In the following we illustrate the implementation of an RDL Pattern Library including an implementation for the *Factory Method* design pattern, and an RDL Script benefiting from the RDL Pattern implementation.

**PATTERN LIBRARY**

GammaPatterns

```
PATTERN __FactoryMethod( IN absCreatorName : STRING , IN facMethodName : STRING , INOUT concreteCreatorClass : CLASS )
-- Create Concrete Creator
```
IF (concreteCreatorClass = NIL)  
concreteCreatorClass =  
CLASS_EXTENSION(  
absCreatorName, ? );  
END_IF  

-- Extends Creator Factory Method  
METHOD_EXTENSION(  
absCreatorName, facMethodName,  
concreteCreatorClass );  
END_PATTERN  
END_PATTERN_LIBRARY  

COOKBOOK myCookBook  
RECIPE main  
conCreator : CLASS;  
CALL_PATTERN(  
FactoryMethod, (  
“AbstractView”,  
”createAlarm”, conCreator));  
END_RECIPE  
END_COOKBOOK  

4.2 The Framework Instantiation Tool (xFIT)  

Our approach is supported by an instantiation tool known as xFIT (Framework Instantiation Tool). xFIT provides a runtime environment for RDL scripts. The framework class diagram and an RDL script are taken as inputs and based on application developers feedback xFIT generates the application instance class diagram. xFIT performs validation tasks over the design elements produced to ensure that its structure is regular and well-formed. As an example, xFIT certifies that all abstract classes and methods have been resolved in the final design since all the hotspots must have been handled.

5. CASE STUDY  

We developed an initial case study to evaluate the feasibility of our approach. The idea was to produce an application instance through transformations over an AO framework design. We developed a Drawing Editor framework in AspectJ which exposed 6 hotspots(CSG Drawing Editor Framework). The Observer design pattern was used allowing Figure objects(Subjects) to notify registered Observers about size changes (e.g., zoom in/out). The Observer, in our case represented by the Display class would handle the notifying events by repainting the drawing in the appropriate canvas window. We used an aspect version of the Observer design pattern that exposes 2 hotspots i) the Display (Observer) reaction to Figure’s (Subjects) resizing, realized by an aspect method extension, and ii)
the Figure object’s that should notify the Display about state changes (e.g., calls to the resize() method of Figure objects), realized by a pointcut extension. We represented the instantiation tasks in an AF-RDL script, mapped some AF-RDL commands to XQuery user-defined functions and performed the corresponding XQuery transformations. At the end we obtained a serialized XMI-like model representing our application instance design. Aspects, pointcuts, aspect methods and advices were represented by specific model elements. The case study showed that the idea of mapping high-level AF-RDL commands to XQuery functions is feasible. Indeed, XQuery turned out to be a powerful language for transformation. Therefore, the combination of AF-RDL and XQuery was seen as very positive and promising. The next steps in our research include i) enhancing the AF-UML expressiveness by defining new models for AO frameworks, ii) specifying a corresponding XMI-compliant description to all model elements in AF-UML, iii) mapping all RDL instantiation commands to related XQuery functions, and iv) developing a tool to manipulate AF-UML models.

6. Conclusion

This paper reported our experience in the definition of an AO generative approach. The goal of this approach is to explore the horizontal domain that MASs represent in order to enable the code generation of agent architectures. During the development process of the generative approach, it was necessary to adapt modeling notations used in generative programming due to the adoption of AOSD. Also, a new notation was proposed to support the representation of AO architectures (section 3). Aspectual components have been used to model crosscutting features from the architectural point of view. We believe that the definition of AO generative approaches can bring important benefits to the development of software families. GP allows: (i) to evolve the problem and solution spaces independently; and (ii) to define clearly the mapping between high-level features and implementation components. The integrated use of GP and AOSD techniques brings additional benefits, such as: (i) clear separation of orthogonal and crosscutting features starting at early design phases; and (ii)
direct mapping of crosscutting features in aspectual components. This latter benefit simplifies the implementation of code generators, because the composition of crosscutting concerns is accomplished by the aspect weavers. Using only OO abstractions, crosscutting agent features need to be hand-coded in the code of classes. This work aimed at identifying relevant techniques and requirements to be considered on the development of AO generative approaches. It represents a significant step in the definition of a method to develop AO generative approaches.

we also presented our approach to framework instantiation, it shows the enhancements that have been made to improve its effectiveness, and a case study to evaluate our approach. Our case study showed that the use of our approach was helpful in raising the level of abstraction of framework adaptation processes.

7. References
Semi-Automatic Generation of Transformation Rules in Model Driven Engineering: The Challenge and First Steps

Slimane Hammoudi, Samyr Vale

ESEO, Ecole Supérieure de l’Ouest
4, Rue Merlet de la Boulaye BP 926,
49009 Angers cedex 01
France
slimane.hammoudi@eseo.fr; samyr.vale@eseo.fr;

ABSTRACT: Recently, Model Driven Engineering (MDE) approaches have been proposed for supporting the development, maintenance and evolution of software systems. Model driven architecture (MDA) from OMG (Object Management Group), “Software Factories” from Microsoft and the Eclipse Modelling Framework (EMF) from IBM are among the most representative MDE approaches. Nowadays, it is well recognized that model transformations are at the heart of these approaches and represent as a consequence one of the most important operations in MDE. However, despite the multitude of model transformation languages proposals emerging from university and industry, these transformations are often created manually, generally a fastidious and error-prone task, and therefore expensive process. We argue that, the semi-automatic generation of transformation rules is an important challenge in future MDE development. In this paper we propose an extended architecture that aims to semi-automate the process of transformation in the context of MDA. This architecture introduces mapping and matching as first class entities in the transformation process, represented by models and metamodels. We will introduce and discuss briefly two main operations “adaptation” and “derivation” which we consider as core techniques for a semi-automatic transformation process in MDA, along with the first two main techniques of matching and mapping. Finally, we will present a classification of the major approaches to matching in the literature and situate them in the context of MDE metamodel matching. These matching techniques are the centerpiece of a semi-automatic transformation process.

KEYWORDS: model driven architecture, transformation language, mapping metamodel, matching metamodel, semi-automatic transformation, transformation architecture and matching techniques.

1. Introduction

The main motivation behind model driven engineering (MDE) is to transfer the focus of work from programming to modeling by treating models as first class entities and consequently the primary artifacts of development. There are nowadays several approaches based on MDE principles, the most well known being MDA [20] by OMG, “Software factories” by Microsoft [10] and the Eclipse Modelling Framework (EMF) from IBM [7]. In the literature, several issues around MDE have been studied and subject of intensive research, e.g. modeling languages [3] [5], model transformation languages [14] [21], mapping between metamodels [11], and design methodologies [1]. Among these issues, model transformation languages occupy a central place and allow to define how a set of elements from a source model are analyzed and transformed into a set of elements of a target model. However, these transformations are created manually, often a fastidious and error-prone task, and therefore an expensive process. These transformations consist of creating a set of rules involving, and in the same time merging mapping and transformation techniques between two metamodels. A semi-automation of the transformation process leads to a real challenge allowing many advantages: It enhances significantly the development time of transformation and decreases the errors that may occur in a manual definition of transformations. In [17], we have initiated a first attempt towards this semi-automation. We have introduced an approach separating mapping specification from transformation definition, and have implemented this approach in a tool called MMT (Mapping Modeling Tool). In this first approach, a mapping specification was created manually to define the relationships between metamodels (i.e. equivalent metamodel elements), while
transformation definition was generated automatically and contained the operational
description of the transformation rules between models. In this work, we propose to push the
semi-automation process one step further by considering matching techniques [15] [22], to
provide semi-automatic mappings between two metamodels. The produced mappings could
be adapted and validated by an expert for the automatic derivation of a transformation model,
as a set of transformation rules. In this paper, we present an extended architecture of the
transformation process in the context of MDA. This architecture introduces the matching and
mapping components as two other important operations in the transformation process. We
will introduce and discuss briefly two main operations “adaptation” and “derivation” which
we consider as core techniques for a semi-automatic transformation process in MDA.

This paper is organized as follows: section 2 introduces the core concepts of model
transformation in MDA and point out the main problems of the transformation process.
Section 3 presents an extended architecture for a semi-automatic transformation process and
discusses the matching and mapping metamodels as two important components in this
process. Section 4 reviews the matching techniques that have been proposed in the literature
and situates metamodel matching in the context of MDE. Finally, section 5 concludes our
work and presents some final remarks and perspectives.

2 Model Transformations: Core concepts and main problems

It is well recognized today that model transformation is one of the most important operations
in MDA. In our discussions here we are concerned with a transformation that takes a
platform-independent model and transforms it into a platform-specific model. In the context
of the basic four levels Metamodeling architecture of MDA [20], various scenarios of model-
to-model transformation have been identified. Figure 2 presents the most common scenario
of these transformations, which is compatible with the MOF2.0/QVT standard [21]. Each
element presented in Figure 2 plays an important role in MDA. In our approach, MOF is the
well-established metatemplate used to create metamodels. Transformation rules specify
how to generate a target model (i.e. PSM) from a source model (i.e. PIM). To transform a
given model into another model, the transformation rules map the source into the target
metamodel.

![Figure 1. Model Transformation in MDA: from PIMs to PSMs](image)

The transformation rules are based on the transformation language, such as the standard
QVT. The transformation engine takes the source model, executes the transformation rules,
and produces the target model as output. We point out two main problems concerning this
main scenario of the MDA transformation process illustrated by figure 2:
The first problem concerns the creation of “transformation rules” between metamodels which, as mentioned in the introduction, are often created manually using a transformation language, generally a fastidious and error-prone task, and therefore expensive process [8].

The second problem concerns the specification of these “transformation rules”, which merge together techniques of mappings and transformations without explicit distinction between them. That is to say, the specification of correspondences between elements of two metamodels and the transformation between them are grouped in the same component at the same level. As discussed in [12], an explicit distinction between techniques of mapping and transformation could be very helpful in the whole MDA process of transformation. Moreover, the separation between the mappings and transformations parts is a first step towards a semi-automatic process, since mappings could be discovered and generated by a matching process.

3 An extended architecture for the transformation process

Figure 3 illustrates our proposal of an extended architecture for the transformation process in MDA, allowing a semi-automatic generation of transformation rules between two metamodels, and the semi-automatic generation of a target model from a source model. The first three main operations of our approach are: Matching, Mapping and Transformation. All the components linked to these operations, and their relationships, are presented in figure 3 based on the four level MDA metamodeling architecture.

The matching operation [2] [13] is the process that produces the potential mappings between two metamodels. Generally, this task implies a search of equivalent or similar elements between two metamodels. In the database domain, this task is called schema matching. In our context, a matching model (Matching M) takes two metamodels designed by source and target (representing respectively a PIM and a PSM metamodel), and produces a mapping model (Mapping M). The matching model conforms to a metamodel of matching (Matching MM) which implements techniques that consist of finding semantically equivalent modeling concepts between two metamodels. Thus, different kinds of relationships between metamodel elements are discovered using the metamodel of matching.

The relationships between metamodel elements are saved in a mapping model which conforms to a mapping metamodel (Mapping MM). This metamodel defines the different kinds of links (relationships) that could be generated by the matching model. Each kind of link corresponds to one transformation pattern specified in the transformation model described hereafter. Given that no generic matching solution exists for different metamodels and application domains, it is recommended to give the human expert the possibility to check the obtained mappings, and, if necessary, update or adapt it. This is one of the steps in the whole process, in which the expert intervenes to complete and validate the obtained results.

Finally, a transformation model (Transformation M), in conformance to its transformation metamodel (Transformation MM), is derived automatically from a mapping model. A transformation model is basically represented by a set of rules that states how elements from source metamodel are transformed into elements of target metamodel. These rules are expressed in a transformation language based on MDA standards (OCL, MOF). This language, such as the standard QVT is described by a metamodel as a general formalism and abstract syntax for model transformation in MDA. Frequently, the transformation model is completed by some information such as those concerning the execution environment, and produces a transformation program ready for the execution. This last part is often achieved by a designer (or software engineer) who implements a business model in a specific platform. Finally, a transformation engine takes a source model as input, and executes the transformation program to transform this source model into the target model.
Figure 3. Architecture for a semi-automatic transformation process in MDA.

The first goal with this architecture is to introduce the matching process into the OMG’s MDA approach in order to increase the degree of automation of the transformation process. This leads to a reduction in manual human tasks often fastidious and error-prone, by the rational choice among the plethora of existing works on matching techniques. These techniques are suitable for the problem of automatic mapping production. Thus, from a software point of view, the transformation process involves three main programs which are at the heart of a semi-automatic development:

- **Matching program**: implements the matching metamodel and produces a first version of a mapping model between two metamodels source and target. This mapping model is adapted and validated by an expert user.
- **Generation program**: takes a mapping model validated by an expert, and derives automatically a transformation model (program) as a set of rules.
- **Transformation program**: takes a source model defined by a designers or engineers and produces an equivalent target model on a specific platform.

Two important operations adaptation and derivation allow to link and complete these main programs in the whole process of transformation. Adaptation is the responsibility of the expert user who should accept, discard or modify the obtained mappings, furthermore, to specify the correspondences which the matcher was unable to find. Loosely speaking, the mapping and matching techniques (models) could be defined with the following intuitive formula:

\[
\text{Mapping} = \text{Matching} + \text{Adaptation}
\]

The mapping model obtained in the previous step after adaptation by the expert user should be completely defined allowing an automatic generation of transformation model. This operation is called derivation and, in the same way as above, transformation and mapping models can be defined with the following intuitive formula:

\[
\text{Transformation} = \text{Mapping} + \text{Derivation}
\]
4 Matching techniques for metamodels in MDE

Matching between metamodels are the centerpieces for a semi-automatic transformation process in MDE and MDA in particular. Matching techniques have been studied in various research domains, including digital libraries, ontologies, agent matchmaking, schema integration and evolution in databases ([13], [22]). In the context of MDE, we can find very few works in the literature that address the problem of metamodels matching. To the best of our knowledge, [15] is the only work that investigate metamodels matching. They propose the application of ontology and schema matching techniques for automatically exploring semantic correspondences between metamodels. Moreover, in this work they introduce a lifting process, which allows to create ontologies from metamodels, and after they apply techniques of matching between ontologies. Schemas in the context of databases and metamodels in our context of MDE are closely related, hence, we propose to review the different approaches of schema matching, and after we situate these approaches in our context of metamodeling matching.

4.1 Classification of schema matching approaches

In the literature, several schema matching approaches have been proposed [13] [22]. Each schema matching approach has its own characteristics that were grouped in a taxonomy illustrated bellow in figure 3 [9] [18]. In addition, each approach has been evaluated through match quality measures discussed in the next section 4.2.

- Individual matcher approaches use only one matching criterion. They are classified in:
  - **Schema-only based**, when they consider only metamodels. They can be classified in:
    - **Element level**, the mapping is realized for each individual element. It can be classified in linguistic and constraint-based. Linguistic are based on name similarity, description, global namespace, while constraint-based are based on type similarity and key properties.
    - **Structure-level**, the mapping is realized considering the combinations of elements related in a structure. It is only classified in constraint-based that use graph matching.
  - **Instance/contents-based**, when they consider only instances (or models). It can also be classified in element-level. This last can be classified in linguistic and constraint-based. In

![Figure 3. Classification of schema matching approaches.](image-url)
this case, linguistic is based on word frequencies and key terms present in the element instances, while constraint-based is based on value pattern and ranges of the element instances.

• Combining matchers use multiple matching criteria. They can be classified in:
  – Hybrid, they combine multiple approaches to create only one matcher in order to produce a result, i.e. the creation of mapping between the elements.
  – Composite, they combine many results obtained from different approaches in order to produce the mapping between elements. This combination of results can be manual or automatic.

4.2 Matching quality Measure

The interrelationships between metamodels can be organized in sets which can be manually or automatically created. A set created manually can contain all needed matches (i.e. matched elements); while a set created automatically can contain valid and non valid matches. The first set is denominated real matches, and the later derived matches (cf. Figure 4).

Figure 4. Comparing real matches and automatically derived matches.

In addition, other subsets are defined as follows [13] [18]:

• A (false negatives) are matches needed but not automatically identified.
• B (true positives) are matches which are needed and have also been correctly matched by the automatic match operation.
• C (false positives) are matches falsely proposed by the automatic match operation.
• D (true negatives) are false matches which have also been correctly discarded by the automatic match operation.

Based on the cardinalities of these sets, the following match quality measures are provided as parameters for benchmarks:

\[
\text{Precision} = \frac{|B|}{|B| + |C|} \]

reflects the share of real correspondences among all found ones.

\[
\text{Recall} = \frac{|B|}{|A| + |B|} \]

specifies the share of real correspondences that are found.

\[
\text{F-Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

\[
\text{Overall} = \text{Recall} \times (1 - \frac{1}{\text{Precision}})
\]

All these measures were developed specifically in the schema matching context [9] [18]. We can notice that F-Measure represents the harmonic mean of Precision and Recall. The main underlying idea of Overall is to quantify the post-match effort needed for adding missed matches and removing false ones.
4.3 From schema matching to metamodel matching

In our MDE context with respect to our extended architecture of figure 3, metamodel matching results in a mapping model that must be conform to a mapping metamodel. In [16] [17], an initial mapping metamodel was proposed and implemented in a tool called MMT. According to model management algebra [2], a mapping is generated using an operator called match which takes two metamodels as input and returns a mapping between them. We have adapted this operator as follows: given two metamodels $Ma$ and $Mb$, and $C_{Ma \rightarrow Mb}$, the mapping model (a set of correspondences) which conforms to the mapping metamodel $MC$. The operator match could be defined formally as:

$$\text{Match} (Ma, Mb) = C_{Ma \rightarrow Mb} / MC.$$

In general, metamodels are created with a specific purpose and by different groups of persons. Each purpose is determined in function of the domain, and each group of persons models a system in different ways. In the modeling task, each group abstracts, classifies and generalizes the reality based on its own knowledge. Consequently, metamodels that were created in the same context and by different groups may have different structure and terminology causing the semantic distance among them [18]. According to our approach, a model can be transformed in another model, only if the metamodel of the former can be mapped in the metamodel of the later. In order to map metamodels, the equivalent or similar elements must be identified, and the semantic distance should be minimized. The notion of semantic distance was developed to cover the notion of “how close is close enough”. A dual for “semantic distance” is schema similarity that is defined as “the ratio between the number of matching elements and the number of all elements from both input schemas” [22]:

$$SS = \frac{Nm}{Nt},$$

where $SS$ is the schema similarity, $Nm$ is the number of matching elements and $Nt$ is the number of all elements). Semantic distance can also be quantified as a numeral value (like schema similarity) or as a subset of a metamodel. By the way, according to the MDA manifesto [6], “one of the primary purposes of automation in MDA is to bridge the semantic gap between domain concepts and implementation technology by explicitly modeling both domain and technology choices in frameworks and then exploiting the knowledge built into a particular application framework”. Moreover, automation, which is the main objective of this work in the context of model transformation, is one of the basic tenets of MDA manifesto [6].

5. Conclusion and Future Work

In this paper, we have presented a first approach for a semi automatic transformation process in MDA using an extended architecture. We argue that a semi-automatic transformation process will be a great challenge in MDA as there is not yet a complete solution that automates the development of model transformation. A semi-automatic process will bring many advantages: it accelerates the development time of transformations; it reduces the errors that may occur in manual coding; and it increases the quality of the final transformation code. The key principle for this process is to consider mapping and matching metamodels as first class entities in MDA. In our previous work [18] [19], we have proposed a first algorithm for metamodel matching based on set theory. In future work, we will propose a methodology for a semi-automatic transformation process. This methodology will enforce our architecture and details the different steps in the semi-automatic process, and the main MDA users involved in this process. We will also implement our proposed matching algorithm in the context of our methodology and we will investigate from the obtained mappings, the adaptation and derivation techniques discussed in section 3.
6. Bibliography


Use of Simulation in Systems Engineering

Hassan Badkoobehi, Ph.D.
National University, San Diego

Abstract

As used in systems engineering, the word simulation refers to the construction of a simplified representation of a process or system in another to facilitate its analysis. Such a representation or model may be quantitative or qualitative. In either case it is characterized by the fact that it does not include all the features and characteristics of the original system or process. Rather the purpose of the simulation is to show the effect of particular factors which are being investigated. The four major areas I will discuss in my paper would be:

1. The Development of Models
2. Stochastic and Deterministic Models
3. The Uses of Simulation
4. Simulation Equipment

All the types as described above may be referred to as simulation, and will be so considered in this paper.

In this paper, various aspects of the simulation method have been presented, the general characteristics with analog, digital, and hybrid techniques has been discussed. The importance of simulation in systems engineering arises from the fact that simulation makes possible the verification of proposed designs before completion of system development, thus resulting in invaluable saving in time and money. The major considerations involved in the use of simulation has been discussed in considerable detail. It has been pointed out that simulation is always partial and requires the selection and isolation of significant variables. Consequently the results of a particular simulation are generally applicable only to particular aspects of a system design. Simulation is a tool used in the verification of design hypothesis as well as being a source of new ideas, new designs and new hypothesis. It forms an invaluable link in the process of systems engineering.

The Nature of Simulation

The word simulation refers to the construction of a simplified representation of a process or system in order to facilitate its analysis. Such a representation that simulation does not
include all the features and characteristics of the original system or process, but it shows the effect of particular factors which are being investigated.

**The Development of Models**

The process of constructing a simulation of a physical system thus involves one or more abstractions from the real world. These abstractions may be of varying degrees of severity. For example, the construction of a scale model of the physical system, a procedure common in hydraulics and the chemical industry. While there are important problems of scaling in construction of such a model, in principle or scale model of a harbor, for example represents a relatively small departure from the real world.

**Stochastic and Deterministic Models**

The construction of a model is based on information obtained from the physical world by observation or measurement. Consequently, measurement errors will result in erroneous models. One of the serious problems in the simulation of a process is the selection of those random elements which one desires to incorporate into the model. Many models are constructed on a purely deterministic basis with the understanding that the results obtained from the models may represent statistical averages of certain variables in the physical system.

**The Uses of Simulation**

Simulation in its various forms is of great importance in systems engineering. Some of the important applications are listed below:

1. **Design evaluation** - a simulated system may be used to evaluate the validity of the preliminary design of a portion of a system.

2. **Interrelationships among the parts** - one of the key uses of simulation is in the evaluation of the effect of various portions or subsystems of a system upon each other and upon the performance of the system as a whole.

3. **Costs** - An important part of simulation is in the attempt to reduce overall costs by the evaluation of alternative designs by means of simulation.

4. **Study of failure sources** - without some type of simulation, it is extremely difficult in many cases to determine possible sources of airplane crashes or controller instability.

5. **Hypothesis testing** - simulation makes possible experiments which in the physical system may be difficult or impossible to test.

Many other uses and applications of simulation can be listed.
• Determination of problem area
• Determination of significant and insignificant variables
• Study of the effect of environmental variation upon performance.

Simulation Equipment

Simulation equipment may be classified in several ways.
• Analog simulation
• Digital simulation
• Physical simulation
• Mathematical simulation
• Manned system simulation
• The computers and hybrid analog - digital simulation analog simulation

When a simulation is characterized primarily by continuous signals or ‘elements it is referred to as analog simulation. When a substantial portion of a simulation involves discrete signals, elements, or processes, it is referred to as hybrid analog - digital simulation, analog simulations can take both the forms outlined above.

Digital Simulation

When simulation consists of the manipulation of phenomena which occur with discrete values, it is referred to as digital simulation. Digital simulation lends itself to the study of systems wherein there are many decision functions to be implemented, and to phenomena which are more easily characterized by a word description than by a set of conventional mathematical equations. Digital simulation is usually carried out on a general-purpose digital computer. The process of preparing the simulation model for a digital computer follows many of the general rules which arise in programming any problem for the digital computer.

Physical Simulation

Simulation may involve the study of a physical system by means of any analogous system whose behavior closely approximates that the system under study for the particular phenomena being investigated through forms of such simulation are used:

• Scale models are used in hydraulics and in the process industries.
• Analogous models are used in heat or gaseous flow.
• Partial-system tests used in the interconnection of an element of a physical system with a general-purpose computer, which represents a mathematical analog of the system.

Mathematical simulation represents a further level of abstraction that physical simulation.

A mathematical simulation of the system to be studied must be formulated. The resulting equations are solved by means of analog computer events which perform specific.
operations such as summation, integration or multiplication. The physical variable in the computer element is generally either an electrical voltage or a shaft rotation. The simulation can be performed using a general-purpose electronic analog computer, or it may be performed using special purpose elements.

**Comparison of Physical and Mathematical Simulation**

<table>
<thead>
<tr>
<th>Mathematical Simulation</th>
<th>Physical Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Easy parameter variations</td>
<td>* Parameter variation may be difficult</td>
</tr>
<tr>
<td>* Mathematical description required of all system elements</td>
<td>* Mathematical description NOT required of all system elements</td>
</tr>
<tr>
<td>* Time scale can be varied by selection of computer components</td>
<td>* Generally designed for a fixed time scale</td>
</tr>
<tr>
<td>* Well suited to fast-time simulation</td>
<td>* Well suited to real-time simulation with human operators</td>
</tr>
<tr>
<td>* Results affected by selection of model and quality of computer components</td>
<td>* Results affected by selection of model and validity of analog</td>
</tr>
<tr>
<td>* Possibility of false solutions due to the characteristics of the equations themselves</td>
<td>* No such possibility</td>
</tr>
</tbody>
</table>

Model for each major system element must be formulated. In cases where such models are not well known or where the performance of particular subsystems is difficult to describe mathematically because of complexity, or where statistical variability of special types is important, it may be necessary to include elements physically in the simulation.

**Manned Systems Simulation**

One of the most important areas of application of the simulation method is in the study of original conceptual design and the final manufacture.

In this paper various aspects of the simulation method have been presented. The general characteristics of simulation have been outlined, and simulation with analog, digital, hybrid, and manned systems has been discussed. Since simulation is a tool used in the verification of design hypotheses as well as being a source of new ideas, new designs, and new hypotheses, it forms an invaluable link in the process of systems engineering.
References


Abstract:

The Internet has evolved in its current status starting from end to end principle as a backbone of design methodology. The end to end principle makes the core network simple and all the intelligence can be moved to network end points or hosts, making core network simplified and optimized for forwarding only. Content Delivery Network (CDN) can be regarded as an upcoming challenge to design a tightly integrated overlay which makes use of web caching, request routing, server load balancing and content aware services to optimize the content delivery. CDN is developed to overcome the performance problem of the Internet such as network congestion, server overload and low bandwidth and thus improves the service quality of Internet. A CDN system replicates/distributes content to a number of CDN nodes from the content provider (web servers) making an overlay network over the Internet. The content is provided to the user requesting for the web page from the closest CDN node. In this paper we first give the overview of CDN followed by a general architecture, a brief survey and comparison of few existing CDNs.

Keywords: Content Delivery Network, Replica server, CDN node, Distributed Nodes

1. Introduction

Content Delivery Networks (CDN) is an upcoming technology used to overcome the inherent limitations of Internet when accessing the web content. This technique improves the speed of accessing web content, maximizes bandwidth, reduces the load on the original server and maintains correctness through content replication and hence improves the internet performance. The word CDN was coined in 1998 and still a major thrust area of research in computer network. Research on the blistering areas of CDN like replica placement, request routing, resource optimization, content pricing, traffic congestion etc. are still in progress and it is believed that the successful solution to these problem will make the future CDN more popular. Most of the popular web sites in the internet use CDN system for the fast delivery of their embedded content. A number of commercial and academic CDN exists; some of them are discussed in this paper.

The rest of the paper is organized as follows. Section 2 describes the overview of CDN. In section 3 a general architecture of CDN is discussed. A brief survey of some of the existing CDNs is discussed in section 4. In section 5 we have compared some of the existing CDNs. Section 6 discusses a study on the performance of CDN. Finally in Section 7 we conclude our paper.

2. Overview of Content Delivery Network

A CDN is a collection of network elements arranged in internet for more effective delivery of data to end users. CDN uses the replica of the servers and places the replicas close to the clients to reduce latency and bandwidth consumption. The replica server store very selective set of content. CDN gives a lot of benefits to
the content providers. The content providers may be the web sites. The popular web sites use the CDN providers for the fast delivery of any digital content. The contents are replicated in the server according the user demand or it can be replicated in advance. The contents are also distributed among the various CDN nodes/replica servers created in the internet. The CDN provider delivers the content to the clients from the nodes in close proximity. This is done to maximize the hit ratio to 100%. CDN system does a number of tasks. It redirects the requests to the closest node to bypass congestion and delivers the content. It is responsible for content outsourcing and distribution services to replicate or cache content to distributed CDN nodes on behalf of its customer (content provider). It is also responsible for content negotiation services to meet specific needs of each individual user. It also manages the network components, handle accounting and monitor and report the content usage.

3. Architecture of CDN

3.1 The general architecture of CDN

The general architecture of a CDN system can have a number of components. Figure 1 shows the general architecture of CDN system. Three basic components are the end users, the content provider and the CDN provider. Internal 3 components (Request Routing System, Distributed System and Accounting System) of the CDN provider are also shown in the figure.

The End user requests for a web page. The content provider is the customer of the CDN provider that has the contents to be delivered to the end user. Here we have taken an example of xyz.com. The CDN provider is a proprietary organization or company that provides infrastructure facilities to content providers in order to deliver contents in a timely and reliable manner. Content provider updates the content in the CDN server. CDN system creates an overlay network over the internet. The system consists of a number of CDN nodes that are either the replica server or the nodes where the content is distributed. Replica server is
the server which holds the replica of the resources but it may act as an authoritative reference for client response. The request routing system of the CDN provider routes the request to the closest CDN node. The distribution system moves content to replica servers. This system also communicates with the request routing system through the feedback to assist in the replica server selection process for client requests. The accounting system aggregates and distills the accounting information into statistics and content detail records for use by the CDN server and billing organization. The figure also shows a typical communication between these components. (1) The end user request for the content of www.xyz.com. The request goes to the xyz.com server. (2a) xyz.com sends the index page to the end user and (2b) the xyz.com server simultaneously send the request to the content provider to provide the contents to the end user on its behalf. (3) The Content provider through the request routing system routes the request to a closest CDN node and that CDN nodes sends the embedded objects of the requested page.

4. Survey of existence CDNs

There are many commercial and educational CDN. Some of the popular existing CDNs are discussed in this section.

4.1 Akamai

This is one of the most successful and commercial CDN that hosts most of the popular websites of yahoo, monster, IBM etc. It also built its own DNS network for fast delivery of requested content by resolving the host name of the URL to IP address. This is developed by MIT. It is the market leader in content delivery. It owns more than 18000 servers over 1000 network in 70 countries.

4.2 SyncCast

SyncCast offers complete solutions from application development, Web hosting and Internet connectivity to deployment and systems integration. It provides solutions for delivering digital content and related data via the Internet and other media. Synccast uses load balancing equipment for load balancing client traffic. SyncCast is also a partner with large companies like Microsoft and Dell. SyncCast’s clients include the Motion Picture Association of America, Walmart Music, Lions Gate Films, Microsoft, EMI Music Group, Technicolor and Billboard Radio.

4.3 Globix

Globix has more than 1200 customers. Globix provide four types of services: Network Services, Hosting Services, Managed Services, and Media Services. Globix services are flexible, scalable, and cost-effective. Globix also provides Media services to capture, store, host and distribute media content from live event production, encoding, presentation tools, and traffic analysis. Globix IP backbone connects the customers to the Internet via a high-capacity network, fully owned and operated by Globix.

4.4 Accellion

Headquarter of this privately held company is California. It provides large scale file delivery service. It is a distributed file storage and transmission infrastructure for enterprise applications. Accellion also provides online desktop and server backup and recovery solutions. Accellion customers are industries such as advertising/Media production, manufacturing, healthcare, consumer goods, higher education etc.

4.4 Coral
Coral is a free, peer-to-peer academic content delivery network. When a user wants to use Coral, the content publisher has to append “.nyud.net:8090” to the hostname in a URL. Clients are redirected to the nearby Coral Web caches. Coral Web caches cooperate to transfer data from nearby peers whenever possible, minimizing both the load on the origin Web server and the latency perceived by the user.

4.5 Codeen

Codeen is also an academic CDN developed at Princeton University, USA. It provides caching of Web content and redirection of HTTP requests. In Codeen users set their internet caches to a nearby high bandwidth proxy that participates in the Codeen system. Then the requests to that proxy are sent to the system where the file is cached. The file is forwarded to the proxy and then to the client.

5. Comparison between the existing CDNs

In this section we have compared some of the existing CDNs on the basis of their type, organization, request routing technique, caching technique, content outsourcing and content type. Table 1(a) and 1(b) shows the comparison. CDN can be organized in two different ways: Overlay approach and Network approach. In the overlay approach, CDN nodes in the network handle the distribution of specific content type. In the network approach, networking devices like routers and switches are equipped with the code for identifying the specific application type and for forwarding the request based on predefined policies. Request routing techniques are of six types: Global Server Load Balancing, DNS-based request routing, HTTP redirection, URL rewriting, Any-casting and CDN peeri ng discussed in [2]. Caching techniques are of two types: Intra-cluster and inter cluster caching. This is also defined in [2]. Content outsourcing can be performed in two ways: Cooperative pull based and non-cooperative pull based approaches. In cooperative pull based approach, the end user requests are directed to the closest CDN node though DNS redirection. In non-cooperative pull based approach end user requests are directed either through DNS redirection or URL rewriting technique.

<table>
<thead>
<tr>
<th>CDN Name and Type</th>
<th>CDN Organization</th>
<th>Routing Technique</th>
<th>Caching Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akamai (Commercial)</td>
<td>Network and Overlay Approach</td>
<td>DNS-based request routing</td>
<td>Intra and Inter-cluster caching</td>
</tr>
<tr>
<td>SyncCast (Commercial)</td>
<td>Network Approach</td>
<td>Global Server Load Balancing</td>
<td>Intra-cluster caching</td>
</tr>
<tr>
<td>Globix (Commercial)</td>
<td>Network and Overlay approach</td>
<td>Global Server Load Balancing</td>
<td>Intra-cluster caching</td>
</tr>
<tr>
<td>Accellion (Commercial)</td>
<td>Network Approach</td>
<td>HTTP redirection</td>
<td>Inter-cluster caching</td>
</tr>
<tr>
<td>Coral (Academic)</td>
<td>Overlay Approach</td>
<td>DNS-based request routing</td>
<td>Intra and Inter-cluster caching</td>
</tr>
<tr>
<td>Codeen (Academic)</td>
<td>Overlay Approach</td>
<td>HTTP redirection</td>
<td>Intra and Inter-cluster caching</td>
</tr>
</tbody>
</table>

Table 1a CDN comparison

<table>
<thead>
<tr>
<th>CDN Name</th>
<th>Content Outsourcing</th>
<th>Content Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akamai</td>
<td>Non-cooperative pull-based</td>
<td>Static Content, Streaming media</td>
</tr>
<tr>
<td>CDN Provider</td>
<td>Network Model</td>
<td>Service Type</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>SyncCast</td>
<td>Non-cooperative pull-based</td>
<td>Streaming Media</td>
</tr>
<tr>
<td>Globix</td>
<td>Non-cooperative pull-based</td>
<td>Internet infrastructure and network services</td>
</tr>
<tr>
<td>Accellion</td>
<td>Non-cooperative pull-based</td>
<td>Large File transfer Services</td>
</tr>
<tr>
<td>Coral</td>
<td>Cooperative pull-based</td>
<td>Static content</td>
</tr>
<tr>
<td>Codeen</td>
<td>Cooperative pull-based</td>
<td>Static content</td>
</tr>
</tbody>
</table>

Table 1b CDN comparison

6. Performance Study

We have examined the performance of the CDN in our study. Performance of CDN can be measured in many ways. One piece of work on the performance found from the simulation of a specific web server log file that there is a diminishing return to placing more replicas so as to minimize total content delivery cost, and that a surprisingly small number of replicas (less then 5) is enough to significantly improve optimized performance gain (82%).[3]. It is also observed that placing replicas among a small number of carefully chosen candidates (top client receiving most content and a few client with the highest out degree) allowed us to achieve the same performance gain as placing replicas among all clients with reduced computation cost. To improve performance perceived by client, replicas are usually chosen from a large number of locations where client’s demands are highly aggregated. The performance can also be measured on the basis of number of requests offloaded from origin server, their impact of client perceived latency and their ability to efficiently load balance request amongst a set of CDN servers. Krishnamurthy et al. [12] examined briefly how content distribution servers improve latency when compared to throughput from the origin servers. In more recent work, on the performance of peer to peer swarming CDN system, an intensive simulation of basic P2P swarming CDN system under different critical parameters setting, the simulation results shows the P2P swarming system can perform well under high bandwidth connection, uniform startup distribution and appropriate block size etc. and the P2P swarm content delivery system can scale well with nodes size and content size [14].

7. Conclusion

To overcome the performance problem of the internet such as network congestion, server overload, low bandwidth that arise when accessing popular content by many users, CDN is developed. Caching and replication technique has improved the performance of CDN and hence of Internet in many ways. In this paper, we have discussed the general architecture of CDN, few existing commercial and existing CDN followed by a performance study of the system. With the emerging new form of Internet content and services such as video on demand that requires high bandwidth, CDN is becoming more popular and their customer is growing rapidly. Research on the blistering areas of CDN like replica placement, request routing, resource optimization, content pricing, traffic congestion etc. are still in progress and it is believed that the successful solution to these problem will make the future CDN more popular.
References


Secure Aroma Transfer

Vibhore Kumar Jain                   Subha Joshi         Chaman Kumar Bansal
Software Engineer                     Software Engineer            Software Engineer
SafeNet InfoTech          SafeNet InfoTech             SafeNet InfoTech
vibhore.b9@gmail.com        subha.joshi@gmail.com     chaman.bansal@gmail.com

Abstract - Have you ever wondered feeling the beautiful smell of Swiss Alps when you are sitting thousands of kilometers away in New York? This seems unreal but exploiting nature is what humans' are good at. Nature has given us many gifts including five senses. Till now Internet inarguably one of the strongest communication mediums at present, uses all senses Viz. Vision, Hearing and touching but two are missing viz. the power of smell and taste. In this paper, we will be venturing into how the smell will be transferred over the Internet and how to secure the system from unwanted smells/compounds. The main focus will be on standardizing a transfer process and controlling the smell transfer at end points.

No one would like to produce Phosgene, Mustard Gas etc at an endpoint that can kill a person or produce smell of burning rubber that can give user a bad experience. This paper will also discuss how to use an Odor firewall to stop unwanted smells/compound formation at an endpoint. All this seems unrealistic but within few years these things will penetrate into our life marking a new revolution.

Index – Intelligent Systems, Odor Transfer, Human-Computer Interaction

I. Introduction
Secure Aroma Transfer outlines the concept of transferring smell over Internet in a secure manner. Following points are being covered in this paper:

1) Science behind Odor: In this, we have discussed about the biological concept behind smell and its reception in nose. This includes all the biological and physical concepts and nature in which smell needs to be handled.
2) Smell System and Working: The working of whole system is discussed in this section. An ad-hoc client-system architect is taken as a reference for discussions later.
3) Database: We have discussed the architecture of database required for pattern storage and handling of packets for transfer of pattern over Internet.

4) Packet Structures: We have discussed about the structure of packets that are required for handling and transference of data over the Internet securely.
5) The need for securing the system and various applications of the system will be discussed in the last part.

II. Science behind Odor

A. The Biological Insight
From single cell organisms to Multi-cell organisms, everyone is formed from amino acids. Amino Acids are building blocks of every substance. Our hairs are made of proteins which in turn are created using amino acids. Odors are also formed of Amino Acids. These odors acts as a stimulant to the 100 million receptors cell existing in out nose. The stimulation depends on many parameters which are defined later in this paper.

B. Odor physiology [11]
Olfaction depends upon the interaction between the odor stimulus and the olfactory epithelium. The olfactory membrane is a sensitive area, covering 4 to 6 square cm in each nostril. Beneath the membrane is a mucous layer. The nerve cells or peripheral receptor cells that primarily sense odors and fragrances are located in the epithelium. Cilia extend from the nerve cells into the mucous layer, which greatly increases the potential receptor area. The cilia are thought to contain the ultimate olfactory receptors, which are specialized protein molecules. Specific anosmia may result from the inability to synthesize the appropriate protein. The receptor cells transmit impulses to the olfactory bulb located at the base of the front brain. At the bulb, fibers from the nose contact with other nerves, which travel onto to various parts of the brain.
For a substance to be detected as an odor by the receptor cells, several criteria must be met:

1) The substance must be volatile enough to permeate the air near the sensory area.
2) The substance must be at least slightly water-soluble to pass through the mucous layer and to the olfactory cells.
3) The substance must be lipid-soluble because olfactory cilia are composed primarily of lipid material; and finally,
4) A minimum number of odorous particles must be in contact with the receptors for a minimum length of time.

Thus, above written characteristics will be responsible for proper production of Odor on Producer end. The odor should be easily sniffed by the nostrils and should activate the olfactory neurons.

C. Primary classes of Olfactory Stimulants

Based on psychological tests, seven primary classes of olfactory stimulants have been found to preferentially excite separate olfactory cells. These classes are:

1) Ethereal
2) Camphoraceous
3) Musky
4) Floral
5) Minty
6) Pungent
7) Putrid

The nervous system integrates the responses from a number of cells to determine the identity of the primary odor stimulus being received. The intensity of the perceived odor class is related to the number of receptors bound and the degree of excitation of the olfactory cells.

A more flexible way of presenting the primary odors to clarify the idea of complex odors is through the use of Henning’s odor prism.

Six primary odors are located at the corners of the prism. All other odors are mixtures of the primary odors and located on the surfaces and edges of the prism. Thus, odors consisting of two components would be represented on the edges of the prism, three component mixtures occupy the triangular surfaces, and four-component mixtures occupy the square surfaces.

III. Smell System – Insight in Bioinformatics

Making computers understand smell is just like making them understand a pattern. Various techniques can be employed to make a system understand smell. But for this we need to attach a smelling device or a sniffer to the computer. The complete smelling system (including the hardware and the software) proposed here is depicted by the figure below.

The various parts of the system are as follows:

1) Smell Receiver or Sniffer
2) Smell Producer
3) Smell Database
4) Smell Encoder/Decoder

All most all parts are present on both the receiver and senders’ systems. Both of them may be located miles apart.

1) Smell Receiver or Sniffer - On the sender side, a sensing system is required, which just like any other chemical composition detector will try to judge the chemical combination of the vapors. Though developing such a system can be very complicated, but there are no barriers in the world of science. This thing can be very similar to a smoke detector.

A very major success has been achieved by the scientists at NASA. They have developed what they call an Electronic Nose or the ENose. It’s a device that can learn to recognize almost any compound or combination of compounds. It can even be trained to distinguish between Pepsi and Coke. Like a human nose, the ENose is amazingly versatile, yet it’s much more sensitive. “ENose can detect an electronic change of 1 part per million.” So with NASA going the ENose way our Smell receiver already seems to be on the line of production soon. The device connects to a PC, which then determines what smell the electronic nose has captured.

2) Smell Producer - A system will be attached to a device which is just like an atomizer which
using a few set of chemicals will be able to produce vapors. Now the smell which as discussed earlier is the composition of chemicals can be produced by a few reactions. Another, solution for the systems which intend to generate only a few types of smell, is to store the smells using particular chemicals. For example, a company having some ten types of perfume can always put very small samples in a machine at its store.

3) Smell Database - Now, the computer system in the model contains a database as backend. This smell database will store various chemicals of particular combinations which can be used to compose different type of smell. Even the database can hold record containing data regarding chemicals that can be used as whole or as substitute constituents to produce a required type of smell (if only the particular chemical are unavailable at the Smell Producer).

4) Smell Encoder/Decoder - It is responsible for the processing of input from the Smell Receptor. It will look in to the database to find suitable chemicals constituting a particular smell. And on the client side it will recommend the Smell Producer as which chemical combination to be used so as to produce the desired smell.

Working
Now as illustrated by the model, the user makes the request for scanning a smell. The required smell is sensed by the Smell Receptor and then the sensed signal is sent to the Encoder. The encoder will search for the exact smell first in the database, and if found then the chemical composition is extracted. If the exact smell chemicals are not found then it can suggests some options to the user. Now this part is pure A.I. and will require the learning capability in the system. Once the smell is recognized and the chemicals components have been sorted out, data will be sent to the client side. Once again here the decoder will look in its database as how can the required smell be produced. If an exact match is found, then the data is sent to the Smell Producer. If an exact match is not found, then based on intelligence the system will try to guess alternative chemical compositions. The middleware on the both ends will be responsible for letting the editing process in the database. The model here leaves all the calculation part on the encoder so as to make it work with a wide variety of interfaces thus reducing the complexity at the front end.

IV. Database
The odor structures and patterns will be stored in a database. The database will be stored at two locations viz. on the hardware and at a centralized server. The database stored at the centralized server will be updated from the information gained at individual hardware. Following is the detailed database structure that will be used.

Table 1 – Pattern
The structure of the main packet table will be like the one shown below. It will have following attributes:

Pattern Index: Pattern Index will refer to the identity of the pattern

Ethereal Index: This attribute will refer to the index of the ethereal compound

Ethereal Intensity: This attribute will let know the device how intense is the concentration of ethereal compound.

Camphor Index: This attribute will refer to the index of the camphor compound

Camphor Intensity: This attribute will let know the device how intense is the concentration of camphor compound.

Musky Index: This attribute will refer to the index of the musky compound

Musky Intensity: This attribute will let know the device how intense is the concentration of musky compound.
Putrid Index: This attribute will refer to the index of the putrid compound.
Putrid Intensity: This attribute will let know the device how intense is the concentration of putrid compound.

Pungent Index: This attribute will refer to the index of the pungent compound
Pungent Intensity: This attribute will let know the device how intense is the concentration of pungent compound.

Floral Index: This attribute will refer to the index of the floral compound
Floral Intensity: This attribute will let know the device how intense is the concentration of floral compound.

Minty Index: This attribute will refer to the index of the minty compound
Minty Intensity: This attribute will let know the device how intense is the concentration of minty compound.

Danger Level: The danger level is associated with each pattern. This danger level is important to disallow usage of certain compounds that are harmful.

To secure the odor transfer system, we need to perform some base security checks, one of which is to block certain IP’s that are repeatedly sending unwanted odor information. This unwanted odor information may be unacceptable to the system and scenario in which it will be used or may be categorized as dangerous odor.

Following are the attributes associated with blacklisted IP table:

Index: Reference number to blocked IP will be stored here in this attribute.
IP: The IP blocked will be stored in this attribute in the table.
ClientName: This attribute will contain the hostname of the client to be blocked and will also contain some OS fingerprinting data so that any IP can be unblocked in case the IP assignment was dynamic.
Reason to block: The reason to block will be mentioned automatically. This may be repeatedly sending unwanted data or dangerous smell or something else.

Fig 4: Table 2 – Blacklisted IPs

Composition Tables

Composition tables will contain the list of various chemical compounds specific to seven odor classes. These chemical compounds will later mix together to form various odors.

Table 3 – Composition table for Odor classes

CompoundID: The CompoundID field will contain a unique reference id specific to various chemical compounds listed in the table.

CompoundName: The CompoundName field will contain the name of the chemical compound referenced above.
Composition: The composition field will tell about the chemical composition of the compound. For example – If compound involved is methane, then simple CH₄ will be written in this tuple against composition field.

<table>
<thead>
<tr>
<th></th>
<th>ETHREAL Property Table</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CompoundID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CompoundName</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Composition</td>
<td></td>
</tr>
</tbody>
</table>

Fig 5: Table 3 – Ethereal Composition Tables – Similar tables need to be created for other odor classes

V. Packet Structure

Odor Transfer Protocol (ODTP) will be the protocol used to transfer data on the Internet. The protocol will support certain signals and packets that will provide status of the system and help in communication between two ends.

Following will be the packets that will be used to communicate over the network:

- Login Packet
- Login OK Packet
- Odor Transfer Start Packet
- Odor Pattern Packet
- Odor Transfer Stop Packet
- Odor Transfer Status Packet

Data Packet:

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>32</td>
</tr>
</tbody>
</table>

OD TP

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ClientID</td>
</tr>
<tr>
<td>4</td>
<td>IP</td>
</tr>
<tr>
<td>2</td>
<td>Listening Port</td>
</tr>
<tr>
<td>1</td>
<td>Encryption</td>
</tr>
<tr>
<td>1</td>
<td>Compression</td>
</tr>
<tr>
<td>2</td>
<td>Max Transfer Speed</td>
</tr>
<tr>
<td>2</td>
<td>Min Transfer Speed</td>
</tr>
<tr>
<td>2</td>
<td>Min Time Interval</td>
</tr>
</tbody>
</table>

Data (0 - 224)

(Ethereal, Musky, Pungent, Minty, Putrid, Floral, Camphoraceous)

Fig 6: Data Packet

ODTP Packet

Above shown is the Data Packet that will contain the details of the smell to be sent. The data packet will consist of two parts viz. header and data. Header will be of 16 Bytes and will be common to each packet.

Header Details:

First four Bytes will point to the type of Protocol that will be used (in this case ODTP). Next four bytes will be used to display the protocol version. Next two bytes will be used to display the length of the payload. Next two bytes will be used to display type of payload viz. Initialization, Login, Odor Payload etc. Next three bytes will be used to display SessionID and last byte will be used to display connection status.

Data:

The Data field will be dynamic and the length will range from 0 to 28 Bytes. It will contain the code for odor that needs to produce at other end.

Login Packet:

This packet is used to make session between two machines by sending ClientID, ip, listening port and other details. Following listed are all the values that will be sent on both ends.

Bytes | Value                  |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ClientID</td>
</tr>
<tr>
<td>4</td>
<td>IP</td>
</tr>
<tr>
<td>2</td>
<td>Listening Port</td>
</tr>
<tr>
<td>1</td>
<td>Encryption</td>
</tr>
<tr>
<td>1</td>
<td>Compression</td>
</tr>
<tr>
<td>2</td>
<td>Max Transfer Speed</td>
</tr>
<tr>
<td>2</td>
<td>Min Transfer Speed</td>
</tr>
<tr>
<td>2</td>
<td>Min Time Interval</td>
</tr>
</tbody>
</table>

Login OK Packet:

This Packet will be send by peer on other side in reply to the Login packet. It will contain information about other end’s IP, listening port for further communication, ClientID, support for encryption and compression etc.

Bytes | Value                  |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ClientID</td>
</tr>
<tr>
<td>4</td>
<td>IP</td>
</tr>
<tr>
<td>2</td>
<td>Listening Port</td>
</tr>
<tr>
<td>1</td>
<td>Encryption</td>
</tr>
<tr>
<td>1</td>
<td>Compression</td>
</tr>
<tr>
<td>2</td>
<td>Max Transfer Speed</td>
</tr>
</tbody>
</table>
**Odor Pattern Packet:**

<table>
<thead>
<tr>
<th>PatternID</th>
<th>ObjectName</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Class</th>
<th>Composition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intensity(in%)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Class</th>
<th>Composition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intensity(in %)</td>
<td></td>
</tr>
</tbody>
</table>

Fig 7: Odor Pattern Packet

The Odor pattern packet will contain the information about the percentage intensity, composition and odor class type. The final smell to be produced will be labeled with a name of the substance it smells like.

**Odor Transfer Start Packet:**

This packet is used to start transfer of odor from other end. This packet will contain only header with payload length zero and service value with code for start odor.

**Odor Transfer Stop Packet:**

This packet is used to stop transfer of odor from other end. This packet will contain only header with payload length zero and service value with code for stop odor.

**Transfer Status Packet**

This Packet will be used to send status of a client to peer on other side. This packet will contain information about odor receive, odor send, how many new smells it got and send in last hour.

**VI. Securing the Smell System**

The smell system in itself is a very unique idea that will change the way we do things. But securing the system will be an utmost important thing to do. If some malicious being is successful in transferring Phosgene over the Internet, he can kill so many peoples. To minimize the danger the concept of Odor Firewalls will come into being. Moreover, many other considerations as discussed below should be taken into consideration.

a) **BlackListed IP’s** – A BlackListed IP table is used to maintain the information of blocked IP’s in database. This table will contain ClientName (which will be unique for each machine [hardware] and will not change even in case of dynamic IP’s), IP and port information. The machines from which something malicious, that may be transferring dangerous odor packets periodically, is being done will be blocked.

b) **Payload Integrity Check** – An integrity check will be done on all the packets that need to be transferred. The techniques that can be used may be hashing or CRC check.

c) **Dangerous Odor Check** – All the odors to be transferred will be first checked for Danger Level of the smell. The danger level if large will not allow encoding of the smell on the sender end.

d) **Database Security** – The database structure can only be modified/update by a service running on a web server that will be responsible for updation of database on clients.

e) **Hardware Intelligence in Atomizer** – At hardware level there is a consideration on Physical Environment Dependency that needs to be taken care of: Whenever odor production will start on the producer end, the hardware should check the outside environment chemical composition and should check if the odor to be produced by the atomizer can react with the environment and result in drastic after-effects.

f) **Hardware Agreements** – All the interaction between two ends shall depend on the negotiations performed before starting the transfer. The negotiation will involve encryption, compression information.

g) **Devices of different Brand** – In case, the two interacting entities have different brand of devices, there may occur a situation where one entity encode a smell but the other end decode it as something different. In that regard, a standardization of the hardware needs to be done.

**VII. Applications**

1) Consider the case of a perfume company. All it needs is to set an outlet, which provides intelligent solutions, which will require only the above discussed Smell Model. The customer will ask for the smell of a particular perfume from the system by accessing the front end. And the system will produce the smell on the spot. Then
depending upon the decision of the customer the product can be ordered at the online store.

2) This technology can be used in mission critical sensitive areas such as sophisticated research facilities. For examples, if the amount of ammonia grows in the air, the system will automatically activate an alarm. This is an idea similar to the detection of amount of ammonia in space crafts, the main aim behind Enose by NASA.

3) Consider an online food order system. User just needs to visit the portal and click on his favorite coffee. He can smell the coffee and can order the coffee online. This can be done with all the eatables and once it is done, online shopping will change forever.

4) Similarly as till now feelings have been transferred over the internet in the form of text, sounds and pictures, smell can also come into picture. Imagine a greeting card which not only has pleasing audio-visuals but also produces a soothing sweet smell. Even user defined smells could be transported. A girl could then make her boyfriend feel special by sending smell laden greeting card.

5) This system can bring new realism to games; imagine smelling blood in a bloody game like Resident Evil or the smell of leaves when you pass through a forest in a game like Jurassic Park.

6) Even for many other areas can benefit from smell over internet. Invent a new smell and then transport it through the internet.

7) Chatting will be the biggest opportunity domain for Smell Systems. The smell system will produce the smell of one end on the other end thus virtually bringing people closer.

VIII. Conclusion

Think of surfing Dominos website and clicking on your favorite pizza for order. With a single click, one can smell a very pleasant smell of the pizza in the room and can change his order or finalizes his order. Integrating smell with computers is not a far fetched impossible concept. Smell promises to add a new dimension to the internet. And in a matter of few years this technology will find place in normal Personal Computers similar to audio-video systems. The power of smell really rocks. And from now multi-media will contain one more media, the ‘Media of Smell’.

IX. References

[1] Science of Odor - A
http://www.extension.iastate.edu/Publications/PM1963A.pdf

http://www.extension.iastate.edu/Publications/PM1963B.pdf

http://www.extension.iastate.edu/Publications/PM1963C.pdf

http://www.extension.iastate.edu/Publications/PM1963D.pdf

[5] Smell System – ‘Enhancing Internet: Transporting Smell over Internet’ published by Abhinav Dhall and Vibhore Jain at ICCSA-2005, National University, California, USA RAFFT-2005, Punjabi University, Patiala, Punjab, India
http://www.universitypunjabi.org/pages/teaching/dcs/rafft05/session5.pdf

[6] ENose
http://science.nasa.gov/headlines/y2004/06oct_enose.htm
RESOURCE DESCRIPTION FRAMEWORK : A SMALL CASE STUDY ON EDUCATIONAL DOMAIN

V. Kiran Kumar
Assistant Professor
Dept of Computer Science
Dravidian University, Kuppam
Andhra Pradesh, India
Kirankumar.v@rediff.com

Abstract

The Resource Description Framework is a W3C standard language for representing information about resources in the world wide web; RDF provides a common framework for expressing this information, so it can be exchanged between applications without loss of meaning. In this paper I used RDF and RDFSchemas to represent Dravidian University information database to facilitate students for getting the required information. In this paper, I used SPARQL as a query language to retrieve information from database.

Keywords

Semantic, RDF, RDFSchema, SPARQL

Introduction

The semantic web vision was conceived by Tim Berners-Lee, the inventor of World Wide Web (WWW). Semantic web is an extension of the current web, in which information is given well-defined meaning. Semantic in the ‘semantic web’ is not that the computers can understand the meaning of anything, but the logical pieces of meaning can be mechanically manipulated by a machine to useful ends. In this paper for stressing of the semantic web, I used Resource Description Framework (RDF) for knowledge representation and SPARQL, a query language for information retrieval. For this purpose, I took a case study of the Dravidian University information. Starting with the description of RDF and RDFSchemas, I analyzed the representation of Dravidian University information and worked on sample queries by using SPARQL for retrieving the information. The paper aims to highlight the importance of semantic web by using the available tools like RDF, RDF Schema and SPARQL.

RDF and RDF Schema

In February 2004, The World Wide Web Consortium released the Resource Description Framework (RDF) as W3C Recommendation. RDF is used to represent information and to exchange knowledge in the Web. It gives you a way to make information machine-processable. RDF uses a general method to decompose knowledge into pieces called triples. The triples can be represented as subject, predicate and object. In RDF, the English statement
“Tim Berners-Lee invented World Wide Web”

Could be represented by RDF statement having

- A subject *Tim Berners-Lee*
- A predicate *invented*
- And an object *World Wide Web*

RDF statements may be encoded using various serialization syntaxes. The RDF statement above would be represented by the graph model as shown below

![RDF Graph](image)

Subject and objects are represented by nodes and predicate is represented by an arc. RDF’s vocabulary description language, RDF Schema, is a semantic extension of RDF. It provides mechanisms for describing groups of related resources and the relationships between these resources. RDF Schema vocabulary descriptions are written in RDF using the terms described in this document. These resources are used to determine characteristics of other resources, such as the *domains* and *ranges* of properties.

**Representing Facts in RDF and RDFS**

The information is categorized under four resources such as department, courses in each department, employees in the departments and research activities of each employee. The RDF Schema of our domain uses simple namespaces like

- **Dept** – represents departments in the university
- **Course** – courses in each department
- **Staff** – employees in each department
- **Projects** – research activities of each employee in the university

The ‘Dept’ class represents all the departments in the university. For ex: “computer science head C.Lokanatha reddy”. In this statement computer science is a subject, head is a predicate and ‘C.Lokanatha Reddy’ is Object. Like this I represented all the departments with concerned heads and year of establishment they started. Here ‘depid’ is an identification code for each department. Similarly the Course class represents all courses that are being offered department-wise. For ex: if the computer science department is offering M.C.A course then this fact can be represented as “depid offering M.C.A” (depid -> subject offering -> predicate M.C.A->object). Similarly the Course resource represents other details such as eligibility to join, admission process whether through entrance test or interview by the university, pattern of the course i.e., yearly or semester, fee particulars and category of the course i.e., degree or PG or Research etc., After understanding the meaning of RDF and RDF Schema, let us now
proceed to see how these two work, by taking the Dravidian University information as a case study.

The ‘Staff’ class represents the information about employees working in each department. This class has a relation with the ‘Dept’ with depid. This class contains the details of employee name, qualification, experience and address. Finally, the ‘Project’ class is related to the research activities in the university. This has connected with Staff class through ‘empid’. This contains information about research projects of each employee in the department and details of funding agency. The detailed graph is represented as above. Circles and rectangles represent subject and object and arcs represent predicate.

**Querying with SPARQL**

SPARQL is a query language for getting information from the RDF Graphs. There are several tools and Application Programming Interface (APIs) that already provide SPARQL
functionality. I made an attempt with ARQ – SPARQL processor for jena for information retrieval.

The example queries are listed below

1) List the departments in the university

```
PREFIX ex: <http://dravidianuniversity.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?name ?head ?established
WHERE {
  ?n ex:depid ?id;
  ex:depname ?name;
  ex:depestablished ?established;
  ex:dephead ?head;
}
ORDER BY ?name
```

2) List the Courses offered by “Computer Science” Department

```
PREFIX ex: <http://dravidianuniversity.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

WHERE {
  ?n ex:depid ?id;
  ex:depname “Computer Science”.
  ?m ex:depid ?id;
  ex:coursename ?cname;
  ex:eligibility ?eligibility;
  ex:admission ?adm;
  ex:pattern ?pattern;
  ex:fee ?fee;
}
```

3) List the Employees in each department

```
PREFIX ex: <http://dravidianuniversity.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

WHERE {
  ?n ex:depid ?id.
  ?m ex:depid ?id;
  ex:empname ?name;
}
```
4) List the PG courses offered in each department

```sql
prefix ex: <http://dravidianuniversity.org/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

WHERE {
  ?n ex:depid ?id;
  ex:deptname ?name
  ?m ex:depid ?id;
  ex:coursename ?cname;
  ex:eligibility ?eligibility;
  ex:admission ?adm;
  ex:pattern ?pattern;
  ex:category "PG";
  ex:fee ?fee.
}
```

5) List the projects done by employees in each department

```sql
prefix ex: <http://dravidianuniversity.org/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

WHERE {
  ?n ex:depid ?id;
  ex:deptname ?dname
  ?m ex:depid ?id;
  ex:empname ?ename;
  ex:projectname ?prname;
  ex:funding ?funding.
}
```

**CONCLUSION AND FURTHER WORK**

In this paper an attempt is made to discuss how the information can be represented using RDF and RDF Schemas and how it can be retrieved it using SPARQL. This is entirely different from existing searching tool, as the existing searching engines are capable of retrieving information with keywords. That may result in many inappropriate results. But by using RDF and SPARQL inappropriate results can be minimized to a great extent. Though the appropriateness of the usability of the tool is significant, it has practical difficult in developing the code. Unlike the other search engines the present tool needs two output
formats, one for designing and another for meaning of the content. However, the merit of usability over shadows the difficulty with regard to code. The usability can be extended by adding additional information about the university and making front-end to facilitate query data for user.

References


[4] where to get SPARQL : C:\ARQ\doc\download.html


Cooperative Intrusion Detection for Detecting Novel Attacks Using Real Time Data Mining Approach

R.Pushparaj, Final Year M.E (CSE), S.Bhuvana M.E, Lecturer, CSE.
VLB Janakiammal College of Engineering and Technology, Coimbatore, TamilNadu, India.
pushparaj_vlb@yahoo.co.in, pusheager@sify.com

Abstract

The Internet is growing everyday and the growth appears to be accelerating. As more numbers of users become connected to Internet provides window of opportunity for the malicious users to do their damage becomes very high and lucrative. We proposed a Cooperative Intrusion Detection System (CIDS) approach that combines the advantages of low false-positive rate of signature-based intrusion detection system (IDS) and the ability of anomaly detection system (ADS) to detect novel unknown attacks. By mining anomalous Internet traffic episodes from Internet connections, we are going to build an ADS that detects anomalies beyond the capabilities of signature-based SNORT or Bro systems. A weighted signature generation scheme is developed to integrate ADS with SNORT by extracting signatures from anomalies detected. CIDS extracts signatures from the output of ADS and adds them into the SNORT signature database for fast and accurate intrusion detection. We hope that the signatures generated by ADS will upgrade the SNORT performance by some percent. The CIDS approach proves the vitality of detecting intrusions and anomalies, simultaneously, by automated data mining and signature generation over Internet connection episodes.

Keywords: Network security, intrusion detection systems, Internet traffic data mining, frequent episode rules, anomaly detection, anomaly and normality scores, signature generation, SNORT systems.

1. INTRODUCTION

Security of network systems is becoming increasingly important as more and more sensitive information is being stored and manipulated online. Intrusion Detection Systems (IDSs) have thus become a critical technology to help protect these systems. Intrusions and anomalies are two different kinds of attacks in an open network environment. An intrusion takes place when an unauthorized access of a host computer system is attempted. An anomaly is observed at the network connection level. Both attack types may compromise valuable hosts, disclose sensitive data, deny services to legitimate users, and pull down network-based computing resources. The intrusion detection system (IDS) offers intelligent protection of networked computers or distributed resources much
better than using fixed-rule firewalls. To protect telnet, http, ftp, SMTP, pop3, Email, and authentication services, early detection of Internet anomalies in routers, gateways, hosts, and servers are a necessity.

1.1 Intrusion versus Anomaly Detection Systems

In this paper, we consider signature-matching intrusion detection systems (IDS), based on the misuse model. We design anomaly detection system (ADS) based on the anomaly-based model characterized in Table 1. The misuse IDS model matches attack signature with pre-stored signatures from known attacks. The signatures are manually constructed by security experts analyzing previous attacks. The collected signatures are used to match with incoming traffic to detect intrusions. These are conventional systems that detect known attacks with low false alarms. The major problem with this approach is that these IDSs fail to generalize to detect new attacks or attacks without known signatures. Furthermore, signature matching performs well only for single-connection attacks. With the sophistication of attackers, more attacks involve multiple connections. This limits the detection range by signature matching.

On the other hand, an anomaly-based system uses a different philosophy. It treats any network connection violating the normal profile as an anomaly. A network anomaly is revealed if the incoming traffic pattern deviates from the normal profiles significantly. Through a data mining approach, anomaly detection discovers temporal characteristics of network traffic. This system can detect unknown attacks and handles multiconnection attacks well. However, anomaly detection may result in higher false alarms.

Table 1. Comparison of Misuse Model and Anomaly Model for Intrusion Detection

<table>
<thead>
<tr>
<th>Attack Characteristics</th>
<th>Misuse Intrusion Detection System (IDS)</th>
<th>Anomaly Detection System (ADS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profiling level</td>
<td>Packet scanning to reveal misuse, abnormal, and guilty patterns</td>
<td>Mining TCP/UDP/ICMP connections to save normal traffic profiles</td>
</tr>
<tr>
<td>Detection mechanisms</td>
<td>Signature matching with known attack patterns and misuse logs</td>
<td>Profile matching with association or episode rules against normal traffic</td>
</tr>
<tr>
<td>Implementation</td>
<td>Signature database and signature update mechanisms</td>
<td>Mining of audit data, episode rule generation, and database update</td>
</tr>
<tr>
<td>requirements</td>
<td>advantages and shortcomings</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Detecting known attacks with low false alarms, fail to detect unknown attacks</td>
<td>Detect all attacks with higher false alarms, adaptive to network changes</td>
</tr>
<tr>
<td>Representative Systems</td>
<td>Snort [35], JAM [22], ADCPAR [13], MADAM ID [23]</td>
<td>Fuzzy Logic IDS [4], ADAM [3], EMERALD [31]</td>
</tr>
</tbody>
</table>

Existing IDSs are built with either signature-based or anomaly-based systems. The design philosophies of these two models are quite different, and they were rarely mixed up in existing IDS products from the security industry.
In this paper, we present a new co-operative intrusion detection system (CIDS) which is network based. This system combines the positive features of both intrusion detection models to achieve higher detection accuracy, lower false alarms, and, thus, a raised level of cyber trust. An adaptive base support threshold is applied on selected axis attributes in mining the Internet episode rules. The episode rules are used to build the CIDS, which detects not only known intrusive attacks but also anomalous connection sequences.

The rest of the paper is organized as follows: Section 2 reviews related works and distinguishes the new approach from previous solutions. Section 3 introduces traffic data mining and describes the CIDS architecture. We present Internet episode rules and pruning techniques in Section 4. The weighted signature generation is specified in Section 5. Finally, in Section 6, we summarize the contributions and comment on further research needed.

2 RELATED WORKS AND OUR APPROACH

In the past, data mining techniques such as using association rules were suggested to build IDS [1], [5]. Lazarevic et al have distinguished the differences between single-connection and multiconnection attacks. Both signature-based and anomaly-based IDSs are sensitive to the attack characteristics, system training history, services provided, and underlying network conditions. Data mining techniques are also used to build classification models from labeled attacks [8], [9]. SNORT [4] and Bro [14] are two widely used IDSs that are based on the misuse model. Intrusion detection must be designed to monitor the connection features at the network, transport, and application layers [3], [6]. The MIT/LL IDS evaluation data set and reported IDS performance results were analyzed in [10], [11]. We use this attack data set with mixed background traffic to test the effectiveness of CIDS. The concept of frequent episode rules (FERs) was first proposed by Mannila and Toivonen [12]. Subsequently, Lee suggested a framework to specify FERs for anomaly detection against normal traffic profiles. They developed a level wise data mining algorithm for building ADS. Fan extended Lee’s work to discover accurate boundaries between known attacks and unknown anomalies. In and Hwang [13] refined the rule formulation procedure with an adaptive base-support algorithm to mine normal traffic records. Different axis attribute values apply different thresholds. Kaleton Internet [7] built a prototype system by combining the two detection systems, but they work independently without interactions. We consider close cooperation between the two subsystems.

In this paper, we propose the CIDS architecture. The CIDS integrates the flexibility of ADS with the accuracy of signature-based IDS. ADS is designed by mining FERs [8], [12] over Internet connections. We have developed a new weighted signature generation algorithm to characterize anomalous attacks and extract their signatures. The new signatures are generated from anomalies detected by ADS. This idea was inspired by earlier works on weighted association rules [15], [16]. This new approach automatically enables CIDS to detect similar anomalous attacks in the future.
3 Co-operative Intrusion Detection System (CIDS)

In this section, we first introduce the data mining concept for hybrid intrusion and anomaly detection. Then, we describe the CIDS architecture, the ADS design, and the connection features used in ADS and automated signature generation.

3.1 Traffic Data Mining for Network Anomaly Detection

Open networks face threats from both system intrusions and anomalies in Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or ICMP connections.

Fig. 1. Data mining scheme for network anomaly detection over Internet connection records.

Fig. 1 shows the three major components of our network anomaly detection process. First, we apply the normal profile database and construct the anomaly detection engine. The detection engine is capable of detecting anomalous episodes that are caused by traffic anomalies. The connection records are extracted from audited Internet traffic. The concept of FER will be covered in Section 4. The episode rule mining engine consists of two phases of development. The training phase is needed to generate the normal traffic database without attacks. Attacks may appear in the detection phase. The anomaly is detected once the episode rule describing the real traffic connections cannot find any match with normal connection rules in the database. With this reasoning, the network anomaly is detected by a normal-use detection model. We have generated an attack data set by a mixture of locally captured Internet trace files and the DARPA 1999 IDS evaluation data set. We stretched the trace file to mix with the Massachusetts Institute of Technology/ Lincoln Laboratory traffic files collected. We use the toolkits by Mahoney and Chan to mix Internet traffic data with the MIT/LL data set. A drawback of the FER-
based approach is caused by the fact that many attacks are triggered by a single connection and may not generate anomalous FERs. In order to solve this problem, we keep a keen interest on rare attributes of single connections. For example, connections with the same source and destination addresses are often attacks. Another problem is that a single attack may last for a long period of time. To solve this problem, we use connection sequence numbers, instead of time stamps, to mine connections heading to the same destination.

### 3.2 The CIDS System Architecture

Anomaly-based systems are supposed to detect unknown attacks. These systems are often designed for offline analysis due to their expensive processing and memory overheads. Signature-based system leverages manually characterized attack signatures to detect known attacks in real-time traffic. The CIDS illustrated in Fig. 2 integrates the flexibility of ADS with the accuracy of a signature-based SNORT. The SNORT is connected in cascade with the custom-designed ADS. These two subsystems join hands to cover all traffic events initiated by both legitimate and malicious users. By 2004, SNORT has accumulated more than 2,400 attack signatures in its database.

![Diagram](image.png)

**Fig. 2.** A co-operative intrusion detection system built with a SNORT and an anomaly detection subsystem (ADS) through automated signature generation from Internet episodes.
In CIDS operations, the first step is to filter out the known attack traffic by SNORT through signature matching with the database. The remaining traffic containing unknown or burst attacks is fed to the episode-mining engine to generate frequent episode rules with different levels of support threshold. This leveling allows the detection of some rare episodes, declared as anomalies. The frequent episodes are compared with precomputed frequent episodes from normal traffic. The episodes that do not match the normal profiles or match them with unusually high frequency are labeled as anomalous. The anomalous episodes are used to generate signatures which capture the anomalous behavior using a weighted frequent item set mining scheme. These signatures are then added to the SNORT database for future detection of similar attacks. Unknown, burst, or multiconnection attacks are detectable by ADS. The signature generation unit bridges two detection subsystems in the shaded boxes. This unit characterizes the detected anomalies and extracts their signatures. We built an ADS by using the FER mining mechanisms to be described in Section 4. The new CIDS detects many novel attacks hidden in common Internet services, such as telnet, http, ftp, SMTP, e-mail, authentication, and so forth. The CIDS deployment appeals particularly to protect network-based clusters of computers, resources inside internal networks (intranets), and computational Grids.

3.3 Internet Connection Features

The performance of ADS is directly affected by the features used in training and rule generation. Lee and Stolfo and Lazarevic used connection features, temporal statistics, and content features for building IDSs. Table 1 summarized the connection and temporal statistics features used in our ADS to generate FERs by training and testing.

### TABLE 1

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature Name</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection Features</strong></td>
<td>timestamp</td>
<td>Time when the connection begins</td>
</tr>
<tr>
<td></td>
<td>duration</td>
<td>Duration of the connection in second</td>
</tr>
<tr>
<td></td>
<td>ip_proto</td>
<td>IP protocol type</td>
</tr>
<tr>
<td></td>
<td>src_ip</td>
<td>Source IP address</td>
</tr>
<tr>
<td></td>
<td>dst_ip</td>
<td>Destination IP address</td>
</tr>
<tr>
<td></td>
<td>service</td>
<td>Network service on the destination, e.g., http, ftp</td>
</tr>
<tr>
<td></td>
<td>icmp_type</td>
<td>ICMP message type</td>
</tr>
<tr>
<td></td>
<td>src_types</td>
<td>Bytes sent by the source</td>
</tr>
<tr>
<td></td>
<td>dst_types</td>
<td>Bytes sent by the destination</td>
</tr>
<tr>
<td></td>
<td>flags</td>
<td>SF: Both SYN and FIN packets are known, S0: Only SYN packet seen in a connection, REJ: Connection rejected by the destination</td>
</tr>
<tr>
<td><strong>Temporal statistical features</strong></td>
<td>src_count</td>
<td>Number of connections from the same source</td>
</tr>
<tr>
<td></td>
<td>dst_count</td>
<td>Number of connections to the same destination</td>
</tr>
<tr>
<td></td>
<td>service_count</td>
<td>Number of connections for the same service</td>
</tr>
<tr>
<td></td>
<td>syn_dst%</td>
<td>% of connections with same feature and SYN errors</td>
</tr>
<tr>
<td></td>
<td>service_dst%</td>
<td>% of connections per service to the same destination</td>
</tr>
<tr>
<td></td>
<td>syn_service%</td>
<td>% of connections with SYN error to the same port</td>
</tr>
<tr>
<td></td>
<td>ave_duration</td>
<td>Average duration of connections for the same service</td>
</tr>
<tr>
<td></td>
<td>ave_src_bytes</td>
<td>Average bytes sent by the source</td>
</tr>
<tr>
<td></td>
<td>ave_dst_bytes</td>
<td>Average bytes sent by the destination</td>
</tr>
</tbody>
</table>
Since we aim at detecting anomalies of network traffic, the content features, which are mainly extracted from system log files, are not used in this work. The connection features and temporal statistics will be used in CIDS construction. Connection level features are extracted from raw TCPdump files. They are used in both FER and signature generation. The flags are used to signal special connection status. We listed three different flags: SF, S0, and REJ.

Temporal statistics are related to connections with the same reference features. They can be used to improve the accuracy of signature generation. For example, by tracking the number of connections initiated from the same source, the Source_Count could be used to set the threshold when we generate new signatures.

4. INTERNET EPISODE RULES AND PRUNING TECHNIQUES

An Internet episode is represented by a sequence of connection events, such as TCP, UDP, ICMP, or other connections. An episode can be generated by legitimate users or malicious attackers. Frequent episodes mostly resulted from normal users. A rare episode is likely caused by intruders. Our purpose is to build an ADS that can distinguish the rare or abnormal episodes from the normal or frequent episodes automatically.

Fig. 3. Generation of a frequent episode rule by scanning stream of Internet traffic connection events from left to right using a small scanning window.

4.1 Generation of Internet Episode Rules

In Fig. 3, we show typical stream of Internet traffic, represented by a sequence of connection events labeled as E1, E2, E3, and so forth. These connection events are related to various Internet service commands such as http, ftp, SMTP, authentication, and so forth. Note that some events may repeat to appear in the sequence. The time instants of these connections, in seconds, are marked below the events. A frequent episode is a set of connection events exceeding the occurrence threshold in a scanning window. A FER is generated out of a collection of frequent episodes. The FER is defined over episode sequences corresponding to multiple connection events in a roll.

For the first window in Fig. 3, we cover a sequence of three connection events: E2, E1, and E3. The event E2 triggers the occurrence of events E1 and E2 in a cascade. This leads to the following FER on these three events. This rule is detected within a window size w.

The rule is backed by a support base s, confidence level c, and minimal occurrence f.
\[ E_2 \rightarrow E_1, E_3 (s, c, w, f) \]  \quad \text{----------> (1)}

\[ s \text{ ----> refers to the probability of event } E_2 \text{ to occur,} \]
\[ c \text{ ----> refers to the probability of the joint connection event } (E_1, E_3) \cup E_2 \text{ to take place after event } E_2. \]

We calculate,

\[ s = \text{Prob} [E_2] \]
\[ c = \text{Prob} [(E_1, E_3) \cup E_2] / \text{Prob} [E_2] \]

The support value \( s \) reflects the percentage of minimum occurrences of the episode rule out of the total number of connection records audited. The confidence level \( c \) is the joint probability of the minimal occurrence of the joint episodes out of the support of the left-hand side (LHS) episode. The window size \( w \) is the scanning period of the window. The minimal occurrence \( f \) indicates the minimum number of occurrences to establish the rule in question.

For a real-life example, event \( E_2 \) could be an authentication service requested at time zero, presented by two attributes (service =) authentication; flag = SF), where the flag is defined in Table 1. Events \( E_1 \) and \( E_3 \) correspond to two consecutive SMTP service requests denoted by: (service =smtp) (service = smtp). We obtain the following FER with a confidence level \( c = 80 \) percent for an authentication service followed by two SMTP services detected within a scanning window \( w = 4 \) s. The three joint traffic events account a support level \( s = 10 \) percent out of all possible network connections. The minimal occurrence \( f = 1; 200 \) implies that 1,200 is the minimum number of occurrence for the rule to be generated into the rule base.

\[ (\text{service = authentication}) \rightarrow (\text{service = smtp}) (\text{service = smtp}) (0:8; 0:1; 4 s; 1; 200) \]

\text{------> (2)}

4.2 A Base-Support Data Mining Scheme

Most mining techniques exclude infrequent traffic patterns. This will make the IDS ineffective in detecting rare network events. If we lower the support threshold, then a large number of uninteresting patterns associated with frequent services will be discovered. We introduce a new base support mining process to handle this problem. The process is specified in Algorithm 1. Our method is improved from the level wise algorithm by Lee.

\textbf{Algorithm 1.} Base-Support Traffic Data Mining Algorithm

1: INPUT: Base-support threshold \( f_0 \), all axis attributes and the set \( T \) of all network connections
2: OUTPUT: New FERs to add into existing rule set \( L \)
3: for each axis item set \( X \) in \( T \), do
4: calculate support(X);
5: end for
6: scan T and compute \( L = \{ \text{itemsetY|} f(Y) \geq f_0 \} \);
7: repeat
8: generate new episode rule sets \( E = \{ e_0; e_1; \ldots; e_n \} \),
   \( \text{Where support (e_0; e_1; \ldots; e_n) } \geq f_0 \times \min \{ \text{base sup (e_i)} \} \);
9: if \( E \) is not empty, then
10: generate FERs from \( E \) with confidence \( e \) above minimum confidence \( e_0 \);
11: add the generated FERs into rule set \( L \);
12: end if
13: until \( E \) is empty.

By using Lee’s algorithm, one iteratively lowers the minimum support value. Initially, a high minimum support value is chosen to find the episodes related to high frequency axis attribute values. Then, the procedure iteratively lowers the support threshold by half. This links each new candidate FER with at least one new axis value. The procedure terminates when a very small threshold is reached. Let \( X \) be an item set. The support base of \( X \) denoted by \( \text{sup base}(X) \) is the support value of the axis item set. For example, when choosing the service and flag as the axis attributes, the support base for item set \( X = (\text{service} = \text{ftp}; \text{flag} = \text{S0}; \text{srchost} = \text{128:1:1:1}; \text{destination} = \text{121:1:1:1}) \) is defined by \( \text{sup base}(X) = \text{support (service = ftp; flag = S0)} \). The base-support fraction \( f \) for item set \( X \) is defined by:

\[
F(X) = \text{support}(X) = \text{base sup}(X): \quad \text{(3)}
\]

Similarly, the base-support fraction of an episode is defined as the percentage of the number of minimal episode occurrences to the total number of records in \( T \), which contains the most uncommon axis attributes embedded in this episode. The minimum support base value of an episode \( e_1; e_2; \ldots; e_n \) is denoted by \( \min \{ \text{base sup (ei)} \} \).

### 4.3 Episode Rule Training from Normal Traffic

Fig. 4 shows the FER generation and rule-matching process in anomaly detection based on Algorithm 1. When attacks are detected by SNORT, their time stamps are passed to the packets eliminator, and the corresponding traffic flows are deleted. The rest of the traffic is passed to the ADS. When a FER generated from the traffic does not match the normal FER database, an unknown FER anomaly is suspected.

When the matched rule occurs beyond the threshold, multiple FER anomalies are suspected. The FER anomalies are confirmed by checking some error flags and temporal statistics listed in Table 1. Otherwise, the traffic connection is considered normal. To generate FERs for normal traffic profiles, the attack free training connection records are fed into the data mining engine. We use the audit data sets collected in weeks 1 and 3 of the 1999 MIT/LL IDS evaluation package.
We generated 92 FERs with the limited training time. We do not use FERs with extremely low support values. After finding FERs from each day’s audit record, we simply merge them into a large rule set by removing all redundant rules based on the pruning techniques discussed in Section 4.4.

### 4.4 Pruning of Ineffective Episode Rules

We consider a FER effective if it is applicable and more frequently used in the anomaly detection process. An episode rule is ineffective if it is rarely used in detecting anomalies. Some FERs differ only at the LHS or at the RHS. Keeping all rules generated will enlarge the search space and thus increase the overhead. The following FER transformation laws will reduce the rule search space significantly.

#### 4.4.1 Elimination of Redundant Episode Rules

In general, rules with shorter LHSs are more effective than rules with longer LHSs. This is because shorter rules are often much easier to compare. For example, in the following rule,

\[(\text{Service} = \text{http})(\text{Service} = \text{authentication}) \rightarrow (\text{service} = \text{smtp}) (0:6; 0:1) :---> (4)\]

The rule above is considered ineffective with the existence of the following rule:

\[(\text{service} = \text{authentication}) \rightarrow (\text{service} = \text{smtp}) (0:65; 0:1) :-------->(5)\]
The authentication is related only to the smtp operation; the http does not affect the other two item sets. Therefore, (service = http) can be ignored. Longer rules may introduce some redundant information. Removing them from the normal profile will reduce the false alarms.

4.4.2 Reconstruction of Episode Rules

Many FERs detected from the network traffic have some transitive patterns. Suppose we have two rules A → B and B → C in the rule set. Then, the longer rule A → B, C is implied. Since we reconstruct this rule from two shorter rules, the longer rule A → B, C becomes redundant. The reconstruction helps us split longer FERs into shorter ones. Rule pruning will reduce the false positive rate in an ADS. We are mainly interested in daily network traffic, like the TCP dump. For example, in the following rule,

(service = ftp, srcbyte = 1, 000) → (service = smtp) (service = authentication) --> (6)

is ineffective, because it can be reconstructed from the following two shorter rules:

(service = ftp, srcbyte = 1, 000) → (service = smtp) --> (7.a)

(service = smtp) → (service = authentication) --> (7.b)

This reconstruction is more powerful if the window size is large. For smaller window sizes, the occurrence of an episode may be longer than the window size, violating the basic assumption in FER formation. This reconstruction may result in fewer false alarms.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Traffic Connections and Their Anomaly and Normality Scores (Dataset-I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_proto</td>
<td>src_ip</td>
</tr>
<tr>
<td>icmp</td>
<td>202.77.162.213</td>
</tr>
<tr>
<td>tcp</td>
<td>172.16.112.100</td>
</tr>
<tr>
<td>icmp</td>
<td>202.77.162.213</td>
</tr>
<tr>
<td>icmp</td>
<td>53.88.213.15</td>
</tr>
<tr>
<td>icmp</td>
<td>202.77.162.213</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>icmp</td>
<td>202.77.162.213</td>
</tr>
<tr>
<td>tcp</td>
<td>194.27.251.21</td>
</tr>
<tr>
<td>icmp</td>
<td>202.77.162.213</td>
</tr>
<tr>
<td>icmp</td>
<td>202.77.162.213</td>
</tr>
</tbody>
</table>

5 WEIGHTED SIGNATURE GENERATIONS FROM DETECTED ANOMALIES

In this section, we present a weighted signature generation algorithm to characterize anomalies detected by the ADS. The signature-based SNORT imports these
signatures and detects the same attacks subsequently. The traffic data set is a normal relation table consisting of N connections. A connection ci has M attribute-value pairs < aj; vi;j > , where 1 <= i <= N and 1 <= j <= M. The attributes are selected from the connection features and temporal statistics summarized in Table 1.

The ADS assigns an anomaly score and a normality score for each connection after processing a traffic data set. The anomaly score indicates the degree of anomaly that a connection is deviated from normal traffic, whereas the normality score indicates how close a connection is related to normal traffic. The ADS assigns the anomaly and normality scores to a given connection by comparing its FERs with the normal profile. In our ADS, the sum of anomaly and normality scores is normalized to be 1. Note that the sum is not necessarily a constant in other anomaly based detection systems. Table 2 shows an example of traffic connections and their anomaly and normality scores. Given a set of connections, as well as their anomaly and normality scores assigned by ADS, the problem is reduced to discover the most specific and discriminative patterns of abnormal connections. A connection pattern corresponds to a subset of < attribute; condition > pairs, which determine possible, attribute values in the episode. We define the anomaly or normality scores of a pattern as the sum of all anomaly or normality scores of all connections matching the pattern. We define signatures as those patterns that have high anomaly scores but relatively low normality scores.

5.1 Weighted Signature Generation

Let I = {i1; i2; . . . ; id} be a set of distinct items and T = {t1; t2; . . . ; tN} be a set of transactions. Each transaction ti consists of a subset of items from I. An item set X is a subset of I, and its support sup(X) is the fraction of connection containing X.
We find the frequent item sets whose support exceeds a user-specified threshold (min_sup). We represent each connection as a transaction and each <attribute; value> pair as an item. The pattern of <attribute; condition> pairs are constructed from the frequent item sets.

Fig. 5 shows the functional components in the signature generation unit introduced in Fig. 2. First, similar abnormal connections are grouped together using a clustering analysis. Then, signatures are extracted for each group sharing some common characteristics. Attributes values of each connection are encoded into item numbers for mining the abnormal attribute-value pairs. After eliminating nondiscriminative item sets, the frequent item sets are decoded into <attribute; condition> pairs to form the anomaly signatures.

5.2 Cluster Analysis and Attribute Preprocessing

In traffic connections, different types of unknown attacks are often mixed together. The anomaly and normality scores assigned by ADSs are not sufficient to classify unknown attacks into different groups. It is difficult to discover accurate patterns or signatures for different types of attacks if they are mixed together. Different attacks have skewed distribution on connection volume. For example, different denial-of-service (DoS) attacks could have a large volume of short connections, whereas remote-to-local (R2L) or user-to-root (U2R) attacks could only have a few connections. Table 3 illustrates the attribute discretization and encoding in a Dataset-I. To separate different types of attacks, we use cluster analysis to group similar attacks together and generate signatures for each attack class.

We use the density-based clustering algorithm by Ester et al. [9] to obtain the partitioned cluster. Connections in low-density regions are classified as noises and thus omitted. Since each connection is characterized by symbolic and continuous values, we discretize the attribute values such as src_bytes in Table 3 by the entropy method introduced by Fayyad and Irani. We encode each pair <attribute; value> with an item number. Each item number has six digits. The first two digits represent the attribute index. The remaining digits are discrete symbolic values. For example, the pair <proto; tcp> could be encoded as 3 x 10000 + 2 = 030002, where 03 is the index of attribute prot and 0002 is the index of the attribute value tcp.

Our signature generation is based on a weighted frequent item set mining (WFIM) framework. Algorithm 2 extends from this framework to generate anomaly scores. Let $w_i$ be the weight of connection $t_i$, the weighted support of an item set $X$ is defined by $wsup(X) = \sum_{t_i \geq X, t_i \in T} w_i / \sum_{t_i \in T} w_i$. The purpose is to discover all item sets whose weighted supports are above the minimum support (min_wsup). We call those item sets weighted frequent item sets (WFIs). The anomaly score weighs a connection. The min_wsup helps select desired signatures.
Algorithm 2. Weighted a Priori Algorithm for Generating Signatures from Anomalies Detected

1: INPUT: A set of items I, a set of connections (that is transactions) T, weight wt of connection t, and minimum weighted support min_wsup
2: OUTPUT: Weighted frequent item sets X with wsup(X) > min_wsup
3: W = \sum_{t \in T} wt; 
4: k = 1; 
5: L1 = \{i | i \in I \land wsup(i) > min_wsup\}; \{ Find all weighted frequent 1 item sets \} 
6: repeat
7: k = k + 1;
8: Ck = apriori_gen(k_1) \{Generate candidate item sets\}
9: for each connection t \in T, do
10: Ct = subset (Ck; t); \{Candidates contained in t\}
11: for each candidate item set c \in Ct, do
12: c.weight += wt; \{Add connection weight\}
13: end for
14: end for
15: Lk = \{c \in Ck|c.weight / W \geq min_wsup\};
16: until Lk = 0;
17: return X = \bigcup Lk.

This algorithm is specified for weighted anomaly signature generation. If the support of an item set exceeds min_sup, then all its subsets must be supported. This principle is used by the a priori algorithm to effectively prune candidate item sets. We add items to item sets that are sufficiently large. We follow a weighted a priori principle: If the weighted support wsup(X) of an item set X exceeds min_wsup, all of its subsets have their weighted support exceeding min_wsup as well.

5.3 Signature Extraction and Mapping

For a given min_wsup threshold, the Weighted Frequent Item Set Mapping (WFIM) scheme discovers all WFI whose weighted supports exceed the threshold. When we consider many features as discussed in Section 3.2, the number of discovered WFI could be too large to construct the precise anomaly signatures. In particular, there are many redundant WFI that have to be pruned before signatures can be generated. Instead of using all WFI, we adopt a notion of maximal weighted frequent item sets (MWFI) as a compact representation of all WFI used. An MWFI is defined as a weighted frequent item set for which none of its immediate supersets has a support above the min_wsup threshold. Since any subset of an MWFI is also a WFI, the number of WFI that an MWFI represents will increase exponentially with the number of items in an MWFI.

After the MWFI of abnormal connections are discovered, we extract signatures by decoding the item numbers in MWFI into < attribute; condition > pairs. The < attribute; condition > pairs represent the abstract signatures of the detected anomaly.
These abstract signatures are then mapped into specific signatures of a target IDS system such as SNORT or Bro.

TABLE 3
Mappings between Connection Attributes and SNORT Rule Keywords

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>SNORT Rule Keyword</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_proto</td>
<td>protocol</td>
<td>IP protocol type</td>
</tr>
<tr>
<td>src_ip</td>
<td>source IP address</td>
<td>source IP address</td>
</tr>
<tr>
<td>dst_ip</td>
<td>destination IP address</td>
<td>destination IP address</td>
</tr>
<tr>
<td>service</td>
<td>destination port No.</td>
<td>Service type, e.g. http</td>
</tr>
<tr>
<td>icmp_type</td>
<td>itype</td>
<td>ICMP message type</td>
</tr>
<tr>
<td>src_bytes</td>
<td>Dsize</td>
<td>Packet payload size</td>
</tr>
<tr>
<td>flags</td>
<td>Flags</td>
<td>TCP flags</td>
</tr>
<tr>
<td>land</td>
<td>Samcip</td>
<td>Source and destination addresses are the same</td>
</tr>
<tr>
<td>src_count &lt;n&gt;</td>
<td>threshold: track by_src, count &lt;n&gt;</td>
<td>No. of connections from the same source</td>
</tr>
<tr>
<td>dst_count &lt;n&gt;</td>
<td>threshold: track by_dst, count &lt;n&gt;</td>
<td>No. of connections to the same destination</td>
</tr>
</tbody>
</table>

Table 4 shows how connection attributes are mapped into the keywords in SNORT rules. Besides the compact representation of frequent item sets, we eliminate indiscriminative ones that appear very often in normal traffic. Since the min_sup threshold is relatively low to discover frequent item sets in normal traffic, the number could be very large. Recall that in the MWFI for Dataset-I, the < attribute; condition > pair is decoded as follows:

(ip_proto = icmp), (icmp_type = echo req), (1, 480 <= src_bytes < 1, 490); (dst_count > 10).

Using the attribute mappings in Table 4, we translate the signature into a SNORT rule as follows.

alert icmp$EXTERNAL NET any <> $HOME NET any (msg :“ possible pod attack”; itype : 8; dsize : 1, 480 <> 1, 490; threshold : type both, track by dst,count 10 seconds 1; sid : 900, 001; rev : 0; ).

------> (9)
6. CONCLUSIONS AND FURTHER RESEARCH

We summarize major contributions and make some suggestions for further work on automated detection of intrusions and anomalies in an open network environment. The co-operative IDS/ADS system applies to protect any networked systems, including LAN-based clusters or intranets, large-scale computational Grids, and peer-to-peer service networks, and so forth.

For further research, we suggest the following approaches for continued research and development effort. Distributed CIDS will advance state of the art in co-operative intrusion detection. Extensive benchmark experiments are needed on the DETER test bed for this purpose. Extending a centralized CIDS to a distributed one is highly recommended with strong collaboration over multiple IDS sites. This offers a logical solution to protect Grids, clusters, intranets, and so forth. Cyber trust negotiations and frequent alert information exchanges among distributed IDS sites are the key research issues yet to be solved.

7. REFERENCES

Abstract

The proposed paper attempts to address issues of personal privacy in a world of computerized databases and information networks from security technology to data protection regulation to Fourth Amendment law jurisprudence -- typically proceed from the perspective of controlling or preventing access to information and argued that this perspective has become inadequate and obsolete, overtaken by the ease of sharing and copying data and of aggregating and searching across multiple data bases, to reveal private information from public sources. To replace this obsolete framework, we propose that issues of privacy protection currently viewed in terms of data access be re-conceptualized in terms of data use. From a technology perspective, this requires supplementing legal and technical mechanisms for access control with new mechanisms for transparency and accountability of data use. In this paper, I present a technology infrastructure -- the Policy Aware Web -- that supports transparent and accountable data use on the World Wide Web, and elements of a new legal and regulatory regime that supports privacy through provable accountability to usage rules rather than merely data access restrictions.

Objective of this document to record solution definition as advised by many corporate legal counsel and provide technological solutions to ensure the data protection and data privacy acts across multinational implementation of applications.

Keywords: Data Privacy, Database, Technical Solutions, Data Protection, Data, Privacy, World Wide Web, Data Privacy Act
Application of Cognitive Styles and Inference to Project Management

Dr. David J. Sudbeck, Washington University

Abstract

Project managers increasingly need better and more-integrated inference tools to handle their complex tasks. In particular, managers deal on a daily basis with uncertainty and inconsistency in their environment and in their facts at hand.

1 Introduction

Managers need better tools to handle uncertainty and inconsistency in project management and development. Traditional approaches have endeavored to quantify uncertainty in terms of metrics such as reliability (Sudbeck et al. 1989) while other approaches have endeavored to reduce uncertainty.

March and Simon (1958) noted the process of inference can be used to reduce uncertainty, and Simon won the Nobel prize in part for his work on bounded rationality (Simon 1960) and the need to infer ‘good-enough’ (and not necessarily optimal) solutions that ‘satisfice’ problems. In particular, Simon promoted the inference mode of ‘discovery’ or ‘abductive logic’ (Simon 1977) as a tool to find these solutions.

Along with the problem of uncertainty comes the problem of inconsistency. The groups involved in project planning have different viewpoints. For example, an analysis of a large corporation identified “12 different functional areas of business addressed by 26 different projects” (Henninger 1996).
Different viewpoints can stem from different dimensions of approach (i.e. leadership dimensions), from different dispositions of participants, and/or from different preferences of participants (thinking styles, etc.). Among these thinking styles are different uses of inference modes (see Figure 1).

A project manager must deal with these views and resultant inconsistency in the project plan. Approaches to inconsistency in planning have included a) preventing inconsistency (by fiat, by template, by threat, etc.) and b) resolving inconsistency as it is encountered.

Cognitive styles including approaches based on inference play an important role in what a project manager does and new tools are now available that project managers should use, such as progressing from old case-based management to newer inference-based computer-aided project management and agile methods.
2 Background

Project management is understood here as the managing of resources to complete projects according to constraints of time, cost, and quality (i.e. how well it meets project specifications).

A new agile version of project management is being promoted by IBM and other industry leaders. In the agile philosophy, there is less emphasis on pre-planning and more on flexible revision of specs by various agents of the team. As a noted agile author has said, the members of a team derive a specification by ‘scrapping it out’ (Ambler). Various researchers have begun to investigate impediments to applying these new methods in traditional development environments with differing management styles (Misra et. al.).

3 Model

Traditional project management has included inspection and review of plan specifications. Such an approach was fallible and computer tools were soon introduced to help with this task. These tools included tools for knowledge-based software development, case-based reasoning tools, and data mining and neural network approaches.

As Russo and Nuseibeh (2000, p. 14) explain

Specifications are assumed to be composed of many partial specifications (typically developed by different stakeholders), related to each other by means of pre-defined “consistency rules”. Each partial specification may or may not contain logical inconsistencies. However, the overall specification is defined to be inconsistent whenever at least one of the pre-defined rules is violated.

There are numerous variants of logic that have been enlisted to combat the problem of inconsistency. Three of the most popular variants are the traditional deductive and inductive logics, and the newer discovery or abductive logic.

Deduction can be thought of as traditional rules of the sort if $A$ then $B$. Deductive rules were used in early expert systems and today are reflected in sophisticated business rule engines.

Induction can be thought of deriving if $A$ then $B$ from numerous observations of if $a$ then $b$, if $a$ then $b$ etc., where $a$ and $b$ are individual instances leading to the general case if $A$ then $B$. Several contemporary business engines incorporate inductive reasoning.

Abduction can be thought of as the ‘reverse’ of deduction (and abduction is sometimes called retroduction). Given an object $B$ and if $A$ then $B$, we conclude $A$ is likely. An example if we observer wet-grass, and we know if rain then wet-grass, we ‘abduce’ or hypothesize that earlier we had rain. The $B$ is sometimes called the goal or observation,
while the $A$ is the explanation (if rain then wet-grass). Abductive systems for business and project management are just now starting to be employed.

3.1 Analysis (Including Detection of Inconsistencies)

Detection of inconsistency in a spec can be seen as detection of a violation of a rule such as “for a sentence $A$ including a system variable or entity, it cannot be the case that both $A$ and not-$A$ can be inferred from the spec” (Russo and Nuseibeh, p. 9).

There are several ways in which abductive reasoning can be used in this detection. One way is to make a goal of $P$ and not-$P$, where $P$ is a predicate or symbol in the specification. If an explanation can be abduced that satisfied this contradictory goal, we know the spec is inconsistent. Moreover, the explanations (technical term ‘abducibles’) found that lead to the contradictory goal indicates to us where the spec is inconsistent. On the other hand, if no explanation can be abduced for any such goal, the spec is consistent.

3.2 Revision

There are a number of researchers who point out “inconsistency is inevitable in real large-scale specifications” (Russo and Nuseibeh, p. 18). Abductive inference is applied by these researchers to either handle the inconsistency (i.e. remove objectionable inconsistencies while leaving inconsequential ones alone) or provide ways to re-establish consistency. The related science of complex event handling claims the “key to understanding events is knowing what caused them” (Luckham, p. 10).

An inference-based example of a specification could be for movement of a locomotive train (example adapted from Menzies). If an event $E$ has a train moving at time $t_2$ we can infer the train started at time $t_0$, was running at time $t_1$, and had no brake on at time $t_1$.

$E$(moving, $t_2$) $\rightarrow$ $E$(started, $t_0$) $\land$ $E$(running, $t_1$) $\land$ not-$E$(brake, $t_1$) (I)

We posit a ground instantiation $EC(S) \rightarrow I$, with EventCondition at underlying time $S$. We then expect

$EC(S) \land I(Sc) \rightarrow I(Sn)$

Or, in other words, we expect ground-conditions for the specification plus the movement of the train at current time $Sc$ results in movement of the train at next time $Sn$. We then try to abduce a $D$ such that

$EC(S) \land I(Sc) \land D \rightarrow not-I(Sn)$
If we find such a D we know the spec is not consistent with the desired train-movement (i.e. perhaps the spec allows blizzard weather that halts the train; this ‘fact’ must be either resolved (i.e. spec for a snow-plow) or recognized as an allowed inconsistency).

If a D can be found to disprove the requirement, the D is an explicit indication of where the problem lies in the specification. The abduction approach thus has the advantage of being able to explain its conclusions.

### 3.3 Autonomous Agents

Work on social systems and agents in those systems have explored various ‘logics’ or ‘thinking styles’ identified in human agents. Abducible agents are proposed in (Kowalski and Sadri).

Such systems are particularly strong in temporal reasoning, where interaction of agents can quickly lead to conflicting requirements. These abductive systems recognize we cannot admit specifications that at the same point in time assert something and also deny that same thing. These systems can handle inconsistency across different times, recognizing a kind of evolution of requirements.

For example (Mello et al., p. 35), from the assertion

\[ Y \text{ will accept a request within 2 to 10 minutes after the request is made} \]

And the contradictory assertion

\[ Y \text{ will not accept a request in the next 5 minutes of the request} \]

An abduction system can infer the coherent

\[ Y \text{ will accept a request within 6 to 10 minutes after the request is made.} \]

These abduction systems can be made efficient enough to run in real-time through incremental operation (detecting inconsistencies as soon as possible), interleaving of assertions/retractions/abductions, and reduction of complex sentences to simpler more atomic sentences whenever possible.

### 4 Conclusion

Limitations of present tools for project management are increasingly being felt. In the realm of logics, abductive inference or the logic of discovery is finding increasing popularity.

An abductive approach, since it is based on logic, can work with just initial specifications, inferring inconsistencies. It can also quantify which inconsistencies are
significant, thus deciding which can be ignored. The well-known Bayesian net approach was designed for abductive inference and is found in several vendor products. (See Rodriguez et. al for applications to project management.)

Data mining and neural network approaches have the disadvantage of not being able to explain why they make their conclusions. A limitation of case-based reasoning (which is sometimes called rule-induction) is it assumes past-cases are available for analysis. Satoh (1996) has shown how case-based reasoning can be translated into a subset of abductive logic programming.

Several efforts have been recently devoted to extending logic programming to perform abductive reasoning. Abduction is a form of reasoning which allows one to compute explanations for observations. Moreover, it is a form of non-monotonic reasoning, since explanations which are consistent in a given context may become inconsistent when additional information is added. In fact, it is well-known that abduction provides an alternative formation of default reasoning.” (Giordano et. al., p. 1-2)

5 References


Luckham, David. 2002. The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems


Misra, S. et. al. 2006. The Organizational Changes Required and the Challenges Involved in Adopting Agile Methodologies in Traditional Software Development


Early Muslim Scholars and Binary Mathematics

Numan Tugan
Milwaukee, Wisconsin, U.S.A.
numantugan@yahoo.com

&

Mohammad Amin & Pradip Peter Dey
National University, San Diego, California, 92127, U.S.A.
mamin@nu.edu ; pdey@nu.edu

ABSTRACT:

Using the concept of dualism, early Muslim scholars envisioned the world as an expression of events in terms of two irreducible opposite elements. Whereas others saw this opposition as a source of conflict, these early Muslim scholars saw it as a source of cooperation and interaction, which led to the development of 0-1 mathematics. Historical evidence of their work, examples of their achievements, the thinking set and mathematical operations on this set will be presented. Illustrations of tables and calculations are included.

Keywords: Dualism, 0-1 Mathematics, Mathematical Thinking Set.

HISTORY

The concept of dualism is mentioned in the Quran, the holy book of Muslims, in many places. The following verse, “And He it is who has created all opposites (or pairs)…” [1], and similar verses clearly indicate that the universe is made of opposites and pairs, and is a place for humans to think, contemplate and learn lessons from these opposites. Contemplating these types of statements, Muslim scholars have expressed many different interpretations regarding this. Some commentators regard the term “pairs” as synonymous in this context with "kinds" [2]. Others see in it a reference to the polarity evident in all creation. Some say that it denotes the concept of opposites in general, as "sweet and sour, or white and black, or male and female,” that everything in creation has
its complement, "like high and low, right and left, front and back, past and future" [3].

The expression of these different views or opinions started with the earliest Muslim scholars of the seventh and eighth centuries, and continues even today. Discussion of their various opinions on the subject of dualism helped them to develop a new form of logical thinking process called "kalam." A rigorous scientific approach to this study allowed the development of a methodology for the description and analysis of creation, nature and human existence. Many valuable books were produced during this period, and these early scholars were succeeded by their equally erudite students [4].

After the first generation of Muslim scholars, the second generation and beyond undertook philosophical studies and translated the works of the early Greek philosophers, whose opinions on dualism varied considerably from their own. They came to consider the two irreducible opposite elements of dual pairs as sources of conflict with each other. One gains superiority over the other through conflict and defeat [5]. Those scholars who regarded philosophy as a foreign ideology and rejected its principles did not follow its methods, but used purely Quranic thinking methods. They understood opposites as the basic structure of things, a source of cooperation and interaction. They identified opposing forces not as enemies but as cooperating pairs, helping and completing each other. This type of thinking process helped them to understand the laws of nature and utilize them for their benefit [6].

During the 8th and 9th centuries, those scholars who followed the path of the Greek philosophers were active in politics and controlled all the education and the decision-making mechanisms of the “House of Power” of Caliph in Baghdad and in Cordoba. The other scholars were persecuted and dispersed all over the empire. They migrated to many different places, especially to Samarkand, Bukhara and Khwarizm, to the east. But the scholarly debate never ceased between these two groups. The 10th century produced a tremendous number of scholars, scientists and mathematicians [7]. In this century, a Muslim scholar named Muhammad ibn Mūsā al-Khwārizmī introduced zero as a number to the study of mathematics, as well as the methodical problem-solving technique called algorithm, which is derived from his name [8]. Territories such as India captured during this period of the 10th century were influenced by these scholars in science and mathematics [9].

This momentum of scholarly studies produced many scientist and academics throughout the centuries, which formed the basis for many of today's discoveries and theories. A few examples of these are the first mechanical robot of Al Jazary[10], the map of Ottoman Naval Captain Piri Reis[11], and the 17th century flying apparatus of Ottoman aviator Hazerfen Celeby[12], who flew with artificial wings more than 4.5 miles. One contemporary scholar even described the atom as a type of small motor or machine [14]. Many extant works of these scholars are collecting dust in libraries throughout the world, like treasures waiting to be rediscovered. Some libraries, such as the Suleimaniye Library in Istanbul, have undertaken massive projects to translate this kind of material and make it available to the public. Other valuable old books have not been read in centuries, and might never be seen again.
Our focus in this paper is (RML), another example of the early Muslim scholars' works upon which binominal mathematics is based. The concept of dualism is represented in 0-1 terminology to describe one aspect of the thinking process, which is decision-making. Muslim scholars were aware of the thinking process and they refer to it many times in their writings. For example, Said Nursi refers to the speed of though relative to the speed of other phenomena, saying, “There are many degrees of speed, such as the speed of sound, the speed of electricity, the speed of light and the speed of thinking or imagination.” [15] The thinking process was treated like any other natural phenomenon by the Muslim scholars. They described this phenomenon through logical expressions while explaining the concept which involves thinking and imagination in the terminology of 0-1 [16-20]. The concept of dualism is deeply embedded in these writings, one part of the framework of their way of looking at the world.

These scholars described the information which goes into the human brain in terms of small discrete entities of information or dots called Nuqat (singular nuqda). Information is stored in the brain also as discrete “dots.” When the information is recalled, it is in the same format, one dot at a time, consecutively. There is no simultaneous process, such that two different pieces of information or dots could go in or out at the same time. The speed of this process is what is referred to above as the speed of thinking or imagining. The information dots are so small and so close to each other, that it might appear that the dots are being processed simultaneously.

In the decision-making process, the stored information must be recalled and acted upon, again as a series of discrete dots. Because it would be impossible to consciously examine the dots of information individually, each dot is represented by a randomly generated symbol which is used for comparison to the symbol following it. A mathematical formulation to assist in the decision-making process is as follows:

The set of Nuqat is expressed by \((\mathbb{N}, \oplus)\), all \(a\) and \(b\) are members of \(\mathbb{N}\), such that there is an operation on this set expressed by \(\oplus\) (=dot operation) so that

\[
 a \oplus b = \{ 1 \text{ if } a = b, \text{ and, } 0 \text{ if } a \neq b \},
\]

(= expressed as same as)

If two dots of information coming out consecutively are the same symbol, then the result is 1. If two consecutive dots are not the the same, then they are opposite, and will negate each other, and then the result is 0. A result of zero is equated with a decision of yes, a result of one is a no. I will present a table of calculations as an example.
Muslim scholars have produced various books expressing the above ideas in different terms and interpretations. [21-24] These scholars approached the subject in two different ways, one of which is deterministic and the other probabilistic. Events in the decision-making process are classified as either deterministic (predictable) or probabilistic (random). There is very little information available on the deterministic concept [25]. Most of the books currently available were written on the probabilistic concept, which is somewhat different from today's study of probability. Further development of these concepts, combined with aspects of Raml, a variant of geomancy, are referred to in many subsequent writings. [21-24].

The study of these concepts could be applied to the areas of artificial intelligence, neuroscience, nanoscience, and robotics.

REFERENCES:
[1] - Quran, chapter (43:12)
[2]- Baghawi, Zamakhshari, Baydawi, Ibn Kathir
[6]- Sayyid Qutb  In the shades of the Qur'an
[7]- 10th century mathematicians (Al-Hajjâj ibn Yûsuf ibn Matar · Muhammad ibn Mûsâ al-Khwârizmî · Al-Abbâs ibn Said al-Jawhari · 'Abd al-Hamîd ibn Turk · Al-Kindi · Humayn ibn Ishaq · Banû Mûsâ · Al-Mahani · Ahmed ibn Yusuf · Thâbit ibn Qurra · Al-Hashimi · Muhammad ibn Jâbir al-Harrânî al-Battânî · Abû Kâmil Shujâ ibn Aslam · Sinan ibn Thabit · Al-Nayrizi · Ibrahim ibn Sinan · Abû Ja'far al-Khâzin · Al-Karabisi · Brethren of Purity · Abu'l-Hasan al-Uqlidisi · Al-Saghani · Abû Sahîl al-Qûhi · Abu-Mahmud al-Khujandi · Abû al-Wafî al-Bûzjânî · Ibn Sahl · Al-Sijzi · Labana of Cordoba · Ibn Yunus · Abu Nasr Mansur · Kushtyar ibn Labban · Al-Karaaji · Ibn al-Haytham · Abû Rayhân al-Bîrûnî · Avicenna · Ibn Tahir al-Baghdadi · Ali ibn Ahmad al-Nasawi · Al-Jayyani · Abû Ishâq Ibrâhîm al-Zarqâlî · Yusuf al-Mu'taman ibn Hud · Omar Khayyâm · Ibn Yahyâ al-Maghribî al-Samaw'al · Sharaf al-Dîn al-Tûsî · Ibn Mun‘im · al-Marrakushi · Nasîr al-Dîn al-Tûsî · Muhyi al-Dîn al-Maghribi · Shams al-Dîn al-Samarqandî · Ibn Baso · Ibn al-Banna · Kamâl al-Dîn al-Fârisî · Al-Khalîli · Ibn al-Shatîr · Qâdî Zâda al-Rûmî · Jamshîd al-Kâshi · Ulugh Beg · Al-Umawi · Al-Qalasadi · Ali Ku”çu · Al-Birjandi · Taqî al-Dîn · Muhammad Baqir Yazdi)

[9]- Vateshvara, Indian Mathematician.

[11]- Piri Reis:
[12]- Hezârfen Ahmed
[14]- Said Nursî

[15]- “The Words, Thirty-First Word, speed of sound, electricity light, and thinking (imagination)…” Said Nursî

[16]- Muhammad ibn Muhammad ibn Mahmud Abu Mansur al-Samarqandi al-Maturidi al-Hanafi (d. 333 AH / 944)

[17]- Jalâl al-Dîn Rûmî (1207-1273) and

[18]- Muhyî al-Dîn Ibn ‘Arabî (1165-1240),


[20]- Risale-i Hesab / Yegani Sinan Çelebi. Suleymaniye Library, Istanbul

[21]- Risale-i ilm-i remil , Suleymaniye Library, Istanbul

[22]- Risale-i Cifr-i Cami / Hüseyin el-Ahlati, Suleymaniye Library, Istanbul

[23]- Risaletün fi Ebvabi'r-Remil / Cemaleddin Ömer b. Osman ez-Zennati, Suleymaniye Library, Istanbul

[24]- Fenn Remil, Suleymaniye Library, Istanbul

[25]- İbalül Temassî'l-Küre li-Nukat Muhtelifî fî Zeman V. / Molla Kasim el-Germiyani el-izari, Suleymaniye Library, Istanbul
Acceptance Test Driven Story Card Development for XP (Agile Software Development)

Chetankumar Patel
School of computing
Software engineering research group
Leeds Metropolitan University
Beckett Park, Headingly
Leeds, LS6 3QS, UK

Muthu Ramachandran
School of computing
Software engineering research group
Leeds Metropolitan University
Beckett Park, Headingly
Leeds, LS6 3QS, UK

Abstract
Requirement elicitation is a process to collect information from the end users or stakeholders of system. In traditional software development models requirements are predefined and sent to the developers to do analysis from various stakeholders. Requirement elicitation and quality related to requirements are the challenging issues in the software development life cycle. XP, recommend an on-site customer to represent their requirements through user stories on story cards. Generally customers have rarely a general picture of the requirements or system in their mind which leads problems related to requirements like requirements conflicts, missing requirements, and ambiguous requirements etc, and not address non-functional requirements from exploration phase. Two third of the projects are failed because of ambiguous and incomplete user requirements, and poor quality of the requirements. Acceptance testing is also one of the key issues, which is often left unsolved or uncovered to verify user requirements in XP (Agile software development methodology). To solve these problems we propose A New Improved Requirement Elicitation (Gathering) Process framework for story cards in XP to improve quality of user stories to capture functional and non-functional user requirements on story cards. Our approach involves adopting acceptance tests right from story cards as soon as requirements are clearly understood. We also aim to demonstrate how acceptance test will improve story card driven agile software development (XP). We believe this will enhance the popularity of Agile based software development.

Keywords
Extreme Programming, Requirement Engineering, Acceptance Test, User Story, Story Card

Introduction
User requirements elicitation is one of the early stage of the software project development which records the description of the system how to it has to work or what system has to do or statement of how it should do it (Somerville and Kotenya 2000). Requirements elicitation process is one of the challenging processes in the software development methods. In traditional software development methods end users or stakeholders predefined their requirements and sent to the development team to do analysis.
and negotiation to produce requirement specification. In traditional software development methodology it is expensive to deal with requirement change after careful analysis and negotiation. This problem is well tackled by the XP, which is one of the agile software development methodologies.

Extreme programming is a lightweight and incremental software development methodology which directly involves their customers and developers. XP is actually consisting of 12 practices based on the four values (Beck 2000). Such as test driven development, continuous integration and refactoring can be practiced by individual developer because they directly gives benefit to the developer while some of them are hard to implement individually like Planning game, acceptance testing, and small releases because they require buy in and participation of people from the team or from outside (Rogers 2004).

In XP, Development starts with planning game where customer writes user stories on story cards. Those cards are estimated by the developer, based on those estimation customer priorities them depends on their needs to establish a timebox of an iteration. Developers develop those story cards through pair programming and test driven development. At last customer provides acceptance test to accept the developed functionality. In between they consider all of the XP practices in mind to improve the quality of the software.

Story cards are playing vital role in XP. It is the unit of functionality and priorization. Story cards are one of the important aspects of the XP. Story cards are written by the customer in XP to articulate their business needs. According to Cohn story cards must be testable, estimatable, valuable to the customer, small and independent (Cohn 2004). These story cards must be written by the customer because they know their business need very well compared to developer. Consider the figure 1 which shows a traditional story card as proposed in (Beck 2000).

![Figure 1 Traditional Story Card Used in XP Projects](image)
XP strongly recommend an onsite customer to write their business need. Business is well understood by the customer. Generally customers have rarely a general picture of the requirements or system in their mind (Kotyana and Sommerville 1997). Different stakeholders have different needs. Often an onsite customer is not complete and precise during story cards writing process and do not predict their working system without seen it, which will lead problems related to requirements like requirements conflicts, missing requirements, and ambiguous requirements etc, and not address non-functional requirements from exploration phase. Incomplete requirements, requirements conflicts, missing requirements, and ambiguous requirements are going to be a defect on working system. Two third of the projects are failed because of ambiguous and incomplete user requirements, and poor quality of the requirements. In most of cases non-functional requirements are not covered in the exploration phase. Non-functional requirements play a significant role with functional requirements in software development methods. Non-functional requirements improve the quality of the software. If they are not considered in early stage then it is difficult to address them in final product. In XP there is a question related to quality is unanswered. There is still scope to improve story cards.

Acceptance tests are the test written by the customer to evaluate that the developed functionality works according to their requirements or business needs. Acceptance testing is one of the challenging issues in the software development lifecycle as it is often neglected on an XP projects. It is perceived as just being to hard to get right (Rogers, R.O. 2004), Which causes a problem related to requirement quality and become difficult to check weather system does what it’s intend to do it or not. Massive development efforts based on unproven business requirements are a waste of time effort and money (Bodell, R. 2003). The CHOAS report published in 1995 shows that almost half of the cancelled projects failed due to a lack of requirements engineering effort and that a similar percentage ascribes good requirements engineering as the main reason for project success([Chaos 95]

As a result of this investigation we propose a new prototype to improve requirement elicitation process in XP and new improved framework of story cards to write user stories. This will help to customer and developer to improve the quality of the user stories or story cards, and to address functional and non-functional requirements on story cards. In this paper we discuss about extreme programming and requirement elicitation process through story cards first and then after the challenges and problems on XP software development methodology. This is followed by discussion of related research regarding to an improvement framework of the story cards for XP based projects.

**A New Improved Requirement Elicitation (Gathering) Process in XP**

Extreme programming relies on customer interaction to identify or to tell which features to implement in next release. Stories are written by customers perhaps with the help of agile development member. Therefore each user stories and story cards must be split
into a unique and independent requirement. In many cases a process to get user-visible requirements is hard to impossible. They do not supply any infrastructure requirements.

As a part of this investigation, in our approach we recommended a customer or stakeholder who is on site has comprehensive application domain and business knowledge to put a developer into the picture.

Application domain knowledge and customer business knowledge from customer will help developer to focus on stakeholders’ business needs and requirements and help them to cover any missing functionality. However it is a question of how much risk is involved by not spending sufficient time selecting the right customers and prioritizing them and their respective requirements. It is impossible to have all requirements just from one person. In the following Figure 2 we propose a new improved requirements elicitation process in XP, which emphasizes on customer and developers communication on customer’s domain and business knowledge. XP recommend that developer and customer discuss about their proposed system. We assume that the onsite customer is a domain expert or he understood their business requirement very well and he or she is able to explain their business needs at a glance to Developers to put them into picture of system. Customer business knowledge helps developers to understand how the system will affect and interact with the different part of the business, and help to identify and understand different stakeholders, who are directly or indirectly affected by the system.

Stakeholders identification is one of the important and challenging parts of the requirement elicitation process in XP (Agile software development) and traditional software development. It is quite difficult and risky to get information or collect requirement for whole system from the single prospective. So we come with the idea of identify all stakeholders and their needs requirements to build a good system. XP most likely assume that an onsite customer already have a wide view of all requirements of the system but in most cases it is not and in this case XP also struggling to find out all stakeholders requirement form one person and priories them according to that persons limited knowledge of requirements. Here developers going to ask a series of question to identify stakeholders of the system who are going to affected directly or indirectly by the system.
At the end of successful discussion between customer and developers, customer starts story elicitation process and writes the draft statement of requirements on story cards. These story cards are further analyzed, which assist to customer and developer to identify any problems and missing functionality. Missing functionalities become defect in working software. This scenario will help developers to focus on non-functional requirements. Also helps to keep in mind what they have to do to improve all the aspects of the business through structured system. Identified problems are being negotiated between developers and customer to acquire story cards.

**Acceptance Test Driven Framework of Story Cards**

We introduce acceptance tests directly on the story cards. Acceptance tests on exploration phase for two reasons 1) to identify the verifiable requirements from different stakeholders and 2) This will help developer in effort estimation. Following figure 3 shows new template of the story cards.
There are certain advantages of the acceptance test to introduce directly on the story cards. Acceptance test do the validation of the requirements, helps to write the user stories on the story cards on a verifiable form instead of vague, ambiguous or best guess requirements. Acceptance test also improve the estimation of the story cards as well. If story cards contain too many acceptance test than that means story card is big to fit in iteration and to do estimation. So acceptance test on story cards give an idea to split stories into a smaller stories to fit in iteration, and smaller stories wit acceptance test have a perfect estimation. Acceptance test directly tell to customer what they have to develop instead of guess.

We introduce a new section called point to consider on the story cards. This will helps to identify the goal of the users and other non-functional requirement related to that story card. This is one kind of the constraint where developers and end user or customer add their requirements which are directly or indirectly apply to the story cards or entire set of story cards of whole software.

We apply this story card to the real world user story and explained how it improves the requirement or user story on story cards. Figure 4 shows the story cards written through a traditional way, which is just a short and vague statement of user requirement.
This traditional story card does not provide enough information. It is really difficult for the customer to estimate and test the user requirement as well. Also this is not come with any kind of the user acceptance test as well. Acceptance testing is remain unsolved in most of the projects at moment, which is need to be consider.

<table>
<thead>
<tr>
<th>STORy CARD NO:</th>
<th>Project Name</th>
<th>E-Commerce</th>
<th>Estimation: 4 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Story Name:** User: Registration  
**Date:** 16/06/2007 1:30 PM

**STORY:**
User needs to register with unique username and password before purchasing anything from the online store

**Acceptance Test:**
1. User Id must be unique
2. Try to register with duplicate user id and Password
3. Try to register user name only
4. Try to register with password only
5. Forget Password Link

**Note:**  
User Can View or Visit store as a Visitor but needs to register before purchasing anything

**Risk:** Low

**Points to be Consider:**  
There isn't any non-functional requirement at this stage

Figure 5 User story captured through the improved requirement elicitation process by using the new acceptance test driven story card template
The figure 5 shows the same user story captured through the improved requirement elicitation process by using the new acceptance test driven story card template, which captures user requirement precise, correct, testable or verifiable and enough sort of information with acceptance test. This will help us to estimate human efforts to convert user story cards into working code through the relationship of acceptance tests and effort estimation.

**Conclusion**

Our previous work on knowledge base support on story cards, which provides guidelines to capture requirements on story cards. In this work we introduced an improved framework of story cards which helps developer and customer to address all functional and non-functional requirements of the software. This will improve the story cards and story card based agile software development. We also focus here to write acceptance test with story cards instead of during the iteration planning.

**Future Work**

This work is just a prototype and we going to adopt this concept on to real time projects. We are also working on guideline based story card development and value oriented prioritization matrix to priorities story cards and customer priority.

**References**


CHAOS, Software development report by the Standish group 1995).


Analysis of Near-infrared Phase Effect on Biometric Iris Data Demonstrates Viability of Iris-scan Biometric Device for Identity Authentication in Public Venues

Brady R. Stevenson and Gordon W. Romney
National University, San Diego, CA

Abstract
Correctly and quickly validating the identity of individuals to permit entry to a venue is an increasingly challenging issue in modern society. One common objective is to allow entry to authorized individuals and to restrain all others. Transportation and border venues, that process a high percentage of human movement, are especially critical and vulnerable from a national safety perspective. In this context, the selection of biometric technology that uniquely identifies a human and meets specific operational criteria is fundamental. Finding a statistically reproducible process, quick to capture data, non-invasive to humans, and which functions in a variety of ambient environments becomes the challenge.

Iris-scan technology was selected as the candidate identity authentication mechanism of this research as it is quick to capture data, is non-invasive and produces statistically verifiable data. Prior research had left unresolved the issue of whether normal, ambient environments with variable levels of fluorescent lighting such as airports, rail stations and passport control kiosks might produce unreliable data due to wavelength sensitivity of iris cameras. Prior studies of iris data variations from infrared wave interference of halogen, incandescent, and sunlight were inconclusive relative to near-infrared wavelengths between 700 and 900 nm. Such wavelengths are produced in fluorescent light sources, frequently found in the targeted venues, that commonly emit 30% of their heat in the near-infrared spectrum. The experiment utilized a Panasonic BM-ET300 iris camera scanning nine hundred fifty-four (954) subjects with a variety of ethnic eye configurations using variable background light of a normal fluorescent lamp. Results showed increased heat from fluorescent illumination has no statistical significance due to the near-infrared waves from the camera and the fluorescent lamp. The results of the test confirm that data loss will not occur as heat is increased in the fluorescent light source. Hence, image corruption which would produce unreliable authentication data would not be caused by increased levels of fluorescent light conditions. Use of this particular iris camera in normal ambient lighting conditions should prove viable for biometric identity authentication purposes.
Challenges in Vendor Collaboration in A Mix Mode Offshore Centre - Coexistence Of Captive And Vendor Offshore

Satyen Vats & Rathi Dasgupta

Abstract

Companies with head quarter in US and Europe more and more are using Offshoring to leverage cost and augment skilled staff. Move to create a wholly owned captive centre is considered to be a smart move to save cost and create reusable competency. The most successful model is the ecosystem of coexistence of captive center and vendor partners to do projects. Due to increasing competition in resourcing companies are using vendors to do staff augmentation for spiked requirement during project life cycle. The current research focuses on the challenges in vendor management and the best practices model.

1. Introduction

Non-availability of knowledgeable resources, reduction of fixed cost, ever increasing cost of wages and attrition are some of the challenges, faced by captive centers [4], which are addressed through engagement of offshore vendors. This engagement promises best of both worlds in terms of flexibility of vendors [1] and control of a captive center. But it also creates a complex set of problems that require a carefully thought, planned and focused approach towards vendor management [2] [3].

The vendor management challenges involve the following root causes:

1. Business objectives –
   a. Engagement of offshore IT vendors with offshore captive centers involves the problem of conflicting business objectives.

2. Ownership and expectations –
   a. Conflicting business objectives for different entities engaged in a typical captive center project can easily lead to, in absence of a universally agreed and accepted project charter, conflicting/different expectations and non-uniform understanding of ownership of various aspects of a project, on part of all entities.

3. Development processes and methodologies –
   a. Almost always, vendor’s development processes and methodologies are different from captive or client’s internal development processes. Absence
of a common process framework during the execution of a project leads to significant cost increase and delays on projects.

4. Corporate culture –
a. Differences in corporate culture present serious obstacles to working relations between the vendor and captive organization –
   i. Organizational structural and reporting
   ii. Organizational values and behavioral norms
   iii. Management approach –
      1. Decision making
      2. Risk appetite
      3. Monitor and control mechanism
   iv. Employee evaluation and skill level assessment

5. Knowledge transfer –
a. Knowledge transfer from vendor resources to captive center resources is considered to be an important success factor for a project as well as long term success of the captive center. Multiple

6. Skill level –
a. Domain knowledge
b. System understanding
c. Technology competency

7. Lack of team camaraderie

2. Conflict of business objectives

Engagement of offshore IT vendors with offshore captive centers involves the problem of conflicting business objectives. It is perceived that captive centers and vendors are competitors and, in fact, this perception is true to certain extent. Management of parent organization does not share and discuss the vision, objectives and long term plans associated with the captive center. Vendors, in the absence of any clarity regarding their future relationship with the parent organization and captive center, become skeptical of the captive center.

   i. Lack of clear accountability
   ii. Captive center is considered as threat to business
      ▪ Vendors create nexus with parent organization's employees who are against captive center
      ▪ Miss-representation of facts to highlight captive center as incompetent
   iii. Lack of support for captive center's growth from vendor
      ▪ Resistance towards sharing of knowledgeable resources
      ▪ Resistance towards working in a partnership as a single team
   iv. Vendor and captive center compete against each other
- In-effective communication
- Lack of co-operation among project teams
- Lack of camaraderie among team members
- Blame game

3. Ownership and expectations
Conflicting business objectives for different entities engaged in a typical captive center project can easily lead to, in absence of a universally agreed and accepted project charter, conflicting/different expectations and non-uniform understanding of ownership of various aspects of a project, on part of all entities.

4. Different processes and methodologies
Vendor’s development processes and methodologies are different from captive or client’s internal development processes. Absence of a common process framework during the execution of a project leads to significant cost increase and delays on projects.

  i. Different software development life-cycle methodology and framework
     - One organization may prefer RUP like approach of exact specification and detailed documentation while other may prefer agile methodologies.
  ii. Different project management approaches
     - Vendors adopt client's management practices while working at client location but their offshore project management practices are very different.
  iii. Different level of process maturity
     - Vendor may be at quality certification of CMM Level 5 where as the captive center may be at Level 2.
  iv. Different quality assurance processes
     - Application of different quality assurance approaches to a single application-code leads to in-consistent quality of code.
  v. Different code libraries, development and testing tools and version maintenance and documentation systems
     - Developers and testers are used to certain libraries, tool and systems. Use of multiple tools and systems, for same set of activities, on a project leads to problems of code integration and knowledge management towards later stages of the project and during maintenance and support activities.

5. Corporate culture differences
Differences in corporate culture present serious obstacles to working relations between the vendor and captive organization.

  i. Organizational structural and reporting
  ii. Management approach –
- Decision making
- Risk appetite
- Monitor and control mechanism

iii Employee behavioral norms and expectations
- Education and dedication are great strengths of offshore, particularly Indian, staff but they lack people skills required for a multi-national and multi-cultural work environment. These people skills are at the root of ineffective communication and miss-match of expectations at the team level.

iv Employee assessment and rewards and recognition
- Performance assessment and rewards and recognition are an important aspect of employee motivation. Captive centers follow parent organizations assessment and recognition policies that are very different from vendors’ practices. Presence of non-uniform assessment and recognition practices and different assessors as well, for team members of the same team also contributes towards lack of team camaraderie.

v Skill evaluation and project allocation
- Domain knowledge
- Technology competency
- System understanding

6. Knowledge transfer
Knowledge transfer from vendor resources to captive center resources is considered to be an important success factor for a project as well as long term success of the captive center.

7. Conclusion & Recommendation
Price is a key differentiator, in evaluation of vendors, for a short term relationship and small piece of work. But outsourcing has moved beyond commodity-based, cost-driven engagements and saved-dollars is not the only value-add provided by offshore vendors. To derive additional benefits from outsourcing, customers need to forge a long-term strategic partnership with the vendors. Even though customers are highly focused in their selection process of vendors but they often miss this relationship aspect of vendor management. Many vendors are large successful companies and they look for a win-win relationship with their customers. Whenever customers take the conventional approach of squeezing the maximum out of vendors and vendors are not able to make their margins from an engagement, they loose focus from such engagements and make other profitable engagements as higher priorities.

Long term partnership with vendors –
Defining space of operations for customer, captive center and vendor, and sharing of long term goals and expectations are the key steps required in establishing a
strategic partnership with the vendors.

i. Customer and vendor need to create blueprint of a win-win relationship, with clearly defined role for the vendor, at early stages of their engagement.

ii. Price bench-marking, SLA compliance and contractual aspect of engagement should be replaced with business value, change management, knowledge transfer, domain knowledge, and process maturity.

iii. Create and maintain a system of accountability on all participants

iv. Vendor relationship should be maintained at all levels.

v. Establish constant dialogue to better understand each others business, business objectives, culture to facilitate value addition and better solutions.

vi. As vendor engagements are changing from SLA based contracts to strategic partnerships, customers and vendors need to innovate on pricing of services. A revenue based payment model can bring greater amount of ownership and accountability from the vendor.

Uniform processes and methodologies:

Customer and vendor teams should evaluate the differences among their processes and agree on a common development framework, project management approach and quality assurance methodology along with common code libraries, version control system and development and testing tools.

i. A governance body should be created to monitor and control progress of projects.

ii. Well defined ownership should be defined and compliance should be ensured.

iii. Tollgate process should be established to ensure completion of each phase before moving to next phase.

iv. Milestones should be clearly defined along with details of deliverables and completion dates.

v. An air-tight communication plan and reporting mechanism should be established to ensure early identification of issues and their quick resolution.
References

1. Vendor Management Tips – Building Relationship
   (http://www.computerworld.com/managementtopics/management/story/0,10801,99816,00.htm )
2. Vendor Management (http://cio.ittoolbox.com/topics/t.asp?t=426&p=426&h1=426)
3. A new way to manage vendors
   (http://www.cio.com/article/2034/A_New_Way_To_Manage_Vendors)
4. Dynamics of Captive IT centers, Rathi Dasgupta, Proceeding of 6th International Conference on Smart Sourcing, Atlantic City, 2008
COMPARATIVE STUDY AND ANALYSIS OF FLANK WEAR OF SINGLE POINT COATED AND UNCOATED TURNING TOOL INSERTS USING ACOUSTIC EMISSION SIGNAL PROCESSING AND NEURAL NETWORK TECHNIQUES

Mr. N.PRASANNA,
Assistant Professor- Department of Industrial and Production Engineering,
Sri Jayachamarajendra College of Engineering, Mysore, Karnataka, INDIA
E-mail: nprrasa52@yahoo.com

Mr. SRINIDHILR,
Assistant Professor- Department of Mechanical Engineering,
Sri Jayachamarajendra College of Engineering, Mysore, Karnataka, INDIA
E-mail: shreenedhe@gmail.com

Dr. M.SUKUMAR,
Professor & Head- Department of Instrumentation Technology,
Sri Jayachamarajendra College of Engineering, Mysore, Karnataka, INDIA

Dr. C.S.VENKATESHA,
Professor-Department of Mechanical Engineering
University B.D.T.College of Engineering, Davangere, Karnataka, INDIA

ABSTRACT

Wear of cutting tools is very important and various output generated by the study and analysis of each tool is extremely useful in the tool condition monitoring process in general and in making efforts to determine the expected tool life in particular. Wear of the tool tip generates poor surface finish and an unexpected tool failure may damage the tool, work piece and sometimes the machine tool itself. Advanced manufacturing demands an optimal machining process with proper control of machining parameters. Many problems that affect optimization are related to the diminished machine performance caused by worn out tools. One of the most promising tool wear analysis and tool condition monitoring techniques is based on the acoustic emission (AE) signals. The generation of the AE signals directly in the cutting zone makes them very sensitive to changes in the cutting process. Experiments have been conducted on EN-31, EN-8, EN2A and Mild Steel specimens using Uncoated and Coated carbide inserts (Kennametal, India make) on a high speed lathe for the most appropriate cutting conditions. The AE signal analysis (considering signal parameters such as, ring down count (RDC), rise time (RTT), event duration (EDT), energy (ENT) and peak amplitude (PA).

Key Words: AE, RCT, RTT, ENT, EDT, PA

1 INTRODUCTION

Reliable monitoring of machining operations facilitates economical uses of modern, capital intensive, minimally manned production systems. Tool wear, chipping or catastrophic tool failures are the major obstacles in producing high quality products. Research during the past few years has led to different methods of monitoring either directly or indirectly the state of single-
point. However, direct monitoring has not reached industrial applications because these methods interrupt the machining process, time consuming and inefficient. Indirect monitoring methods for detecting signals related to cutting tool state have become successful in recent studies. Acoustic Emission (AE) methods, monitoring stress wave propagation’s released by material undergoing deformation or fractures in the cutting zone, are one of the most promising tool monitoring techniques.

2 RESEARCH WORK

The purpose of research work was to monitor tool wear (Flank wear) by direct method and to compare and correlate the results from other indirect method and training of neural network. Artificial neural network toolbox under MATLAB V7 was used to develop the program. Experiments using single point Coated Carbide inserts were carried out on En 2A, En8, EN-31 and Mild Steel (MS) specimens. The depth of cut was fixed at 0.5 mm at two speeds namely 500 rpm (94 m/min) and 1000 rpm (188 m/min) at feeds of 0.07 and 0.09 mm/rev respectively. Various parameters like acoustic emission, cutting forces, temperature, surface roughness, vibration are known to have an affect on degree of tool wear during cutting. Acoustic emission and force signals were selected as sensing fields. Since sound signals emitted during turning process are of a very high frequency, relatively distinct signal with almost zero is obtained. Acoustic emission sensors were mounted near the tool to collect acoustic emission signals released during cutting, which is passed through a pre-amplifier and then converted into electric signal and is collected in a computer. The flank wear measured are tabulated in the wear data sheet. The AE signals readings were taken for around 20 trials for which, considerable wear were observed during preliminary work. The AE parameters and flank wear are tabulated...

Experimental Set Up

Experimental work is shown in the figure. AE sensor is placed on the tool holder and connected with AET-5500 (Acoustic Emission Analyzer) through the pre amplifier and the AET-5500 main frame connected with the host computer for data acquisition and for online display of the data. Frequencies that are commonly used for AE testing lie between 50 kHz and 2 MHz. Various acoustic emission parameters were considered for analysis which are inputs to train the neural
After training the network, the weights obtained were saved. These weights are then used to compute the flank wear for a given trial. This program can be used to control the turning operations by deciding values for tool wear depending on cutting conditions and its application by experienced personnel.

3 ACOUSTIC EMISSION TECHNIQUES - FOR MONITORING TOOL WEAR

The acoustic emission technique (AET) is a nondestructive evaluation [9,10] and has shown the possibility for characterization and wear assessment in machining operations using conventional and non-conventional materials. This method has also been widely used to detect process changes like tool wear in metal cutting. In-process tool wear monitoring has presently acquired more importance because of flexibility and provides scope for the optimal utilization. The AE response from metal cutting changes as soon as the cutting tool starts the first cut and continues to change as the cutting tool wears. The major advantage of using AE to detect the condition of tool wear is that the frequency range of the AE signal is much higher than that of the machine vibrations and other interferences and relatively uncontaminated signal can be easily obtained by the use of a high pass filter. In addition, AE can be measured by simply mounting a piezoelectric transducer on the tool holder without being interrupted during the cutting operation.

4 MEASUREMENTS AND PREDICTION OF TOOL WEAR

Many types of tool wear are practically seen and some of them are measurable. Prominent amongst them are flank wear and crater wear. Tool wear [1] starts immediately when the new tool is put into operation. The other consequences of tool wear like increase in surface roughness, increase in temperature, increased vibration, decrease in the production efficiency, decrease in component quality, loss of tool, increase in cutting forces, etc could be identified only after some time.

Tool wear analysis research establishes the relationship with the machining parameters. Further, prediction and detection of tool failure based on the wear measurement is beneficial to avoid material losses and tool. A key prerequisite to the development of systems of such a capacity is reliable definition of the tool failure. One obvious criterion is that of the gross tool breakage. A better criterion than breakage for tool failure is based on the measurement of the tool wear and checking the tool condition. Wear measurement [2] is by far more popular, although other quantities such as tool surface finish have also been found to be sensitive measures of the tool life. The most commonly studied type wear is the flank wear, which is the scar that forms at the tip of the tool and easy to measure. The other types of tool wear are difficult to measure and are more
often encountered on the steel metal cutting tools than the carbide based tools.

![Flank Wear Curve](image)

**Fig.** A typical tool wear curve

Each type of tool wear tends to progress at varying rate. An initial period of the rapid wear, known as wear-in (stage-1), is followed by an extended period of the stable wear (stage-2). As the tool nears the end of its useful life, wear increases rapidly (stage-3) until the catastrophic failure occurs. In the stage-1, the wear occurs or grows rapidly with in the short period of time because during the initial contact of the sharp cutting edge with the work piece, the peaks of the micro unevenness at the cutting edge are rapidly broken away. In the stage-2 the wear progresses at the uniform rate. In the stage-3, the wear rate [4] is rapid and may lead to catastrophic failure of the tool. In general the most economical wear at which to remove the tool and re-sharpen is just before the start of the rapidly increasing portion of the curve, i.e. at the start of the region-3. The wear land on the flank will not generally be uniform along the entire cutting edge length. Although the flank wear measurements are popular for determining the tool condition, they are somewhat ambiguous. When tool wear is a complex geometric and metallurgical process, which can not be completely characterized by any signal numerical measure. Still, in the interest of obtaining meaningful results in a timely fashion, wear measurements can be considered useful points of reference to judge potential tool life estimation as a part of tool monitoring techniques.

### 6 APPLICATION OF NEURAL NETWORKS

Artificial Intelligence techniques are being used for tool wear analysis [13] Application of neural network by training of the network using suitable data is the basis for obtaining the conformity of analysis. Acoustic emission data such as ring down count, rise time, energy, event duration, peak amplitude were obtained from the experiments conducted and was used as input patterns. Further cutting forces were collected and were also used as input patterns. Data for each trial were also used as input patterns. The model of MLFFNN [7] used in the analysis which consists of the final storage after the network was trained. The training of the neural network was conducted for training space generated from the experiments conducted for Ceramics inserts on EN-31. Each training space consisted of 26 of seven parameters each thus forming a [26 x 7] input matrix. The training was conducted varying the momentum constant and hidden neurons, fixing the minimum error at 0.001mm to 0.003. Once the target was reached the weight were save and these weights can be used for testing for any given set of parameters. The network calculated the flank wear and also gives the conditions of the tool. These conditions are pre defined in the testing program. Depending on the flank wear the program interprets the tool condition and report as one of the cases for checking the condition of the tool namely Good, Average, Alarming State and Replace. This is a [7x26] training space pattern generated for ceramic at .09 mm/sec feed, .5 mm depth of cut and 500 rpm i.e., 7 parameters or input neurons and 26 input patterns.
6.1 Sample Training Space for Ceramic Tool at 0.09 Feed

Function \([p, t] = \text{tti1data}()\):

\[
p = [p01 \ p02 \ p03 \ p04 \ p05 \ p06 \ p07 \ p08 \ p09 \ p10 \ p11 \ p12 \ p13 \ p14 \ p15 \ p16 \ p17 \ p18 \ p19 \ p20 \ p21 \ p22 \ p23 \ p24 \ p25 \ p26];
\]

\[
t = [0.09 \ 0.12 \ 0.14 \ 0.15 \ 0.15 \ 0.15 \ 0.15 \ 0.17 \ 0.17 \ 0.2 \ 0.2 \ 0.2 \ 0.2 \ 0.22 \ 0.22 \ 0.22 \ 0.22 \ 0.23 \ 0.23 \ 0.23 \ 0.24 \ 0.24 \ 0.24 \ 0.24 \ 0.25 \ 0.25];
\]

This is a [7x26] training space pattern generated for a ceramic tool at .09 mm/sec feed, .5 mm depth of cut and 500 rpm i.e., 7 parameters or input neurons and 26 input patterns.

7 DATA and ANALYSIS

Experiments were conducted to record the data. AE data recorded include Ring Down Counts (RDC), Rise Time (RTT), Event Duration (EDT), Energy (ENT) and Peak Amplitude (PA) for all the trials which also recorded Flank Wear for each trial along with the cutting forces measured with the help of lathe tool Dynamometer. Each trial also recorded any changes of the tool surface and material surface finish along with the measured parameter of Flank Wear.

7.1 A Sample data used for flank wear analysis

TOOL NAME: TTI15 (Ceramic) MATERIAL: EN31, FEED: 0.09 mm
DEPTH OF CUT: 0.5 mm SPEED: 500 rpm
<table>
<thead>
<tr>
<th>Trial no</th>
<th>Forces in Kgf</th>
<th>Flank wear (mm)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X 12</td>
<td>Z 7</td>
<td>0.09</td>
</tr>
<tr>
<td>2</td>
<td>X 14</td>
<td>Z 9</td>
<td>0.12</td>
</tr>
</tbody>
</table>

7.2 A Sample data used for Acoustic Emission analysis

<table>
<thead>
<tr>
<th>RDC</th>
<th>RTT</th>
<th>ENT</th>
<th>PA</th>
<th>EDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>5474</td>
<td>31312</td>
<td>41420</td>
<td>32076</td>
<td>179536</td>
</tr>
<tr>
<td>8956</td>
<td>57063</td>
<td>79400</td>
<td>62158</td>
<td>298896</td>
</tr>
</tbody>
</table>

8 RESULTS AND DISCUSSION

Flank wear occurs primarily as a result of rubbing action between flank face of the tool and work piece surface. Specially, flank wear is chosen to study because of the fact that flank is directly involved in metal contact and is more prone to wear. It is strongly dependent on the speed, fed depth of cut, etc. in this experimental investigation, throughout; speed and depth of cut were maintained constant. The variety of AE signal produced is constant upon temperature and the extent of deformation in the zone. Therefore, AE signals are collectively influenced by the extent of tool wear, either directly or indirectly. The use of AE signal for general purpose monitoring of machining process is limited because the mechanism of generation of AE signal is not completely understood, and the analytical techniques for AE signals are still being evaluated.

Mere few trials may not be sufficient for a complete analysis. Also a short duration for machining operation does not suffice as AE parameter usually show higher value in the beginning when the tool comes in contact with the work piece and then stabilize during the cutting operation. The force significantly varies with respect to flank wear. Machining operation is speed dependent and tool wear is influenced by the cutting forces. As the cutting cycle is repeated, higher values of the cutting forces are recorded showing steady and progressive change in them.

RDC and Energy as seen in the comparison of graphs, which were generated for various tool inserts have close relation with the flank wear, at discerning points it can be seen that all the three parameters rise and drop simultaneously. Parameters of an acoustic signal which were considered for analysis i.e., energy and RDC were more sensitive to the progress of wear. Experimental investigation was carried out to assist the feasibility of AE signals for measuring and analyzing tool wear and estimate wear as a part of tool monitoring technique in the process of turning.

Acoustic emission signals are very sensitive to the sequence of signals during tool contacts, chip formation and tool engagement/disengagement leads to entry and exit peaks in AE signal during its cutting cycle. The change in the AE signal is due to more deformation and rubbing of tool materials. The AE signal clearly demarcates the cutting action between sharp edge and worn out tools. In the present study, it is observed that beyond 0.2mm flank wear; there is a rapid increase in AE parameters. Among the AE parameters RDC and Energy are more sensitive to the conditions of cutting tool.
Graph 1 and 2 shows the comparison of ring down counts of various materials at various feeds (i.e. 0.09 mm/rev, and 0.07 mm/rev). It can be seen from the graphs that RDC values are higher for higher feed rates which is because of higher wear of the tool tip.

Graph 3 shows the Comparison of Flank wear vs. Trial no. for various tools for one of the feed rates a feed rate of. It can be seen from the graph that the wear is higher at higher feed rates. Therefore it can be inferred that as the feed rate increases the wear rate also increases or in simple words the tip wears out quickly.

8.1 Comparative RDC graphs for different tool materials at same feeds

Graph 1: Comparative Cumulative RDC vs. trial no at .09 feed

Graph 2: Comparative Cumulative RDC vs. trial no at .07 feed
8.2 Training the Network for Ceramic Tool at .09 mm/s feed

The initial result of training the network could be obtained by using the error versus the number of epochs, which illustrated that the graph moves closer to X-axis as the number of epochs increases thus suggesting that probability of reaching the performance goal increases with increasing number of epochs.

Number of neurons in Hidden layer = 40
Momentum constant = 0.85
Maximum epochs = 10000
Error goal = 0.001

The performance goal was reached after 4616 epochs. The network thus trained and the final weights were saved for future testing.

8.3 Flank wear analysis of Ceramic Tool at .09mm/s feed

For a particular test it was found that the tested parameters tallied with that of the results obtained by training of neural network using AE signals. Actual flank wear recorded by direct measurement was 0.23mm, result obtained by training of Neural network was 0.2299mm and physical examination and result obtained based on the attributes of the trained network indicated the condition of the tool as Average The flank wear obtained from the experiment measured directly using tool makers microscope were correlated with computed values of flank wear for 19th trial here. The results tally here and for all cases that were examined. With a suitable provision for online measurement of flank wear, Neural Network model may successfully be implemented in Computer Integrated Manufacturing Systems.

9 CONCLUSION

Experimental results obtained on measurement and the trained network generated attributes correlate for one complete experiment. Higher degree of accuracy may be achieved when the
training of network is taken up with more number of trials and use of larger number of epochs. Cutting tool wear proceeds the same way as the standard wear pattern indicating the right direction of the experiment. More than one method [8] of wear analysis and correlatives studies is being considered for obtaining higher levels of accuracy and for optimization.

10 REFERENCES

Multi-model Multi-strategy Teaching/Learning in Science, Engineering and Technology

Pradip Peter Dey¹, Mohammad Amin¹, Michelle Bright¹, Arun Datta¹,², Shakil Akhtar³, Albert Cruz¹, Mudasser F. Wyne⁴, Patrick Kennedy¹, Patrick Olson¹, Rell Snyder¹, Alireza Farahani¹, Samuel Afuwape¹, Hassan Badkoobehi¹ and Amber Lo¹

¹National University, 3678 Aero Court, San Diego, CA 92123, U.S.A.
²National University Community Research Institute, 11255 North Torrey Pines, La Jolla, CA 92037
³Clayton State University, Morrow, Georgia, U.S.A.
⁴University of Michigan, Flint, Michigan, USA,
pdey@nu.edu; mamin@nu.edu; mbright@nu.edu; adatta@nu.edu; sakhtar@clayton.edu; acruz@nu.edu; mfwyne@umflint.edu; pkennedy@nu.edu; polson@nu.edu; rsnyder@nu.edu;
afarahan@nu.edu; safuwape@nu.edu; hbadkoob@nu.edu; alo@nu.edu

Abstract: Critical teaching/learning issues are raised considering all instructional strategies including lectures, transformations, experimentations, problem solving, analogical, case-based and mathematical reasoning. Recent trends and best practices are considered.

1 INTRODUCTION
Students as well as educators enjoy Problem Based Learning (PBL). With PBL, learners are usually organized into groups, one or more problems are given to each group for solving the problems under the supervision of the instructors and learning is driven by the problems [1, 2, 3, 4, 5]. Do we need to consider any alternatives given that PBL is so successful in science, technology and engineering? Is there anything to gain from combining multiple learning strategies? Are multi-model multi-strategy learning approaches feasible in science, engineering and technology?

2 CHALLENGES
Although PBL is highly successful in certain environments, it is not appropriate for all learners for all topics, since teaching methods are not dynamically adjusted on the basis of different challenges faced by different learners for different topics. Adjusting teaching methods based on learner feedbacks may be appropriate in multi-model multi-strategy learning environments. Agility in teaching/learning and grading helps to overcome the different challenges faced by different learners for different topics. According to Glickman "Effective teaching is not a set of generic practices, but instead is a set of context-driven decisions about teaching. Effective teachers do not use the same set of practices for every lesson . . . Instead, what effective teachers do is constantly reflect about their work, observe whether students are learning or not, and, then adjust their practice accordingly [6]. Our ability to teach is enhanced with our agility to adjust. In education, the same size does not fit all. In technologically advanced societies, problem based learning is usually practiced in combination with technology based learning [7, 8] at least in the medical domain because changes in medical technology were significant during the past 80 years that it is impossible to imagine solving problems in the medical domain without technology. Technology enhanced learning is receiving reasonable considerations around the world [9]. In any domain, in addition to PBL and technology based learning, one may consider any combination of game-based learning [10, 11, 12], community-based learning [13, 14, 15], work-based learning [16, 17], inquiry-based learning [18, 19], project based learning [20, 21, 22], brain-based learning [23, 24, 25], team-based learning [26, 27], internet-based learning
Understanding of the mechanisms underlying human information processing system is crucial for comprehension of learning. Mathematical reasoning is used in science, engineering and technology with advantages in multi-cultural environments with diverse learners. Christians, Muslims, Hindus, Jews and atheists may differ on many social or religious issues, but must agree with mathematical reasoning such as:

\[
\text{if } 3X = 12 \\
\text{then } X = 4
\]

which is crucial for science, engineering and technology. Humans use analogical reasoning in learning for certain topics in certain circumstances. However, its use may or may not bring disagreements among diverse learners or community of learners based on their religion, culture tradition or beliefs. Case-based reasoning and anecdotal reasoning are also used in learning.

3 IMPORTANT STEPS
The goals of the 21st-century academia are to provide the flexibility, improve the learning outcomes and equip learners with appropriate knowledge for solving more complex problems never experienced before. As we learn more about learning we understand scientific aspects of learning based on the recent contributions from neuroscience, psychology and cognitive science. It would be wonderful if knew more about learning. Can excellent teaching abilities be learned? Can these abilities be taught? What parameters characterize symbolic learning that can be isolated? Can we positively correlate and verify organic level learning? We are struggling with many such questions about teaching and learning. “Recent test results show that U.S. 10th-graders ranked just 17th in science among peers from 30 nations, while in math they placed in the bottom five. Research suggests that a good teacher is the single most important factor in boosting achievement, more important than class size, the dollars spent per student or the quality of textbooks and materials”[51]. Our discussion of learning considers all instructional strategies including lectures, transformations, experimentations, problem solving, analogical, case-based and mathematical reasoning for affective learning utilizing tools and technologies in an innovative way. We may combine several strategies in creative ways for the benefit of diverse learners. The practitioners realize that certain propositions are more important than others:

1. Learning outcomes are more important than tools.
2. Analyzing feedbacks from students is more important than declaring student centered environments.
3. Dynamically adjusting teaching strategies to learner’s goals and preferences is more important than following a teaching plan.
4. Teaching activities are driven by realistic problem solving, feedbacks from students and learning outcomes rather than by a schedule.
5. Demonstrating problem solving strategies on sample problems is more effective for teaching/learning than lecturing on them.
6. Dynamically combining multiple strategies in multiple models is more effective for teaching/learning than relying on a single pre-planned strategy.
7. Mathematical reasoning is more important than case-based reasoning.
One other aspect of education using multimodal techniques is how best to use technology for instructional purposes. There are several inherent constraints that educators must take into account when lecturing, presenting or designing learning environments. Three of which are listed in *Beyond Bullet Points* [57]:

1. Limits of working memory
2. Address two channels of learning
3. Attention guiding

This reference gives a model on how to construct a convincing and longer remembered presentation taking advantage of educational research in how humans learn. Knowing these constraints, how to avoid them and take advantage of how the human mind learns increases the fulfillment of the learning objectives in any modality.

One of the important challenges every teacher faces is how to combine multiple strategies in multiple models in order to serve 21st century learners in a reasonable way so that their learning goals are met within the scope of learning theories [58-68].

**References:**

[14] Community-Based Learning, [http://www.nwrel.org/scpd/sirs/10/t008.html](http://www.nwrel.org/scpd/sirs/10/t008.html) Retrieved February 10, 2008.


