STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
*Stat., Optim. Inf. Comput.*, Vol. **2**, December 2014, pp 368–383.
Published online in International Academic Press (www.IAPress.org)

# Innovation Goals in Software Development for Business Applications

Bob Arnold [1] and M. Reza Shadnam [1, *]

[1] *R&D Incentives Group, KPMG Vancouver, Canada*

Received: 19 December 2013; Accepted: 18 April 2014
Editor: Yashwant Malaiya

**Abstract** Having been in touch with technical side of many companies in various sectors, we know industry is facing a period of unprecedented change. We have compiled a list of common technological challenges in the sector that companies are facing in adapting to the change. The purpose of this contribution is to discuss and communicate areas where technological challenges in the Software Development Sector lie. We see this kind of inventory beneficial to the academic community as it provides an account of industry challenges. The analysis would be necessary to really assist players in the sector in being better prepared for formalizing and documenting their learning and know-how development. Beside knowledge management benefits, the analysis also would help in taking advantage of research and development funding and attracting investors; both academic and industrial organizations can take advantage of this aspect. We are calling this type of analysis "capabilities analysis". A single company might not solve the world's technical problems, but just being aware of them and measuring steps taken to advance, even slightly, in the direction of solving the problems presented in this analysis would make the company stand out. When data is formulated this way, strong evidence is created that the project goes beyond standard engineering by distinguishing risk that can be eliminated through experiment from standard engineering risk.

**Keywords** standard engineering, current research, web services, technological uncertainty, experimental development.

**DOI:** 10.19139/soic.v2i4.53

*Correspondence to: Email:mshadnam@kpmg.ca

ISSN 2310-5070 (online) ISSN 2311-004X (print)

## 1. Introduction

Companies are facing challenges to access new technology, deliver process improvements, create added value products and develop new business models. At the R&D group of KPMG, we are always interested in assessing the emerging trends in the industry. We work with a number of many major companies in various sectors. We help our clients understand emerging trends, build for the future and stay ahead of their competitors by taking optimal advantage of available governmental funding mechanisms.

This contribution is intended to serve as an account of technological uncertainties, challenges and difficulties in the computing science and software development industrial research and development. Having worked with the R&D side of hundreds of IT product development and systems integrators and IT departments within companies it is amazing to see that number of common technological challenges in the sector is limited. Companies are trying to solve similar problems, which clearly define state of the art in the sector; but in many cases they do not share their solutions with each other because of IP ownership issues. Such a list of current technological problems in the industry sector would be helpful to both academics and students focus on more industry relevant projects, which would entail more interest and investment from industry and governmental granting agencies. Some of the funding agencies/programs require industrial partners and/or partial funding through industry and for some others formal expression of interest from industry/companies in form of a letter of interest is required (e.g. some of NSERC funding mechanisms). The authors would be pleased to put research groups in touch with possible industrial partners. Last but not least would be better absorption of students in the industry job market.

It has been shown that technology projects for companies could be grouped into 4 categories (see Table 1):

Table1- Innovative business project classification

| Business Efficiency | | |
|---|---|---|
| | **Process (Quadrant I)**<br>Improving processes, adding new capabilities:<br>• R&D<br>• Design<br>• Analytical capabilities<br>• Low-cost production/development | **Business Models (Quadrant II)**<br>Innovating revenue generation:<br>• Web and mobile technologies<br>• Improve motivation<br>• Use of big-data in support of decisions, monitor choices and outcomes |
| | **Technology (Quadrant III)**<br>New technologies in all aspects:<br>• Improved delivery<br>• Product stability<br>• models | **Added Value Products (Quadrant IV)**<br>Additional product features or services that go beyond expectations:<br>• monitoring<br>• automatic testing<br>• automatic monitoring |
| | | Business Value |

We see current challenge areas in the industrial sector are:

1 AI: TRUSTED SYSTEMS (Added Value Products (Quadrant IV);
2 APPLIANCES: APPLICATIONS (Added Value Products (Quadrant IV);
3 APPLIANCES: USER INTERFACES (Added Value Products (Quadrant IV);
4 AUTOMATED FAILURE DETECTION AND CORRECTION (Technology (Quadrant III));
5 DATA WAREHOUSE/BIG DATA DESIGN (Process (Quadrant I));
6 DATA WAREHOUSE/BIG DATA DESIGN: ANALYTICS (Business Models (Quadrant II);
7 DATA WAREHOUSE/BIG DATA DESIGN: DATA QUALITY (Technology (Quadrant III));
8 DECISION SUPPORT SYSTEMS (Business Models (Quadrant II));
9 DIGITAL ASSET STORAGE/RETRIEVAL (Technology (Quadrant III));
10 DIGITAL ASSET TOOL DEVELOPMENT (Technology (Quadrant III));
11 MODEL DRIVEN ENGINEERING (Process (Quadrant I));
12 NEUROMARKETING RESEARCH (Business Models (Quadrant II));
13 REMOTELY MANAGED SYSTEMS DEVELOPMENT (Process (Quadrant I));
14 SERVICE ORIENTED ARCHITECTURES (Process (Quadrant I));
15 SERVICE ORIENTED ARCHITECTURES: EDI (Business Models (Quadrant II));
16 SERVICE ORIENTED ARCHITECTURES: SECURITY (Process (Quadrant I));
17 STOCHASTIC MODELING (Process (Quadrant I));
18 SYSTEM DEVELOPMENT TOOLS (Process (Quadrant I));
19 USER INTERFACE DESIGN (Technology (Quadrant III));
20 USER INTERFACE DESIGN: DESKTOP WIDGETS (Technology (Quadrant III));
21 WEB SEARCH OPTIMIZATION: GENERAL AND LOCAL (Business Models (Quadrant II));
22 ISTM-IMPLEMENTATION OF SERVICE LEVEL AGREEMENTS, ACCOUNTING, MONITORING OF QUALITY OF SERVICE PARAMETERS (Added Value Products (Quadrant IV));
23 APPLICATION DEPLOYMENT AUTOMATION, RELEASE AUTOMATION (Process(Quadrant I));
24 AGENT BASED MODELING TO CAPTURE VOLATILITY IN THE CONSUMER PACKAGED GOODS INDUSTRY (Business Models (Quadrant II));
25 SCHEDULING IN DISTRIBUTION NETWORKS IN A DYNAMIC ROUTING ENVIRONMENT (Business Models (Quadrant II));
26 DEMAND FORECASTING IN A SUPPLY CHAIN (Business Models (Quadrant II)).

Here we would focus on 3 of the above listed areas. Two categories of approaches are out there for solving problems in these challenging areas:

**1)** There are some good approaches to challenging problems that just need use of routine engineering approaches to tackle with (such as trial and error);

**2)** There are approaches that require moving beyond standard methods and merit research and development. These problems are the focus of this contribution.

## 2. Service Oriented Architectures: Security

The Table 2 below evaluates technical risks that can be mitigated through standard engineering approaches (first category). Where a standard engineering approach is not completely effective in mitigating the specified risk, further non-routine engineering may be necessary and we have noted this by adding the tag, ”see below” to the mitigation.

Table 2. Standard engineering approaches.

| Area | Routine Risk | Mitigation |
|---|---|---|
| Authorization & authentication | General | Transport layer security is a common approach for mitigating authorization and authentication risks. This may be sufficient for simple applications. Internal configuration details may be hidden by using XML to rewrite URLs and other information exposed to the web (filtering). |
| Authorization & authentication | Denial of service attacks | Detail of service attacks may be thwarted by using a proxy XML security gateway which check and limits messages on the basis of connection duration, and message size. In addition to schema validation, checks should be made on formed-ness, identity or resource references, protocol (e.g. SOAP) validity and other message validity checks. |
| | Message Repudiation | By signing messages, to prevent modification of content, message repudiation is prevented, and the transaction history (augmented by synchronizing all network nodes using the NTP (Network Time Protocol) will be useful for verifying the sequence of steps comprising a transaction. Although this is effective,it may not be efficient as the processing requirements may be quite high (see below). |
| | Eavesdropping | Message encryption can be accelerated XML encryption computations using either hardware or software XML- accelerated devices, however, the general approach is to parse the XML transaction first, select the portions to encrypt and then to apply a set of XML and crypto functions. |
| | General attacks | Practices include use of secure sockets layer for sensitive messages limiting connect permissions to specific users or groups, firewalls, explicitly disabling or dropping endpoints and using endpoint defaults which limit processing of unexpected messages. Kerberos authentication is one of the best practices for securing XML Web services. |
| | Revocation of credentials | Method of synchronizing revocation of credentials among jurisdictions in a federation, which is able to handle non-responding jurisdictions and similar errors, has not been developed. |
| | Issues exist in the area of distributed synchronization | Method for securely distributing configuration files and federation keys has not been developed. |
| | Revocation of federation keys on withdrawal of a jurisdiction | The browser is the weak spot |

| Area | Routine Risk | Mitigation |
|---|---|---|
| Attacks | Cross site scripting and potential theft of cookies | DACS, single sign-on and role-based security system for Apache or server-based software, which provides authentication and rule-based authorization for any web service or CGI program. The custom role model is only a concept and not implemented by DACS. Although there is a workaround by modifying DACS to invoke HTTP requests for external DACS services to handle validation, no translation between DACS credentials and access control rules in different Web applications exists. |
| Revocation of federation keys | Modification of web services to accept DACS cookies non uniform and complex. | AJAX, JSON do not rely on cookies. |
| Attacks | DACS relies on cookies | See Table 3 |

### *2.1. Experimental development approaches*

The Table 3 below outlines current research in the area, including any conclusions and limitations on those conclusions. Metrics for the problem are given. A first further research step is outlined. Problems in second category represent current research in the area. As noted above, the approaches outlined in Table 3 should not be amenable to standard engineering techniques i.e. appearing in the Table 3.

Table 3. Experimental development approaches

| Federated Security Models | |
|---|---|
| Current research | Transport layer security is a common approach used by many which may be sufficient for less complicated uses. The standard practice is to use a security architecture which is multi-tiered. A web client sends a request to a middle-tier application which is redirected to a distributed access control system. The middle tier then re-directs the request with the token received to the web service. Issues exist in the area of distributed synchronization, Revocation of federation keys on withdrawal of a jurisdiction and weakness at the level of the browser open the system to cross site scripting and potential theft of cookies. Modification of web services to accept cookies for the purposes of authorization is non uniform and complex.<br><br>Other models of SOA security are detailed in the literature. Hunhs et. al. discuss the relationship between service oriented computing and multi-agent systems and present a research agenda for the next 15 years on service-oriented multi-agent systems [1]. The computing environment resembles the grid based computing environment [2] shown in Figure 1 below.<br><br>Confidentiality and message integrity may be promoted in more complex environments characterized by more than two parties, or multiple web services. |

| | |
|---|---|
| | Messages or portions of messages may be signed and encrypted and tokens may also be added to messages to assert claims, such as those made about the identity of the message sender by a trusted authority [3]. |
| | Access control models using the concept of stacking or delegation of authority (distribution of a capability), which is done by passing a WSDL description, creating a new authority by generating a restricted capability based on an original capability, (stacking an object on an original object), have been presented, but a application in the domains of interest that the writers are familiar with has not been demonstrated [4]. |
| Metrics | The metrics that would be appropriate would be the same metrics that would be applicable to code created using one architectural approach vs. another. These would include the following; parameters, exceptions thrown, cyclomatic complexity, lines of code, paths, static invocations, and anonymous classes. When designing distributed web services, it has been suggested that there are three properties that are commonly desired: consistency, availability, and partition tolerance. It has also been hypothesized that it is impossible to achieve all three and the authors of the referenced paper have gone on to prove this conjecture in the asynchronous network model, and then discuss solutions to this involving a relaxation of requirements to a partially synchronized model [5]. |
| First test | We will implement a system whereby users will be attached to user proxies and resources attached to resource proxies via long lived credentials. Resource proxies will be attached to user proxies and other resource proxies through short-lived credentials. The tests of the architecture will be compared with the architecture of the current distributed access control system. |

| Creating Secure Operating Systems From Insecure Components | |
|---|---|
| Current research | A new operating system has been proposed which prevents confidential processes from conveying any information to non-confidential processes [6]. Message passing systems require communication with trusted user level services such as file servers in order to perform data transfer operations making it difficult to enforce one-way information flow. |

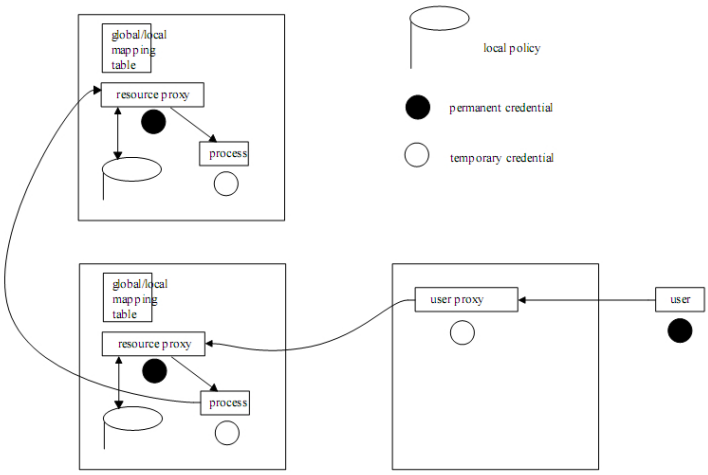


Figure 1. Computational Grid Security Architecture

## 3. Application Deployment Automation, Release Automation

The Table 4 below evaluates all technical risks that can be mitigated through standard engineering approaches (first category).

Table 4. Standard engineering approaches

| Area | Routine risk | Mitigation |
|---|---|---|
| Knowledge | Increasing testing time or more escaped bugs. | Hire someone with experience. |
| Automation | Manual testing is time consuming. Automated testing is too difficult and too expensive to implement | Other more recent tools like Ant or MSBuild are frequently used in Continuous Integration environments. (Agile) |
| | Inability to test all technologies Windows, .NET, Java, WPF (XAML) applications, web pages, web servers and web services together. | Right software e.g. TestComplete (Agile) |
| Project complexity | Inability to test often. | Right software e.g.TestComplete |
| Conflicting changes | Checking in weeks worth of work runs the risk of conflicting with other features and can be very difficult to solve. | Other more recent tools like Ant or MSBuild are frequently used in Continuous Integration environments |

| Area | Routine risk | Mitigation |
|---|---|---|
| Excessive processing time | Incomplete testing or interference with other applications. | Buy more hardware. |
| Environmental complexity | Having a test environment can lead to failures in tested systems when they are deployed to the production environment, because the production environment may differ from the test environment in a significant way. | Test in a clone of the production environment. using Virtual server technology, a copy of the production server can be sent as a file to the test lab-significantly decreasing setup time. |
| Environmental/project complexity | The state of the art analysis does not capture the complexity of systems under test with more than two languages or layers and the use of stubs to reduce problem complexity is not safe. | See Table 5 |
| GUIs, B2B and other asynchronous communicating systems | The standard approach in which generation of test cases which model all states becomes a difficult task when a GUI is the module under test. Although A solution is proposed to allow the tester to generate and save naive scripts this is at the stage of bench top. | See Table 5 |
| Test Tools for Service Oriented Architectures | Most testing tools are incapable of building composite interdependent tests across technology platforms, languages and systems, rendering end to end testing of SOAs impossible using existing approaches. | See Table 5 |

The Table 5 below outlines current research in the area, including any conclusions and limitations on those conclusions. Metrics for the problem are given. A first further research step is outlined. Problems in second category represent current research in the area.

Table 5. Experimental development approaches

| Environmental/project complexity | |
|---|---|
| Current research | Recent literature describes an algorithm for automatic test input generation for database applications and offers an analysis based a view with two languages or layers: Java programs and the database. It is noted that most enterprise applications are built in several different layers, including JavaScript code, browser forms, and a server such as Tomcat that mediates data flow. The second recent piece of academic |

work describes the application of automated environment generation to commercial software [7]. Environment generation is found to be a non-trivial task. A test environment should be general enough to subject the module under test to a range of conditions, in combination, yet limited so that both the development of the test environment and the testing itself could be carried out in a limited amount of time with limited resources. It would not be surprising to find that a portal type system which funneled goods and payments, including partial deliveries and payments could generate as many as three billion test states such that it would be impractical to address each one. Constraints are introduced to promote a scalable solution. The static analyses of a typical test environment generator do not take into account all possible dependencies between a module and their environment and because of this the testing would not be safe. Java.sql has also been used to implement the test bed and JDBC were modeled as empty stubs, a significant limitation on environment generation.

Metrics A metric would be the number of errors discovered or in the case of verified code, test coverage.

First test Developers wanting to compare their current hand coded test environment versus automatically generated test environment would therefore measure the number of errors discovered or in the case of verified code, test coverage. The numbers for states and transitions, path coverage, and branch coverage would also be measured.

---

GUIs, B2B and other asynchronous communicating systems

---

Current research This problem is particularly acute in asynchronous systems; notably GUIs or B2B. The general reference article [7] alludes to two problems of testing these types of systems. 1) The number of features is large and 2) the non-sequential nature of a transaction leading to a large number of test scenarios. A third problem lies in the regression testing of GUIs whereby, over time, the input-output mapping does not remain constant over successive versions of the software [8]. The standard approach in which generation of test cases which model all states becomes a difficult endeavor when a GUI is the module under test. The approach is only as feasible as the degree to which the number of states can be limited.

The next article also alludes to complexity of the path leading to a testable state as being one of the problems with GUI testing as a consequence of the multiple dialog sequences available concurrently in a GUI-based application. Both the application program and the test set grow when concurrent dialog resulting from exits from uncompleted functions, and interleaved or asynchronous user events are involved. Approaches based on UI Management System (UIMS) and formal process specifications were bypassed because of the need to reverse engineer a UIMS or formal model from an existing system in order to generate tests automatically [9]. A solution is proposed comprised of i) a way of simulating user inputs and binding

| | |
|---|---|
| | the input generated by genetic algorithms to the UI during execution), ii) processing logic layer to capture the state of the user interface during application execution, iii) a method to allow the tester to generate and save naive scripts. Although the ability to use a small number of inputs to generate a large number of naïve test scripts was demonstrated, and the ability of this approach to simulate the perambulations of the naïve user were subjectively better than for automated test tools and expert test scripts, the applications under test were small. |
| Metrics | The metric would be the number of errors discovered or in the case of verified code, test coverage. As with other test technologies measures of the numbers for states and transitions, path coverage, and branch coverage are relevant. |
| First test | A first test would be to create three test platforms using hand coded, versus two forms of automated test environment; one based on formal process specifications or UIMS and another which would require building a driver into the GUI so that commands or events could be sent to the software from another program [10]. The metric will be the number of errors discovered or in the case of verified code, test coverage. We will measure the numbers for states and transitions, path coverage, and branch coverage. |

| Test Tools for Service Oriented Architectures | |
|---|---|
| Current research | Most testing tools are incapable of building composite interdependent tests across technology platforms, languages and systems. Testing may be considered part the development of service oriented architectures, however, if system testing means end-to-end testing, this is made difficult by the distribution of components, the diversity of implementation platforms, the high number of system states, deployment on distributed platforms, some of which may not event be available at the time of testing [11]. Some researchers have found that although tools are available to perform testing at multiple levels; testing for qualities such as availability, performance, and security, the tools that they have analyzed cannot perform composite tests across technologies. The test tools that the authors are aware of, presuppose control over all components of the service-oriented system and do not contemplate fall back to the interfaces. They suggest overcoming the limitation of access to interfaces only through the use of gray-box testing and this is the subject of current research. |
| Limitations of current research | As above the tools that have been considered cannot perform compositetests across technologies, and assume as a pre-condition of control over all constituent parts of the system under test, not just the interfaces to the parts of the system and although current research is being carried out with respect to the applicability of gray-box testing to the problem of testing SOAs, namely, simulation of service-oriented system environments and practices for exception handling. There are more research avenues to be followed; dynamic testing in distributed, heterogeneous environments; service certification and the possibility of service repositories that |

| | |
|---|---|
| | provide test cases for services; test-aware interfaces. |
| Metrics | As with the other areas of test platform research, the metric would be the number of errors discovered though testing compared with the number found in the ”wild” or in operation or in the case of verified code, test coverage. As with other test technologies measures of the numbers for states and transitions, path coverage, and branch coverage are relevant. |
| First test | SOA test platform developers wanting to compare their test environment versus a test environment utilizing techniques such as gray-box would measure the number of errors discovered though testing compared with the number found in the ”wild” or in operation. The numbers for states and transitions, path coverage, and branch coverage would also be measured. |

## 4. Data Warehouse/Big Data Design: Data Quality

The Table 6 below evaluates all technical risks that can be mitigated through standard engineering approaches (first category).

Table 6. Standard engineering approaches

| Area | Routine risk | Mitigation |
|---|---|---|
| Data quality | Production data quality problems. | Verify that a database refactoring is required, choose the most appropriate database refactoring, deprecate the original schema, write unit tests, modify the database schema, migrate the source data, update external access programs, update your data migration script(s), run your regression tests, announce the refactoring, version control your work. |
| | Patterns in the data which suggest that some cleanup may be required, missing data, non-conformed fields, outliers in the data, duplicates in the data (i.e.) one customer may have two addresses and two customer IDs | Extract, transform, load, three database functions that are combined into one tool to pull data out of one database and place it into another database. |
| | Patterns in the data which suggest that some cleanup may be required, missing data, non-conformed fields, outliers in the data, duplicates in the data (i.e.) one customer may have two addresses and two customer IDs | Centralized Storage of Control Totals<br><br>Individual control totals are required for what was received, adjusted, excluded (errors and filters), included (corrected errors from previous cycle), output (extracts and data marts) |
| | Inability to calculate summary statistics without having to perform complex joins | De-normalization and a star schema design |

| Area | Routine risk | Mitigation |
|---|---|---|
| Data quality | Need to decompose the data subjects into data entities comprising facts and multiple dimensions. | Dimensional (logical) model (cube). Entity-relationship model, star schemas, snowflake schemas, fact-constellation schemas, persistent multidimensional stores, summary tables. The challenge, given a set of databases that are already decomposed into data entities is to transpose this model to a star schema arrangement of facts and dimensions without losing the meaning of the data. |
| | Another risk is having multiple sources of the same data. | Mitigation is through analysis to find the best source, and in the longer term, build synchronization functions or work towards consolidation to a single "source of truth" database. |
| | Missing or incorrect data leads to errors in information derived from data | Profiling to discover anomalies, validity assessment by measuring conformance with business rules, and accuracy assessment through sampling. to a single "source of truth" database. |
| | Missing or incorrect data leads to errors in information derived from data | Assign trust factors to each data source and include a "decay factor." As data gets older, the trust factor declines. (Sanlam gets a new look on life with customer data) |

The Table 7 below outlines current research in the area, including any conclusions and limitations on those conclusions. Metrics for the problem are given. A first further research step is outlined. Problems in second category represent current research in the area.

Table 7. Experimental development approaches

| Data Quality | |
|---|---|
| Current research | Statistical process control (SPC) can be used for early identification of data anomalies; an automated statistical control framework could be developed to detect when the process variables or inputs to the second stage of the data warehouse in this case were out of control. Although bands of routine variation for both the individual values and the moving ranges, the choice of which processes to monitor, and the frequency, natural batch and which variables to observe has not been studied much in practice [10].<br><br>Some authors feel that no tools for data mining exist or that the existing tools are not effective and propose a rethinking of methodologies, models, and techniques and of course a set of requirements for the technology for implementing the data flow process [11]. The main components introduced are described as; an integrator which integrates in a coordinated fashion data from operational databases, from the DW, and from other data streams; a repository capable of storing short-term data for quick retrieval, for the purpose of rule application and mining; a module that computes hierarchies of indicators to feed dashboards and reports; tools for extracting patterns out of the data streams; a module which monitors the events and sends messages to the users.<br><br>Data latency or the interval between an event through a transforming process of the data coming from databases as well as from data input streams is a process variable that we seek to minimize. Although this might be achieved through the techniques of dynamic integration, such as by query writing on heterogeneous sources which has been reported to have been implemented in prototype, however it is conjectured by the authors of one paper that most of the cleaning techniques devised (purge/merge and duplicate detection) rely on a materialized integrated level. |

| | |
|---|---|
| Limitations of current research | As stated above, although reduction in data latency might be achieved through the techniques of dynamic integration, it is conjectured that most of the cleaning techniques devised (purge/merge and duplicate detection) rely on a materialized integrated level.<br><br>Notable events are not limited to patterns which may be detected in short term information. Events of interest include time dependant patterns deviate fromthe norm by definition arise over time i.e. processes which may arise over time, and may only be detected by relying on some historical data. Work on high-performance time series mining has been carried out; however, the problem of storing data for fast retrieval arises. Since data will be accessed in different ways by different modules concurrently, straightforward buffering techniques will not be work in this context. |
| Metrics | Measures of technologies for ETL, specifically, data cleaning are therefore the latency in the cleaning process and the depth of the sample in which patterns that deviate from the norm may be detected. |
| First test | A next step would be to prototype, in memory, merge/purge and duplicate detection, measuring the latency in the process. A technique for buffering the input data stream to a certain depth could also be prototyped. |
| ETL - Web Services, Real-time data | |
| Current research | One paper suggests that web services might be used to pack a data warehouse containing real time data of an electric power company [16] This work touches on data transformation and flow (ETL) with the update strategy based on message queue and XML. |
| Limitations | Although it seems that strategies capable of ETLing real time data would be superior, the message from this research is that the traditional methods (above) are increasingly inadequate as volumes and immediacy of data increase. It is not clear whether the strategies could be implemented in the domain of interest and how. |
| Metrics | The number of edge cases (non-linear behavior) associated with a specific domain. |
| Data mining | |
| Current research | Applications of data mining are ubiquitous in the textbooks and, the metrics vary with each application. For example, many papers explore the classification problem; a major sub-field of data mining, through the use of mathematical programming based methods. The paper we choose to reference offers a good comparison of methods, including LDA, Decision Tree, SVMLight, and LibSVM which is compared with the proposed proposes a Multi-criteria Convex Quadratic Programming model (MCCQP). Although the experimental results indicate that the proposed MCCQP model achieves as good as or even better classification accuracies than other methods, the work does not address the objective of the work which was also reviewed,namely the development of a learning system, which allows the non-confidential aspects of the multiple data sets to be shared. Problems of this nature are more likely in the future.<br><br>Statistical process control (SPC) can be used for early identification of data anomalies, an automated statistical control framework could be developed at the beginning of the information chain to monitor the inputs. |
| Limitations | Although the limits on the data can define bands of routine variation for both the individual values and the moving ranges, the choice of which processes to monitor, and the frequency, natural batch and which variables to observe has not been studied much.[17] |
| Metrics | Metrics in one of the following areas will be chosen depending on the most likely use for the dataset; 1) Predictive accuracy 2) Classification accuracy 3) Rank accuracy and 4) Non-accuracy. [18] |
| First test | Datasets used for classification such as Medical:Appendicitis, Breast cancer (Wisconsin), Heart disease(Cleveland) and Other datasets:Ionosphere, Satellite image dataset (Statlog version), Sonar, Telugu, Vovel will be used to compare classification methods, broadening the work above. Metrics will be chosen as above. |

## 5. Data Warehouse/Big Data Design: Analytics

The Table 8 below evaluates all technical risks that can be mitigated through standard engineering approaches (first category).

Table 8. Standard engineering approaches

| | | |
|---|---|---|
| Complexity | Inability to process large amounts of complex data in a timely way More complex queries that involve a variety of different data sets | Use columnar data stores to enhance performance. General use of query optimizers. Technologies such as Hadoop and MapReduce |
| Complexity | Inability to process large amounts of complex data in a timely way More complex queries that involve a variety of different data sets | Use of real time data integration technologies such as data virtualization |
| Data Visualization | Lack of visual and other novel UI for bringing data immediately to the attention of the user | Use of analytics applications and data visualization tools tailored to specific industries. This tailored solution is only a partial solution to the general problem. |

Experimental Development

Table 9. Experimental development approaches

Analytics

Current research

Applications of data mining are ubiquitous in the textbooks and, the metrics vary with each application. For example, many papers explore the classification problem; a major sub-field of data mining, through the use of mathematical programming based methods. The paper we choose to reference offers a good comparison of methods, including LDA, Decision Tree, SVMLight, and LibSVM which is compared with the proposed proposes a Multi-criteria Convex Quadratic Programming model (MCCQP). Although the experimental results indicate that the proposed MCCQP model achieves as good as or even better classification accuracies than other methods, the work does not address the objective of the work which was also reviewed, namely the development of a learning system, which allows the non-confidential aspects of the multiple data sets to be shared. Problems of this nature are more likely in the future.

Statistical process control (SPC) can be used for early identification of data anomalies; an automated statistical control framework could be developed at the beginning of the information chain to monitor the inputs [10]. Although the limits on the data can define bands of routine variation for both the individual values and the moving ranges, the choice of which processes to monitor, and the frequency, natural batch and which variables to observe has not been studied much.

Golfarelli et al [11] characterize the BPM solutions proposed by software vendors as classical OLAP tools with some specialized ETL and data integration systems and cite two examples [12], [13]. They go on to say that such solutions or permutations of them will not solve the problem.

They suggest a complete functional framework for synthesizing Business Intelligence (BI), one which combines the functionality of Data Warehousing (DW) with Business Activity Monitoring (BAM) [14]. The main components introduced by BAM are:

- Right-Time Integrator (RTI) that integrates at right-time data from all sources; operational databases, DW, and from data streams;
- Dynamic Data Store (DDS), short term data store, to support rule inference
- KPI manager that computes variables to feed dashboards and reports;
- mining tools, to extract relevant patterns out of the data streams;
- rule engine that monitors the events raised by the RTI or mining tools and sends messages to users

The authors claim that although BAM reduces data latency through the provision of the above functionality, chiefly the RTI, the adoption of dynamic techniques, raises

problems of data quality and integration.

The authors conclude that while they have identified a functional framework for synthesizing Business Intelligence (BI), most of these elements, albeit the subject of active research, are not mature enough to be implemented in mature commercial products.

| | |
|---|---|
| Limitations | Golfarelli et al claim that although BAM reduces data latency through the provision of the above functionality, chiefly the RTI, the adoption of dynamic techniques, raises problems of data quality and integration. They hypothesize that although integration by query rewriting on heterogeneous sources has been widely investigated and in at least one case [15] implemented in research prototypes, most of the methods detecting and correcting (or removing) corrupt or inaccurate records (e.g. purge/merge problem and duplicate detection) rely on the presence of a materialized views of which only a fraction could be replaced by data structures in memory. They further state that current technology limits on-line queries to simple filtering, with other more complex queries would be implemented in batch.<br><br>Golfarelli et al. [11] conclude with four research challenges including reduction of data latency discussed in the paragraph above;<br><br>• The informative value of a BPM system is chiefly related to the types of rules and variables supported, the relationships between variables, inference of patterns/limits using historical data, and detection of out of control processes. High volume processing and storage techniques will also be needed.<br>• Visual and other novel UI for bringing data immediately to the user.<br>• Development of light tools to streamline the transformation of business process models to executable code. |
| Metrics | Data latency defined by the interval between an event and its perception by the user. |

## 6. Conclusions

We have demonstrated a type of analysis that we are calling "capabilities analysis" which consists of an inventory of common technical challenges in the computing science and software development sector summarizing standard engineering practice in the area, and also what is not known about the area. When data is formulated this way, strong evidence is created that the project goes beyond standard engineering by distinguishing risk that can only be eliminated through experiment; therefore, technical merit would formally be established for further research and more R&D funding through governmental and non-governmental sources would be justified. The analysis would be necessary to assist players in the sector in being better prepared for formalizing and documenting their learning and know how development.

## REFERENCES


1. Huhns, M. N. & Singh, M.P. (2005). Research Directions for Service-Oriented Multiagent Systems, IEEE Internet Computing. 9, 65-70.
2. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S. (1998). A Security Architecture for Computational Grids, 5th ACM Conference on Computer and Communication Security, pp 83- 92.
3. Kearney, P. (2005). Message Level Security for Web Services, Information Security Technical Report. 10, 41-50.
4. Mabuchi, M., Shinjo, Y., Sato, A. & Kato, K. (2008). An Access Control Model for Web-Services That Supports Delegation and Creation of Authority, Proceedings of the Seventh International Conference on Networking, 213-222.
5. Gilbert, S. & Lynch, N. (2002). Brewer's Conjecture and the Feasibility of Consistent Available Partition-Tolerant Web Services, ACM SIGACT News, 33.
6. Zeldovich, N. (2007). Securing Untrustworthy Software using Information Flow Control, Thesis submitted to the Department of Computing Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy.
7. Tkachuk, O. & Rajan, S. P. (2006). Application of Automated Environment Generation to Commercial Software, Proceedings of the 2006 international Symposium on Software Testing and Analysis (ISSTA '06), 203-214.
8. http://en.wikipedia.org/wiki/GUI software testing accessed at July 22, 2010.
9. Kasik, D.J. & George, H.G. (1996). Toward Automatic Generation of Novice User Test Scripts, Proceedings of the Conference on Human Factors in Computing Systems : Common Ground, 244-251.
10. Maurer, C. (2007). Data Warehousing: Ensuring Data Integrity with End-to-End and Statistical Process Controls, Information Management, 7.
11. Golfarelli, M., Rizzi, S. & Cella, I. (2004). Beyond data warehousing: what's next in business intelligence? Proceedings of the 7th ACM international workshop on Data warehousing and OLAP, 1-6.
12. Istante Software. Istante: Product Overview. http://www.istantesoftware.com
13. Sonnen, D., and Morris, H. (2004), BusinessFactor: Event-Driven Business Performance Manager. TIBCO White Paper
14. Dresner, H. (2003) Business Activity Monitoring: BAM Architecture. Gartner Symposium ITXPO (Cannes, France).
15. Beneventano, D., et al. (2000) Information Integration: The MOMIS Project Demonstration. In Proceedings VLDB Conference, Cairo, Egypt.
16. Singh, T. et al. (2012), Service Oriented Architecture Based Electric Power Real-time Data Ware House, International Journal of Engineering and Management Research, Vol.2, Issue 6, pp 48-56.
17. Maurer, C. (2007) Data Warehousing: Ensuring Data Integrity with End-to-End and Statistical Process Controls
18. Schroder, G., Thiele, M., Lehner, W. (2010) Setting Goals and Choosing Metrics for Recommender System Evaluations, European Social Fund and Free State of Saxony under grant agreement 080954843, http://ucersti.ieis.tue.nl/files/papers/4.pdf